

Joan Cabestany
Ignacio Rojas
Gonzalo Joya (Eds.)

LNCS 6691

Advances in Computational Intelligence

11th International Work-Conference
on Artificial Neural Networks, IWANN 2011
Torremolinos-Málaga, Spain, June 2011, Proceedings, Part I

1
Part I

 Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Joan Cabestany Ignacio Rojas
Gonzalo Joya (Eds.)

Advances in Computational Intelligence

11th International Work-Conference
on Artificial Neural Networks, IWANN 2011
Torremolinos-Málaga, Spain, June 8-10, 2011
Proceedings, Part I

Volume Editors

Joan Cabestany

Universitat Politècnica de Catalunya (UPC)

Departament d'Enginyeria Electrònica

Campus Nord, Edificio C4, c/ Gran Capità s/n, 08034 Barcelona, Spain

E-mail: cabestan@eel.upc.es

Ignacio Rojas

University of Granada

Department of Computer Architecture and Computer Technology

C/ Periodista Daniel Saucedo Aranda, 18071 Granada, Spain

E-mail: irojas@ugr.es

Gonzalo Joya

Universidad de Málaga, Departamento Tecnología Electrónica

Campus de Teatinos, 29071 Málaga, Spain

E-mail: gjoya@uma.es

ISSN 0302-9743

e-ISSN 1611-3349

ISBN 978-3-642-21500-1

e-ISBN 978-3-642-21501-8

DOI 10.1007/978-3-642-21501-8

Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011928243

CR Subject Classification (1998): J.3, I.2, I.5, C.2.4, H.3.4, D.1, D.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

© Springer-Verlag Berlin Heidelberg 2011

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

We are proud to present the set of final accepted papers for the eleventh edition of the IWANN conference “International Work-Conference on Artificial Neural Networks” held in Torremolinos (Spain) during June 8–10, 2011.

IWANN is a biennial conference that seeks to provide a discussion forum for scientists, engineers, educators and students about the latest ideas and realizations in the foundations, theory, models and applications of hybrid systems inspired by nature (neural networks, fuzzy logic and evolutionary systems) as well as in emerging areas related to the above items. As in previous editions of IWANN, this year’s event also aimed to create a friendly environment that could lead to the establishment of scientific collaborations and exchanges among attendees. Since the first edition in Granada (LNCS 540, 1991), the conference has evolved and matured. The list of topics in the successive Call for Papers has also evolved, resulting in the following list for the present edition:

1. **Mathematical and theoretical methods in computational intelligence:** Mathematics for neural networks; RBF structures; Self-organizing networks and methods; Support vector machines and kernel methods; Fuzzy logic; Evolutionary and genetic algorithms
2. **Neurocomputational formulations:** Single-neuron modelling; Perceptual modelling; System-level neural modelling; Spiking neurons; Models of biological learning
3. **Learning and adaptation:** Adaptive systems; Imitation learning; Reconfigurable systems; Supervised, non-supervised, reinforcement and statistical algorithms
4. **Emulation of cognitive functions:** Decision making; Multi-agent systems; Sensor mesh; Natural language; Pattern recognition; Perceptual and motor functions (visual, auditory, tactile, virtual reality, etc.); Robotics; Planning motor control
5. **Bio-inspired systems and neuro-engineering:** Embedded intelligent systems; Evolvable computing; Evolving hardware; Microelectronics for neural, fuzzy and bioinspired systems; Neural prostheses; Retinomorphic systems; Brain–computer interfaces (BCI) nanosystems; Nanocognitive systems
6. **Hybrid intelligent systems:** Soft computing; Neuro-fuzzy systems; Neuro-evolutionary systems; Neuro-swarm; Hybridization with novel computing paradigms: Quantum computing, DNA computing, membrane computing; Neural dynamic logic and other methods; etc.
7. **Applications:** Image and signal processing; Ambient intelligence; Biomimetic applications; System identification, process control, and manufacturing; Computational biology and bioinformatics; Internet modeling, communication and networking; Intelligent systems in education; Human–robot interaction. Multi-agent systems; Time series analysis and prediction; Data mining and knowledge discovery

At the end of the submission process, we had 202 papers on the above topics. After a careful peer-review and evaluation process (each submission was reviewed by at least 2, and on average 2.4, Program Committee members or additional reviewer), 154 papers were accepted for oral or poster presentation, according to the recommendations of reviewers and the authors' preferences.

It is important to note that for the sake of consistency and readability of the book, the presented papers are not organized as they were presented in the IWANN 2011 sessions, but classified under 21 chapters and with one chapter on the associated satellite workshop. The organization of the papers is in two volumes and arranged following the topics list included in the call for papers. The first volume (LNCS 6691), entitled *Advances in Computational Intelligence. Part I* is divided into ten main parts and includes the contributions on:

1. Mathematical and theoretical methods in computational intelligence
2. Learning and adaptation
3. Bio-inspired systems and neuro-engineering
4. Hybrid intelligent systems
5. Applications of computational intelligence
6. New applications of brain-computer interfaces
7. Optimization algorithms in graphic processing units
8. Computing languages with bio-inspired devices and multi-agent systems
9. Computational intelligence in multimedia processing
10. Biologically plausible spiking neural processing

In the second volume (LNCS 6692), with the same title as the previous volume, we have included the contributions dealing with topics of IWANN and also the contributions to the associated satellite workshop (ISCIF 2011). These contributions are grouped into 11 chapters with one chapter on the satellite workshop:

1. Video and image processing
2. Hybrid artificial neural networks: models, algorithms and data
3. Advances in machine learning for bioinformatics and computational biomedicine
4. Biometric systems for human-machine interaction
5. Data mining in biomedicine
6. Bio-inspired combinatorial optimization
7. Applying evolutionary computation and nature-inspired algorithms to formal methods
8. Recent advances on fuzzy logic and soft computing applications
9. New advances in theory and applications of ICA-based algorithms
10. Biological and bio-inspired dynamical systems
11. Interactive and cognitive environments
12. International Workshop of Intelligent Systems for Context-Based Information Fusion (ISCIF 2011)

During the present edition, the following associated satellite workshops were organized:

1. **4th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2011)**. CISIS aims to offer a meeting opportunity for academic and industry-related researchers belonging to the various vast communities of computational intelligence, information security, and data mining. The corresponding selected papers are published in an independent volume (LNCS 6694).
2. **International Workshop of Intelligent Systems for Context-Based Information Fusion (ISCIF 2011)**. This workshop provides an international forum to present and discuss the latest scientific developments and their effective applications, to assess the impact of the approach, and to facilitate technology transfer. The selected papers are published as a separate chapter in the second volume (LNCS 6692).
3. **Third International Workshop on Ambient-Assisted Living (IWAAL)**. IWAAL promotes the collaboration among researchers in this area, concentrating efforts on the quality of life, safety and health problems of elderly people at home. IWAAL papers are published in LNCS volume 6693.

The 11th edition of IWANN was organized by the Universidad de Malaga, Universidad de Granada and Universitat Politecnica de Catalunya, together with the Spanish Chapter of the IEEE Computational Intelligence Society. We wish to thank to the Spanish Ministerio de Ciencia e Innovacion and the University of Malaga for their support and grants.

We would also like to express our gratitude to the members of the different committees for their support, collaboration and good work. We specially thank the organizers of the associated satellite workshops and special session organizers. Finally, we want to thank Springer, and especially Alfred Hofmann, Anna Kramer and Erika Siebert-Cole, for their continuous support and cooperation.

June 2011

Joan Cabestany
Ignacio Rojas
Gonzalo Joya

Organization

IWANN 2011 Organizing Committee

Honorary Chairs

Alberto Prieto	University of Granada
Francisco Sandoval	University of Malaga

Conference Chairs

Joan Cabestany	Polytechnic University of Catalonia
Ignacio Rojas	University of Granada
Gonzalo Joya	University of Malaga

Technical Program Chairs

Francisco Garcia	University of Malaga
Miguel Atencia	University of Malaga

Satellite Workshops Chairs

Juan M. Corchado	University of Salamanca
Jose Bravo	University of Castilla la Mancha

Publicity and Publication Chairs

Pedro Castillo	University of Granada
Alberto Guillen	University of Granada
Beatriz Prieto	University of Granada

IWANN 2011 Program Committee

Plamen Angelov	University of Lancaster
Cecilio Angulo	Polytechnic University of Catalonia
A. Artes Rodriguez	University of Carlos III, Madrid
Antonio Bahamonde	University of Oviedo
R. Babuska	Delft University of Technology
Sergi Bermejo	Polytechnic University of Catalonia
Piero P. Bonissone	GE Global Research
Andreu Catala	Polytechnic University of Catalonia
Gert Cauwenberghs	University of California, San Diego
Jesus Cid-Sueiro	University of Carlos III, Madrid
Rafael Corchuelo	University of Seville

Óscar Córdón	European Centre for Soft Computing
Carlos Cotta	University of Malaga
Marie Cottrell	University of Paris I
Alicia D'Anjou	University of the Basque Country
Luiza De Macedo Mourelle	State University of Rio de Janeiro (UERJ)
Dante Del Corso	Polytechnic of Turin
Angel P. del Pobil	University of Jaume I, Castellon
Richard Duro	University of A Coruña
Marcos Faundez-Zanuy	Polytechnic University of Mataro
J. Manuel Ferrández	Polytechnic University of Cartagena
Kunihiko Fukushima	Takatsuki, Osaka
Chistian Gamrat	CEA, Gif sur Yvette
Patrik Garda	University Paris Sud, Orsay
F. Javier Gonzalez Cañete	University of Malaga
Karl Goser	University of Dortmund
Manuel Graña	University of the Basque Country
Anne Guerin-Dugue	Institut National Polytechnique de Grenoble
Hani Hagra	University of Essex
Alistar Hamilton	University of Edinburgh
Jeanny Hérault	GIPSA-Lab, INPG, Grenoble
Luis Javier Herrera	University of Granada
Francisco Herrera	University of Granada
Cesar Hervás	University of Cordoba
Tom Heskes	Radboud University Nijmegen
Pedro Isasi	University of Carlos III, Madrid
Simon Jones	University of Loughborough
Christian Jutten	GIPSA-lab/DIS - CNRS - Grenoble University
Kathryn Klemic	Yale University
Amaury Lendasse	Helsinki University of Technology
Kurosh Madani	University of Paris XII
Jordi Madrenas	Polytechnic University of Catalonia
Luís Magdalena	ECSC Mieres
Dario Maravall	Polytechnic University of Madrid
Bonifacio Martín Del Brio	University of Zaragoza
Francesco Masulli	University of La Spezia, Genoa
Jose M. Molina	University of Carlos III, Madrid
Augusto Montisci	University of Cagliari
Claudio Moraga	European Centre for Soft Computing
Juan M. Moreno	Polytechnic University of Catalonia
Klaus-Robert Muller	FIRST, Berlin
Jose Muñoz	University of Malaga
Alan F. Murray	Edinburgh University
Jean-Pierre Nadal	Normal Superior School, Paris

Nadia Nedjah	State University of Rio de Janeiro
Erkki Oja	Helsinki University of Technology
Madalina Olteanu	University of Paris I
Julio Ortega	University of Granada
Kevin M. Passino	The Ohio State University USA
Witold Pedrycz	University of Alberta
Francisco Pelayo	University of Granada
Vincenzo Piuri	University of Milan
Hector Pomares	University of Granada
Carlos G. Puntonet	University of Granada
Leonardo Reyneri	Polytechnic of Turin
Eduardo Ros	University of Granada
Ulrich Rueckert	University of Paderborn
Eduardo Sanchez	LSI, EPFL
Jordi Solé-Casals	University of Vic
Peter Szolgay	Pazmany Peter Catholic University
John Taylor	Kings College London, UK
Carme Torras	Polytechnic University of Catalonia
I. Burhan Turksen	TOBB Econ Technol. University, Ankara
Mark Van Rossum	University of Edinburgh
Marley Vellasco	Pontif. Catholic University of Rio de Janeiro
Alfredo Vellido	Polytechnic University of Catalonia
Michel Verleysen	Catholic University of Louvain-la-Neuve
Thomas Villmann	University of Leipzig
Changjiu Zhou	Singapore Polytechnic
Ahmed Zobaa	University of Cairo
Pedro Zufiria	Polytechnic University of Madrid

IWANN 2011 Reviewers

Carlos Affonso	Nove de Julho University
Vanessa Aguiar	University of A Coruña
Arnulfo Alanis Garza	Instituto Tecnológico de Tijuana
Amparo Alonso-Betanzos	University of A Coruña
Juan Antonio Alvarez	University of Seville
Jhon Edgar Amaya	University of Tachira
César Andrés	Complutense University of Madrid
Anastassia Angelopoulou	University of Westminster
Plamen Angelov	Lancaster University
Davide Anguita	University of Genoa
Cecilio Angulo	Polytechnic University of Catalonia
Angelo Arleo	CNRS - University Pierre and Marie Curie Paris VI
Manuel Atencia	IIIA-CSIC
Miguel Atencia	University of Malaga

Jorge Azorin	University of Alicante
Davide Bacciu	IMT Lucca School for Advanced Studies
Antonio Bahamonde	University of Oviedo at Gijón, Asturias
Halima Bahi	University of Annaba
Javier Bajo	Pont. University of Salamanca
Juan Pedro Bandera	University of Malaga
Cristian Barriué	Polytechnic University of Catalonia
Bruno Baruque	University of Burgos
David Becerra	University of the West of Scotland
Lluís A. Belanche-Munoz	Polytechnic University of Catalonia
Sergi Bermejo	Polytechnic University of Catalonia
Nicu Bizdoaca	University of Craiova
Juan Botia	University of Murcia
Julio Bregáins	University of A Coruña
Gloria Bueno	University of Castilla-La Mancha
Joan Cabestany	Polytechnic University of Catalonia
Inma P Cabrera	University of Malaga
Tomasa Calvo	University of Alcala
Jose Luis Calvo-Rolle	University of A Coruña
Mariano Carbonero-Ruz	ETEA - Cordoba University
Carlos Carrascosa	GTI-IA DSIC Universidad Politecnica de Valencia
Luis Castedo	University of A Coruña
Pedro Castillo	University of Granada
Ana Cavalli	GET/INT
Miguel Cazorla	University of Alicante
Raymond Chiong	Swinburne University of Technology
Jesus Cid-Sueiro	University of Madrid
Máximo Cobos	Universidad Politecnica de Valencia
Valentina Colla	Scuola Superiore S. Anna
Feijoo Colomine	University of Tachira
Pablo Cordero	University of Malaga
Óscar Córdón	European Centre for Soft Computing
Francesco Corona	TKK
Ulises Cortes	Polytechnic University of Catalonia
Carlos Cotta	University of Malaga
Marie Cottrell	Universite Paris I
Mario Crespo-Ramos	University of Oviedo
Raúl Cruz-Barbosa	Universidad Tecnológica de la Mixteca
Manuel Cruz-Ramírez	Departamento de Informática y Análisis Numérico
Erzsébet Csuha-j-Varjú	Hungarian Academy of Sciences
Daniela Danciu	University of Craiova
Adriana Dapena	University of A Coruña
Alberto De La Encina	Universidad Complutense

Luiza De Macedo Mourelle	State University of Rio de Janeiro (UERJ)
Suash Deb	C.V. Raman College of Engineering
José Del Campo-Ávila	University of Malaga
Angel P. Del Pobil	Jaume-I University
Enrique Dominguez	University of Malaga
Julian Dorado	University of A Coruña
Richard Duro	University of A Coruña
Gregorio Díaz	University of Castilla-La Mancha
Marta Díaz	Polytechnic University of Catalonia
Emil Eirola	Helsinki University of Technology
Patrik Eklund	Umea University
Pablo Estevez	University of Chile
Marcos Faundez-Zanuy	Escola Universitaria Politecnica de Mataro
Carlos Fernandez	University of A Coruña
J. Fernandez De Cañete	University of Malaga
Alberto Fernandez Gil	University Rey Juan Carlos
E. Fernandez-Blanco	University of A Coruña
J.C. Fernández Caballero	University of Cordoba
M. Fernández Carmona	University of Malaga
F. Fernández De Vega	University of Extremadura
Antonio Fernández Leiva	University of Malaga
F. Fernández Navarro	University of Cordoba
J. Manuel Ferrández	Universidad Politecnica de Cartagena
Anibal R. Figueiras-Vidal	Universidad Politecnica de Madrid
Oscar Fontenla-Romero	University of A Coruña
Leonardo Franco	University of Malaga
Ana Freire	University of A Coruña
Ramón Fuentes	Universidad Publica de Navarra
Colin Fyfe	University of the west of scotland
José Gallardo	University of Malaga
Jose Garcia Rodríguez	University of Alicante
Francisco Garcia-Lagos	University of Malaga
Maite Garcia-Sebastian	University of the Basque Country
Juan Miguel García	Universidad Politecnica de Valencia
Patricio García Báez	University of La Laguna
Pablo García Sánchez	University of Granada
Maribel García-Arenas	University of Granada
Esther García-Garaluz	University of Malaga
Patrick Garda	UPMC (France)
Marcos Gestal	University of A Coruña
Peter Gloesekotter	University of Applied Sciences Münster
Juan Gomez	University of Madrid
Luis González Abril	University of Seville
Jesús González Peñalver	University of Granada
Juan Gorriz	University of Granada

Karl Goser	University of Dortmund
Bernard Gosselin	Université de Mons
Jorge Gosálbez	Universidad Politecnica de Valencia
Manuel Grana	University of the Basque Country
Bertha Guijarro-Berdiñas	University of A Coruña
Nicolás Guil	University of Malaga
Alberto Guillen	University of Granada
Pedro Antonio Gutiérrez	University of Cordoba
Vanessa Gómez-Verdejo	University of Madrid
Andrei Halanay	Polytechnic University of Bucharest
Alistair Hamilton	University of Edinburgh
Francisco Herrera	University of Granada
Álvaro Herrero	University of Burgos
Cesar Hervás	University of Cordoba
Tom Heskés	Radboud University Nijmegen
M. Hidalgo-Herrero	Universidad Complutense
Rob Hierons	Brunel University
Wei-Chiang Hong	School of Management, Da Yeh University
Jeanny Hérault	GIPSA-Lab, INPG, Grenoble
José Jerez	University of Malaga
M.D. Jimenez-Lopez	University of Rovira i Virgili
J.L. Jiménez Laredo	University of Granada
Simon Jones	University of Loughborough
Gonzalo Joya	University of Malaga
Vicente Julian	GTI-IA DSIC UPV
Christian Jutten	GIPSA-lab/DIS - CNRS - Grenoble University
Jorma Laaksonen	Helsinki University of Technology
Alberto Labarga	University of Granada
Vincent Lemaire	Orange Labs
Amaury Lendasse	HUT
Paulo Lisboa	Liverpool John Moores University
Ezequiel Lopez	University of Malaga
Rafael Luque	University of Malaga
Otoniel López	Miguel Hernandez University
Guillermo López Campos	Institute of Health "Carlos III"
M.A. López Gordo	University of Granada
Kurosh Madani	LISSI / Université PARIS XII
Jordi Madrenas	Polytechnic University of Catalonia
Luis Magdalena	ECSC Mieres
Enric Xavier Martin Rull	Polytechnic University of Catalonia
Luis Martí	University of Madrid
Mario Martín	Polytechnic University of Catalonia
Bonifacio Martín Del Brio	University of Zaragoza
José Martín Guerrero	University of Valencia

José Luís Martínez	University of Castilla-La Mancha
F.J. Martínez-Estudillo	ETEA
Francesco Masulli	University of Genova
Montserrat Mateos	Pont. University of Salamanca
Jesús Medina-Moreno	University of Cadiz
Mercedes Merayo	Complutense University of Madrid
Juan J. Merelo	University of Granada
Gustavo J. Meschino	National University of Mar del Plata
Jose M. Molina	University of Madrid
Carlos Molinero	Complutense University of Madrid
Federico Montesini-Pouzols	HUT
Augusto Montisci	University of Cagliari
Antonio Mora	University of Granada
Angel Mora Bonilla	University of Malaga
Claudio Moraga	European Centre for Soft Computing
Gin Moreno	University of Castilla la Mancha
Juan M. Moreno	Polytechnic University of Catalonia
Juan Moreno García	University of Castilla-La Mancha
Jose Muñoz	University of Malaga
Susana Muñoz Hernández	Technical University of Madrid
E. Mérida-Casermeiro	University of Malaga
Nadia Nedjah	State University of Rio de Janeiro
Pedro Nuñez	University of Extremadura
Manuel Núñez	UCM
Salomon Oak	California State Polytechnic University
Manuel Ojeda-Aciego	University of Malaga
Madalina Olteanu	SAMOS, Université Paris 1
Jozef Oravec	PF UPJS
Julio Ortega	University of Granada
A. Ortega De La Puente	Autonomous University of Madrid
Juan Miguel Ortiz	University of Malaga
Inma P. De Guzmán	University of Malaga
Oswaldo Pacheco	Universidade de Aveiro
Esteban Palomo	University of Malaga
Diego Pardo	Polytechnic University of Catalonia
Miguel Angel Patricio	University of de Madrid
Fernando L. Pelayo	University of Castilla-La Mancha
Francisco Pelayo	University of Granada
Vincenzo Piuri	University of Milan
Hector Pomares	University of Granada
Alberto Prieto	University of Granada
Mar Prueba	University of Malaga
Aleka Psarrou	University of Westminster
Francisco Pujol	University of Alicante
Carlos G. Puntonet	University of Granada

José Manuel Pérez	University of Jaen
Pablo Rabanal	Complutense University of Madrid
Juan Rabuñal	University of A Coruña
Ander Ramos	University of Tübingen
Daniel Rivero	University of A Coruña
Ismael Rodríguez Laguna	Complutense University of Madrid
A. Rodríguez-Molinero	Hospital Sant Antoni Abat
Juan Antonio Rodríguez	University of Malaga
Sara Rodríguez	University of Salamanca
David Rodríguez Rueda	University of Tachira
Ignacio Rojas	University of Granada
Fernando Rojas	University of Granada
Enrique Romero	Polytechnic University of Catalonia
Samuel Romero Garcia	University of Granada
Ricardo Ron	University of Malaga
Eduardo Ros	University of Granada
Fabrice Rossi	TELECOM ParisTech
Peter Roth	Graz University of Technology
Leonardo Rubio	University of Granada
Fernando Rubio Díez	Complutense University of Madrid
Ulrich Rueckert	University of Paderborn
Nicolás Ruiz Reyes	University of Jaen
Amparo Ruiz Sepúlveda	University of Malaga
Joseph Rynkiewicz	University of Paris I
Vladimir Râsvan	University of Craiova
Addisson Salazar	Universidad Politecnica de Valencia
Sancho Salcedo-Sanz	University of Alcalá
Albert Samà	Polytechnic University of Catalonia
Miguel A. Sanchez	Pontifical University of Salamanca
Francisco Sandoval	University of Malaga
Jose Santos	University of A Coruña
J.A. Seoane Fernández	University of A Coruña
Eduardo Serrano	Autonomous University of Madrid
Olli Simula	Helsinki University of Technology
Evgeny Skvortsov	Simon Fraser University
Sergio Solinas	Università degli studi di Pavia
Jordi Solé-Casals	Universitat de Vic
Adrian Stoica	Polytechnic University of Bucharest
José Luis Subirats	University of Malaga
Peter Szolgay	Pazmany Peter Catholic University
Javier Sánchez-Monedero	University of Cordoba
Ana Maria Tomé	Universidade de Aveiro
Carme Torras	Polytechnic University of Catalonia
Claude Touzet	Université de Provence
Gracián Triviño	University of Malaga

Ricardo Téllez	Pal Robotics
Raquel Ureña	University of Granada
Olga Valenzuela	University of Granada
Germano Vallesi	Università Politecnica delle Marche - Ancona
Agustín Valverde	University of Malaga
Pablo Varona	Autonomous University of Madrid
M.A. Veganzones	University of the Basque Country
Sergio Velastín	Kingston University
Marley Vellasco	PUC-Rio
Alfredo Vellido	Polytechnic University of Catalonia
Francisco Veredas	University of Malaga
Michel Verleysen	Université catholique de Louvain
Bart Wyns	Ghent University
Vicente Zarzoso	University of Nice Sophia Antipolis
Carolina Zato	University of Salamanca
Ahmed Zobaa	University of Exeter

IWANN 2011 Invited Speakers

Hani Hagrais	The Computational Intelligence Centre School of Computer Science and Electronic Engineering, University of Essex, UK
Francisco Herrera	Head of Research Group SCI2S (Soft Computing and Intelligent Information Systems), Department of Computer Science and Artificial Intelligence, University of Granada, Spain
Tom Heskes	Head of Machine Learning Group, Intelligent Systems Institute for Computing and Information Sciences (iCIS) Faculty of Science Radboud University Nijmegen, The Netherlands

IWANN 2011 Special Sessions Organizers

New Applications of Brain–Computer Interfaces

Francisco Pelayo	University of Granada
M.A. López Gordo	University of Granada
Ricardo Ron	University of Malaga

Optimization Algorithms in Graphic Processing Units

Antonio Mora	University of Granada
Maribel García-Arenas	University of Granada
Pedro Castillo	University of Granada

Computing Languages with Bio-inspired Devices

M. D. Jimenez-Lopez	University of Rovira i Virgili
A. Ortega De La Puente	Autonomous University of Madrid

Computational Intelligence in Multimedia

Adriana Dapena	University of A Coruña
Julio Bregáins	University of A Coruña
Nicolás Guil	University of Malaga

Biologically Plausible Spiking Neural Processing

Eduardo Ros	University of Granada
Richard R. Carrillo	University of Almeria

Video and Image Processing

Enrique Domínguez	University of Malaga
José García	University of Alicante

Hybrid Artificial Neural Networks: Models, Algorithms and Data

Cesar Hervás	University of Cordoba
Pedro Antonio Gutiérrez	University of Crdoba

Advances in Machine Learning for Bioinformatics and Computational Biomedicine

Paulo J.L. Lisboa	Liverpool John Moores University
Alfredo Vellido	Polytechnic University of Catalonia
Leonardo Franco	University of Malaga

Biometric Systems for Human–Machine Interaction

Alexandra Psarrou	University of Westminster
Anastassia Angelopoulou	University of Westminster
C.M. Travieso-Gonzlez	University of Las Palmas de Gran Canaria
Jordi Solé-Casals	University of Vic

Data Mining in Biomedicine

Julián Dorado	University of A Coruña
Juan R. Rabuñal	University of A Coruña
Alejandro Pazos	University of A Coruña

Bio-inspired Combinatorial Optimization

Carlos Cotta Porras	University of Malaga
Antonio J. Fernández Leiva	University of Malaga

Applying Evolutionary Computation and Nature-Inspired Algorithms to Formal Methods

Ismael Rodríguez	Complutense University of Madrid
------------------	----------------------------------

Recent Advances on Fuzzy Logic and Soft Computing Applications

Inma P. Cabrera	University of Malaga
Pablo Cordero	University of Malaga
Manuel Ojeda-Aciego	University of Malaga

New Advances in Theory and Applications of ICA-Based Algorithms

Addison Salazar	Polytechnic University of Valencia
Luis Vergara	Polytechnic University of Valencia

Biological and Bio-inspired Dynamical Systems

Vladimir Rasvan	University of Craiova
Daniela Danciu	University of Craiova

Interactive and Cognitive Environments

Andreu Catalá	Polytechnic University of Catalonia
Cecilio Angulo	Polytechnic University of Catalonia

Table of Contents – Part I

Mathematical and Theoretical Methods in Computational Intelligence

Gaze Gesture Recognition with Hierarchical Temporal Memory Networks	1
<i>David Rozado, Francisco B. Rodriguez, and Pablo Varona</i>	
Feature Selection for Multi-label Classification Problems	9
<i>Gauthier Doquire and Michel Verleysen</i>	
A Novel Grouping Heuristic Algorithm for the Switch Location Problem Based on a Hybrid Dual Harmony Search Technique	17
<i>Sergio Gil-Lopez, Itziar Landa-Torres, Javier Del Ser, Sancho Salcedo-Sanz, Diana Manjarres, and Jose A. Portilla-Figueras</i>	
Optimal Evolutionary Wind Turbine Placement in Wind Farms Considering New Models of Shape, Orography and Wind Speed Simulation	25
<i>B. Saavedra-Moreno, S. Salcedo-Sanz, A. Paniagua-Tineo, J. Gascón-Moreno, and J.A. Portilla-Figueras</i>	
Multi-Valued Neurons: Hebbian and Error-Correction Learning	33
<i>Igor Aizenberg</i>	
Multi-label Testing for CO ² RBFN: A First Approach to the Problem Transformation Methodology for Multi-label Classification	41
<i>A.J. Rivera, F. Charte, M.D. Pérez-Godoy, and María Jose del Jesus</i>	
Single Neuron Transient Activity Detection by Means of Tomography	49
<i>Carlos Aguirre, Pedro Pascual, Doris Campos, and Eduardo Serrano</i>	
Estimate of a Probability Density Function through Neural Networks	57
<i>Leonardo Reyneri, Valentina Colla, and Marco Vannucci</i>	

Learning and Adaptation

A Neural Fuzzy Inference Based Adaptive Controller Using Learning Process for Nonholonomic Robots	65
<i>Ting Wang, Fabien Gautero, Christophe Sabourin, and Kurosh Madani</i>	

A Multi-objective Evolutionary Algorithm for Network Intrusion Detection Systems	73
<i>J. Gómez, C. Gil, R. Baños, A.L. Márquez, F.G. Montoya, and M.G. Montoya</i>	
A Cognitive Approach for Robots' Vision Using Unsupervised Learning and Visual Saliency	81
<i>Dominik M. Ramík, Christophe Sabourin, and Kurosh Madani</i>	
Fusing Heterogeneous Data Sources Considering a Set of Equivalence Constraints	89
<i>Manuel Martín-Merino</i>	
A Novel Heuristic for Building Reduced-Set SVMs Using the Self-Organizing Map	97
<i>Ajalmar R. Rocha Neto and Guilherme A. Barreto</i>	
An Additive Decision Rules Classifier for Network Intrusion Detection	105
<i>Tommaso Pani and Francisco de Toro</i>	
Multi-modal Opponent Behaviour Prognosis in E-Negotiations	113
<i>Ioannis Papaioannou, Ioanna Roussaki, and Miltiades Anagnostou</i>	
Bio-inspired Systems and Neuro-engineering	
An AER to CAN Bridge for Spike-Based Robot Control	124
<i>M. Domínguez-Morales, A. Jimenez-Fernandez, R. Paz, A. Linares-Barranco, D. Cascado, J.L. Coronado, J.L. Muñoz, and G. Jimenez</i>	
Neuromorphic Real-Time Objects Tracking Using Address Event Representation and Silicon Retina	133
<i>F. Gómez- Rodríguez, L. Miró-Amarante, M. Rivas, G. Jimenez, and F. Diaz-del-Rio</i>	
Performance Study of Software AER-Based Convolutions on a Parallel Supercomputer	141
<i>Rafael J. Montero-Gonzalez, Arturo Morgado-Estevez, Alejandro Linares-Barranco, Bernabe Linares-Barranco, Fernando Perez-Peña, Jose Antonio Perez-Carrasco, and Angel Jimenez-Fernandez</i>	
Frequency Analysis of a 64x64 Pixel Retinomorphic System with AER Output to Estimate the Limits to Apply onto Specific Mechanical Environment	149
<i>Fernando Perez-Peña, Arturo Morgado-Estevez, Alejandro Linares-Barranco, Gabriel Jimenez-Moreno, Jose Maria Rodriguez-Corral, and Rafael J. Montero-Gonzalez</i>	

An AER Spike-Processing Filter Simulator and Automatic VHDL Generator Based on Cellular Automata	157
<i>Manuel Rivas-Perez, A. Linares-Barranco, Francisco Gomez-Rodriguez, A. Morgado, A. Civit, and G. Jimenez</i>	
A Biologically Inspired Neural Network for Autonomous Underwater Vehicles	166
<i>Francisco García-Córdova and Antonio Guerrero-González</i>	
Hybrid Intelligent Systems	
A Preliminary Study on the Use of Fuzzy Rough Set Based Feature Selection for Improving Evolutionary Instance Selection Algorithms	174
<i>Joaquín Derrac, Chris Cornelis, Salvador García, and Francisco Herrera</i>	
Forecasting Based on Short Time Series Using ANNs and Grey Theory – Some Basic Comparisons	183
<i>Jelena Milojković, Vančo Litovski, Octavio Nieto-Taladriz, and Slobodan Bojanić</i>	
Short-Term Wind Power Forecast Based on Cluster Analysis and Artificial Neural Networks	191
<i>Javier Lorenzo, Juan Méndez, Modesto Castrillón, and Daniel Hernández</i>	
Back Propagation with Balanced MSE Cost Function and Nearest Neighbor Editing for Handling Class Overlap and Class Imbalance	199
<i>R. Alejo, J.M. Sotoca, V. García, and R.M. Valdovinos</i>	
Combination of GA and ANN to High Accuracy of Polarimetric SAR Data Classification	207
<i>Ataollah Haddadi G. and Mahmoodreza Sahebi</i>	
Gradient Descent Optimization for Routing in Multistage Interconnection Networks	215
<i>Mehran Ghaziasgar and Armin Tavakoli Naeini</i>	
The Command Control of a Two-Degree-of-Freedom Platform by Hand Gesture Moment Invariants	223
<i>Chih-Lyang Hwang and Chen-Han Yang</i>	
Network Intrusion Prevention by Using Hierarchical Self-Organizing Maps and Probability-Based Labeling	232
<i>Andres Ortiz, Julio Ortega, Antonio F. Díaz, and Alberto Prieto</i>	

Applications of Computational Intelligence

Human/Robot Interface for Voice Teleoperation of a Robotic Platform	240
<i>L. Gallardo-Estrella and A. Poncela</i>	
Graph Laplacian for Semi-supervised Feature Selection in Regression Problems	248
<i>Gauthier Doquire and Michel Verleysen</i>	
Detection of Transients in Steel Casting through Standard and AI-Based Techniques	256
<i>Valentina Colla, Marco Vannucci, Nicola Matarese, Gerard Stephens, Marco Pianezzola, Izaskun Alonso, Torsten Lamp, Juan Palacios, and Siegfried Schiewe</i>	
Oesophageal Voice Harmonic to Noise Ratio Enhancement over UMTS Networks Using Kalman-EM	265
<i>Marouen Azzouz, Begoña García Zapirain, Ibon Ruiz, and Amaia Méndez</i>	
Study of Various Neural Networks to Improve the Defuzzification of Fuzzy Clustering Algorithms for ROIs Detection in Lung CTs	273
<i>Alberto Rey, Alfonso Castro, and Bernardino Arcay</i>	
Differential Evolution Optimization of 3D Topological Active Volumes	282
<i>J. Novo, J. Santos, and M.G. Penedo</i>	
Genetic Algorithms Applied to the Design of 3D Photonic Crystals	291
<i>Agustín Morgado-León, Alejandro Escuin, Elisa Guerrero, Andrés Yáñez, Pedro L. Galindo, and Lorenzo Sanchis</i>	
Sliding Empirical Mode Decomposition for On-line Analysis of Biomedical Time Series	299
<i>A. Zeiler, R. Faltermeier, A.M. Tomé, C. Puntonet, A. Brawanski, and E.W. Lang</i>	
Suitability of Artificial Neural Networks for Designing LoC Circuits	307
<i>David Moreno, Sandra Gómez, and Juan Castellanos</i>	
Aeration Control and Parameter Soft Estimation for a Wastewater Treatment Plant Using a Neurogenetic Design	315
<i>Javier Fernandez de Canete, Pablo del Saz-Orozco, and Inmaculada Garcia-Moral</i>	

Pulse Component Modification Detection in Spino Cerebellar Ataxia 2 Using ICA	323
<i>Rodolfo V. García, Fernando Rojas, Jesús González, Luis Velázquez, Roberto Rodríguez, Roberto Becerra, and Olga Valenzuela</i>	
Early Pigmentary Retinosis Diagnostic Based on Classification Trees ...	329
<i>Vivian Sistachs Vega, Gonzalo Joya Caparrós, and Miguel A. Díaz Martínez</i>	

New Applications of Brain-Computer Interfaces

Audio-Cued SMR Brain-Computer Interface to Drive a Virtual Wheelchair	337
<i>Francisco Velasco-Álvarez, Ricardo Ron-Angevin, Leandro da Silva-Sauer, Salvador Sancha-Ros, and María José Blanca-Mena</i>	
A Domotic Control System Using Brain-Computer Interface (BCI)	345
<i>Rebeca Corralejo, Roberto Hornero, and Daniel Álvarez</i>	
A Dictionary-Driven SSVEP Speller with a Modified Graphical User Interface	353
<i>Ivan Volosyak, Anton Moor, and Axel Gräser</i>	
Non-invasive Brain-Computer Interfaces: Enhanced Gaming and Robotic Control	362
<i>Reinhold Scherer, Elisabeth C.V. Friedrich, Brendan Allison, Markus Pröll, Mike Chung, Willy Cheung, Rajesh P.N. Rao, and Christa Neuper</i>	
An EEG-Based Design for the Online Detection of Movement Intention	370
<i>Jaime Ibáñez, J. Ignacio Serrano, M. Dolores del Castillo, Luis Barrios, Juan Álvaro Gallego, and Eduardo Rocon</i>	
Auditory Brain-Computer Interfaces for Complete Locked-In Patients ..	378
<i>M.A. Lopez-Gordo, Ricardo Ron-Angevin, and Francisco Pelayo Valle</i>	
Brain-Computer Interface: Generic Control Interface for Social Interaction Applications	386
<i>C. Hintermüller, C. Guger, and G. Edlinger</i>	

Optimization Algorithms in Graphic Processing Units

Variable Selection in a GPU Cluster Using Delta Test	393
<i>A. Guillén, M. van Heeswijk, D. Sovilj, M.G. Arenas, L.J. Herrera, H. Pomares, and I. Rojas</i>	

Towards ParadisEO-MO-GPU: A Framework for GPU-Based Local Search Metaheuristics	401
<i>N. Melab, T.-V. Luong, K. Boufaras, and E.-G. Talbi</i>	
Efficient Simulation of Spatio-temporal Dynamics in Ultrasonic Resonators	409
<i>Pedro Alonso-Jordá, Jesús Peinado-Pinilla, Isabel Pérez-Arjona, and Victor J. Sánchez-Morcillo</i>	
GPU Implementation of a Bio-inspired Vision Model	417
<i>Raquel Ureña, Christian Morillas, Samuel Romero, and Francisco J. Pelayo</i>	
Bipartite Graph Matching on GPU over Complete or Local Grid Neighborhoods	425
<i>Cristina Nader Vasconcelos and Bodo Rosenhahn</i>	
GPU Computation in Bioinspired Algorithms: A Review	433
<i>M.G. Arenas, A.M. Mora, G. Romero, and P.A. Castillo</i>	
Computing Languages with Bio-inspired Devices and Multi-Agent Systems	
About Complete Obligatory Hybrid Networks of Evolutionary Processors without Substitution	441
<i>Artiom Alhazov, Gemma Bel-Enguix, Alexander Krassovitskiy, and Yuri Rogozhin</i>	
Chemical Signaling as a Useful Metaphor for Resource Management	449
<i>Evgeny Skvortsov, Nima Kaviani, and Veronica Dahl</i>	
Distributed Simulation of P Systems by Means of Map-Reduce: First Steps with Hadoop and P-Lingua	457
<i>L. Diez Dolinski, R. Núñez Hervás, M. Cruz Echeandía, and A. Ortega</i>	
Hierarchy Results on Stateless Multicounter $5' \rightarrow 3'$ Watson-Crick Automata	465
<i>Benedek Nagy, László Hegedüs, and Ömer Eğecioğlu</i>	
Towards a Bio-computational Model of Natural Language Learning	473
<i>Leonor Becerra-Bonache</i>	
Computing Languages with Bio-inspired Devices and Multi-Agent Systems	481
<i>M. Dolores Jiménez-López</i>	

Computational Intelligence in Multimedia Processing

A Novel Strategy for Improving the Quality of Embedded Zerotree Wavelet Images Transmitted over Alamouti Coding Systems	489
<i>Josmary Labrador, Paula M. Castro, Héctor J. Pérez-Iglesias, and Adriana Dapena</i>	
Applying Data Mining Techniques in a Wyner-Ziv to H.264 Video Transcoder	497
<i>José Luis Martínez, Alberto Corrales-García, Pedro Cuenca, and Francisco José Quiles</i>	
On the Use of Genetic Algorithms to Improve Wavelet Sign Coding Performance	505
<i>Ricardo García, Otoniel López, Antonio Martí, and Manuel P. Malumbres</i>	
Kernel-Based Object Tracking Using a Simple Fuzzy Color Histogram	513
<i>Juan Villalba Espinosa, José María González Linares, Julián Ramos Cózar, and Nicolás Guil Mata</i>	
Computational Intelligence in Multimedia Processing	520
<i>Nicolás Guil, Julio C. Bregáins, and Adriana Dapena</i>	

Biologically Plausible Spiking Neural Processing

Isometric Coding of Spiking Haptic Signals by Peripheral Somatosensory Neurons	528
<i>Romain Brasselet, Roland S. Johansson, and Angelo Arleo</i>	
Context Separability Mediated by the Granular Layer in a Spiking Cerebellum Model for Robot Control	537
<i>Niceto R. Luque, Jesús A. Garrido, Richard R. Carrillo, and Eduardo Ros</i>	
Realistic Modeling of Large-Scale Networks: Spatio-temporal Dynamics and Long-Term Synaptic Plasticity in the Cerebellum	547
<i>Egidio D'Angelo and Sergio Solinas</i>	
Event and Time Driven Hybrid Simulation of Spiking Neural Networks	554
<i>Jesus A. Garrido, Richard R. Carrillo, Niceto R. Luque, and Eduardo Ros</i>	
Author Index	563

Table of Contents – Part II

Video and Image Processing

Lossy Image Compression Using a GHSOM	1
<i>E.J. Palomo, E. Domínguez, R.M. Luque, and J. Muñoz</i>	
Visual Features Extraction Based Egemotion Calculation from a Infrared Time-of-Flight Camera	9
<i>Diego Viejo, Jose Garcia, and Miguel Cazorla</i>	
Feature Weighting in Competitive Learning for Multiple Object Tracking in Video Sequences	17
<i>R.M. Luque, J.M. Ortiz-de-Lazcano-Lobato, Ezequiel López-Rubio, E. Domínguez, and E.J. Palomo</i>	
The Segmentation of Different Skin Colors Using the Combination of Graph Cuts and Probability Neural Network	25
<i>Chih-Lyang Hwang and Kai-Di Lu</i>	
Reduction of JPEG Compression Artifacts by Kernel Regression and Probabilistic Self-Organizing Maps	34
<i>María Nieves Florentín-Núñez, Ezequiel López-Rubio, and Francisco Javier López-Rubio</i>	
An Unsupervised Method for Active Region Extraction in Sports Videos	42
<i>Markos Mentzelopoulos, Alexandra Psarrou, and Anastassia Angelopoulou</i>	
6DoF Egemotion Computing Using 3D GNG-Based Reconstruction	50
<i>Diego Viejo, Jose Garcia, and Miguel Cazorla</i>	
Fast Image Representation with GPU-Based Growing Neural Gas	58
<i>José García-Rodríguez, Anastassia Angelopoulou, Vicente Morell, Sergio Orts, Alexandra Psarrou, and Juan Manuel García-Chamizo</i>	
Texture and Color Analysis for the Automatic Classification of the Eye Lipid Layer	66
<i>L. Ramos, M. Penas, B. Remeseiro, A. Mosquera, N. Barreira, and E. Yebra-Pimentel</i>	
Quantitative Study and Monitoring of the Growth of Lung Cancer Nodule Using an X-Ray Computed Tomography Image Processing Tool	74
<i>José Luis García Arroyo, Begoña García Zapirain, and Amaia Méndez Zorrilla</i>	

A Geometrical Method of Diffuse and Specular Image Components Separation 83
Ramón Moreno, Manuel Graña, and Alicia d’Anjou

Optical Flow Reliability Model Approximated with RBF 90
Agis Rodrigo, Díaz Javier, Ortigosa Pilar, Guzmán Pablo, and Ros Eduardo

Video and Image Processing with Self-organizing Neural Networks 98
José García-Rodríguez, Enrique Domínguez, Anastassia Angelopoulou, Alexandra Psarrou, Francisco José Mora-Gimeno, Sergio Orts, and Juan Manuel García-Chamizo

Hybrid Artificial Neural Networks: Models, Algorithms and Data

Parallelism in Binary Hopfield Networks 105
José Muñoz-Pérez, Amparo Ruiz-Sepúlveda, and Rafaela Benítez-Rochel

Multi-parametric Gaussian Kernel Function Optimization for ϵ -SVMr Using a Genetic Algorithm 113
J. Gascón-Moreno, E.G. Ortiz-García, S. Salcedo-Sanz, A. Paniagua-Tineo, B. Saavedra-Moreno, and J.A. Portilla-Figueras

Face Recognition System in a Dynamical Environment 121
Aldo Franco Dragoni, Germano Vallesi, and Paola Baldassarri

Memetic Pareto Differential Evolutionary Neural Network for Donor-Recipient Matching in Liver Transplantation 129
M. Cruz-Ramírez, C. Hervás-Martínez, P.A. Gutiérrez, J. Briceño, and M. de la Mata

Studying the Hybridization of Artificial Neural Networks in HECIC 137
José del Campo-Ávila, Gonzalo Ramos-Jiménez, Jesús Pérez-García, and Rafael Morales-Bueno

Processing Acyclic Data Structures Using Modified Self-Organizing Maps 145
Gabriela Andrejková and Jozef Oravec

On the Performance of the μ -GA Extreme Learning Machines in Regression Problems	153
<i>A. Paniagua-Tineo, S. Salcedo-Sanz, E.G. Ortiz-García, J. Gascón-Moreno, B. Saavedra-Moreno, and J.A. Portilla-Figueras</i>	
A Hybrid Evolutionary Approach to Obtain Better Quality Classifiers	161
<i>David Becerra-Alonso, Mariano Carbonero-Ruz, Francisco José Martínez-Estudillo, and Alfonso Carlos Martínez-Estudillo</i>	
Neural Network Ensembles with Missing Data Processing and Data Fusion Capacities: Applications in Medicine and in the Environment ...	169
<i>Patricio García Báez, Carmen Paz Suárez Araujo, and Pablo Fernández López</i>	
Hybrid Artificial Neural Networks: Models, Algorithms and Data	177
<i>P.A. Gutiérrez and C. Hervás-Martínez</i>	

Advances in Machine Learning for Bioinformatics and Computational Biomedicine

Automatic Recognition of Daily Living Activities Based on a Hierarchical Classifier	185
<i>Oresti Banos, Miguel Damas, Hector Pomares, and Ignacio Rojas</i>	
Prediction of Functional Associations between Proteins by Means of a Cost-Sensitive Artificial Neural Network	194
<i>J.P. Florido, H. Pomares, I. Rojas, J.M. Urquiza, and F. Ortuño</i>	
Hybrid (Generalization-Correlation) Method for Feature Selection in High Dimensional DNA Microarray Prediction Problems	202
<i>Yasel Couce, Leonardo Franco, Daniel Urda, José L. Subirats, and José M. Jerez</i>	
Model Selection with PLANN-CR-ARD	210
<i>Corneliu T.C. Arsene, Paulo J. Lisboa, and Elia Biganzoli</i>	

Biometric Systems for Human-Machine Interaction

Gender Recognition Using PCA and DCT of Face Images	220
<i>Ondrej Smírg, Jan Mikulka, Marcos Faundez-Zanuy, Marco Grassi, and Jiri Mekyska</i>	
Efficient Face Recognition Fusing Dynamic Morphological Quotient Image with Local Binary Pattern	228
<i>Hong Pan, Siyu Xia, Lizuo Jin, and Liangzheng Xia</i>	

A Growing Neural Gas Algorithm with Applications in Hand Modelling and Tracking 236
Anastassia Angelopoulou, Alexandra Psarrou, and José García Rodríguez

Object Representation with Self-Organising Networks 244
Anastassia Angelopoulou, Alexandra Psarrou, and José García Rodríguez

Data Mining in Biomedicine

SNP-Schizo: A Web Tool for Schizophrenia SNP Sequence Classification 252
Vanessa Aguiar-Pulido, José A. Seoane, Cristian R. Munteanu, and Alejandro Pazos

MicroRNA Microarray Data Analysis in Colon Cancer: Effects of Normalization 260
Guillermo H. López-Campos, Alejandro Romera-López, Fernando Martín-Sánchez, Eduardo Díaz-Rubio, Victoria López-Alonso, and Beatriz Pérez-Villamil

Automatic Handling of Tissue Microarray Cores in High-Dimensional Microscopy Images 268
G. Bueno, M. Fernández, O. Déniz, and M. García-Rojo

Visual Mining of Epidemic Networks 276
Stéphan Cléménçon, Hector De Arazoza, Fabrice Rossi, and Viet-Chi Tran

Bio-inspired Combinatorial Optimization

Towards User-Centric Memetic Algorithms: Experiences with the TSP 284
Ana Reyes Badillo, Carlos Cotta, and Antonio J. Fernández-Leiva

A Multi-objective Approach for the 2D Guillotine Cutting Stock Problem 292
Jesica de Armas, Gara Miranda, and Coromoto León

Ant Colony Optimization for Water Distribution Network Design: A Comparative Study 300
C. Gil, R. Baños, J. Ortega, A.L. Márquez, A. Fernández, and M.G. Montoya

A Preliminary Analysis and Simulation of Load Balancing Techniques Applied to Parallel Genetic Programming 308
F. Fernández de Vega, J.G. Abengózar Sánchez, and C. Cotta

A Study of Parallel Approaches in MOACOs for Solving the Bicriteria TSP	316
<i>A.M. Mora, J.J. Merelo, P.A. Castillo, M.G. Arenas, P. García-Sánchez, J.L.J. Laredo, and G. Romero</i>	
Optimizing Strategy Parameters in a Game Bot	325
<i>A. Fernández-Ares, A.M. Mora, J.J. Merelo, P. García-Sánchez, and C.M. Fernandes</i>	
Implementation Matters: Programming Best Practices for Evolutionary Algorithms	333
<i>J.J. Merelo, G. Romero, M.G. Arenas, P.A. Castillo, A.M. Mora, and J.L.J. Laredo</i>	
Online vs Offline ANOVA Use on Evolutionary Algorithms	341
<i>G. Romero, M.G. Arenas, P.A. Castillo, J.J. Merelo, and A.M. Mora</i>	
Bio-inspired Combinatorial Optimization: Notes on Reactive and Proactive Interaction	348
<i>Carlos Cotta and Antonio J. Fernández-Leiva</i>	
Applying Evolutionary Computation and Nature-inspired Algorithms to Formal Methods	
A Preliminary General Testing Method Based on Genetic Algorithms	356
<i>Luis M. Alonso, Pablo Rabanal, and Ismael Rodríguez</i>	
Tackling the Static RWA Problem by Using a Multiobjective Artificial Bee Colony Algorithm	364
<i>Álvaro Rubio-Largo, Miguel A. Vega-Rodríguez, Juan A. Gómez-Pulido, and Juan M. Sánchez-Pérez</i>	
Applying a Multiobjective Gravitational Search Algorithm (MO-GSA) to Discover Motifs	372
<i>David L. González-Álvarez, Miguel A. Vega-Rodríguez, Juan A. Gómez-Pulido, and Juan M. Sánchez-Pérez</i>	
Looking for a Cheaper ROSA	380
<i>Fernando L. Pelayo, Fernando Cuartero, and Diego Cazorla</i>	
A Parallel Skeleton for Genetic Algorithms	388
<i>Alberto de la Encina, Mercedes Hidalgo-Herrero, Pablo Rabanal, and Fernando Rubio</i>	
A Case Study on the Use of Genetic Algorithms to Generate Test Cases for Temporal Systems	396
<i>Karnig Derderian, Mercedes G. Merayo, Robert M. Hierons, and Manuel Núñez</i>	

Experimental Comparison of Different Techniques to Generate Adaptive Sequences 404
Carlos Molinero, Manuel Núñez, and Robert M. Hierons

Recent Advances on Fuzzy Logic and Soft Computing Applications

An Efficient Algorithm for Reasoning about Fuzzy Functional Dependencies 412
P. Cordero, M. Enciso, A. Mora, I. Pérez de Guzmán, and J.M. Rodríguez-Jiménez

A Sound Semantics for a Similarity-Based Logic Programming Language 421
Pascual Julián-Iranzo and Clemente Rubio-Manzano

A Static Preprocess for Improving Fuzzy Thresholded Tabulation 429
P. Julián, J. Medina, P.J. Morcillo, G. Moreno, and M. Ojeda-Aciego

Non-deterministic Algebraic Structures for Soft Computing 437
I.P. Cabrera, P. Cordero, and M. Ojeda-Aciego

Fuzzy Computed Answers Collecting Proof Information 445
Pedro J. Morcillo, Ginés Moreno, Jaime Penabad, and Carlos Vázquez

Implication Triples Versus Adjoint Triples 453
M^a Eugenia Cornejo, Jesús Medina, and Eloisa Ramírez

Confidence-Based Reasoning with Local Temporal Formal Contexts 461
Gonzalo A. Aranda-Corral, Joaquín Borrego Díaz, and Juan Galán Páez

New Advances in Theory and Applications of ICA-Based Algorithms

Application of Independent Component Analysis for Evaluation of Ashlar Masonry Walls 469
Addisson Salazar, Gonzalo Safont, and Luis Vergara

Fast Independent Component Analysis Using a New Property 477
Rubén Martín-Clemente, Susana Hornillo-Mellado, and José Luis Camargo-Olivares

Using Particle Swarm Optimization for Minimizing Mutual Information in Independent Component Analysis 484
Jorge Igual, Jehad Ababneh, Raul Llinares, and Carmen Igual

Regularized Active Set Least Squares Algorithm for Nonnegative Matrix Factorization in Application to Raman Spectra Separation	492
<i>Rafal Zdunek</i>	
A Decision-Aided Strategy for Enhancing Transmissions in Wireless OSTBC-Based Systems	500
<i>Tiago M. Fernández-Caramés, Adriana Dapena, José A. García-Naya, and Miguel González-López</i>	
Nonlinear Prediction Based on Independent Component Analysis Mixture Modelling	508
<i>Gonzalo Safont, Addison Salazar, and Luis Vergara</i>	
Biological and Bio-inspired Dynamical Systems	
Robustness of the “Hopfield Estimator” for Identification of Dynamical Systems	516
<i>Miguel Atencia, Gonzalo Joya, and Francisco Sandoval</i>	
Modeling Detection of HIV in Cuba	524
<i>Héctor de Arazoza, Rachid Lounes, Andres Sánchez, Jorge Barrios, and Ying-Hen Hsieh</i>	
Flexible Entrainment in a Bio-inspired Modular Oscillator for Modular Robot Locomotion	532
<i>Fernando Herrero-Carrón, Francisco B. Rodríguez, and Pablo Varona</i>	
Dengue Model Described by Differential Inclusions	540
<i>Jorge Barrios, Alain Piétrus, Aymée Marrero, Héctor de Arazoza, and Gonzalo Joya</i>	
Simulating Building Blocks for Spikes Signals Processing	548
<i>A. Jimenez-Fernandez, M. Domínguez-Morales, E. Cerezuela-Escudero, R. Paz-Vicente, A. Linares-Barranco, and G. Jimenez</i>	
Description of a Fault Tolerance System Implemented in a Hardware Architecture with Self-adaptive Capabilities	557
<i>Javier Soto, Juan Manuel Moreno, and Joan Cabestany</i>	
Systems with Slope Restricted Nonlinearities and Neural Networks Dynamics	565
<i>Daniela Danciu and Vladimir Răsvan</i>	
Bio-inspired Systems. Several Equilibria. Qualitative Behavior	573
<i>Daniela Danciu</i>	

Interactive and Cognitive Environments

Biologically Inspired Path Execution Using SURF Flow in Robot Navigation 581
Xavier Perez-Sala, Cecilio Angulo, and Sergio Escalera

Equilibrium-Driven Adaptive Behavior Design 589
Paul Olivier and Juan Manuel Moreno Arostegui

Gait Identification by Using Spectrum Analysis on State Space Reconstruction 597
Albert Samà, Francisco J. Ruiz, Carlos Pérez, and Andreu Català

Aibo JukeBox A Robot Dance Interactive Experience..... 605
Cecilio Angulo, Joan Comas, and Diego Pardo

International Workshop of Intelligent Systems for Context-Based Information Fusion (ISCIF’11)

On Planning in Multi-agent Environment: Algorithm of Scene Reasoning from Incomplete Information..... 613
Tomasz Grzejszczak and Adam Galuszka

Research Opportunities in Contextualized Fusion Systems. The Harbor Surveillance Case 621
Jesus Garcia, José M. Molina, Tarunraj Singh, John Crassidis, and James Llinas

Multiagent-Based Middleware for the Agents’ Behavior Simulation 629
Elena García, Sara Rodríguez, Juan F. De Paz, and Juan M. Corchado

A Dynamic Context-Aware Architecture for Ambient Intelligence 637
José M. Fernández, Rubén Fuentes-Fernández, and Juan Pavón

Group Behavior Recognition in Context-Aware Systems..... 645
Alberto Pozo, Jesús Gracia, Miguel A. Patricio, and José M. Molina

Context-Awareness at the Service of Sensor Fusion Systems: Inverting the Usual Scheme 653
Enrique Martí, Jesús García, and Jose Manuel Molina

Improving a Telemonitoring System Based on Heterogeneous Sensor Networks 661
Ricardo S. Alonso, Dante I. Tapia, Javier Bajo, and Sara Rodríguez

Supporting System for Detecting Pathologies	669
<i>Carolina Zato, Juan F. De Paz, Fernando de la Prieta, and Beatriz Martín</i>	
An Ontological Approach for Context-Aware Reminders in Assisted Living Behavior Simulation	677
<i>Shumei Zhang, Paul McCullagh, Chris Nugent, Huiru Zheng, and Norman Black</i>	
Author Index	685

Gaze Gesture Recognition with Hierarchical Temporal Memory Networks

David Rozado, Francisco B. Rodriguez, and Pablo Varona

Grupo de Neurocomputación Biológica (GNB)

Dpto. de Ingeniería Informática, EPS, Universidad Autónoma de Madrid

Calle Francisco Tomás y Valiente, 11, 28049 - Madrid (Spain)

{david.rozado,f.rodriguez,pablo.varona}@uam.es

Abstract. Eye movements can be consciously controlled by humans to the extent of performing sequences of predefined movement patterns, or 'gaze gestures'. Gaze gestures can be tracked non-invasively employing a video-based eye tracking system. Gaze gestures hold great potential in the context of Human Computer Interaction as low-cost gaze trackers become more ubiquitous. In this work, we build an original set of 50 gaze gestures and evaluate the recognition performance of a Bayesian inference algorithm known as Hierarchical Temporal Memory, HTM. HTM uses a neocortically inspired hierarchical architecture and spatio-temporal coding to perform inference on multi-dimensional time series. Here, we show how an appropriate temporal codification is critical for good inference results. Our results highlight the potential of gaze gestures for the fields of accessibility and interaction with smartphones, projected displays and desktop computers.

Keywords: Neural Network Architecture, Soft Computing, Pattern Recognition, Time series analysis and prediction.

1 Introduction

The usage of predefined gestures in human-computer interaction, HCI, often employs the hands, head or mouse [1]. The arrival of smartphones and tabletop computers with often touch sensitive surfaces as their only input modality has prompted recent interest in the subject of gestures for HCI purposes. In this work, we explore the feasibility of using predefined, recognisable and repeatable paths of gaze movements, i.e. gaze gestures, as an input modality in HCI.

Video-based gaze tracking systems can determine where a user is looking at on a screen. Gaze tracking is a very convenient technology for pointing but problematic when trying to distinguish whether the user looks at an object to examine it or to interact with it. This is known as the Midas touch problem, and it highlights the need for additional gaze interaction methods beyond dwell time selections [2]. Gaze gestures hold great potential in HCI due to the fast nature of eye saccadic movements, and its robustness to inaccuracy problems, calibration shifts and the Midas problem. The main concern with gaze gestures is the accidental detection of a gaze gesture during normal gaze activity.

The use of gaze gestures in HCI is a relatively new concept and the amount of research done on it is rather limited [3]. Gaze gestures can be employed by people with severe disabilities, who use gaze as a mono-modal input in their HCI. Gaze gestures can also provide an additional input channel in multi-modal interaction paradigms providing a new venue of interaction with small screen devices such as smartphones or in scenarios where traditional interaction methods are out of reach such as media center devices or surgery rooms. In this work, we have created an in-house data set of 50 gaze gestures and used a neuroinspired bayesian pattern recognition paradigm known as Hierarchical Temporal Memory, HTM, to learn them. HTMs are appropriate for this problem due to their robustness to noise and ability to analyze patterns with a multi-dimensional temporal structure. The temporal structure of gaze gestures that unfolds over time requires an appropriate temporal codification for HTMs to properly perform inference, hence, we analyze the impact of different temporal codifications on performance. We also carry out a user study to elucidate time requirements, training constraints and specificity, learning effects, and users perceived satisfaction with gaze gestures.

2 Eye Tracking, Gaze Gestures and the HTM Formalism

Gaze tracking video-oculography determines a person Point of Regard or PoR (i.e. where a person is looking at) by gathering information from eye position and movements [4]. Infrared illumination is used to improve iris to pupil contrast and to create a reflection on the cornea, or glint. Due to the spherical shape of the eyeball, this glint remains stationary as the eye moves in its orbit and it is used as a reference point from which to estimate gaze direction. This is done by calculating the vector distance from the corneal reflection and the center of the pupil. Video-oculography is limited by some optical and anatomical constraints and as of yet its maximum accuracy is limited to about 0.5° .

We define a gaze gesture as an ordered sequence of gaze positions over time. Different conceptualizations of gaze gestures exist. Gaze gestures can be relative, i.e. they can be performed anywhere on the screen, or absolute, requiring the user to direct gaze to a sequence of absolute positions on the screen. Due to the limited accuracy of eye tracking technology, fine discrimination between close points on the screen is often not possible. This limitation and the discomfort that continuous micro-movements generate on users, advocates the merits of absolute gaze gestures. Here, we consider a modality of gaze gesture consisting on gliding the gaze along a predefined path on the screen using microsaccadic gaze movements and employing the cursor position as feedback to ensure that the path is followed correctly, Figure 1(a).

Hierarchical Temporal Memory, HTM, is a computational model of cortical micro-circuits based on Bayesian belief propagation. HTMs incorporate the hierarchical organization of the mammalian neocortex into their topology and use spatio-temporal codification to encapsulate the structure of problems' spaces [5]. Conceptually, HTM uses a set of nodes organized in a tree-like structure,

Figure 1(a), a generative model, Figure 1(b) and Bayesian belief propagation, Table 1. Each node contains a set of coincidence patterns or CPs: $c_1, c_2, \dots, c_n \in C$ and a set of Markov chains or MCs: $g_1, g_2, \dots, g_n \in G$. CPs represent co-occurrences of sequences from their children nodes. Each MC is defined as a subset of the set of CPs in a node. CP capture the spatial structure of nodes or sensors underneath in the hierarchy by representing vectorially the co-activation of MC in a node’s children. A MC activated in a parent node concurrently activates its constituent MC in the node’s children. The MCs capture the temporal structure of a set of CP, i.e., the likelihood of temporal transitions among them. The incoming vectors to an HTM node encapsulate the degree of certainty over the child MCs. With this information the node calculates its own degree of certainty over its CPs. Based on the history of messages received, it also computes a degree of certainty in each of its MCs. This information is then passed to the parent node. Feedback information from parent nodes toward children nodes takes place by parent nodes sending to children nodes their degree of certainty over the children node’s MCs, Table II

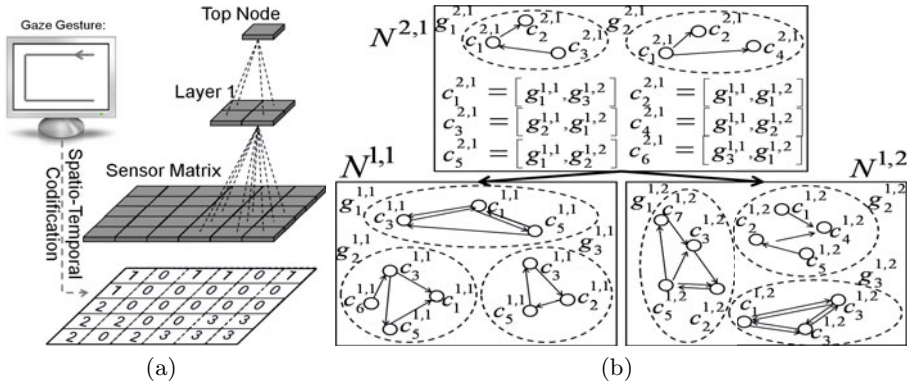


Fig. 1. In Panel *a*, the HTM topology employed during experiments. The bottom level nodes are fed with a spatio-temporal codification of a gaze gesture. Nodes in upper layers receive input vectors from the receptive field formed by its children nodes and emit output vectors encapsulating the spatio-temporal properties of their receptive field towards their parent node. Panel *b* illustrates the inner-workings of HTM nodes as a Generative Model. A simple two level hierarchy consisting of 3 nodes is shown. Each node contains a set of CPs, c 's and a set of MCs, g 's, defined over the set of CPs. A CP in a node represents a co-activation of a subset of MCs in its children nodes.

3 Experimental Setup

Gaze data acquisition was carried out using the ITU Gaze Tracker [4] software in a remote setup. We used a Sandberg webcam with no infrared filter, a 16mm lens and two infrared lamps. Image resolution was set to 640×480 pixels, and the frame rate was 30 fps. The distance from the eye to the camera was approximately 60 cm. A filter algorithm was employed on the raw gaze data to

Table 1. Belief propagation equations for HTM nodes. The reader is encouraged to take the *Node*^{2,1} from Fig. [1\(b\)](#) as reference. *Node*^{2,1} contains 6 CPs and two MCs. Each MC is composed of 3 CP. In this table c_i is i^{th} coincidence in node. g_r is r^{th} MC in node. $-e_t$ indicates the bottom up evidence at instant t . $-e_0^t$ indicates the evidence sequence from time $0 \dots t$. $+e$ stands for Top-down evidence. λ is the feed-forward output of the node. λ^{m_i} represents Feed-forward input to node from child node m_i . π is the feedback input to node. π^{m_i} is feedback output of node to child node m_i . y is the bottom-up likelihood over CPs in a node. α is a bottom-up state variable for the MCs in a node. β is a state that combines bottom-up and top-down evidence for MC in node. B_{c_i} represents belief in the i^{th} CP in a node.

Likelihood over CPs:	$y_t(i) = P(-e_t c_i(t)) \propto \prod_{j=1}^M \lambda_t^{m_j}(r_i^{m_j})$ where CP c_i is the co-occurrence of $r_j^{m_1}$ 'th MC from child 1, $r_i^{m_2}$ 'th MC from child 2, ..., and $r_i^{m_M}$ 'th MC from child M.
Feed-forward likelihood of MCs	$\lambda_t(g_r) = P(-e_0^t g_r(t)) \propto \sum_{c_i(t) \in C^k} \alpha_t(c_i, g_r)$ $\alpha(c_i, g_r) = P(-e_t c_i(t)) \sum_{c_j(t-1) \in C^k} P(c_i(t) c_j(t-1), g_r) \alpha_{t-1}(c_j, g_r)$ $\alpha_0(c_i, g_r) = P(-e_0 c_i(t=0)) P(c_i(t=0) g_r)$
Belief distribution over CP	$Bel_t(c_i) \propto \sum_{g_r \in C^k} \beta_t(c_i, g_r)$ $\beta_t(c_i, g_r) = P(-e_t c_i(t)) \sum_{c_j(t-1) \in C^k} P(c_i(t) c_j(t-1), g_r) \beta_{t-1}(c_j, g_r)$ $\beta_0(c_i, g_r) = P(-e_0 c_i(t=0)) P(c_i g_r) \pi_0(g_r)$
Mesassage to be sent to children nodes	$\pi^{m_i}(g_r) \propto \sum_i I(c_i) Bel(c_i)$, where $I(c_i) = \begin{cases} 1 & \text{if } g_r^{m_i} \text{ is even} \\ 0 & \text{otherwise} \end{cases}$

smooth out microsaccades and involuntary jerks while maintaining an acceptable real-time latency. The gaze accuracy achieved with the setup was about 1.5° .

Gaze gestures data for training the HTM networks was generated with no black or special purpose background, nor markers to attract or better position the gaze in specific coordinates of the screen. The data set was designed trying to minimize the spatial overlap by maximizing orthogonality. Yet, some gestures were intentionally designed with complete overlap, but different start/end points, to illustrate the importance of temporal coding. 30 instances for each of the 50 categories in the data set, Figure [2\(a\)](#), were generated by a user experienced with eye-tracking. Test data to measure the performance of HTM inference was gathered by 5 test subjects. All of them were male, regular computer users, not familiar with eye tracking and with ages ranging from 25 to 59 years. Participants were instructed to perform the data set in Figure [2\(a\)](#) as fast and accurately as possible. One participant with no prior experience on eye-tracking repeated the task over 5 blocks to study learning effects. After completing the experiments, participants filled out a short questionnaire rating speed, accuracy, fatigue and ease of use of their experience with gaze gesture as a form of HCI.

We used Numenta's Nupic package (v1.7.1) [5](#) to construct and train the HTM networks. The raw gaze data, consisting on a time series of x, y coordinates, was transformed into a $m \times n$ matrix representing the screen on which the gaze gesture was performed. The matrix was initially filled with 0s but those areas of the screen/matrix scanned by the gaze during the performance of a gaze gesture were assigned a number codifying its temporal structure, Figure [1\(a\)](#).

We used three types of temporal codification: **No temporal codification**: by storing in the corresponding matrix elements only the spatial representation of a gaze gesture, i.e. a 1, **Temporal codification in seconds** by storing in the corresponding matrix element the second in which the gaze passed over that particular area of the screen and **Three Temporal Stages codification**: by dividing the total time employed during performance of a gesture in 3 slices (beginning, middle, end) and assign correspondingly to the matrix the numbers 1, 2 3 depending when the gaze hovered over the corresponding area.

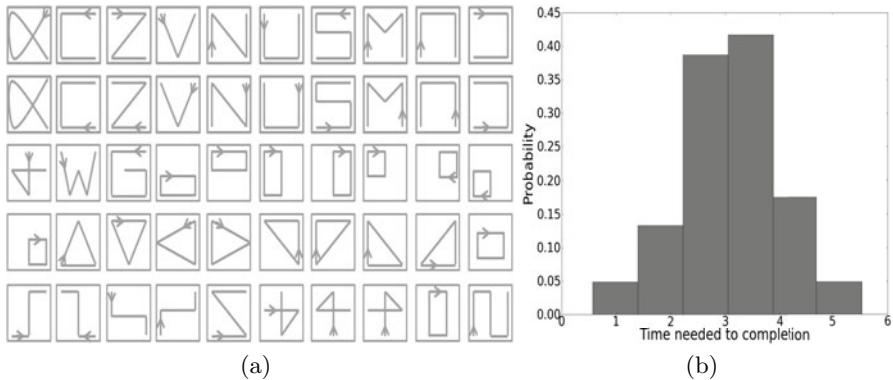


Fig. 2. Panel *a* shows the set of gestures employed in the user study to evaluate performance. Arrowheads indicate the start of a gesture. Panel *b* shows the histogram of times per gesture employed by users while performing the gaze gesture set.

4 Results

We studied the amount of time needed to complete a gaze gesture by plotting on a histogram the distribution of time lengths per gesture required by users while performing the set of 50 gaze gestures, Figure 2(b). Inference accuracy for gaze gestures recognition varied for different network topologies with a two layer network showing the best performance, Figure 3(a).

Several data representations of the gaze gestures were tried out, Figure 3(b), on the optimal 2-Layer HTM network determined above. Namely, no temporal codification, a finely grained temporal codification in terms of seconds and a simpler temporal codification with just three temporal stages that turned out to be the optimal one with up to 96% recognition accuracy.

Experiments were carried out to determine HTM decreasing performance with larger gaze gestures vocabularies, Figure 4(a). The 3 temporal stages data representation proved to be the most resistant to increasing vocabulary size.

HTM performance improved markedly with increasing number of training instances, Figure 4(b). The 3 temporal stages data representation proved to perform better than the others for the whole range of training instances availability.

A user with no prior experience in gaze tracking repeated the 50 gaze gestures data set over five blocks in order to determine learning effects, Figure 5(a). Upon

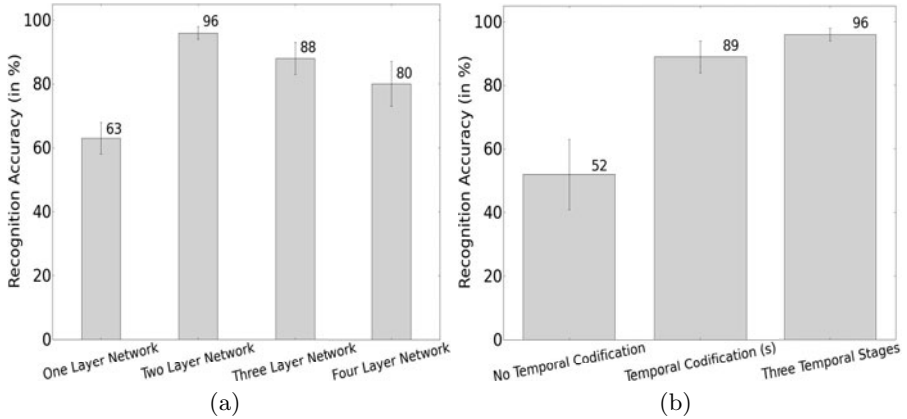


Fig. 3. Panel *a* shows accuracy performance for different network topologies using 3 stages temporal codification. The bars show the percentage of correct classifications achieved by 1, 2, 3 and 4 layer networks. Panel *b* shows the accuracy performance of different data representations on the optimal 2 layer HTM network: no temporal codification, temporal codification in seconds and 3 stages temporal codification.

completing the experiments, participants filled up a questionnaire to provide feedback about gaze gestures as an input modality in HCI. On a score scale from 0 to 5 ranging from strongly disagree to strongly agree, participants reported the averages 4.5, 4.5, 2.5, and 4 for the respective categories: 'Easy to use', 'Fast', 'Tiring', and 'Accurate'.

To study gaze gesture recognition in real time, we measured HTM inference scores obtained on gaze data during normal computer use and scores obtained when specifically performing inferences over gaze gestures data, Figure 5(b).

5 Discussion

For optimal performance, HTMs require training instances to be composed of a complete spatial structure at any time instant. This creates a challenge for data structures with a temporal component unfolding over time. To address this issue, data representation and coding become key when designing and training an HTM network.

Gaze gestures patterns as existing in the original data set are not fit to be learned by a traditional HTM network since at any moment in time the complete spatial representation of the pattern is not complete. The original data set consisted of just a temporal series of (x,y) coordinate pairs. The original time-series were transformed into a 2 dimensional matrix containing the on-screen path performed by the gaze during the performance of a sign 1(a). This data structure captured the complete spatial structure of a gaze gesture. However, a gaze gesture possesses a temporal structure as well. This aspect is critical since the temporal order of a gaze gesture differentiated several gestures with complete overlap in their 2D spatial representation such as the gaze gesture in the

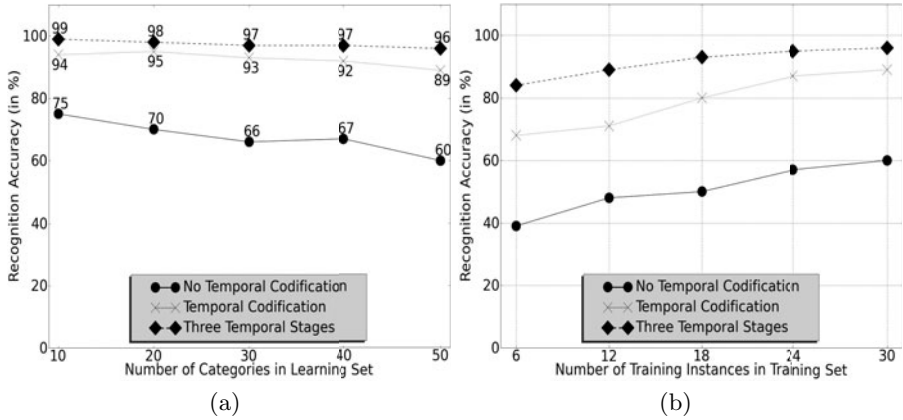


Fig. 4. Panel *a* shows the decreasing performance of the optimal 2-layer HTM network using different data representations as the vocabulary size of the gaze gestures set to be learnt increases. Panel *b* shows the improving performance of HTM networks using different data representations as the number of training instances increases.

first row, 2nd column and the gaze gesture in the last column of the same row in Figure 2(a). In order to codify the temporal information of a gesture, several approaches were explored, Figure 3(b).

Time needed to complete gaze gestures is an important design and constraint parameter for gaze-computer interaction. Our experiments show that gaze

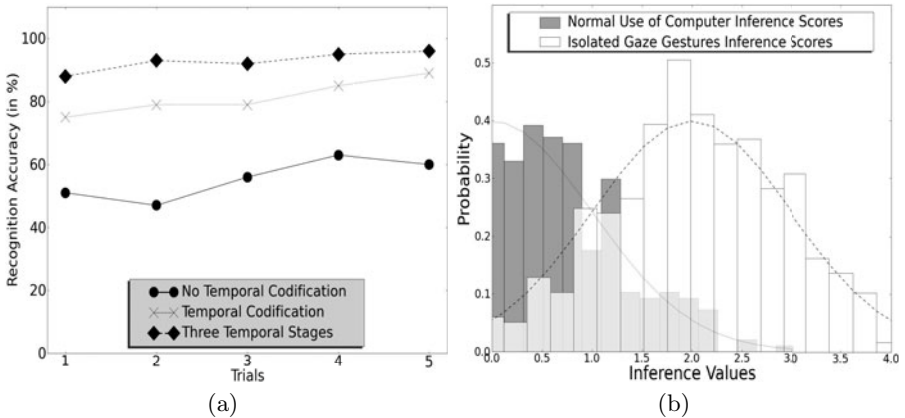


Fig. 5. Panel *a* shows the recognition accuracy of the 2-layer HTM network using different data representations over 5 blocks for a single user. Panel *b* shows the HTM inference scores during normal gaze activity while using a computer and the HTM scores obtained on consciously performed gaze gestures. The area of overlap between both types of inference scores is shown in light grey.

gestures are fast to complete, Figure 2(b), easy to learn, Figure 5(a), and they do not occupy screen-real state since no visual markers to aid in gaze trajectory were used. Furthermore, in our user study, participants showed through questionnaires satisfaction with this innovative input modality.

Gaze gestures however present problems of their own, the main one being accidental gesture completion during normal gaze activity. As Figure 5(b) shows, it is a challenge for the HTM algorithm to perfectly partition consciously performed gaze gestures from unintended gaze gestures completed during normal gaze activity while using a computer. Furthermore, gaze gestures also generate a cognitive load on the user who is forced to memorize and reproduce sequences of eye movements to recreate a gesture without an immediate feedback.

Our results suggest that our system can be expanded to a larger vocabulary set with still acceptable recognition performance, Figure 4(a). Figure 4(b) illustrates the importance of having a large number of training instances for the HTMs to achieve good recognition performance.

A trade-off emerges from our study between complex gaze gestures and simpler ones. Simpler gaze gestures are easy on the user, yet the possibilities of accidental gesture completion are high. Complex gaze gestures decrease the possibility of accidental recognition during normal gaze activity and augment the interaction vocabulary space, imposing however, a cognitive load on the end user.

Our work shows that humans adapt quickly and comfortably to this innovative modality of HCI. The good recognition results achieved by the HTM algorithm and the positive feedback from users, illustrate that using gaze gestures recognized through HTMs constitutes an innovative, robust, easy-to-learn and viable approach to HCI for several environments and device combinations.

References

1. Rozado, D., Rodriguez, F., Varona, P.: Optimizing hierarchical temporal memory for multivariable time series. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010. LNCS, vol. 6353, pp. 506–518. Springer, Heidelberg (2010)
2. Mollenbach, E., Hansen, J.P., Lillholm, M., Gale, A.G.: Single stroke gaze gestures. In: Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems, CHI 2009, pp. 4555–4560. ACM, New York (2009), <http://doi.acm.org/10.1145/1520340.1520699>
3. Drewes, H., Schmidt, A.: Interacting with the computer using gaze gestures. In: Baranauskas, C., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007. LNCS, vol. 4663, pp. 475–488. Springer, Heidelberg (2007)
4. San Agustin, J., Skovsgaard, H., Mollenbach, E., Barret, M., Tall, M., Hansen, D.W., Hansen, J.P.: Evaluation of a low-cost open-source gaze tracker. In: ETRA 2010: Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, pp. 77–80. ACM, New York (2010)
5. George, D., Hawkins, J.: Towards a mathematical theory of cortical micro-circuits. PLoS Comput. Biol. 5(10), e1000532 (2009), <http://dx.doi.org/10.1371> (online)

Feature Selection for Multi-label Classification Problems

Gauthier Doquire* and Michel Verleysen

Université catholique de Louvain, Machine Learning Group - ICTEAM
Place du Levant, 3, 1348 Louvain-la-Neuve, Belgium
{gauthier.doquire,michel.verleysen}@uclouvain.be
<http://www.ucl.ac.be/mlg>

Abstract. This paper proposes the use of mutual information for feature selection in multi-label classification, a surprisingly almost not studied problem. A pruned problem transformation method is first applied, transforming the multi-label problem into a single-label one. A greedy feature selection procedure based on multidimensional mutual information is then conducted. Results on three databases clearly demonstrate the interest of the approach which allows one to sharply reduce the dimension of the problem and to enhance the performance of classifiers.

Keywords: Feature selection, Multi-Label, Problem Transformation, Mutual Information.

1 Introduction

Multi-label Classification is the task of assigning data points to a set of classes or categories which are not mutually exclusive, meaning that a point can belong simultaneously to different classes. This problem is thus more general than the traditional single-label classification which assumes each point belongs to exactly one category; it is therefore often encountered in practice. As an example, in text categorization problems, an article about the Kyoto Protocol can be labelled with both *politics* and *ecology* categories. In scene classification, a picture could as well belong to different classes such as *beach* and *mountain* [1]. Other domains for which multi-label classification has proved useful also include protein function classification [2] and classification of music into emotions [3].

Due to its importance, multi-label classification has been studied quite extensively since a few years, leading to the development of numerous classification algorithms. Some of them are extensions of existing single-label classification methods such as AdaBoost [4], support vector machines (SVM) [5] or K nearest neighbors [6] among others.

Another popular approach to multi-label classification consists in transforming the problem into one or more single-label classification tasks. State of the art algorithms such as SVM can then be used directly. The most popular transformation method is the binary relevance (BR) which consists in learning a different

* Gauthier Doquire is funded by a Belgian FRiA grant.

classifier for each label. In other words, the original problem is transformed into C two classes single-label classification problems, where C is the number of possible labels. The i^{th} ($i = 1 \dots l$) classifier decides whether or not a point belongs to the i^{th} class. The union of predicted labels for each point is the final output. One of the major drawbacks of BR is that it does not take into account the dependence which could exist between labels.

Label powerset (LP) is a different problem transformation method which does consider this dependence. It treats each unique set of labels in the training set as a possible class of a single-label classifier. The number of classes created this way being potentially huge, Read et al. [7] recently proposed to prune the problem, by considering only classes represented by a minimum number of data points. Points with a too rare label are either removed from the training set or are given a new label and kept. They called this approach pruned problem transformation (PPT). See [7] for details.

Surprisingly, feature selection for multi-label classification has not received much attention yet. Indeed, to the best of our knowledge, one of the few proposed approach is the one by Trohidis et al. [3] which consists in transforming the problem with the LP method, before using the χ^2 statistic to rank the features. However, feature selection is an important task in machine learning and pattern recognition, as it can improve the interpretability of the problems, together with performances and learning time of prediction algorithms [8].

This paper proposes to use mutual information (MI) to achieve feature selection in multi-label classification problems. More precisely, the problem is first transformed with the PPT method and a greedy forward search strategy is then conducted with multidimensional MI as the search criterion. This approach thus considers dependencies between labels as well as dependencies between features, which ranking approaches such as [3] do not.

The remaining of the paper is organized as follows. Section 2 briefly recalls some basic concepts about MI, and introduces the estimator used. The methodology is described in Section 3, and the interest of the approach is experimentally shown in section 4. Section 5 concludes the work and gives some future work perspectives.

2 Mutual Information

2.1 Definitions

MI [9] is a measure of the quantity of information two variables contain about each other. It has been widely used for feature selection [10] mainly because of its ability to detect non-linear relationships between variables. This not the case, as an example, for the correlation coefficient. Moreover, MI is naturally defined for groups of variables, which allows one to take feature dependence into account during the feature selection process.

MI of a couple of random variables X and Y is formally defined in terms of the probability density functions (PDF) of X , Y and (X, Y) , respectively denoted as f_X , f_Y and $f_{X,Y}$:

$$I(X; Y) = \int \int f_{X,Y}(\zeta_X, \zeta_Y) \log \frac{f_{X,Y}(\zeta_X, \zeta_Y)}{f_X(\zeta_X)f_Y(\zeta_Y)} d\zeta_X d\zeta_Y. \quad (1)$$

In practice none of the PDF's are known for real-world problems, and MI has to be estimated from the dataset.

2.2 Estimation

In this paper, an MI estimator introduced by Gomez et al. [11] is used. It is based on the Kozachenko-Leonenko estimator of entropy [12]:

$$\hat{H}(X) = -\psi(K) + \psi(N) + \log(c_d) + \frac{d}{N} \sum_{n=1}^N \log(\epsilon(n, K)) \quad (2)$$

where ψ is the digamma function, K the number of nearest neighbors considered, N the number of samples in X , d the dimensionality of these samples, c_d the volume of a unitary ball of dimension d and $\epsilon(n, K)$ twice the distance from the n^{th} observation in X to its K^{th} nearest neighbor. Throughout this paper, the metric used in the X space is the Euclidean distance.

Basing their developments on [2], Kraskov et al. previously proposed two estimators for regression problems. See [13] for details.

For classification problems, the probability distribution of the (discrete) class vector Y is $p(y = y_l) = n_l/N$, with n_l the number of points whose class value is y_l . Rewriting MI in terms of entropies:

$$I(X; Y) = H(Y) - H(Y|X), \quad (3)$$

it is possible to derive the following estimator:

$$\hat{I}_{cat}(X; Y) = \psi(N) - \frac{1}{N} n_l \psi(n_l) + \frac{d}{N} \left[\sum_{n=1}^N \log(\epsilon(n, K)) - \sum_{l=1}^L \sum_{n \in y_l} \log(\epsilon_l(n, K)) \right] \quad (4)$$

where L is the total number of classes. $\epsilon_l(n, K)$ has the same meaning as $\epsilon(n, K)$ but the set of possible neighbors for the n^{th} observation is limited to the points whose class label is y_l .

This estimator has the crucial advantage that it does not require directly the estimation of any PDF which is a particularly hard task when the dimension of the data increases, due to the so-called *curse of dimensionality*. This curse reflects the fact that the number of points needed to sample a space increases exponentially with the dimension of the space. Histograms or kernel density estimators [14] are thus not likely to work well in high dimensional spaces. Because

it avoids such unreliable estimations, (4) is expected to be less sensitive to the dimension of the data; this family of estimators has already been used successfully for feature selection [11, 15].

3 Methodology

This section describes the methodology followed to achieve feature selection. First, the multi-label problem is transformed using the PPT method defined above [7], and the data points with a class label encountered less than t times in the training set are discarded. The result of this transformation is thus a multi-class *single-label* classification problem. The pruning has a double interest here; it leads to a simplified version of the problem and ensures that all classes are represented by at least t points. This last observation is crucial since the MI estimator (4) requires the distance between each point and its K^{th} nearest neighbor from the same class. It is thus needed that $K < t$.

Once the problem has been transformed, traditional feature selection techniques can be used. In this paper, a greedy forward feature selection algorithm based on MI is employed. It begins with an empty set of features and first selects the feature whose MI with the class vector is the highest. Then, sequentially, the algorithm selects the feature not yet selected whose addition to the current subset of selected features leads to the set having the highest MI with the output. This choice is never questioned again, hence the name *forward*. Of course, other search strategies could also be considered such as backward elimination, which starts with the set of all features and recursively removes them one at a time.

The procedure can be ended when a predefined number of features have been chosen. Another strategy is to rank all the features and then to choose the optimal number to keep on a validation set.

4 Experiments and Results

This section begins by introducing the performance criterias considered since they differ from those used for single-label classification. The databases used in the experiments are then briefly described and the results are eventually shown.

4.1 Evaluation Criteria

Two very popular evaluation criterias are considered in this study: the Hamming loss and the accuracy. Let $|M|$ be the number of points in a test set M , Y_i , $i = 1 \dots |M|$, the sets of true class labels and \hat{Y}_i the sets of labels predicted by a multi-label classifier h .

The Hamming loss is then defined as:

$$HL(h, M) = \frac{1}{|M|} \sum_{i=1}^{|M|} \frac{1}{|C|} |Y_i \Delta \hat{Y}_i| \quad (5)$$

where Δ denotes the symmetric difference between two sets, i.e. the difference between the union and the intersection of the two sets. $|C|$ is the number of possible labels.

The accuracy is defined as:

$$Accuracy(h, M) = \frac{1}{|M|} \sum_{i=1}^{|M|} \frac{|Y_i \cap \hat{Y}_i|}{|Y_i \cup \hat{Y}_i|}. \quad (6)$$

Of course, the smaller the Hamming loss and the higher the accuracy, the better the classifier's performances. It is important to note that all data points are assumed to belong to at least one class. If it was not the case, the accuracy as defined above (6) would be infinite.

4.2 Datasets

Three datasets are used for experiments in this paper.

The first one is the Yeast dataset. It is concerned with associating each gene a set of functional classes. For the sake of simplicity, the data have been pre-processed by Elisseff and Weston [5] to consider only the known structure of the functional classes. Eventually, the sample sizes of the training set and the test set are 1500 and 917 respectively. There are 103 features and 14 possible labels. The Scene dataset is also considered [1]. The goal here is the semantic indexing of scenes. The number of samples is 1211 for the training set and 1196 for the test set. There are 294 features and 6 labels. The last dataset is called Emotions and is about the classification of music into emotions [3]. Among the 593 samples, 391 are used as the training set and the 202 other as the test set. The number of features and of labels is 72 and 6 respectively. The proposed training set/testing set splittings are the ones traditionally used in the multi label classification literature.

The three datasets are available for download in ARFF format at the web page of the Mulan project¹.

4.3 Experimental Results

The k nearest neighbors based multi-label classification algorithm introduced by Zhang and Zhou [6] is used to illustrate the interest of the proposed approach. To determine the set of labels of a new instance, the algorithm first identifies its k nearest neighbors. Then, based on their labels, the maximum a posteriori principle is used to predict the label set of the new instance. As suggested by the authors, the value of the parameter k has been set to 7.

The problem transformation is only used to achieve feature selection. Once the features have been ranked, the original multi-label problem is considered again with all the samples. The number of neighbors considered in the MI estimator (4) is $K = 4$ and the t parameter for the PPT is fixed to 6. These values have been

¹ <http://mulan.sourceforge.net/datasets.html>

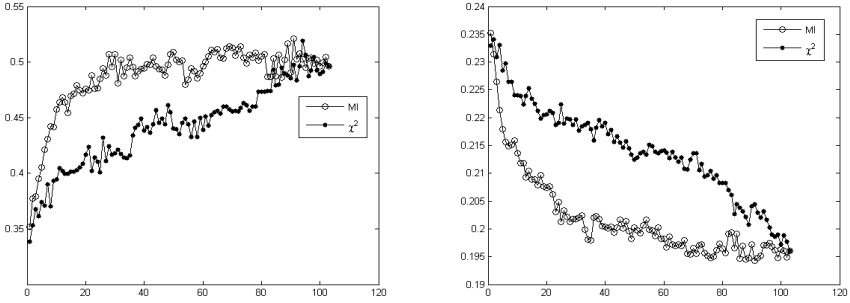


Fig. 1. Accuracy (left) and Hamming loss (right) of the k nearest neighbors classifier as a function of the number of selected features for the Emotions dataset

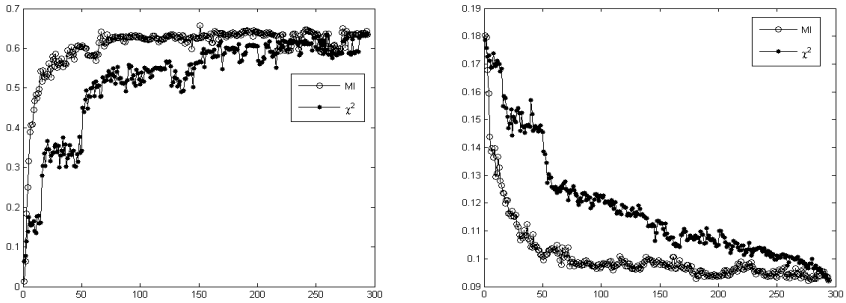


Fig. 2. Accuracy (left) and Hamming loss (right) of the k nearest neighbors classifier as a function of the number of selected features for the Emotions dataset

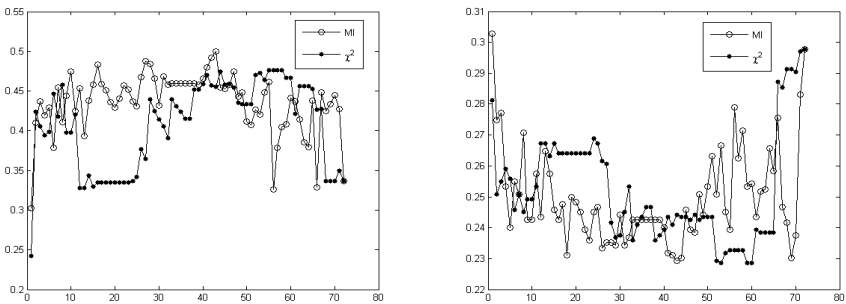


Fig. 3. Accuracy (left) and Hamming loss (right) of the k nearest neighbors classifier as a function of the number of selected features for the Emotions dataset

chosen as a compromise between the need to consider a sufficiently large number of neighbors in the MI estimator and the fact that in [7], better performances are usually obtained with a small value of t .

Figures 1 to 3 show the Hamming loss and the accuracy of the classifier on the test set of the three datasets with the MI based feature selection (denoted as MI). For comparison, the results obtained with the approach by Trohidis et al. [3] are also presented (and are denoted as χ^2).

The results clearly demonstrate the interest of the MI based approach and its advantage over the method based on the χ^2 statistic. Particularly, the proposed approach always leads to an increase in performance both for the Hamming loss and the accuracy compared with the case no feature selection is considered. This is not the case for the Hamming loss with the χ^2 based approach on the Yeast and Scene datasets and the differences between the two approach performances are particularly obvious for those two datasets. The results are much more comparable for the Emotions dataset for which both methods result in a large improvement of the classifier performances.

As already stated, the good behaviour of the proposed methodology can be explained by the use of a powerful criterion combined with an approach taking relations between features into account. Indeed, the forward selection procedure described above is expected to better handle redundancy between features than simple ranking methods do. This is fundamental since a feature, even with a high predictive power, is useless if it carries the same information about the output than another selected feature; it should therefore not be selected.

The same experiments have been carried out with a SVM classifier working directly on the transformed and pruned problem. The results obtained confirm those presented in this paper. However, due to space limitations, they are not presented here.

5 Conclusions

This paper is concerned with feature selection for multi-label classification, a problem which has up to now received little attention despite its great importance and the amount of work recently proposed on multi-label classification.

It is suggested to use multidimensional mutual information after the transformation of the multi-label problem to a single-label one through the pruned problem transformation method. To this end, a nearest neighbors based MI estimator is used, as it is believed to behave well when dealing with high-dimensional data. The estimator is combined with a simple greedy forward search strategy to achieve feature selection.

Results on three real-world datasets coming from different domains show the interest of this new approach compared with a strategy based on the χ^2 statistic in terms of the Hamming loss and the accuracy of a classifier.

Future work should include the study of the influence of the pruning parameter; it could be possible to tune this parameter to maximise the performances of

the classifiers. A trade-off should then be found between the increase in performances and the computation load of validation procedures such as k -fold cross validation.

Besides the basic stopping criteria proposed in Section 3, much sophisticated strategies can be thought of. As an example, Damien et al. proposed a stopping criterion based on resampling and the permutation test [16]. The basic idea is to halt the procedure when the addition of a new feature does not improve significantly the MI between the selected features and the output. This approach could be applied to the problem considered in this paper.

References

1. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning Multi-Label Scene Classification. *Pattern Recogn.* 37, 1757–1771 (2004)
2. Diplaris, S., Tsoumakas, G., Mitkas, P., Vlahavas, I.: Protein Classification with Multiple Algorithms. In: Bozanis, P., Houstis, E.N. (eds.) *PCI 2005. LNCS*, vol. 3746, pp. 448–459. Springer, Heidelberg (2005)
3. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multi-Label Classification of Music into Emotions. In: 9th International Conference on Music Information Retrieval (ISMIR 2008), Philadelphia, pp. 325–330 (2008)
4. Schapire, R.E., Singer, Y.: Boostexter: A Boosting-Based System for Text categorization. *Machine Learning* 39, 135–168 (2000)
5. Elisseeff, A., Weston, J.: A Kernel method for Multi-Labelled Classification. *Advances in Neural Information Processing Systems* 14, 681–687 (2001)
6. Zhang, M.-L., Zhou, Z.-H.: ML-KNN: A Lazy Learning Approach to Multi-Label Learning. *Pattern Recogn.* 40, 2038–2048 (2007)
7. Read, J.: A Pruned Problem Transformation Method for Multi-label Classification. In: *New Zealand Computer Science Research Student Conference (NZCSRS 2008)*, pp. 143–150 (2008)
8. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. *J. Mach. Lear. Res.* 3, 1157–1182 (2003)
9. Shannon, C.E.: A mathematical Theory of Communication. *Bell Syst. Tech. J.* 27, 379–423, 623–656 (1948)
10. Battiti, R.: Using Mutual Information for Selecting Features in Supervised Neural Net Learning. *IEEE T. Neural. Networ.* 5, 537–550 (1994)
11. Gomez-Verdejo, V., Verleysen, M., Fleury, J.: Information-Theoretic Feature Selection for Functional Data Classification. *Neurocomputing* 72, 3580–3589 (2009)
12. Kozachenko, L.F., Leonenko, N.: Sample Estimate of the Entropy of a Random Vector. *Problems Inform. Transmission* 23, 95–101 (1987)
13. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating Mutual Information. *Phys. Rev. E* 69, 066138 (2004)
14. Parzen, E.: On Estimation of a Probability Density Function and Mode. *Ann. Math. Statist.* 33, 1065–1076 (1962)
15. Benoudjit, N., François, D., Meurens, M., Verleysen, M.: Spectrophotometric Variable Selection by Mutual Information. *Chemometr. Intell. Lab.* 74, 243–251 (2004)
16. Francois, D., Rossi, F., Wertz, V., Verleysen, M.: Resampling Methods for Parameter-free and Robust Feature Selection with Mutual Information. *Neurocomputing* 70, 1276–1288 (2007)

A Novel Grouping Heuristic Algorithm for the Switch Location Problem Based on a Hybrid Dual Harmony Search Technique

Sergio Gil-Lopez¹, Itziar Landa-Torres¹, Javier Del Ser¹,
Sancho Salcedo-Sanz², Diana Manjarres¹, and Jose A. Portilla-Figueras²

¹ TecNALIA Research and Innovation, 48170 Zamudio, Spain

{sergio.gil,javier.delser}@tecnalia.com

² Universidad de Alcalá. 28871 Alcalá de Henares, Madrid, Spain

sancho.salcedo@uah.es

Abstract. This manuscript proposes a novel iterative approach for the so-called Switch Location Problem (SLP) based on the hybridization of a group-encoded Harmony Search combinatorial heuristic (GHS) with local search and repair methods. Our contribution over other avantgarde techniques lies on the dual application of the GHS operators over both the assignment and the grouping parts of the encoded solutions. Furthermore, the aforementioned local search and repair procedures account for the compliancy of the iteratively refined candidate solutions with respect to the capacity constraints imposed in the SLP problem. Extensive simulation results done for a wide range of network instances verify that statistically our proposed dual algorithm outperforms all existing evolutionary approaches in the literature for the specific SLP problem at hand. Furthermore, it is shown that by properly selecting different yet optimized values for the operational GHS parameters to the two parts comprising the group-encoded solutions, the algorithm can trade statistical stability (i.e. lower standard deviation of the metric) for accuracy (i.e. lower minimum value of the metric) in the set of performed simulations.

Keywords: Switch Location Problem, Genetic Algorithm, Harmony Search, grouping encoding.

1 Introduction

Due to its massive deployment, sharply-increasing number of users, and wide expansion in the Telecommunications market, most of the communication technologies emerging in the last decade (e.g. wireless sensor networks (WSN), Internet access networks or mobile telephony) have encountered a common design problem: to derive network topologies that ensure a resilient connection among interconnected nodes while optimizing the number and cost of the network resources deployed thereon. This problem can be exemplified by a typical network planning problem, where a dense deployment of access nodes embodies a communication-efficient albeit cost-inefficient topology. In summary: a well-balanced trade-off must be met between 1) the economical cost associated with

the deployment and maintenance of network resources, and 2) the allocated communication resources necessary for serving the traffic generated by the nodes.

This article emphasizes on network planning problems; however, all the derivations herein can be easily extrapolated to any other problem of similar nature. Among the variety of related planning paradigms, it is worth mentioning the design of fixed network topologies [1], the optimum positioning of base stations [2] and the Terminal Assignment (TA) problem [3], all of which render by themselves NP-complete optimization problems. Let us concentrate on the Switch Location Problem (SLP), which is based on assigning a set of N nodes with distinct rate demands to a set of M concentrators subject to capacity constraints. The $N - M$ nodes that are not chosen as concentrators are defined as *terminals*, each of which can be assigned to only one concentrator. The assignment must be performed such that the sum of distances between each terminal and its concentrator is minimized, whereas the overall rate demanded from each concentrator must be kept below its maximum capacity. To efficiently solve the SLP problem, a large amount of exact, heuristic and meta-heuristic algorithms have been presented in the literature, such as those proposed by Ceselli et al. in [4,5,6], Simulated Annealing and Tabu Search by Osman and Christofides in [7], a bionomic approach by Maniezzo et al. in [8], and a Set Partitioning Algorithm was proposed in [9]. Analogously, Lorena et al. proposed a Constructive Genetic Algorithm (CGA) for grouping problems [10], which can also be adapted to the SLP paradigm herein considered. In a similar approach, Lorena et al. presented a Column Generation Algorithm in [11]. Also, Scatter Search approaches have been applied to solve this problem, such as the works by Scheuerer et al. in [12] and Diaz et al. in [13], the latter incorporating path relinking to further enhance the performance of the algorithm. Likewise, a guided construction search heuristic was introduced by Osman et al. in [14]. Finally, Santos-Correa et al. [15] proposed an evolutionary algorithm with novel operators specifically designed for the SLP scenario. In a more recent approach this problem was also tackled by means of hybrid evolutionary approaches, mixing global search techniques with local techniques, such as those proposed in [16], [3] and [17].

This manuscript advances over this upsurge of research on the SLP by proposing an improved version of the Grouping Harmony Search (GHS) algorithm first published in [17]. Specifically, as opposed to [17] our novel approach applies the operators to the two compounding parts of the grouping-encoded solutions, hence the algorithm is coined as Dual Grouping Harmony Search (DGHS). Secondly, we present an exhaustive comparative study of the aforementioned DGHS scheme with 1) a hybrid GGA and GHS with similar encoding and local search methods; and with 2) all alternative existing evolutionary techniques reported so far for this particular optimization scenario. As the obtained results will clearly show, our approach dominates – in terms of accuracy – the aforementioned evolutionary techniques, thus producing network topologies with a better balance between rate service and cost of deployment.

The rest of the paper is structured as follows: first, Section 2 formally poses the SLP, whereas Section 3 elucidates the proposed DGHS technique. Next, an exhaustive simulation-based comparison among all existing evolutionary techniques for the SLP problem is presented in Section 4 and, finally, Section 5 summarizes the paper by drawing some conclusions.

2 Problem Statement

The SLP can be formulated as: given N nodes (terminals) denoted with the set of indices $\mathcal{T} \triangleq \{1, \dots, N\}$, select M nodes out of the N terminals to perform as *concentrators*. Based on the widely-utilized mathematical definition of Revelle and Swain [18], let us define a $1 \times N$ binary vector \mathbf{y} such that $y_i = 1$ ($i \in \{1, \dots, N\}$) if node i is set as a concentrator, and $y_i = 0$ otherwise. In addition, let \mathbf{X} denote a $N \times M$ binary matrix, where $x_{im} = 1$ means that terminal i is assigned to concentrator m , and $x_{im} = 0$ otherwise. For the sake of mathematical completion, linear indices $m \in \{1, \dots, M\}$ are mapped to node indices $\Lambda(m) \in \{1, \dots, N\}$ by means of an injective mapping $\Lambda: 1 \times M \mapsto 1 \times N$, e.g. $\Lambda(1) = 12$ will denote that the first concentrator is node index 12. Obviously, $\Lambda(m)$ will equal n for some m iff (if and only if) $y_n = 1$. We further define a $1 \times N$ capacity vector \mathbf{c} , whose element c_i ($i \in \{1, \dots, N\}$) establishes the total capacity of node i when acting as a concentrator. Similarly, a $1 \times N$ weight vector $\mathbf{w} \triangleq \{w_1, \dots, w_N\}$ is further assumed to account for the rate demanded by node i when it acts as a terminal. A $N \times M$ matrix \mathbf{D} can also be constructed so as to account for the distance between nodes and concentrators, i.e. each element d_{im} represents the Euclidean distance from node i to concentrator m , with $m \in \{1, \dots, M\}$. Therefore, given a vector \mathbf{y} one can arbitrarily sort the selected concentrators and construct the mapping $\Lambda(\cdot)$ accordingly, but this mapping must be set before arranging the distance and assignment matrices \mathbf{D} and \mathbf{X} . With the above definitions, the SLP can be then stated as

$$\min_{\mathbf{y}, \mathbf{X}} \left(\sum_{i=1}^N \sum_{m=1}^M d_{im} \cdot x_{im} \right), \quad (1)$$

subject to

$$x_{im} \in \{0, 1\}, \quad i = 1, \dots, N, \quad m = 1, \dots, M, \quad (2)$$

$$y_i \in \{0, 1\}, \quad i = 1, \dots, N, \quad \sum_{i=1}^N y_i = M, \quad (3)$$

$$\sum_{i=1}^N w_i \cdot x_{im} \leq c_{\Lambda(m)}, \quad m = 1, \dots, M. \quad (4)$$

Expression (1) represents the *metric* or fitness function for quantifying the cost of each network configuration, whereas Expression (4) stands for the capacity constraint, i.e. the sum of rate requirements of the terminals associated with a

given concentrator cannot exceed its corresponding capacity. The addition of this constraint to the distance-based minimization problem grounds the rationale for combining a global search technique with a local yet exhaustive restricted search method dealing with such a constraint in a computationally-efficient manner. We delve into such a hybrid approach in next section.

3 Proposed Hybrid Dual Grouping Harmony Search (DGHS) Algorithm

The SLP problem comprises by itself a grouping problem, in the sense that different groups of elements belonging to a set must be split into several subsets subject to certain constraints. In this context, Falkenauer in [19,20] first showed that the concept of *grouping* should be included in the problem encoding, since it significantly reduces the redundancy, thus the dimensions of the solution space and, ultimately, the complexity of the underlying NP-hard problem. As such, the grouping encoding procedure divides each proposed solution into two parts, namely $\mathbf{S} = (\mathbf{s}_x | \mathbf{s}_y)$. The assignment part \mathbf{s}_x indicates to which concentrator is assigned each of the N terminals; therefore, it consists of N integer indices drawn from the set $\{1, \dots, M\}$. On the other hand, the grouping part \mathbf{s}_y denotes the node indices which act as concentrators in the current solution; consequently, it is furnished by M integer indices from the set $\{1, \dots, N\}$. Encouraged by the results of [19,20], Alonso-Garrido et al. in [3], and more recently Gil-Lopez et al. in [17] applied the grouping encoding strategy to the SLP problem by using a Genetic and Harmony Search Algorithm, respectively.

Let us elaborate further on the Harmony Search (HS) algorithm, which is a meta-heuristic population-based algorithm mimicking the behavior of a music orchestra when aiming at composing the most harmonious melody. Assuming the classical notation related to HS, we hereafter denote each possible candidate for a solution as *melody*, whereas we will refer to each of its compounding elements as *note*. The algorithm essentially iterates on a set of φ possible candidates or melodies (*Harmony Memory*), which are refined – in terms of their metric – by means of a double *improvisation* procedure applied note-wise. The process is repeated until a maximum assumed complexity is reached or until the quality of the best proposed harmony falls below a certain threshold. Having said this, it is important to note that the novel Grouping Harmony Search approach in [17] – specially tailored to solve the SLP – only executes the improvisation procedure to the \mathbf{s}_x part of the grouping-encoded set of melodies, whereas the \mathbf{s}_y part is deterministically built based on the minimization of the sum-distance between clustered nodes.

On the contrary, in this manuscript we propose to apply the Harmony Search operators not only to the assignment part \mathbf{s}_x , but also to the grouping part \mathbf{s}_y . In this scheme the algorithm works sequentially, at each iteration, by first operating on all notes conforming the assignment part of the φ stored melodies, followed by the same process on the grouping part \mathbf{s}_y . Once both grouping parts \mathbf{s}_x and \mathbf{s}_y have been *improvised* (i.e. refined), and right before the metric of the newly

improvised melodies is evaluated, the node selected to perform as concentrator for a given group is the one minimizing the total distance between itself and all the other nodes belonging to the group. The algorithm can be summarized by the following four steps:

1. The initialization process is only considered at the first iteration. At this point, and since no *a priori* knowledge of the solution is assumed, the set of φ harmonies in the Harmony Memory is filled randomly with N integer indices from the set $\{1, \dots, M\}$ in the s_x .
2. The improvisation process is applied sequentially to each note of the complete set of harmonies, first in s_x , and second in s_y . The proposed method is controlled by three different parameters:
 - The Harmony Memory Considering Rate, $\text{HMCR} \in [0, 1]$, which sets the probability that the new value for a note is drawn uniformly from the values of the same note in all the other $\varphi - 1$ harmonies in the Harmony Memory.
 - The Pitch Adjusting Rate, $\text{PAR} \in [0, 1]$, establishes the probability that the new value for a given note is taken from its neighboring value in the respective alphabet, i.e. $\{1, \dots, M\}$ for s_x and $\{1, \dots, N\}$ in s_y .
 - The uniformly random selection of the new value for a note based on the corresponding alphabet is controlled by another probabilistic parameter RSR (Random Selection Rate), different than the complementary HMCR probability used in the naïve implementation of the Harmony Search algorithm.
3. The **repair criterion** in [3] is hybridized with the previous global search dual method and applied to the stored melodies when the capacity constraints in expression (4) are not satisfied. This procedure is executed until all the compounding groups satisfy the capacity constraint, or until a number of evaluations are done. In the latter case, the capacity constraints are not satisfied, and thus the repair of the harmony is delegated to the next described local search method. If this local search does not succeed, the metric value of the solution is penalized.
4. The **local search** method hinges on the *GreedyExp* algorithm proposed by Salcedo-Sanz *et al.* in [3], which reduces to an optimized version of the *Greedy* algorithm [21]. The local search is only applied to the s_x part.
5. The evaluation of the new generated candidate solutions and the update of the Harmony Memory is made at each iteration based on the fitness function (1). At each iteration φ new harmonies are improvised and evaluated, but they will be included in the Harmony Memory iff they improve the quality of the harmonies remaining from the previous iteration.
6. The algorithm stops when a fixed number of iterations \mathcal{I} is reached.

4 Computational Experiments and Results

Extensive computer simulations have been done by considering 7 networks of different size, number of nodes N and of concentrators M randomly deployed

over a 400×400 grid (see Table 1). The novel hybrid DGHS scheme is compared with other published evolutionary schemes based on Genetic Algorithms [3,15,11] and on the Harmony Search heuristic on which the present work is built [17]. For the sake of fairness, several parameters are fixed for all simulated schemes: the physical locations of all nodes, rate demands \mathbf{w} (drawn from a normal distribution with mean 10 and standard deviation 5), concentrator models, maximum capacities of the nodes \mathbf{p} and maximum number of iterations \mathcal{I} . Feasibility in the scenarios (i.e. $\sum_{i=1}^N w_i < \sum_{j=1}^M c_j$) is guaranteed by setting $c_i = 1.1 \sum_{n=1}^N w_n / M$. Statistical results for all approaches are obtained for the same number of fitness evaluations per execution (10^4) and by averaging over 20 realizations of each algorithm. The operator values used in the case of the proposed hybrid DGHS are based on an exhaustive optimization study for a total of 177 combinations of the parameter set

$$\lambda \triangleq [\text{PAR}(\mathbf{s}_x), \text{HMCR}(\mathbf{s}_x), \text{RSR}(\mathbf{s}_x), \text{HMCR}(\mathbf{s}_y), \text{PAR}(\mathbf{s}_y), \text{RSR}(\mathbf{s}_y)], \quad (5)$$

based on which two parameter sets are selected: $\lambda_1 = [0.01, 0.1, 0.1, 0.1, 0, 0]$ and $\lambda_2 = [0.2, 0.2, 0.01, 0.01, 0.1, 0.1]$. Note that these operators always refer to a probability for every node (i.e. not for the whole solution). Thereby, although their values are smaller than those of the GGA operators, it does not mean that they act fewer times.

Table 1. Parameters of the considered SLP instances, and results from previous works

ID	N	M	\mathcal{I}	CGA [10]	GA [15]	GA+LP [3]	GA+Greedy_Exp [3]
1	60	5	200	3841/3856/18	3800/3812/12	3800/3816/17	3841/3877/30
2	80	8	200	3819/3822/4	3860/3874/29	3806/3831/26	3819/3842/27
3	90	9	200	3883/3914/21	3792/3845/66	3792/3837/39	3891/3931/38
4	100	10	200	4452/4567/75	4452/4508/29	4532/4637/90	4455/4464/13
5	110	10	200	4776/4785/25	4744/4794/53	4727/4768/53	4772/4835/55
6	150	15	200	5216/5296/57	5210/5383/81	5059/5210/114	5258/5414/106
7	200	15	200	No data	No data	No data	No data

The obtained results are presented in the format $rnd(best/mean/std)$ value of the metric in Tables 1 and 2. Two sets of results can be identified in such tables: the first compiles all evolutionary yet non-grouping approaches for the SLP (Table 1), and the second corresponding to the grouping approaches published by the authors in [3,17], along with the hybrid DGHS herein proposed (Table 2). In the simulated scenarios the second result set clearly outperforms the first one, whereas the best performance in terms of statistical stability (i.e. lowest standard deviation) is attained by the proposed DGHS scheme. This fact evidences that the proposed dual technique is more exploitative than previous approaches, conclusion that is further buttressed by two particular observations: 1) in the fourth scenario, DGHS with parameter set λ_1 yields a higher mean but smaller standard deviation than their non-dual counterparts, whereas DGHS with λ_2 trades a higher standard deviation for a slightly lower mean; and 2) in the sixth simulated network instance,

the minimum value achieved by HGGA and GHS (namely, 5059) is not reached by the proposed scheme with neither λ_1 nor λ_2 , but the mean and standard deviation values are significantly outperformed by DGHS with both parameter sets. As a matter of fact, DGHS with λ_2 renders a more explorative behavior – lower *min* value and higher *mean* and *std* – with respect to DGHS with λ_1 .

Table 2. Comparison between the proposed DGHS and other evolutionary algorithms

ID	HGGA [3]	GHS [17]	DGHS, λ_1	DGHS, λ_2
1	3800/3813/17	3800/3800/0	3800/3800/0	3800/3800/0
2	3806/3819/18	3806/3807/1	3806/3806/0	3806/3806/0
3	3792/3807/16	3792/3795/10	3792/3792/0	3792/3792/0
4	4455/4498/26	4455/4464/13	4455/4468/4	4455/4464/11
5	4724/4745/28	4724/4728/4	4724/4724/0	4724/4724/0
6	5059/5127/47	5059/5080/19	5062/5067/6	5061/5070/11
7	7054/7058/10	7054/7057/14	7054/7054/0	7054/7054/0

5 Conclusions

In this paper a novel Dual Grouping Harmony Search (DGHS) algorithm has been presented for dealing with the so-called Switch Location Problem (SLP). The main difference of this work with respect to previous GHS approaches in the literature is the duality characterizing the application of the GHS refining procedures, which operate not only on the assignment part but also on the grouping part comprising the grouping-encoded solutions. In addition, local search and repair methods are inserted into the algorithm thread so as to allow improving the search capability of the proposed algorithm. The statistical performance of our proposal is analyzed in 7 instances of the SLP problem, with different network sizes and densities. The obtained results are compared to different evolutionary schemes found in the literature related to the SLP paradigm. Based on this simulation-based study we conclude that the duality of the GHS operators is crucial for enhancing the exploitative behavior of the algorithm, outperforming – in terms of a lower standard deviation and similar or lower mean and minimum values – other existing approaches in all the simulated scenarios.

References

1. Pierre, S., Elgibaoui, A.: Improving Communications Networks' Topologies using Tabu Search. In: 22nd Annual Conference on Local Computer Networks (LCN), pp. 44–53 (1997)
2. Calegari, P., Guidec, F., Kuonen, P., Wagner, D.: Genetic Approach to Radio Network Optimization for Mobile Systems. In: IEEE Vehicular Technology Conference, vol. (2), pp. 755–759 (1997)
3. Alonso-Garrido, O., Salcedo-Sanz, S., Agustin-Blas, L.E., Ortiz-Garcia, E.G., Perez-Bellido, A.M., Portilla-Figueras, J.A.: A Hybrid Grouping Genetic Algorithm for the Multiple-Type Access Node Location Problem. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 376–383. Springer, Heidelberg (2009)

4. Ceselli, A.: Two Exact Algorithms for the Capacitated P-Median Problem. *4OR: A Quarterly Journal of Operations Research* 1(4), 319–340 (2003)
5. Ceselli, A., Righini, G.: A Branch-and-Price Algorithm for the Capacitated P-Median Problem. *Networks* 45(3), 125–142 (2005)
6. Ceselli, A., Liberatore, F., Righini, G.: A Computational Evaluation of a General Branch-and-Price Framework for Capacitated Network Location Problems. *Annals of Operations Research* 167, 209–251 (2009)
7. Osman, I., Christofides, N.: Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search. *International Transactions on Operational Research* 1(3), 317–336 (1994)
8. Maniezzo, V., Mingozzi, A., Baldacci, R.: A Bionomic Approach to the Capacitated P-Median Problem. *Journal of Heuristics* 4, 263–280 (1998)
9. Baldacci, R., Hadjiconstantinou, E., Maniezzo, V., Mingozzi, A.: A New Method for Solving Capacitated Location Problems based on a Set Partitioning Approach. *Computers & Operations Research* 29(4), 365–386 (2002)
10. Lorena, L.A.N., Furtado, J.C.: Constructive Genetic Algorithm for Clustering Problems. *Evolutionary Computation* 9(3), 309–327 (2001)
11. Lorena, L.A.N., Senne, E.: A Column Generation Approach to Capacitated P-Median Problems. *Computers & Operations Research* 31(6), 863–876 (2004)
12. Scheuerer, S., Wendolsky, R.: A Scatter Search Heuristic for the Capacitated Clustering Problem. *European Journal of Operational Research* 169(2), 533–547 (2006)
13. Diaz, J.A., Fernandez, E.: Hybrid Scatter Search and Path Relinking for the Capacitated P-Median Problem. *European Journal of Operational Research* 169(2), 570–585 (2006)
14. Osman, I., Ahmadi, S.: Guided Construction Search Metaheuristics for the Capacitated P-Median Problem with Single Source Constraint. *Journal of the Operational Research Society* 18, 339–348 (2006)
15. Santos-Correa, E., Steiner, M.T., Freitas, A.A., Carnieri, C.: A Genetic Algorithm for Solving a Capacitated P-Median Problem. *Numerical Algorithms* 35(4), 373–388 (2004)
16. Salcedo-Sanz, S., Yao, X.: A Hybrid Hopfield Network-Genetic Algorithm Approach for the Terminal Assignment Problem. *IEEE Transactions on Systems, Man, and Cybernetics, PartB: Cybernetics* 34(6), 2343–2353 (2004)
17. Gil-Lopez, S., Del Ser, J., Landa-Torres, I., Garcia-Padrones, L., Salcedo-Sanz, S., Portilla-Figueras, J.A.: On the Application of a Novel Grouping Harmony Search Algorithm to the Switch Location Problem. In: Chatzimisios, P., Verikoukis, C., Santamaría, I., Laddomada, M., Hoffmann, O. (eds.) *MOBILIGHT 2010. LNICS*, vol. 45, pp. 662–672. Springer, Heidelberg (2010)
18. Reville, C., Swain, R.: Central Facilities Location. *Geographical Analysis* 2, 30–42 (1970)
19. Falkenauer, E.: The Grouping Genetic Algorithms-Widening the Scope of the GAs. *Belgian Journal of Operations Research, Statistics and Computer Science* 33, 79–102 (1993)
20. Falkenauer, E.: A New Representation and Operators for Genetic Algorithms Applied to Grouping Problems. *Evolutionary Computation*, 123–144 (1994)
21. Abuali, F.N., Schoenefeld, D.A., Wainwright, R.L.: Terminal Assignment in a Communications Network using Genetic Algorithm. In: *22nd Annual ACM Computer Science Conference*, pp. 74–81 (1994)

Optimal Evolutionary Wind Turbine Placement in Wind Farms Considering New Models of Shape, Orography and Wind Speed Simulation

B. Saavedra-Moreno, S. Salcedo-Sanz, A. Paniagua-Tineo,
J. Gascón-Moreno, and J.A. Portilla-Figueras

Department of Signal Theory and Communications, Universidad de Alcalá,
28871 Alcalá de Henares, Madrid, Spain
`sancho.salcedo@uah.es`

Abstract. In this paper we present a novel evolutionary algorithm for optimal positioning of wind turbines in wind farms. We consider a realistic model for the wind farm, which includes orography, shape of the wind farm, simulation of the wind speed and direction, and costs of installation, connection and road construction among wind turbines. Several experiments show that the proposed evolutionary approach obtains very good solutions which maximize power production, and takes into account the different constraints of the problem.

1 Introduction

Wind power is one of the most promising sources of renewable energy in the world. As an example, wind power installed worldwide by the end of 2009 reaches a total of 157 GW, of which about 76 GW correspond to Europe, and 19 GW only to Spain [1]. Thus, wind power represents over 12% of the total power consumed in countries such as USA, Germany or Spain, and it is expected that this percentage grows up to an amazing 20% by 2025. This figure situates wind energy as one of the main actors in the energetic mix of different countries, which are definitely betting for its development [2].

The majority of the wind power consumed in the world is generated in large facilities, known as *wind farms*. Automatic wind farm design is a topic gaining popularity among wind farm designers and engineers in the last few years. There is an increasing number of articles tackling this problem, applying successfully computational intelligence techniques, mainly evolutionary algorithms [3]. The seminal paper in the use of evolutionary computation techniques to wind farm design is the work by Mosetti et al., [4]. This paper proposed a genetic algorithm to tackle the problem of the optimal positioning of turbines in a wind farm. The model proposed in [4] consists in modeling the wind farm as a square divided into cells in which turbines can be situated. A useful wake model was proposed and several experiments considering different average wind speed and direction were presented. This initial work has been the base of different recent approaches which have improved the initial model. For example, in [5] Grady et al. showed

that better results can be obtained in the problem by improving the genetic algorithm used, using the same model as in [4]. Another improvement with the same model has been recently proposed by Emami et al. in [6]. That paper proposed a modification of the objective function of the problem, to take into account deployment cost and efficiency of the turbines. The authors showed that this modification leads to better design results than previous approaches using a standard genetic algorithm. Another interesting and recent work is the paper by Riquelme et al. [7], where a variable-length genetic algorithm with novel procedures of crossover is applied to solve a problem of optimal positioning of wind turbines, considering monetary cost as the objective optimization function. The authors showed that their variable-length evolutionary approach is able to obtain good results in terms of the objective function, considering different types of wind turbines to be used. A similar approach using a hybrid evolutionary algorithm was previously presented in Martínez et al. [8]. This approach has been further studied recently in [1] and [9]. It is also significant the work by Wang et al. [10], where a new improved wind and turbine models have been considered within a genetic algorithm. The authors have shown that this new model is able to produce better results than previous approaches in the literature.

In this paper we present a modified wind farm optimization problem, which includes several novelties in order to make the problem closer to reality than previous approaches. A wind farm shape model, an orography model and the inclusion of simulation to model the wind speed are the main new points included in this paper. In addition, we present a novel evolutionary algorithm to look for the optimal solution of the problem. We will show that the proposed approach is able to obtain good and feasible solutions for the problem, with a balance between the computational cost and the quality of the solution obtained.

The rest of the paper is structured as follows: next section presents the main novelties included in the optimization model considered in this paper. Section 3 presents the evolutionary algorithm proposed. The experimental part of the paper is shown in Section 4, and Section 5 closes the paper giving some final remarks.

2 Novel Optimization Model Proposed

This section presents the optimization model assumed in this paper. We have tried to include several points to make the problem closer to a real wind farm design than previous approaches. The main novelties in our model are the inclusion of the wind farm shape, an orography model, wind speed simulation and finally the inclusion of a cost model based on benefit/investment terms.

2.1 Wind Farm Shape Model

Previous approaches in the literature have not taken into account the problem of the wind farm shape. Basically the majority of previous approaches consider

square wind farms, divided into cells where turbines could be positioned [4], [5]. This square-based approach is interesting since it introduces a nice way of managing the different possible points where a turbine can be installed, but the problem is that it cannot model in a realistic way the design of a real wind farm. In this paper we propose an easy way to consider different shapes for the wind farm, keeping the idea of the square cells to model a possible point to locate a wind turbine. The idea is really simple: over a square of length $K \times K$ cells which serves as a background, we define a binary template \mathcal{T} , also $K \times K$ which describes the *allowed zones* to install turbines, in such a way that $\mathcal{T}_{ij} = 1$ stands for a point included in the wind farm area, and $\mathcal{T}_{ij} = 0$ stands for a point outside of the wind farm. Note that this way we can consider almost any shape for the wind farm.

2.2 Wake, Orography Model and Wind Speed Simulation

In this paper we consider as wake model the one previously described in [5]. Though it is a simple model, it works really well to simulate a real turbine's wake, obtaining a good balance between model's complexity and final performance. Moreover, we have introduced several new points in this work, in order to make the problem more realistic. First of all, note that none of the previous approaches to the problem takes into account the wind farm's orography or variations on wind speed. The existence of hills within the wind farm makes that the wind speed is different at the top of the hill or at the bottom of the corresponding valley. In order to take this important point into account, we introduce the concept of *wind speed multipliers*, in such a way that a higher point will be characterized by having a larger wind multiplier. Thus, when the wind speed associated with a given point in the wind farm is modified by means of the wind multiplier, we are taking into account the orography of the wind farm.

The wind speed in a given point of the wind farm has been calculated in the following way in our model: we consider a given wind rose for the wind farm, and then, for each direction of the wind rose, a set of Monte Carlo simulations of t years wind using a Weibull probability density function for the wind speed module. The result of the Monte Carlo simulation is weighted by the corresponding probability extracted from the wind rose and by the wind speed multipliers, for including the orography in each point. Following this methodology, we obtain a more realistic model for the wind speed in the wind farm than previous approaches. Then a standard power curve for the turbine considered is used to obtain the power production associated with the wind speed in a wind turbine.

2.3 Cost Model

The cost model used in this work is based on a simplified model of investment/benefit considerations, similar to the one proposed in [7]. Specifically, our cost model includes wind turbines installation cost (C_i) and connection between turbines and road construction costs (C_{ij}^C), modeled as the Euclidean distance

between turbine i and j , and also the net benefit obtained from the energy produced in t years (B_t). All this parameters are measured in Euros. The objective function to be maximized is:

$$\varphi(\Xi) = B_t - N \cdot C_i - \sum_{i=1}^N \sum_{j<i} C_{ij}^C \quad (1)$$

where N stands for the number of wind turbines installed in the wind farm. Note that we do not consider alternative costs such as the operational costs (OPEX) in this objective function. However, it is good enough to show the performance of the different compared algorithms.

3 Proposed Evolutionary Algorithm

Evolutionary algorithms (EAs) [11], are robust problems' solving techniques based on natural evolution processes. They are population-based techniques which codify a set of possible solutions to the problem, and evolve it through the application of the so called *evolutionary operators* [11]. Next we describe the main characteristics of the evolutionary algorithm proposed in this paper, including the algorithm's initialization and selection, crossover and mutation operators proposed.

1. Generate an initial population of μ individuals (solutions). Let t be a counter for the number of generations, set it to $t = 1$. Each individual is taken as a matrix of integer vectors $\Xi = (x_i, y_i)$, $i = 1, \dots, N$, where each x_i stands for the x -coordinate of turbine i in the background square considered, and each y_i stands for the y -coordinates ($x_i = 1, \dots, K$, $y_i = 1, \dots, K$). Note that every location point of a given solution Ξ , lets say (x, y) , must fulfil a number of requirements to be considered as feasible: first, all the location points should be within the wind farm surface, i.e, the associate value in matrix \mathcal{T} must be 1 ($\mathcal{T}_{xy} = 1, \forall x, y$). Second, a given turbine situated at a point (x, y) must be at least at a distance \mathcal{D} of any other turbine. The initial individuals of the population are generated in such a way that these constraints are fulfilled.
2. Evaluate the fitness value for each individual Ξ of the population using the problem's objective function φ .
3. Generate an offspring population, of length μ , applying one-point crossover operator [11] and mutation operator. The crossover operator is applied in the traditional way. On the other hand, mutation operator is carried out by randomly changing couples of specific points (x_i, y_i) to (x'_i, y'_i) .
4. Correct the offspring population in such a way that all their individuals are feasible (fulfil the problem's constraints). Note that the crossover operator may produce solutions within a distance \mathcal{D} of another turbine, and mutation operator, in addition, may produce solutions which are outside the wind farm

points defined by matrix \mathcal{T} . In order to correct these unfeasible solutions, a modification of unfeasible points is applied after each round of crossover and mutation, by using two random numbers $r, s \in [-2\mathcal{D}, 2\mathcal{D}]$: a given unfeasible point (x, y) is modified to $(x + r, y + s)$ until it is feasible.

5. Selection: Pass the best individual found so far in the evolution to the next generation (elitism process). Conduct then pairwise comparison over the union of parents and offspring remaining: for each individual, p opponents are chosen uniformly at random from all the parents and offspring. The best individual in these p is selected to survive for the next generation. This process is repeated until a new parent generation of μ individuals is obtained.
6. Evaluate the fitness value for each individual Ξ of the new parent population using the problem's objective function φ .
7. Stop if the stopping criterion is satisfied, and if not, set $t = t + 1$ and go to Step 3. In this case, the stopping criterion established is that the best solution found by the algorithm is not improved during \mathcal{K} generations, or, alternatively, the algorithm reaches to a maximum number of generations *max_ite*.

4 Experiments and Results

This section presents the simulations and results carried out in this paper to show the effectiveness of the proposed EA. First we describe a greedy-constructive approach used for comparison purposes, and then we show the results obtained by the EA in several randomly-generated instances.

4.1 A Greedy-Constructive Approach for Comparison Purposes

For comparison purposes, we have implemented a greedy-constructive approach which is also able to tackle the problem. It consists in the following steps:

1. Locate the point in the wind farm with maximum wind and set there the first wind turbine.
2. Modify the wind speed in neighbor points to turbines taking into account the wake model and wind rose considered.
3. Locate the point in the wind farm which maximizes Equation (II)
4. If N wind turbines have been installed, then stop. Otherwise go to step 2.

This greedy approach is able to come up with a feasible solution to the problem. It is fast, and its performance in terms of objective function will be evaluated together with the proposed EA.

4.2 Results

We have run a number of simulations with the following parameters: Figure 1 shows the wind rose considered, extracted from a real wind farm in southern Spain. Weibull's distribution parameters used in the model of the wind speed are $\lambda = 10$ and $\beta = 1.6$. The positioning of $N = 20$ wind towers in the wind farm is considered. The cost of each tower has been estimated to be $C_i = 10^6$ Euros. Other parameters of the simulations are the following: we consider a basic cell size of $10m \times 10m$, with $K = 500$, $\mathcal{D} = 90m$. The wind farm shape has been randomly obtained, and can be seen in Figure 2, where we show some greedy and EA results. We estimate a cost for the connection of wind towers and road construction (C_{ij}^c) of 10^5 Euros/Km. Regarding the calculation of the estimated benefit $B_i(\Xi)$, we have considered 75 Euros/Mwatt-h, 3000 effective-hours/year and a period of $t = 10$ years of simulation. Regarding the parameters of the EAs, values of $\mu = 50$, $p = 10$, $\mathcal{K} = 75$ have been set.

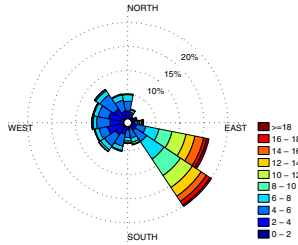


Fig. 1. Example of a real wind rose in a wind farm of southern Spain used in the simulations of wind farm design in this paper

Table 1 shows the comparison between the greedy-constructive approach and the EA, for one wind farm shape, and 10 different (randomly generated) wind speed multipliers. It is easy to see that the EA is able to obtain better solutions for the problem in all the instances run. This shows that, as has also been shown in other works in the literature with different optimization models, evolutionary computation techniques are shown to be a really good option in wind farm design problems.

Figure 2 shows an example of the final wind turbines disposition obtained by the greedy heuristic and proposed Evolutionary Algorithm (instance 10), over the wind farm with wind multipliers depicted. Circles of length \mathcal{D} are also shown in this figure to show the feasibility of the solution. Note that these solutions fulfil the problem's constraints (the wind turbines should be positioned within the wind farm surface, at a minimum distance \mathcal{D} of the nearest turbine). Note also that these solutions are optimized following Equation (1), so they take into account the wind farm orography and the different costs considered.

Table 1. Objective function values (in Euros/ 10^7), in the 10 different simulations performed, obtained by the Greedy-constructive heuristic and proposed Evolutionary Algorithm (EA)

# instance	Greedy heuristic	Proposed EA
1	2.283	3.613
2	3.305	5.029
3	1.002	3.741
4	1.697	4.364
5	2.171	3.818
6	1.849	3.964
7	1.683	3.334
8	1.776	4.530
9	1.201	4.046
10	2.813	3.934

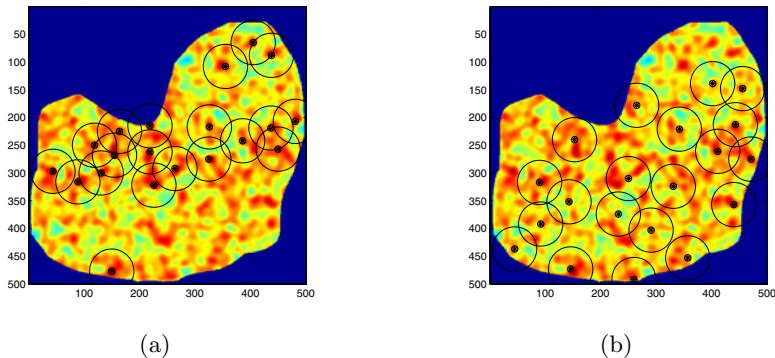


Fig. 2. Final wind turbines location, security radius and wind multipliers; (a) Greedy heuristic; (b) Evolutionary Algorithm

5 Conclusions

In this paper we have presented a novel evolutionary algorithm to solve a problem of optimal location of wind turbines in wind farms. A novel optimization model has also been proposed, which includes some new aspects such as wind farm shape, orography and different costs in the objective function. We have carried out several experiments where we have shown the good performance of the proposed evolutionary algorithm in the design of wind farms.

Acknowledgement

This work has been partially supported by Spanish Ministry of Industry, Tourism and Trading, under an Avanza 2 project, number TSI-020100-2010-663.

References

1. Serrano-González, J., González-Rodríguez, A.G., Castro-Mora, J., Riquelme-Santos, J., Burgos-Payán, M.: Optimization of wind farm turbines layout using an evolutive algorithm. *Renewable Energy* 35(8), 1–11 (2010)
2. Swisher, R., Real de Azua, C., Clendenin, J.: Strong winds on the horizon: wind power comes of age. *Proceedings of the IEEE* 89(12) (2001)
3. Réthoré, P.E.: State of the art in wind farm layout optimization. *Wind Energy Research*, <http://windenergyresearch.org/?p=979>
4. Mosetti, G., Poloni, C., Diviacco, B.: Optimization of wind turbine positioning in large wind farms by means of a genetic algorithm. *Journal of Wind Engineering and Industrial Aerodynamics* 51(1), 105–116 (1994)
5. Grady, S.A., Hussaini, M.Y., Abdullah, M.M.: Placement of wind turbines using genetic algorithms. *Renewable Energy* 30(2), 259–270 (2005)
6. Emami, A., Noghreh, P.: New approach on optimization in placement of wind turbines within wind farm by genetic algorithms. *Renewable Energy* (2010) (in press)
7. Riquelme-Santos, J., Burgos-Payan, M., Calero, J.M., Castro Mora, J.: An evolutive algorithm for wind farm optimal design. *Neurocomputing* 70(16-18), 2651–2658 (2007)
8. Martínez-Ramos, J.L., Castro, J., Riquelme-Santos, J., Burgos-Payán, M.: A hybrid evolutive algorithm for wind farm optimum network design. In: *Artificial Intelligence in Energy Systems and Power*, Madeira, Portugal, pp. 1–5 (2006)
9. Serrano-González, J., Riquelme-Santos, Burgos-Payán, M.: Wind farm optimal design including risk. In: *Proc. of the Modern Electric Power Systems*, Wroclaw, Poland (2010)
10. Wang, C., Yang, G., Li, X., Zhang, X.: Optimal micro-siting of wind turbines by genetic algorithms based on improved wind and turbine models. In: *Proc. of the 48th IEEE Conference on Decision and Control*, Shanghai, China, pp. 5092–5096 (2009)
11. Eiben, A.E., Smith, J.E.: *Introduction to evolutionary computing*. Springer, Heidelberg (2003)

Multi-Valued Neurons: Hebbian and Error-Correction Learning

Igor Aizenberg

Department of Computer Science,
Texas A&M University-Texarkana,
P.O. Box 5518, Texarkana, TX 75505-5518, USA
igor.aizenberg@tamut.edu

Abstract. In this paper, we observe some important aspects of Hebbian and error-correction learning rules for the multi-valued neuron with complex-valued weights. It is shown that Hebbian weights are the best starting weights for the error-correction learning. Both learning rules are also generalized for a complex-valued neuron whose inputs and output are arbitrary complex numbers.

Keywords: complex-valued neural networks, derivative-free learning, multi-valued neuron.

1 Introduction

The discrete multi-valued neuron (MVN) was introduced in [1]. This neuron is based on the concept of multiple-valued logic over the field of complex numbers, which was introduced in [2], then presented in detail in [3], and further developed in [4]. The continuous MVN was presented in [5]. MVN with a periodic activation function (MVN-P) was recently introduced in [6].

MVN is a neuron with complex-valued weights and inputs and output located on the unit circle. MVN has a number of remarkable properties. Its main property, which is simultaneously its main advantage over other artificial neurons, is its higher functionality. A single MVN-P can easily learn non-linearly separable input/output mappings, and, for example, classical XOR and Parity n non-linearly separable problems are about the simplest, which can be learned by a single MVN-P.

Perhaps, the most interesting application of MVN is MLMVN (a multilayer neural network based on multi-valued neurons), which was introduced in [5] and then developed in [7]. It outperforms many other machine learning techniques in terms of generalization capability, number of parameters employed and network complexity. Other interesting applications of MVN are, for example, associative memories [8-9].

Although MVN was comprehensively observed, for example, in [4], and [5, 6], there is still at least one interesting open problem in its learning theory. This is specificity of its Hebbian learning and the relation between its Hebbian and error-correction learning. We would like to consider these problems in this paper. We also will generalize both Hebb and error-correction MVN learning rules for a complex-valued neuron whose inputs and outputs are arbitrary complex numbers.

2 Multi-Valued Neuron (MVN)

The discrete MVN was proposed in [1]. It is based on the principles of multiple-valued threshold logic over the field of complex numbers. The discrete MVN implements a mapping between n inputs and a single output. This mapping is described by a multiple-valued (k -valued) threshold function of n variables $f(x_1, \dots, x_n)$. It is important to specify that we consider here multiple-valued logic over the field of complex numbers [3, 4]. While in traditional multiple-valued logic its values are encoded by the integers from the set $K = \{0, 1, \dots, k-1\}$, in the one over the field of complex numbers they are encoded by the k th roots of unity $E_k = \{\varepsilon^0, \varepsilon, \varepsilon^2, \dots, \varepsilon^{k-1}\}$, where $\varepsilon^j = e^{i2\pi j/k}$, $j = 0, \dots, k-1$, (i is an imaginary unity). A k -valued threshold function $f(x_1, \dots, x_n): O^n \rightarrow E_k$ (here O is a set of complex numbers located on the unit circle), which presents an input/output mapping implemented by the discrete MVN, is represented using $n+1$ complex-valued weights as follows

$$f(x_1, \dots, x_n) = P(w_0 + w_1 x_1 + \dots + w_n x_n), \quad (1)$$

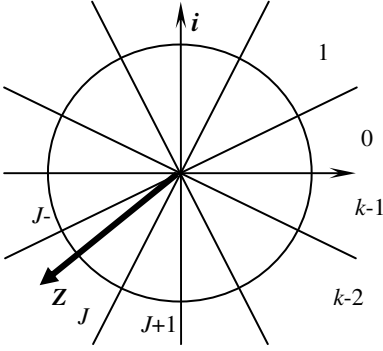
where x_1, \dots, x_n are the neuron inputs and w_0, w_1, \dots, w_n are the weights. The values of this function are the k th roots of unity: $\varepsilon^j = e^{i2\pi j/k}$, $j \in \{0, 1, \dots, k-1\}$, i is an imaginary unity. P is the activation function

$$P(z) = e^{i2\pi j/k}, \text{ if } 2\pi j/k \leq \arg z < 2\pi(j+1)/k, \quad (2)$$

where $j=0, 1, \dots, k-1$ are values of k -valued logic, $z = w_0 + w_1 x_1 + \dots + w_n x_n$ is the weighted sum, $\arg z$ is the argument of the complex number z . It is important to mention that function (2), which was introduced in [2], is historically the first known complex-valued activation function. Function (2) divides a complex plane into k equal sectors and maps the whole complex plane into a set of k th roots of unity (see Fig. 1). The continuous MVN has been presented in [5]. The continuous case corresponds to $k \rightarrow \infty$ in (2). If the number of sectors $k \rightarrow \infty$ (see Fig. 1), then the angular size of a sector tends to 0. Hence, an activation function in this case becomes simply a projection of the weighted sum $z = w_0 + w_1 x_1 + \dots + w_n x_n$ onto the unit circle:

$$P(z) = \exp(i(\arg z)) = e^{i \text{Arg } z} = z/|z|, \quad (3)$$

where z is the weighted sum, $\text{Arg } z$ is a main value of its argument (phase) and $|z|$ is the absolute value of the complex number z . Activation function (3) is illustrated in Fig. 2. It maps the whole complex plane into the unit circle.



$$P(z) = \exp(j \cdot i 2\pi / k)$$

Fig. 1. Geometrical interpretation of the discrete MVN activation function

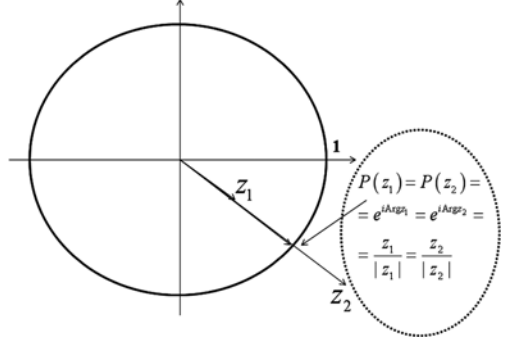


Fig. 2. Geometrical interpretation of the continuous MVN activation function

3 Generalized Learning Rules for Complex-Valued Neurons

3.1 Generalized Hebbian Learning

Hebbian learning for MVN has never been studied in detail. It was just slightly outlined in [1] and [8], but its mechanism was not discussed. Moreover, Hebbian learning for a complex-valued neuron whose inputs and outputs are arbitrary complex numbers was not yet considered at all.

The mechanism of the Hebbian learning for a complex-valued neuron is the same as the one for the classical threshold neuron and as it was described by D. Hebb in his seminal book [10]. This is the mechanism of the association.

Let us consider the Hebbian learning rule for a complex-valued neuron whose inputs and outputs are arbitrary complex numbers except 0. Let us have a learning set containing N n -dimensional learning samples

$(x_1^j, \dots, x_n^j); x_i^j \in C \setminus \{0\}; j = 1, \dots, N$; Let $\mathbf{f} = (f_1, \dots, f_N)^T$ be an N -

dimensional vector-column of the desired outputs. Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be N -dimensional vectors-columns of all the inputs $\mathbf{x}_r = (x_r^1, x_r^2, \dots, x_r^N)^T; r = 1, \dots, n$. Let

$\tilde{\mathbf{x}}_r = \left((x_r^1)^{-1}, (x_r^2)^{-1}, \dots, (x_r^N)^{-1} \right)^T; r = 1, \dots, n$. Let also $\mathbf{x}_0 = (1, 1, \dots, 1)^T$. Then

the generalized Hebbian learning rule for finding the weights w_1, \dots, w_n for a complex-valued neuron is as follows.

$$w_i = \frac{1}{(n+1)} \left(f_1 (x_i^1)^{-1} + f_2 (x_i^2)^{-1} + \dots + f_N (x_i^N)^{-1} \right); i = 0, \dots, n. \quad (4)$$

Particularly, for MVN whose inputs and output are located on the unit circle, the generalized Hebbian rule (4) is transformed to

$$w_i = \frac{1}{(n+1)} (\mathbf{f}, \mathbf{x}_i) = \frac{1}{(n+1)} (f_1 \bar{x}_i^1 + f_2 \bar{x}_i^2 + \dots + f_N \bar{x}_i^N), i = 0, \dots, n, \quad (5)$$

where “bar” is a symbol of complex conjugation. Equation (5) is easily obtained from (4) when $x_i \in E_k$ or $x_i \in O; i = 1, \dots, n$ because in such a case $|x_i| = 1; i = 1, \dots, n$ and $\bar{x}_i = (x_i)^{-1}; i = 1, \dots, n$.

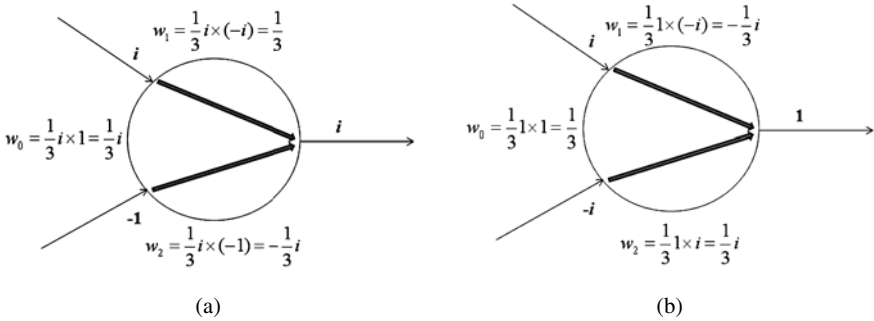


Fig. 3 Calculation of the MVN weights using the Hebbian rule (5) for the two neuron inputs and for a single learning sample: the weight is equal to the product of the desired output and the complex-conjugated input

Let us now consider the following examples. Let $k=4$ in the discrete MVN activation function (2). Thus, our MVN works in 4-valued logic whose values are encoded by the elements of the set $E_4 = \{1, i, -1, -i\}$ ($i = \varepsilon_4 = e^{i2\pi/4}$ is an imaginary unity, which is also a primitive 4th root of a unity). Let us now consider MVN with the two inputs.

In Fig. 3a, the desired MVN output is i , while its two inputs are i and -1 , respectively. According to (5) $w_0 = \frac{1}{3} i; w_1 = \frac{1}{3} f_1 \bar{x}_1 = \frac{1}{3} i \cdot (-i) = \frac{1}{3}$, and $w_2 = \frac{1}{3} f_1 \bar{x}_2 = \frac{1}{3} i \cdot (-1) = -\frac{1}{3} i$. The weighted sum is $z = \frac{1}{3} i + \frac{1}{3} i - \frac{1}{3} i \cdot (-1) = i$, and according to (2) the neuron output is $P(i) = i$.

Thus, the weight w_1 pass the input x_1 to the output, while the weight w_2 “rotates” the input x_2 passing it to the output.

In Fig. 3b, the desired MVN output is 1, while its two inputs are i and $-i$, respectively. According to (5) $w_0 = \frac{1}{3}$; $w_1 = \frac{1}{3} f_1 \bar{x}_1 = \frac{1}{3} \cdot (-i) = -\frac{1}{3}i$, and $w_2 = \frac{1}{3} f_1 \bar{x}_2 = \frac{1}{3} \cdot i = \frac{1}{3}i$. The weighted sum is $\frac{1}{3} - \frac{1}{3}i \cdot i + (\frac{1}{3}i) \cdot (-i) = 1$, and according to (2) the neuron output is $P(1) = 1$. Thus, both weights w_1 and w_2 “rotate” the inputs x_1 and x_2 passing them to the output.

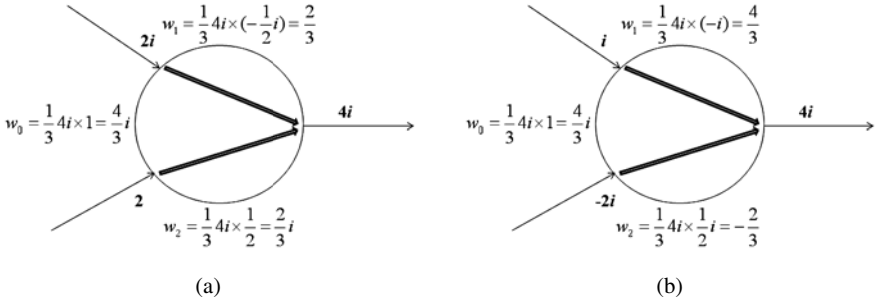


Fig. 4 Calculation of the weights for a complex-valued neuron with arbitrary complex inputs and outputs using the generalized Hebbian rule (4)

Fig. 4 presents how the generalized Hebb rule (4) is used for a complex-valued neuron whose inputs and output are arbitrary complex numbers (except 0) and whose activation function is linear.

In Fig. 4a, the desired neuron output is $4i$, while its two inputs are $2i$ and 2 , respectively. According to (4)

$$w_0 = \frac{1}{3} 4i \cdot 1 = \frac{4}{3}i; w_1 = \frac{1}{3} f_1(x_1)^{-1} = \frac{1}{3} 4i \cdot (-\frac{1}{2}i) = \frac{2}{3}; w_2 = \frac{1}{3} f_1(x_2)^{-1} = \frac{1}{3} 4i \cdot \frac{1}{2} = \frac{2i}{3}.$$

The weighted sum and the output of the neuron is $z = \frac{4}{3}i + \frac{2}{3}2i + \frac{2}{3}i \cdot 2 = 4i$.

In Fig. 4b, the desired MVN output is $4i$, while its two inputs are i and $-2i$, respectively. According to (4)

$$w_0 = \frac{1}{3} 4i \cdot 1 = \frac{4}{3}i; w_1 = \frac{1}{3} f_1(x_1)^{-1} = \frac{1}{3} 4i \cdot (-i) = \frac{4}{3}; w_2 = \frac{1}{3} f_1(x_2)^{-1} = \frac{1}{3} 4i \cdot \frac{1}{2} = \frac{2i}{3}.$$

The weighted sum and the output of the neuron is $z = \frac{4}{3}i + \frac{4}{3}i - \frac{2}{3} \cdot (-2i) = 4i$.

Thus in all examples, the weights obtained according to generalized Hebb rule guarantee that the actual neuron output coincides with its desired output.

When there are more learning samples in the learning set, the Hebbian learning rule usually does not lead to a weighting vector, which implements the corresponding input/output mapping. However, the learning algorithm, which is based on the error-correction learning rule, converges much faster when the learning process starts from this (Hebbian) vector than when it starts from a random vector.

3.2 Generalized Error-Correction Learning

There are several approaches to the MVN learning. MVN learning algorithm, which was most comprehensively presented in [4], is based on the error-correction learning rule. It is identical for both discrete and continuous neurons. The most important property of MVN learning is that it is derivative-free. Let D be the desired neuron output and Y be the actual one. Then $\delta = D - Y$ is the error, which determines the adjustment of the weights performing as follows

$$w_0^{r+1} = w_0^r + \frac{C_r}{(n+1)} \delta; \quad w_i^{r+1} = w_i^r + \frac{C_r}{(n+1)} \delta \bar{x}_i; \quad i = 1, \dots, n, \quad (6)$$

where \bar{x}_i is the i th input complex-conjugated, n is the number of neuron inputs, δ is the neuron's error, r is the number of the learning iteration, w_i^r is the current i th weight (to be corrected), w_i^{r+1} is the following i th weight (after correction), C_r is the learning rate (it may always be equal to 1).

Let us now again consider a complex-valued neuron whose inputs and output are arbitrary complex numbers (except 0). For simplicity, but without loss of generality, we may consider that the activation function of this neuron is identical (equals the weighted sum $z = w_0 + w_1 x_1 + \dots + w_n x_n$). It is important that in the error-correction learning rule (6) the adjusting term, which is added to the i th weight to correct it, contains a factor \bar{x}_i . Let us generalize rule (6) for a complex-valued neuron whose inputs and output are arbitrary complex numbers (except 0):

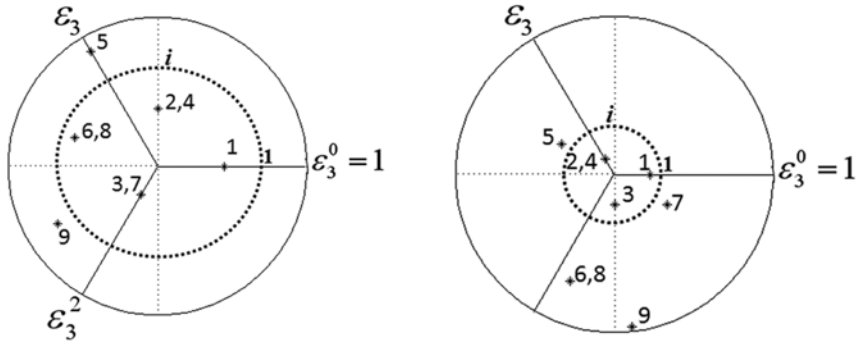
$$w_0^{r+1} = w_0^r + \frac{C_r}{(n+1)} \delta; \quad w_i^{r+1} = w_i^r + \frac{C_r}{(n+1)} \delta x_i^{-1}; \quad i = 1, \dots, n. \quad (7)$$

Let us show that if the weights are corrected according to the generalized error-correction rule (7), then the updated neuron output is equal exactly to the desired value. Let D be the desired neuron output and Y be the actual one. Then $\delta = D - Y$ is the error. Let us use (7) to adjust the weights:

$$\begin{aligned} \tilde{z} &= \tilde{w}_0 + \tilde{w}_1 x_1 + \dots + \tilde{w}_n x_n = \\ &= \left(w_0 + \frac{1}{n+1} \delta \right) + \left(w_1 + \frac{1}{n+1} \delta x_1^{-1} \right) x_1 + \dots + \left(w_n + \frac{1}{n+1} \delta x_n^{-1} \right) x_n = \\ &= \underbrace{w_0 + w_1 x_1 + \dots + w_n x_n}_z + \underbrace{\frac{1}{n+1} \delta + \dots + \frac{1}{n+1} \delta}_{n+1 \text{ times}} = z + \delta = Y + \delta + D. \end{aligned} \quad (8)$$

Thus, a single learning step with the modified error-correction rule makes it possible to reach the desired output immediately. Equation (8) shows that after the weights are corrected, the weighted sum is changed exactly by δ , that is by the error.

Since MVN inputs x_1, \dots, x_n are located on the unit circle ($x_i \in E_k$ or $x_i \in O; i = 1, \dots, n$), then $|x_i| = 1; i = 1, \dots, n$ and $\bar{x}_i = (x_i)^{-1}; i = 1, \dots, n$, and taking this into account we can easily conclude that the MVN learning rule (6) follows from the generalized error-correction learning rule (7).



(a) Distribution of the weighted sums with the Hebbian weighting vector

(b) Iteration 1

Fig. 5. Movement of the weighted sum z after the correction of the weights according to (6) starting from the Hebbian weighting vector

Let us illustrate how the error-correction learning can follow the Hebbian one. Let us consider the input/output mapping described by the function $f_{\max}(x_1, x_2) = \max(x_1, x_2)$ for $k=3$. Evidently, $\mathbf{f} = (\epsilon^0, \epsilon, \epsilon^2, \epsilon, \epsilon, \epsilon^2, \epsilon^2, \epsilon^2, \epsilon^2)^T$.

Let us find the Hebbian weights for $f_{\max}(x_1, x_2)$. According to (5) we obtain the following Hebbian weighting vector $W_H = (-0.11 + 0.06i, 0.167 - 0.032i, 0.167 - 0.032i)$. This weighting vector does not implement the function $f_{\max}(x_1, x_2)$.

Distribution of the weighted sums with the weighting vector W_H is shown in Fig. 5a. The outputs for five learning samples out of nine (samples 2, 4, 6, 8, 9) are incorrect. However, they can easily be corrected using the learning algorithm based on the error-correction rule (6). After just a single learning iteration the actual outputs for all the learning samples coincide with the desired outputs (see Fig. 5b). If the same learning process starts from the random weights, more than a single iteration is always needed to complete it.

Acknowledgement

This work is supported by the National Science Foundation under Grant 0925080.

References

1. Aizenberg, N.N., Aizenberg, I.N.: CNN Based on Multi-Valued Neuron as a Model of Associative Memory for Gray-Scale Images. In: The Second IEEE Int. Workshop on Cellular Neural Networks and their Applications, October 1992, pp. 36–41. Technical University Munich, Germany (1992)
2. Aizenberg, N.N., Ivaskiv, Y.L., Pospelov, D.A.: About one generalization of the threshold function. *Doklady Akademii Nauk SSSR (The Reports of the Academy of Sciences of the USSR)* 196(6), 1287–1290 (1971) (in Russian)
3. Aizenberg, N.N., Ivaskiv, Y.L.: *Multiple-Valued Threshold Logic*. Naukova Dumka Publisher House, Kiev (1977) (in Russian)
4. Aizenberg, I., Aizenberg, N., Vandewalle, J.: *Multi-valued and universal binary neurons: theory, learning, applications*. Kluwer Academic Publishers, Boston (2000)
5. Aizenberg, I., Moraga, C.: Multilayer Feedforward Neural Network Based on Multi-Valued Neurons (MLMVN) and a Backpropagation Learning Algorithm. *Soft Computing* 11(2), 169–183 (2007)
6. Aizenberg, I.: A Periodic Activation Function and a Modified Learning Algorithm for a Multi-Valued Neuron. *IEEE Transactions on Neural Networks* 21(12), 1939–1949 (2010)
7. Aizenberg, I., Paliy, D., Zurada, J.M., Astola, J.: Blur Identification by Multilayer Neural Network based on Multi-Valued Neurons. *IEEE Transactions on Neural Networks* 19(5), 883–898 (2008)
8. Jankowski, S., Lozowski, A., Zurada, J.M.: Complex-Valued Multistate Neural Associative Memory. *IEEE Trans. Neural Networks* 7(6), 1491–1496 (1996)
9. Muezzinoglu, M.K., Guzelis, C., Zurada, J.M.: A New Design Method for the Complex-Valued Multistate Hopfield Associative Memory. *IEEE Trans. Neural Networks* 14(4), 891–899 (2003)
10. Hebb, D.O.: *The Organization of Behavior*. John Wiley & Sons, New York (1949)

Multi-label Testing for CO²RBFN: A First Approach to the Problem Transformation Methodology for Multi-label Classification

A.J. Rivera, F. Chartre, M.D. Pérez-Godoy, and María Jose del Jesus

Dep. of Computer Science, University of Jaén, Jaén, Spain

arivera@ujaen.es

<http://simidat.ujaen.es>

Abstract. While in traditional classification an instance of the data set is only associated with one class, in multi-label classification this instance can be associated with more than one class or label. Examples of applications in this growing area are text categorization, functional genomics and association of semantic information to audio or video content. One way to address these applications is the Problem Transformation methodology that transforms the multi-label problem into one single-label classification problem, in order to apply traditional classification methods. The aim of this contribution is to test the performance of CO²RBFN, a cooperative-competitive evolutionary model for the design of RBFNs, in a multi-label environment, using the problem transformation methodology. The results obtained by CO²RBFN, and by other classical data mining methods, show that no algorithm outperforms the other on all the data.

Keywords: Multi-label Classification, RBFNs, Problem Transformation.

1 Introduction

Recently, applications where an instance of the data set is associated with several labels or classes have been growing. For example in text categorization, each document can be classified as belonging to different predefined topics, such as *education* and *health*, a movie may belong to the classes *action* and *thriller*, or a song can be categorized as *rock* and *pop*. These data sets are called multi-label data sets and the related classification task is called multi-label classification [1, 3].

The first applications [1] in this area dealt with text categorization problems but other examples are: functional genomics, semantic association of images, scene classification, medical diagnosis or directed marketing.

The different approaches that address multi-label classification can be categorized into two groups: Problem Transformation and Algorithm Adaptation. The first group of algorithms transforms the multi-label problem into one single-label classification problem. In the second group, classical algorithms are adapted to handle multi-label data directly.

Radial Basis Function Networks (RBFNs) are one of the most important Artificial Neural Network (ANN) paradigms in the machine learning field. An RBFN is a feed-forward ANN with a single layer of hidden units, called radial basis functions (RBFs) [1]. The overall efficiency of RBFNs has been proved in many areas [2] like pattern classification, function approximation and time series prediction.

An important paradigm for RBFN design is Evolutionary Computation [6]. There are different proposals in this area with different scheme representations: Pittsburgh [8], where each individual is a whole RBFN, and cooperative-competitive [12], where an individual represents a single RBF.

Authors have developed an algorithm for the cooperative-competitive design of Radial Basis Functions Networks, CO²RBFN [10], that has been successfully used in classical and imbalanced classification.

The purpose of the present paper is to test CO²RBFN in multi-label classification, exploring this field. For this initial approach and based on the first group of techniques mentioned, multi-label data sets are transformed into single-label data sets. The results obtained are compared with other traditional techniques in data mining.

The text is organized as follows. In Section 2, multi-label classification and the solutions provided for it in the specialized bibliography are described. The cooperative-competitive evolutionary model for the design of RBFNs applied to classification problems, CO²RBFN, is described in Section 3. The analysis of the experiments and the conclusions are shown in Sections 4 and 5.

2 Multi-label Classification

Classification is one of the most important applications of data mining. In a classification environment, a mapping from an input space X^n to a finite set of classes L with $L = \{l_1, l_2, \dots, l_k\}$, must be established. Considering a training set D with p patterns or instances:

$$D = \{(\mathbf{x}_u, l_u) | \mathbf{x}_u \in X^n, l_u \in L, u = 1, \dots, p\} \quad (1)$$

where \mathbf{x}_u is the feature vector and l_u is the class it belongs to. When $|L| = 2$ the classifier is binary. If $|L| > 2$ a multi-class classifier is needed. In any case, each instance is only associated with one of the classes.

However, there is an important number of problems where each instance can be simultaneously associated with a subset of classes or labels $Y \subseteq L$. These problems are known as multi-label classification problems. Even binary classification and multi-class classification can be seen as special cases of multi-label problems where the number of labels assigned to each instance is 1.

As mentioned previously, there are two main ways to address multi-label classification problems [11]: Problem Transformation and Algorithm Adaptation approaches. With the problem transformation (algorithm independent) method, the original problem is transformed into a set of single-label problems. The most popular of these transformations are:

- Label powerset (LP): This method considers as a single label the subset of labels associated with each instance of the data set. Drawbacks of this method include the fact that the data set obtained can contain a large number of classes, and some of these classes can be associated with a limited number of examples.
- Binary relevance (BR): This method, based on the one-against-all techniques, creates a new data set for each label of the original data set. Thus, for example, in the i – th data set, each instance associated with the label i is labelled as positive and the other instances are labelled as negative. As a drawback, this method may not be capable of handling correlations between labels.

Despite their possible drawbacks, BR and LP can achieve reasonably good results and we will use them in our experimentation.

On the other hand, algorithm adaptation approaches modify existing algorithms to manage multi-label data. For example, ML-kNN [15], a modification of the well-known kNN algorithm, uses prior and posterior probabilities for the frequency of labels within the k nearest neighbours, in order to determine the label set of a test instance. In [4] the C4.5 algorithm was adapted by modifying the calculation of its formula of entropy in order to manage multi-label data. BP-MLL [16] introduces a new error function, in the Back-propagation algorithm, in order to take into account multiple labels. A modification of the SVM algorithm that minimizes the ranking loss measure is proposed in [5]. ML-RBF [14] uses a clustering-based analysis for each label in order to place the neurons of the net, and there is an output in the RBFN for each label.

3 CO²RBFN: An Evolutionary Cooperative-Competitive Hybrid Algorithm for RBFN Design

CO²RBFN [10] is an evolutionary cooperative-competitive hybrid algorithm for the design of RBFNs. In this algorithm each individual of the population represents, with a real representation, an RBF and the entire population is responsible for the final solution.

The individuals cooperate towards a definitive solution, but they must also compete for survival. In this environment, in which the solution depends on the behaviour of many components, the fitness of each individual is known as credit assignment.

In order to measure the credit assignment of an individual, three factors have been proposed: the RBF contribution to the network output, the error in the basis function radius, and the degree of overlapping among RBFs.

The application of the operators is determined by a Fuzzy Rule-Based System. The inputs of this system are the three parameters used for credit assignment and the outputs are the operators' application probability.

The main steps of CO²RBFN, explained in the following subsections, are shown in the pseudocode, in Algorithm 1. For a wider explanation of the algorithm see reference [10].

Algorithm 1. Main steps of CO²RBFN

1. Initialize RBFN
 2. Train RBFN
 3. Evaluate RBFs
 4. Apply operators to RBFs
 5. Substitute the eliminated RBFs
 6. Select the best RBFs
 7. If the stop condition is not verified go to step 2
-

RBFN initialization. To define the initial network a specified number m of neurons (i.e. the size of population) is considered. The center of each RBF is randomly allocated to a different pattern of the training set. The RBF widths, d_i , will be set to half the average distance between the centres. Finally, the RBF weights, w_{ij} , are set to zero.

RBFN training. The Least Mean Square algorithm [13] is used to calculate the RBF weights.

RBF evaluation. A credit assignment mechanism is required in order to evaluate the role of each RBF ϕ_i in the cooperative-competitive environment. For an RBF, three parameters, a_i , e_i , o_i are defined:

- The contribution, a_i , of the RBF ϕ_i , is determined by considering the weight, w_i , and the number of patterns of the training set inside its width, pi_i :

$$a_i = \begin{cases} |w_i| & \text{if } pi_i > q \\ |w_i| * (pi_i/q) & \text{otherwise} \end{cases} \quad (2)$$

where q is the average of the pi_i values minus the standard deviation of the pi_i values.

- The error measure, e_i , for each RBF ϕ_i , is obtained by counting the wrongly classified patterns inside its radius:

$$e_i = \frac{pibc_i}{pi_i} \quad (3)$$

where $pibc_i$ and pi_i are the number of wrongly classified patterns and the number of all patterns inside the RBF width respectively.

- The overlapping of the RBF ϕ_i and the other RBFs is quantified by using the parameter o_i . This parameter is computed by taking into account the fitness sharing methodology [6], whose aim is to maintain the diversity in the population.

Applying operators to RBFs. In CO²RBFN four operators have been defined in order to be applied to the RBFs:

- Operator Remove: eliminates an RBF.
- Operator Random Mutation: modifies the centre and width of an RBF in a random quantity.

- Operator Biased Mutation: modifies, using local information, the RBF trying to locate it in the centre of the cluster of the represented class.
- Operator Null: in this case all the parameters of the RBF are maintained.

The operators are applied to the whole population of RBFs. The probability for choosing an operator is determined by means of a Mandani-type fuzzy rule based system [9] which represents expert knowledge about the operator application in order to obtain a simple and accurate RBFN. The inputs of this system are parameters a_i , e_i and o_i used for defining the credit assignment of the RBF ϕ_i . These inputs are considered as linguistic variables va_i , ve_i and vo_i . The outputs, p_{remove} , p_{rm} , p_{bm} and p_{null} , represent the probability of applying Remove, Random Mutation, Biased Mutation and Null operators, respectively. Table 1 shows the rule base used to relate the antecedents and consequents described.

Table 1. Fuzzy rule base representing expert knowledge in the design of RBFNs

Antecedents			Consequents				Antecedents			Consequents			
v_a	v_e	v_o	p_{remove}	p_{rm}	p_{bm}	p_{null}	v_a	v_e	v_o	p_{remove}	p_{rm}	p_{bm}	p_{null}
R1	L		M-H	M-H	L	L	R6	H		M-H	M-H	L	L
R2	M		M-L	M-H	M-L	M-L	R7	L		L	M-H	M-H	M-H
R3	H		L	M-H	M-H	M-H	R8	M		M-L	M-H	M-L	M-L
R4	L		L	M-H	M-H	M-H	R9	H		M-H	M-H	L	L
R5	M		M-L	M-H	M-L	M-L							

Introduction of new RBFs. In this step, the eliminated RBFs are substituted by new RBFs. The new RBF is located in the centre of the area with maximum error or in a randomly chosen pattern with a probability of 0.5 respectively.

Replacement strategy. The replacement scheme determines which new RBFs (obtained before the mutation) will be included in the new population. To do so, the role of the mutated RBF in the net is compared with the original one to determine the RBF with the best behaviour in order to include it in the population.

4 Experimentation

The objective of this paper is to test our present evolutionary cooperative-competitive algorithm for RBFN design, CO²RBFN, in the new multi-label classification field while taking into account other typical data mining methods. With the conclusions obtained we can draw lines for future development.

With this purpose in mind, we have used the multi-label data mining software and repository Mulan (<http://mulan.sourceforge.net/index.html>). In this site you can find different multi-label methods, tools and data sets as well as the possibility of using classical Weka learning methods [7].

In order to test CO²RBFN the data sets Emotions and Scene have been chosen. In Emotions a piece of music must be classified in more than one class and in Scene an image may belong to multiple semantic classes. Emotions has 593 instances, 72 numeric attributes and 6 labels. Scene has 2407 instances, 294 numeric attributes and 6 labels. As a first conclusion the high dimensionality of the multi-label data sets must be highlighted.

Typical data-mining methods have been chosen for comparisons, specifically: C4.5, KNN, Naive Bayes, MLP, PART, RBFN and SVM. Their implementations and references can be found in Weka [7]. These methods have been run with the parameters recommended by their authors. For CO²RBFN the iterations of the main loop have been established to 100 and the number of neurons in the range between 10 and 20. These parameter values have been heuristically chosen.

To run CO²RBFN and the other classical data mining techniques with the above data sets, we use the problem transformation methodology and concretely the popular Binary Relevance and Label Powerset techniques. In this way, both Emotions and Scene have been transformed with BR and LP.

General experimentation parameters, set up in MULAN, are ten-fold cross validation (90% for training data set, 10% for test data set) and three repetitions for obtaining the means values of the tables of test results. The measures used in the results are the ones returned by Mulan software and are described in [11]. For the measure Hamming Loss the lower the value the better, and for the other the higher the value, the better. The best result appears in bold.

In Table 2 the average test results for BR transformation and the two data sets are shown. Table 3 shows the results for the LP transformation.

Table 2. Average test results with Binary Relevance transformation

Data set Emotion								
	C4.5	CO ² RBFN	KNN	MLP	Naive Bayes	PART	RBFN	SVM
Hamming Loss	0.247	0.204	0.235	0.215	0.252	0.257	0.229	0.244
Subset Accuracy	0.184	0.270	0.268	0.270	0.206	0.157	0.213	0.180
Example-Based Recall	0.599	0.612	0.626	0.646	0.773	0.614	0.630	0.441
Example-Based Accuracy	0.462	0.514	0.514	0.525	0.529	0.456	0.494	0.391
Data set Scene								
	C4.5	CO ² RBFN	KNN	MLP	Naive Bayes	PART	RBFN	SVM
Hamming Loss	0.137	0.141	0.111	0.100	0.242	0.119	0.139	0.126
Subset Accuracy	0.427	0.365	0.629	0.566	0.169	0.477	0.369	0.306
Example-Based Recall	0.634	0.457	0.693	0.706	0.858	0.668	0.484	0.325
Example-Based Accuracy	0.535	0.419	0.674	0.647	0.453	0.578	0.437	0.323

As can be observed, from the tables of results there is no one a method that outperforms the others, neither for BR transformation nor for the LP transformation. CO²RBFN achieves its best results for the Emotions data set (independently of the transformation used), outperforming the other methods in four measures. For the BR transformation of Scene, CO²RBFN achieves results

Table 3. Average test results with Label Powerset transformation

Data set Emotions								
	C4.5	CO ² RBFN	KNN	MLP	Naive Bayes	PART	RBFN	SVM
Hamming Loss	0.277	0.243	0.235	0.234	0.233	0.293	0.217	0.281
Subset Accuracy	0.207	0.301	0.268	0.278	0.268	0.209	0.298	0.271
Example-Based Recall	0.541	0.653	0.626	0.630	0.630	0.526	0.647	0.595
Example-Based Accuracy	0.438	0.522	0.514	0.518	0.512	0.424	0.542	0.473
Data set Scene								
	C4.5	CO ² RBFN	KNN	MLP	Naive Bayes	PART	RBFN	SVM
Hamming Loss	0.144	0.186	0.111	0.114	0.137	0.139	0.116	0.095
Subset Accuracy	0.547	0.427	0.629	0.641	0.537	0.563	0.621	0.688
Example-Based Recall	0.609	0.454	0.693	0.701	0.678	0.626	0.677	0.720
Example-Based Accuracy	0.589	0.454	0.674	0.684	0.615	0.605	0.662	0.720

similar to other methods. The worst results for CO²RBFN are for the LP transformation of Scene. It must be highlighted the right accuracy achieved by the other RBFN design method and therefore the good behaviour of the RBFN models in multi-label classification tasks. In any case, CO²RBFN is the method with more best results (bold) in individual measures, along with SVM.

In summary, when transformations are applied to multi-label data sets in order to solve the associated classification problem, no algorithm outperforms the other on all the data.

5 Conclusions

In many real classification data sets, instances can be associated to more than one class. These data sets are called multi-label data sets. Examples of related applications are text categorization and association of semantic information to audio or video content. We can distinguish two ways to solve a multi-label problem: Problem Transformation and Algorithm Adaptation. With the first approach the original data set is transformed into single-label data-sets in order to apply traditional classification methods. The other method involves adapting classical algorithms in order to manage multi-label data.

In this paper a first approach to multi-label classification, CO²RBFN, a cooperative-competitive evolutionary model for the design of RBFNs, is tested with multi-label data sets. The results of CO²RBFN, and other data mining methods chosen for comparison, show that no algorithm outperforms the other on all the data. This behaviour may be due to the drawbacks described for transformation problem methods or to the intrinsic characteristics of the multi-label data sets.

As a future line of research we propose an in-deep analysis of the multi-label problem in order to carry out our developments, taking into account characteristics such as high dimensionality, correlations among labels and the interpretability of the results obtained.

Acknowledgments. Supported by the Spanish Ministry of Science and Technology under the Project TIN2008-06681-C06-02, FEDER funds, and the Andalusian Research Plan TIC-3928.

References

1. Broomhead, D., Lowe, D.: Multivariable functional interpolation and adaptive networks. *Complex Systems* 2, 321–355 (1988)
2. Buchtala, O., Klimek, M., Sick, B.: Evolutionary optimization of radial basis function classifiers for data mining applications. *IEEE Transactions on System, Man and Cybernetics B* 35(5), 928–947 (2005)
3. Carvalho, A.C.P.L.F., Freitas, A.A.: Foundations of Computational Intelligence. In: Abraham, A., Hassanien, A.-E., Snášel, V. (eds.) *Foundations of Computational Intelligence Volume 5*. SCI, vol. 205, pp. 177–195. Springer, Heidelberg (2009)
4. Clare, A., King, R.: Knowledge discovery in multi-label phenotype data. In: Siebes, A., De Raedt, L. (eds.) *PKDD 2001*. LNCS (LNAI), vol. 2168, pp. 42–53. Springer, Heidelberg (2001)
5. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. *Advances in Neural Information Processing Systems* 14 (2002)
6. Goldberg, D.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading (1989)
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. *SIGKDD Explorations* 11(1) (2009)
8. Harpham, C., Dawson, C.W., Brown, M.R.: A review of genetic algorithms applied to training radial basis function networks. *Neural Computing and Applications* 13, 193–201 (2004)
9. Mandani, E., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies* 7(1), 1–13 (1975)
10. Pérez-Godoy, M.D., Rivera, A.J., del Jesus, M.J., Berlanga, F.J.: *CO²RBFN*: An evolutionary cooperative-competitive RBFN design algorithm for classification problems. *Soft Computing* 14(9), 953–971 (2010)
11. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining Multi-label Data. In: *Data Mining and Knowledge Discovery Handbook*, 2nd edn., pp. 667–668. Springer, Heidelberg (2010)
12. Whitehead, B., Choate, T.: Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction. *IEEE Transactions on Neural Networks* 7(4), 869–880 (1996)
13. Widrow, B., Lehr, M.A.: 30 years of adaptive neural networks: perceptron, madaline and backpropagation. *Proceedings of the IEEE* 78(9), 1415–1442 (1990)
14. Zhang, M.L.: Ml-rbf: Rbf neural networks for multi-label learning. *Neural Processing Letters* 29(2), 61–74 (2009)
15. Zhang, M.L., Zhou, Z.H.: Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition* 40, 2038–2048 (2007)
16. Zhang, Y., Burer, S., Street, W.N.: Ensemble pruning via semi-definite programming. *Journal of Machine Learning Research* 7, 1315–1338 (2006)

Single Neuron Transient Activity Detection by Means of Tomography

Carlos Aguirre¹, Pedro Pascual², Doris Campos¹, and Eduardo Serrano¹

¹ GNB

² GAA

Escuela Politécnica Superior, Universidad Autónoma de Madrid,
28049 Madrid, Spain

Carlos.Aguirre@uam.es, Pedro.Pascual@uam.es, Doris.Campos@uam.es,
Eduardo.Serrano@uam.es

Abstract. Spectral methods allow the estimation of the firing frequency in the activity of a single neuron. However, transient periods, changes in the neuron firing frequency or even changes in the neuron activity regime (rest, tonic firing or spiking) due to different inputs or to the presence of neurotransmitters are not well detected by means of these methods due to the fact that frequency and time are not commutable operators. Some other methods have been developed to deal with local transients, for example the localized Fourier transform or the Wigner distribution. Unfortunately these localized methods need fine tuning to find an adequate working resolution and the resulting coefficients are hard to interpret. In this work we propose the use of the tomographic transforms to detect and characterize transient components in the behaviour of a single neuron.

1 Introduction

Most neurobiological signals are rhythmic in nature and, therefore, spectral methods, such as the Fourier Transform, have been very useful for the processing of many types of biological signals and, in particular, for the analysis of neuronal activity. However, spectral methods does not perform well for non-stationary signals or transient time-varying signals. This is the case of signals produced by neuron action potentials where many signal properties can vary in time. For example, it is well known that neurons in a tonic spiking regime can increase or decrease its firing rate or go to a quiet state depending on the input received by other neurons. Furthermore, CA1 and CA3 neurons of the hippocampus can switch from a bursting behaviour to a tonic spiking behaviour in the presence of several neurotransmitters and neuromodulators [1].

This change in the regime of the neuron behaviour cannot be detected by the use of Fourier transform as this transform only provides perfect information about the frequency content of the signal but no about transient behaviour of the signal. Wavelet transforms have also been used for the analysis of neuronal activity, for example for the detection of multiple encoding rhythms in a network

of connected neurons [2]. Wavelet transforms (or in general any time-resolution transform) provide accurate information about the transient resolution content of the signal, as wavelet basis are located in both time and resolution. Unfortunately wavelets basis are not well located in frequency and, therefore, wavelet transforms does not provide information about transient frequency components.

Several time-frequency transforms for component detection have been proposed in the bibliography, for example the Cohen's Class [3] obtained as a convolution with the Wigner distribution. In general this is the method followed by most authors [4,5,6] and the identification of signal transient components is done by looking for amplitude concentrations in the time–frequency plane. The drawback for the use of these methods is that they present negative terms, cross terms or, in general, they do not behave correctly in the marginal time and frequency components or present many artifacts even selecting clever kernels or filtering operations. These are some consequences of the fact that time and frequency are non-commuting operators, making impossible the existence of joint probability distributions on the time-frequency plane.

A recently proposed family of transforms, named the tomographic transform [10], uses a different approach to obtain a representation of the signal in the time-frequency plane. The idea of tomography is to decompose the signal by using the eigenfunctions of linear combinations of operators, for example, time and frequency, time and resolution or time and conformal operator. Even when this transform does not provide a joint probability distribution in the plane determined by the two operators, it fulfills the condition that the sum of the squares of the projection on the eigenfunctions has the same norm than the signal, providing so an exact probabilistic interpretation. This approach has been used for the analysis of reflectometry data [7,8].

In this paper we propose the use of the tomographic transform for the detection of transient components in the activity of a single neuron.

2 Tomograms for Signal Analysis

In this work we will focus in the analysis of neuronal signals considering a linear functional operator named the time–frequency operator $B(\mu, \nu)$ defined in the following way:

$$B(\mu, \nu) = \mu t + \nu \omega = \mu t + \nu i \frac{\partial}{\partial t} \quad (1)$$

where t is the time operator and ω is the frequency operator. This linear combination of operators produces a new linear operator obtained by the interpolation between two linear operators, the time operator and the frequency operator.

From a given operator the transform of a signal $s(t)$ can be defined:

$$M_s^B(X, \mu, \nu) = \int s^*(t) \delta(B(\mu, \nu) - X) s(t) dt \quad (2)$$

where, for a normalized signal s and for each pair of values (μ, ν) $M_s^B(X, \mu, \nu)$ provides a probability distribution on the variable X corresponding to the corresponding time–frequency operator $B(\mu, \nu)$.

Using the previously defined operators $B(\mu, \nu)$ and their unitary exponentiations $U(\mu, \nu) = \exp(iB(\mu, \nu))$ a complete description of the time–frequency operator can be obtained [10] and an explicit expression for the tomogram may be found at [11]. In this paper we will consider the pair (μ, ν) as coming from a single parameter θ of the form $(\mu, \nu) = (\cos \theta, \sin \theta)$, $\theta \in [0, \frac{\pi}{2}]$. Thus, this transform (the Radon transform, actually) interpolates between time ($\theta = 0$) and frequency ($\theta = \frac{\pi}{2}$).

Following these ideas, a family of distributions $M_s(x, \theta)$ is defined for a signal $s(t)$, $t \in [0, T]$, by

$$M_s(x, \theta) = \left| \int s(t) \Psi_x^{\theta, T}(t) dt \right|^2 \quad (3)$$

with

$$\Psi_x^{\theta, T}(t) = \frac{1}{\sqrt{T}} \exp\left(\frac{-i \cos \theta}{2 \sin \theta} t^2 + \frac{ix}{\sin \theta} t\right) \quad (4)$$

Notice that the $\Psi_x^{\theta, T}(t)$ are generalized eigenfunctions for any value x of the operator $U(\theta)$. In fact, this observation is true for other kind of tomogram operators like the time-resolution operator or the time-conformal operator and it allows to obtain the distributions for these operators as the eigenfunctions of the given operator. Observe also that for $\theta = \frac{\pi}{2}$ we obtain the usual Fourier basis. In $\theta = 0$ this eigenfunctions are not well defined, and this case requires an special study, see [11].

Now we select a set of x_n such that the set $\Psi_{x_n}^{\theta, T}(t)$ is orthonormal. This is possible considering the set

$$x_n = \frac{2n\pi}{T} \sin(\theta) \quad (5)$$

Now we consider the projection of the signal $s(t)$ over the basis $\Psi_{x_n}^{\theta, T}(t)$ as

$$c_{x_n}^\theta(s) = \int s(t) \Psi_{x_n}^{\theta, T}(t) dt \quad (6)$$

The original signal $s(t)$ can now be recovered from the tomogram coefficients $c_{x_n}^\theta$ by the sum

$$s(t) = \sum_n c_{x_n}^\theta \Psi_x^{\theta, T}(t) \quad (7)$$

For denoising or component selection, only a restricted set F_n of the whole set of coefficients can be considered. In this case the reconstructed signal becomes

$$s_k(t) = \sum_{n \in F_n} c_{x_n}^\theta \Psi_x^{\theta, T}(t) \quad (8)$$

3 Examples

In this section we discuss the method presented in the previous section in two particular neuronal signals. The first example shows that the tomographic transform is able to disentangle a signal that alternates a rhythmic pattern with a resting state. In the second example we analyze a signal with several rhythmic patterns present at different time intervals.

Both signals are generated by a phenomenological neuronal model presented in [9]. This model is implemented as a two-variable coupled map and presents most of the characteristics of the behaviour of real neurons such as, tonic spiking, bursting or subthreshold oscillations with a minimal computational consumption. The model also provides an accurate response to external current injection or coupling with other neurons.

3.1 Single Rhythmic Pattern

Here we consider a neuronal signal $x(t)$ generated by a phenomenological model. The values of the parameters for this neuron are (see [9]) $L = .01$, $B = .15$, $C = .3$, $D = .9$, $R = 3.0$, $S = .0001$, $H0 = .01$, $K0 = .29$, $T0 = .75$, $H1 = .14$, $K1 = .02$, $T1 = .4$. and $E = 0.0$ and the signal runs for 20000 time steps. In time intervals $[0 : 4000]$, $[8000 : 12000]$ and $[16000 : 20000]$ the neuron receives a constant external input with value $\sigma = .07$ that induces the neuron to a tonic spiking state with a characteristic rhythmic pattern. In intervals $[4000 : 8000]$ and $[12000 : 16000]$ no external input is present ($\sigma = .0$) resulting in a resting state. The neuron output is presented in figure 1.

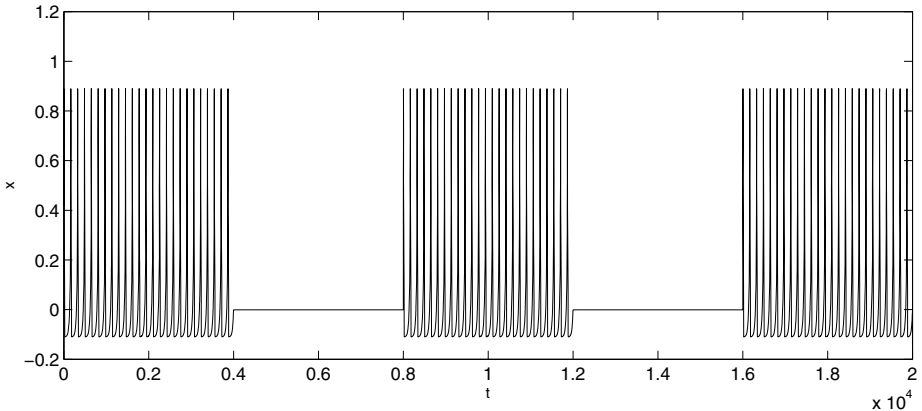


Fig. 1. Neuronal signal with silent and tonic spiking transient activity components

In figure 2, the most significant values of the spectrogram and the tomogram of the previous signal are depicted.

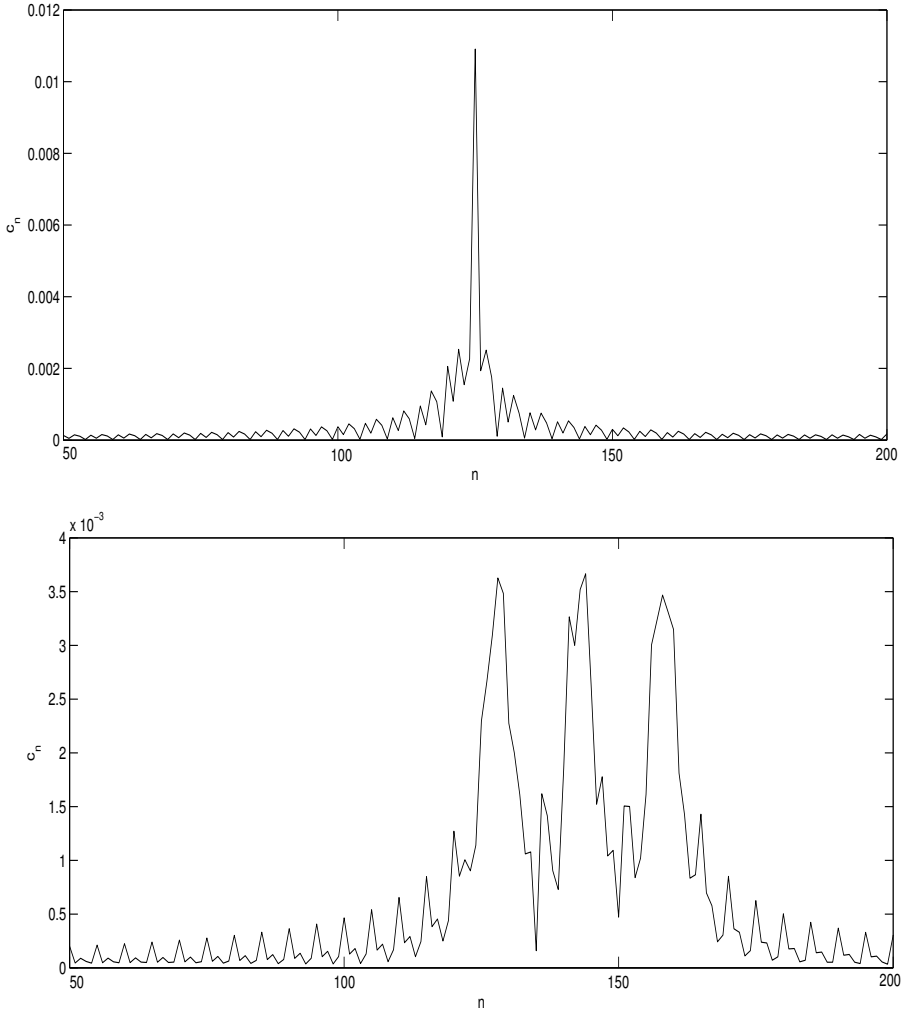


Fig. 2. Spectrogram (top) and Tomogram (bottom) of a neural signal with spiking and silent regimes

The spectrogram presents a clear single high significant value at the Fourier coefficient $n = 126$. If we consider that the signal runs for a second it means that the signal is mainly composed by a single $126Hz$ frequency, as expected the spectrogram does not indicate the presence of several transitory components in the signal. In the lower part of figure 2 the tomogram presented in the previous section for angle $\theta = \frac{\pi}{3}$ is shown. In this plot three clear relevant values are present at coefficients $n = 129$, $n = 145$ and $n = 159$ that indicates the presence of three transient components of the same or a very similar frequency content.

3.2 Different Rhythmic Patterns

In this section we inject to the neuron model an external input of value $\sigma = .007$ at time intervals $[4000 : 8000]$ and $[12000 : 16000]$ resulting in a signal with two rhythmic regimes. A fast regime when a higher external input is present and a slow regime when low external input is present. The neuron output is presented in figure 3.

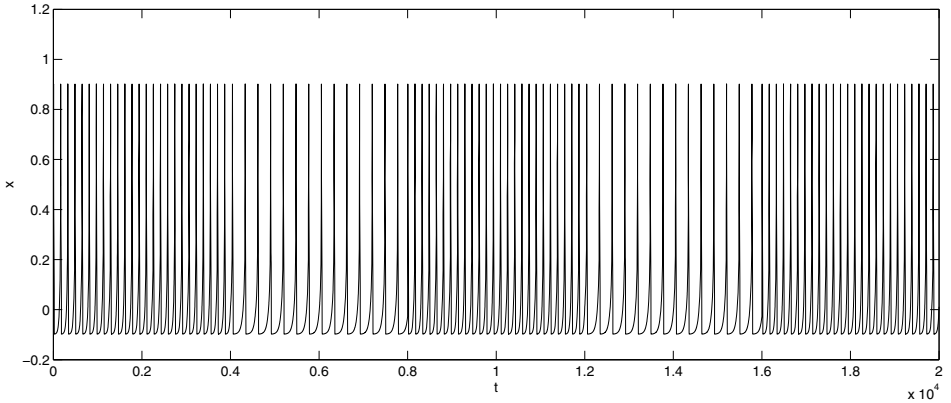


Fig. 3. Neuronal signal with several transient tonic spiking rhythmic patterns

In this case, the spectrogram (see top of figure 4) shows again a very significant value at the coefficient $n = 124$. A second significant lower frequency value appears now at coefficient $n = 70$ indicating so that a second $70Hz$ frequency component is present at the signal. This value corresponds to the slow spiking rhythm. On the other hand, the tomogram (see bottom of figure 4) shows three components of high frequency similar to the ones in the previous example but it also shows the two additional low frequency components that correspond with the two low spiking rhythms at time intervals $[4000 : 8000]$ and $[12000 : 16000]$.

In both experiments the selected value of the parameter of the tomogram transform $\theta = \frac{\pi}{3}$ is not critical. Any value of θ in the interval $(\frac{\pi}{2}, \frac{\pi}{4})$ produces a similar result.

4 Conclusions

Here we have applied the tomographic transform to find transient components in a neural signal. The method seems quite robust to disentangle components of similar frequency present at different time intervals and to detect different transitory rhythmic patterns present in the signal. This method could not only be useful to detect signal components localized in time, it can be also used as filtering tool by just preserving the most significant values of the tomographic transform, in a similar way as in the Fourier transform but with the advantage of preserving time information.

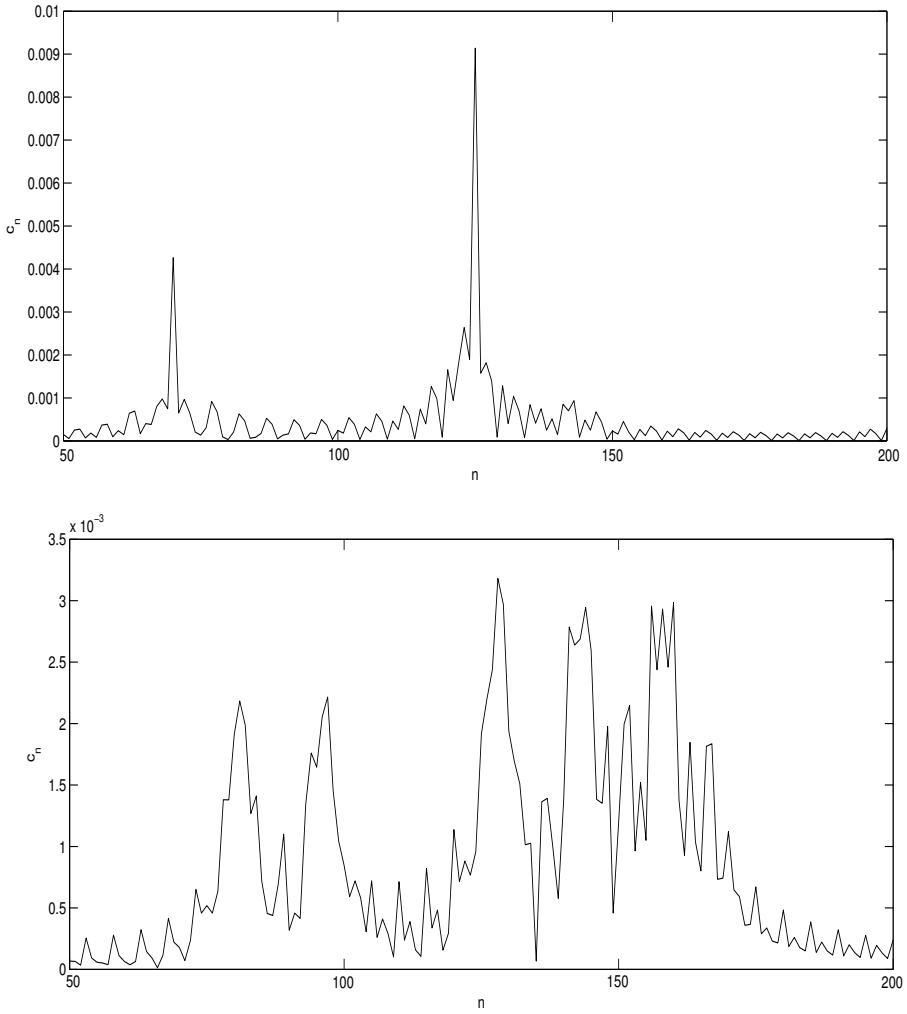


Fig. 4. Spectrogram (top) and Tomogram (bottom) of a neural signal with two rhythmic patterns

Acknowledgments

(CA) and (ES) are supported by BFU2009-08473. (CA) and (PP) are partially supported by AYA2009-14212-05. (PP) is partially supported by TIN2010-21575-C02-01.

References

1. Szucs, A., Varona, P., Volkovskii, A.R., Abarbanel, H.D.I., Rabinovich, M.I., Selverston, A.I.: Interacting biological and electronic neurons generate realistic oscillatory rhythms. *Neuro Report* 11(3), 563–569 (2000)
2. Aguirre, C., Pascual, P.: A Wavelet Based Method for Detecting Multiple Encoding Rhythms in Neural Networks. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) *IWANN 2009*. LNCS, vol. 5517, pp. 9–16. Springer, Heidelberg (2009)
3. Cohen, L.: Time-frequency distributions - A review. *Proc. IEEE* 77, 941–981 (1989)
4. Khadra, L.M.: Time-frequency distribution of multi-component signals. *Int. J. of Electronics* 67, 53–57 (1989)
5. Choi, H.I., Williams, W.J.: Improved time-frequency representation of multi-component signals using exponential kernels. *IEEE Trans. Acoust. Speech Signal Process* 37, 862–871 (1989)
6. Fineberg, A.B., Mammone, R.J.: Detection and classification of multicomponent signals. In: *Proceedings of 25th Asilomar Conference on Computer Signal and Systems*, pp. 1093–1097 (1991)
7. Briolle, F., Lima, R., Manko, V.I., Vilela Mendes, R.: A tomographic analysis of reflectometry data I: Component factorization. *Meas. Sci. Technol.* 20(10), 105501–105511 (2009)
8. Briolle, F., Lima, R., Manko, V.I., Vilela Mendes, R.: A tomographic analysis of reflectometry data II: phase derivative. *Meas. Sci. Technol.* 20(10), 105512–105522 (2009)
9. Aguirre, C., Campos, D., Pascual, P., Serrano, E.: Neuronal behavior with sub-threshold oscillations and spiking/bursting activity using a piecewise linear two-dimensional map. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (eds.) *ICANN 2005*. LNCS, vol. 3696, pp. 103–108. Springer, Heidelberg (2005)
10. Manko, M.A., Manko, V.I., Vilela Mendes, R.: Tomograms and other transforms: a unified view. *Journal of Physics A: Math. Gen.* 34, 8321–8332 (2001)
11. Manko, V.I., Vilela Mendes, R.: Noncommutative time frequency tomography. *Physics Letters A* 263, 53–59 (1999)

Estimate of a Probability Density Function through Neural Networks

Leonardo Reyneri¹, Valentina Colla², and Marco Vannucci²

¹ Politecnico di Torino, Electronic Department
C.so Duca degli Abruzzi 24, 10129, Torino, Italy

² Scuola Superiore Sant'Anna, PERCRO-TECIP
Viale Rinaldo Piaggio, 56025 Pontedera, Italy
leonardo.reyneri@polito.it, colla@sssup.it

Abstract. A correct estimate of the probability density function of an unknown stochastic process is a preliminary step of utmost importance for any subsequent elaboration stages, such as modelling and classification. Traditional approaches are based on the preliminary choice of a mathematical model of the function and subsequent fitting on its parameters. Therefore some a-priori knowledge and/or assumptions on the phenomenon under consideration are required. Here an alternative approach is presented, which does not require any assumption on the available data, but extracts the probability density function from the output of a neural network, that is trained with a suitable database including the original data and some ad hoc created data with known distribution. This approach has been tested on a synthetic and on an industrial dataset and the obtained results are presented and discussed.

Keywords: probability density function, weighted radial basis function networks, industrial databases.

1 Introduction

The probability density function (PDF) in statistics identifies the probability of occurrence of each possible value of a random variable, if it is discrete, or the probability of the value falling within a particular interval, if the variable is continuous [1]. If the considered random variable is derived from a unknown stochastic process, its PDF is not available and must be estimated. The first step of the PDF estimate consists in the selection of its generic mathematical expression among a quite large varieties of known options (often referred as “families”). Afterwards, suitable values of the parameters characterising the particular shape of such function must be selected. When the PDF is totally unknown and poor a-priori knowledge on the phenomenon under consideration is available, the above operations must be iterated several times for different mathematical formulations of the PDF and finally an assessment of goodness of fit must be pursued through some specific tests, which aim at understanding how reasonable is the assumption that the available dataset comes from the selected PDFs once the

associated parameters have been suitably chosen. The whole above-described procedure is commonly referred as *Distribution fitting*.

When coping with data coming from real world applications, PDF estimate and distribution fitting are often the first necessary step for the development of any kind of model for any kind of application such as classification, clustering, prediction, etc... Application of distribution fitting can be found in a wide variety of different fields, such as economy [2], engineering [3] and natural sciences [4].

The approach proposed here is totally different, as it exploits the capability of Weighted Radial Basis Function (WRBF) [5] Neural Networks (NN) to fit any kind of function and does not require any predefined parametric mathematical formula to express the generic form of the PDF.

2 Theoretical Background

Given a continuous scalar random variable x , its *Probability Density Function* (PDF) is a function $p(x) \geq 0$ for all x such that, given an interval $I = [a, b]$, the integral $\int_a^b p(x)dx$ expresses the probability that $x \in I$. The extension of such definition to vector random variables \mathbf{x} is straightforward.

The most commonly adopted PDF (e.g. Normal, Chi-square, Dagum, Log-Logistic, etc..) are parametric functions, namely they have a mathematical formulation in the form $p(x|\theta_1, \theta_2, \dots, \theta_k)$, where $\theta_1, \theta_2, \dots, \theta_k$ are k constant parameters. Given a set x_1, \dots, x_n of n independent identically distributed observations sampled from a continuous random variable x , the estimate of the PDF is composed of the following two operations:

- the choice of the particular mathematical model, which is normally made by exploiting some a-priori knowledge on the process under consideration and/or on the flexibility and generality of the selected models;
- the selection of the particular set of parameters which allows the final PDF to best fit the available data. Such selection is performed by solving an optimization problem on θ_i where a pre-defined performance index is maximized;

One of the most widely used methods for searching the parameters of a PDF that match at best a set of observations is the *Maximum Likelihood Estimation* (MLE) [6]. The so-called *Likelihood Function* (LF) is defined as:

$$L(x_1, \dots, x_n | \theta_1, \theta_2, \dots, \theta_k) = \prod_{i=1}^n p(x_i | \theta_1, \theta_2, \dots, \theta_k)$$

and the MLEs of the parameters are calculated by maximizing L or its natural logarithm $\Lambda = \ln L$ (which is often easier) by simultaneously solving the k equations $\frac{\partial \Lambda}{\partial \theta_j} = 0$, ($j = 1, 2, \dots, k$). The best estimate of a distribution parameter is thus defined to be the one that maximizes the probability of obtaining the observed data. The complexity of the above-described system of equations depends on the adopted PDF and may be non-trivial: specialized software for numerical resolution may be required and, if this is the case, the obtained MLE may be sensible to the choice of the starting values.

3 Distribution Fitting Based on Neural Networks

We propose here an innovative technique to estimation of PDF based on the intrinsic properties of NN to solve optimization and minimization problems.

We start from a given set \mathcal{S} of Q points $\mathbf{X}^q = [x_1^q, x_2^q, \dots, x_N^q] \in \mathcal{S}; \forall q \in [1, Q]$ whose statistical distribution $p(\mathbf{X}) : R^N \rightarrow [0, 1]$ has to be found. We suppose that $p(\mathbf{X})$ is continuous and “sufficiently” smooth to be approximated by an appropriate NN $f(\mathbf{X}; \theta) \approx p(\mathbf{X})$ where θ is the vector of the free parameters of the NN (namely, its weights, centers and biases). The problem is therefore to find the best θ such as to reduce the approximation error between $f(\mathbf{X}; \theta)$ and $p(\mathbf{X})$. In our case, we decide to minimize the RMS error:

$$E = \sqrt{\sum_q (f(\mathbf{X}^q; \theta) - p(\mathbf{X}^q))^2} \quad (1)$$

Unfortunately, the real probability distribution of data $p(\mathbf{X})$ is unknown, otherwise minimizing (1) would be a trivial NN training. We therefore propose the following training algorithm to train the NN to approximate as much as possible the unknown $p(\mathbf{X})$. The algorithm is organized in five steps:

1. Establish the *universe set* \mathcal{U} of points for the data set \mathcal{S} , for instance by finding the extreme values for each component of all patterns \mathbf{X}^p , namely: $l_i = \min_{q \in [1, Q]} \{x_i^q\}$ and $u_i = \max_{q \in [1, Q]} \{x_i^q\}$ such that:

$$\mathcal{S} \subset \mathcal{U} = [l_1, u_1] \times [l_2, u_2] \times \dots \times [l_N, u_N] \quad (2)$$

2. Create a *training set* by associating to each point of \mathcal{S} a unitary target value

$$t^q = 1; \forall q \in [1, Q] \quad (3)$$

3. Augment the *training set* by generating as many points as are the points in the given data set \mathcal{S} , namely Q additional points \mathbf{X}_R^q for $\forall q \in [Q + 1, 2Q]$ The new points shall be generated *randomly* within the universe set \mathcal{U} with a given *reference* statistical distribution $p_R(\mathbf{X})$. The shape of the reference distribution is in theory immaterial, although some distribution may in practice offer better performance, as will be discussed later on. At this stage, one can use, for instance, either a uniform or a gaussian distribution.
4. Associate to each additional point in the training set a target value:

$$t^q = 0; \forall q \in [Q + 1, 2Q] \quad (4)$$

5. Train the NN with the augmented training set using any training algorithm based on the minimization of the RMS error as defined above (e.g. any *delta rule*).

For each small region $\mathcal{R} \subset \mathcal{U}$ containing at least a few patterns, the NN sees Q_1 patterns $\mathbf{X}^q \in \mathcal{R}$ associated to $t^q = 1$ and a Q_2 patterns $\mathbf{X}_R^q \in \mathcal{R}$ associated to $t^q = 0$. The mathematically expected number of patterns is respectively:

$$Q_1 = Q \int_{\mathbf{X} \in \mathcal{R}} p(\mathbf{X}) dx_1 dx_2 \dots dx_N ; Q_2 = Q \int_{\mathbf{X} \in \mathcal{R}} p_R(\mathbf{X}) dx_1 dx_2 \dots dx_N \quad (5)$$

If both $p(\mathbf{X})$ and $p_R(\mathbf{X})$ are continuous and smooth, i.e. they are sufficiently constant in \mathcal{R} , the two counts can be approximated as, respectively:

$$Q_1 \approx Q \cdot p(\mathbf{X}) \cdot \int_{\mathbf{X} \in \mathcal{R}} dx_1 dx_2 \dots dx_N \quad Q_2 \approx Q \cdot p_R(\mathbf{X}) \cdot \int_{\mathbf{X} \in \mathcal{R}} dx_1 dx_2 \dots dx_N \quad (6)$$

The NN $y(\mathbf{X}^q; \Theta)$ is alternatively trained towards targets $t^q = 1$ and $t^q = 0$ and the training rule tends to minimize the RMS cost function:

$$E = \sqrt{\sum_q (y(\mathbf{X}^q; \Theta) - t^q)^2} \quad (7)$$

which is known to have a minimum when $y(\mathbf{X}^q; \Theta)$ coincides with the average value of the targets:

$$y(\mathbf{X}; \Theta) \approx t^q(\mathbf{X}) = \frac{Q_1}{Q_1 + Q_2} \quad (8)$$

as there are Q_1 (respectively Q_2) targets equal to 1 (respectively 0).

By combining formulae (6) and (8) and simplifying, we get that, once trained, the NN output approximates:

$$y(\mathbf{X}) \approx \frac{p(\mathbf{X})}{p(\mathbf{X}) + p_R(\mathbf{X})} \quad (9)$$

from which the $p(\mathbf{X})$ can be obtained, as the $p_R(\mathbf{X})$ is known apriori:

$$p(\mathbf{X}) \approx p_R(\mathbf{X}) \frac{y(\mathbf{X})}{1 - y(\mathbf{X})} \quad (10)$$

Thus, training the NN with the purposely augmented data set produces a neural function $y(\mathbf{X})$ from which the unknown PDF $p(\mathbf{X})$ can be inferred, by using an intermediate arbitrary PDF $p_R(\mathbf{X})$.

Despite formula (10) shows that the unknown $p(\mathbf{X})$ can be inferred from any arbitrary reference distribution $p_R(\mathbf{X})$, a sensitivity analysis on formula (10) proves that the approximation error is minimized when $p(\mathbf{X}) \approx p_R(\mathbf{X})$, which corresponds to $y(\mathbf{X}) \approx 0.5$. There is therefore an optimal $p_R(\mathbf{X}) \approx p(\mathbf{X})$ which reduces approximation error to the minimum, although it is obvious that this cannot be used apriori, as $p(\mathbf{X})$ is supposed to be unknown.

It is yet possible to use the following *successive approximation* strategy:

1. generate Q additional patterns with a uniform distribution $p_R(\mathbf{X})$;
2. train the NN as described in previous section. Once the NN is trained, infer a first rough estimate of $p'(\mathbf{X}) \approx p(\mathbf{X})$ as from formula (10);
3. remove the Q additional patterns introduced at the first step and substitute them with as many patterns but having a distribution $p'(\mathbf{X})$. In practice, one can repeatedly generate a new random pattern \mathbf{X}^q , with uniform distribution, and accept it into the training set with a probability $p'(\mathbf{X}^q)$; repeat until exactly Q patterns have been added;

4. train again a second NN, while keeping the first one as the reference distribution $p_R(\mathbf{X}) = p'(\mathbf{X})$. After training, a better estimate for the unknown distribution of input data can be obtained by appropriately modifying formula (10):

$$p''(\mathbf{X}) \approx p'(\mathbf{X}) \frac{y(\mathbf{X})}{1 - y(\mathbf{X})} \quad (11)$$

5. if desired, repeat this operation once more, for a better approximation.

4 Numerical Results

The NN-based approach that is described in Sec.3 has been tested to estimate the PDF of two different series of data: the former database has been *ad-hoc* created and consists of 1100 bidimensional data, while the second one derives from an industrial application and consists of 3500 monodimensional data. In the following, a subsection is devoted to each test.

In order to test the goodness of the distribution fitting, the *Mean Absolute Error* (MAE) has been adopted as performance index. The MAE [7] is a generic measure of the disagreement between any kind of predictive model and the actual values of the predicted variable. Here the MAE is the average of the absolute values of the differences between the estimated values $\hat{f}(\mathbf{X}_i)$ of the PDF and its corresponding real values $f(\mathbf{X}_i)$ on a predetermined set of N points:

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{f}(\mathbf{X}_i) - f(\mathbf{X}_i)| \quad (12)$$

In the case of the synthetic dataset of the first test the real PDF is known, thus $f(\mathbf{X}_i)$ can be the values of the real PDF on a predetermined regular grid of points. In the second test, which is more realistic because the actual PDF is not known, the MAE is employed to compute error between the histogram calculated on the observed data and the predicted PDF evaluated in the bin points. The number of bins is calculated by means of the Sturges' formula $k = \lceil \log_2 n + 1 \rceil$ where n is the number of samples in the dataset [8].

4.1 Test on a Synthetic Database

The first database is composed of 2 joint subsets of bidimensional points formed by 2 independent random components drawn from a normal distribution. The first data subset is formed by 1000 samples with average $\boldsymbol{\mu}_1 = [3 \ 0]^T$ and diagonal covariance matrix $\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$, while the second one is composed of 100 data with average $\boldsymbol{\mu}_2 = [-3 \ -3]^T$ and diagonal covariance matrix $\boldsymbol{\Sigma}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix}$. The overall synthetic database is therefore composed of 1100 data.

A two layered WRBF network has been adopted with 2 gaussian neurons in the hidden layer and 1 neuron in the output linear layer. According to Sec.3, the

network training database is composed of 2200 data; the last 1100 data derive from a known distribution $p'(\mathbf{X})$, in this case a normal distribution centred in the origin of the axes with covariance matrix $\Sigma_r = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$.

The initial values of the weight of the neurons in the hidden layer of the network are set to 1, the centers are randomly initialised and the thresholds initial values are null. As far as the output layer is concerned, the initial value of the weight is unitary, while the center and the threshold values are initially null. The network has been trained for 1000 epochs with a standard backpropagation algorithm. The PDF that is obtained through eq. (10) is shown in Fig. 1 a).

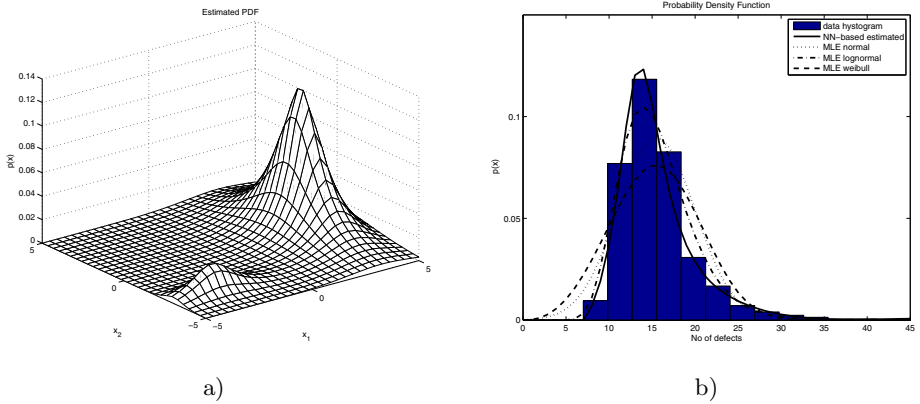


Fig. 1. a) Estimated PDF of the synthetic data obtained through the NN approach; b) Hystogram of the industrial database and the different estimates of the PDF

As a term of comparison, the MLE procedure has also been applied to the above-described problem by adopting as a mathematical model the following function, arising by the composition of two bi-variate normal distributions:

$$p(\mathbf{X}) = A \frac{1}{2\pi|\Sigma_1|} e^{-\frac{1}{2}(\mathbf{X}-\mu_1)^T \Sigma_1^{-1}(\mathbf{X}-\mu_1)} + B \frac{1}{2\pi|\Sigma_2|} e^{-\frac{1}{2}(\mathbf{X}-\mu_2)^T \Sigma_2^{-1}(\mathbf{X}-\mu_2)}$$

This is a strong assumption, which denotes a considerable *a-priori* knowledge on the problem under consideration: the same does not apply to a large variety of real world applications; however here it is useful to demonstrate the efficiency of the proposed approach. There are 14 unknown parameters in the PDF above: for the sake of simplicity, one might assume that the independency of each other of the two components of \mathbf{X} is known (which implies that the two covariance matrices are diagonal); moreover $B = 1 - A$. Finally 9 parameters must be estimated. As a numerical procedure must be adopted to search for the maximum of the likelihood function, the results of the MLE procedure strongly depend on the initial choice of the parameter values: in general, the mean vectors μ_1 and μ_2 are fitted better than the two covariance matrices. For instance, for the

available synthetic dataset after several attempts, the following best MLEs of the distribution parameters have been found: $\hat{A} = 0.9087$, $\hat{\boldsymbol{\mu}}_1 = [3.00 \ 0.06]^T$, $\hat{\boldsymbol{\Sigma}}_1 = \begin{bmatrix} 1.31 & 0 \\ 0 & 1.99 \end{bmatrix}$, $\hat{\boldsymbol{\mu}}_2 = [-2.84 \ -3.17]^T$ and $\hat{\boldsymbol{\Sigma}}_2 = \begin{bmatrix} 1.40 & 0 \\ 0 & 0.59 \end{bmatrix}$.

The MAE of the PDFs estimated through NN and MLE with respect to the actual PDF have been evaluated on a regular grid of 30×30 points where both components of \boldsymbol{x} lie in the range $[-5, 5]$. The MAE related to the PDF estimated through the approach proposed here is equal to 0.0028 while the MAE for the PDF estimated via MLE is equal to 0.0035. In relative terms, the NN-based approach improves the PDF estimate of 20%. This result is even more relevant if one considers that for MLE-based approach some quite strong assumptions have been exploited and a careful choice of the initial point for the search procedure has been made, while no assumptions are required for the NN-based approach and no special initialisation procedure is applied to the NN.

4.2 Test on an Industrial Application

The proposed approach has been tested also on real data related to particular surface defects in the production of metal objects. In this application defects are detected by dedicated inspection systems and subsequently classified through sophisticated image processing techniques. Usually a report is associated to each product, where the number of defects belonging to each particular defect class is recorded. An excessive increase in the number and/or gravity of some defects is considered as an indication of malfunctioning or, at least, of the need for maintenance operations. Therefore, an accurate knowledge of the distribution of each class of defects (namely the PDF associated to the number of defects per single product) is useful for the analysis of the overall qualitative trend of the production, for the implementation of techniques related to the statistical control of the process as well as for product classification and eventual destination.

A database composed of 3500 data have been processed, which contains the number of defects n that belong to a particular category per product: such number lies between $N_{min} = 5$ and $N_{max} = 40$. Also in this case, a two layered WRBF network has been adopted with 5 gaussian neurons in the hidden layer and 1 neuron in the output linear layer. The network training database is composed of 7000 samples, where the last 3500 data have been randomly generated and derive from a known distribution $p_R(n)$, in this case a normal distribution centred on $\eta = \frac{N_{min} + N_{max}}{2}$ with standard deviation $\sigma = N_{max} - N_{min}$. The same initialisation procedure of the test described in Sec. 4.1 has been adopted.

Figure 4.1b compares the histogram obtained from the real data, the PDF that is estimated through the NN-based approach and some widely adopted PDF whose parameters are estimated through MLE. Table 4.1 quantifies through the MAE the goodness of fit of the different estimated PDFs: the NN-based estimate shows the best fit of the histogram extracted from the real data. Moreover, it must be underlined that the NN-based approach does not need the construction of the histogram itself.

Table 1. MAE of the different estimates of the PDF for the industrial database

PDF type	Formula	MAE
NN-based	-	0.0027
normal	$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	0.0086
log-normal	$f(x) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-\frac{(\log x - \mu)^2}{2\sigma^2}}$	0.0035
Rayleigh	$f(x) = \frac{x}{b^2} e^{-\frac{x^2}{2b^2}}$	0.0156
Weibull	$f(x) = ba^{-b}x^{b-1}e^{-\left(\frac{x}{a}\right)^b}$	0.0111

5 Conclusions and Future Work

A NN-based approach has been presented for the estimation of the PDF of an unknown stochastic process, which does not require neither an a-priori choice of a mathematical model nor the elaboration of the data histogram, but only the computation of the variability range of each components of available data samples. Noticeably, the proposed algorithm is also capable of finding if the available data are clustered or not, as, in case of clustered data, the estimated PDF turns out to be multimodal. The proposed method have been tested on two dataset and the results shows that it is capable to outperform a standard MLE-based approach. In the future, further tests will be performed on other dataset coming from real word applications and the performance will be compared also to other literature approaches.

References

1. Devore, J.L.: Probability and Statistics for Engineering and the Sciences, Enhanced Review Edition. Duxbury Press, UK (2008)
2. Combes, C., Dussauchoy, A.: Generalized extreme value distribution for fitting opening/closing asset prices and returns in stock-exchange. Springer Operational Research 6(1), 3–26 (2006)
3. Zaharim, A., Razali, A.M., Abidin, R.Z., Sopian, K.: Fitting of Statistical Distributions to Wind Speed Data in Malaysia. European Journal of Scientific Research 26(1), 6–12 (2009)
4. Wendell Cropper Jr., P., Anderson, P.J.: Population dynamics of a tropical palm: use of a genetic algorithm for inverse parameter estimation. Ecological Modelling 177, 119–127 (2004)
5. Reyneri, L.M.: Unification of neural and wavelet networks and fuzzy systems. IEEE Trans. on Neural Networks 10(4), 801–814 (1999)
6. Le Cam, L.: Maximum likelihood – an introduction. ISI Review 58(2), 153–171 (1990)
7. Wun, L.-M., Pearn, W.L.: Assessing the statistical characteristics of the mean absolute error of forecasting. International Journal of Forecasting 7, 335–337 (1991)
8. H.A.: The Choice of a Class Interval. Journal of the American Statistical Association 21(153), 65–66 (1926)

A Neural Fuzzy Inference Based Adaptive Controller Using Learning Process for Nonholonomic Robots

Ting Wang, Fabien Gautero, Christophe Sabourin, and Kurosh Madani

Images, Signals and Intelligence Systems Laboratory (LISSI / EA 3956),
University PARIS-EST Creteil (UPEC), Senart-FB Institute of Technology,
Bât.A, Av. Pierre Point, F-77127 Lieusaint, France
madani@univ-paris12.fr

Abstract. In this paper, we propose a control strategy for a nonholonomic robot which is based on using the learning process of an Adaptive Neural Fuzzy Inference System (ANFIS). The proposed neuro-controller allows the robot track a desired reference trajectory. After a short reminder about Adaptive Neural Fuzzy Inference System, we describe the control strategy which is used on our virtual nonholonomic robot. And finally, we give the simulations' results where the robot have to pass into a narrow path as well as the first validation results concerning the implementation of the proposed concepts on real robot.

Keywords: Neuro-Fuzzy, Learning process, Nonholonomic robot, Controller, Real-Time.

1 Introduction and Problem Stating

The wheeled robots constitute a major class of robots used in a wide range of applications (as transportation, logistic and warehouse applications, security and military applications, etc...).

Most of the control approaches concerning this kind of robots are based on asymptotic stabilization with the feedback controls. Different methods have been used to reduce or to transform the nonlinear kinematics issued equations into a linear approximation system. The approach proposed by Samson in 1995 transforms the nonlinear system into a chained system with the feedback control to solve the path-following problem [1]. Several authors have addressed the problem of tracking admissible trajectory by applying dynamic feedback linearization techniques ([2] to [5]). In [6] Morin and Samon are certainly the firsts to address the problem of tracking arbitrary trajectories (i.e., not necessarily for the controlled robot) based on the conception of transverse functions. And in [7], Barfoot and Clark propose a feedback control law inheriting strong robustness properties associated with stable linear systems. However, the above-mentioned approach yields slow convergence, making it quite inappropriate for real-time applications.

This short overview emphasizes the fact that all proposed approaches are based on a kinematical modeling and most of them have a slow convergence. Their main drawback is related to the fact that they have been designed for specific usage and could not always be generalized. The difficulties are inherent to the nonlinear nature

of the robot's dynamics which is sensitive as well to the given trajectory's shape as to the applicative context (e.g. the nature of the robot, the nature of ground, etc...).

An alternative solution to the kinematical modeling is to use "Machine Learning" based approaches ([8] to [11]). In this paper, we propose a new approach based on using the learning process of an Adaptive Neural Fuzzy Inference System (ANFIS) [11]. This approach may be decomposed in two parts: the first one allows rotting an arbitrary (e.g. given) path into several trajectories (characterized as "desired trajectories"), and the second is composed of two ANFIS controllers (one controlling the position and the other the orientation). In fact, the suggested ANFIS based control approach doesn't depend on kinematics issues, and although in the present work it is applied for nonholomic robot's control, the concept may be generalized as a global control strategy to another kind of wheeled robots.

The paper is organized in six sections. In the next section, we introduce the Adaptive Neural Fuzzy Inference System. In the third section, we give the kinematical model of the nonholonomic robot in order to state the control parameters, describing how we can control the wheeled robot with ANFIS. In section 4, we present the suggested control strategy. Simulations' and validation's results are shown in the fifth section. We give the simulations' results where the robot have to pass into a narrow path as well as the first validation results concerning the implementation of the proposed concepts on real robot. At last, we get some brief conclusions.

2 Brief Overview of Adaptive Neuro-Fuzzy Inference System

The main advantage of a Fuzzy Inference Systems is that they allow dealing with linguistic rules, making such systems appropriate to design control strategy where it is difficult to use the mathematical modeling. However, its main disadvantage is that it needs knowledge of an expert. It also needs a relatively long time to get the accurate membership functions. Neural network based approaches, or more generally adaptive systems based on learning process, can overcome this disadvantage improving the basic fuzzy inference system. ANFIS, combining neural networks and fuzzy inference systems, is a class of such adaptive fuzzy inference systems. In this section, we briefly remind the ANFIS architecture initially proposed by Jang in 1995 [8].

Let us assume a control system with m inputs X_1, X_2, \dots, X_m and one output Y ; let suppose n linguistic rules, where each rule R_i , where $i \in \{1, 2, \dots, n\}$ is the index of the rule, can be expressed as relation (1).

$$\text{If } X_1 \text{ is } A_{i1} \text{ and } X_2 \text{ is } A_{i2} \text{ and } \dots \text{ and } X_m \text{ is } A_{im} \text{ Then } Y \text{ is } w_i \quad (1)$$

$$\mu_{ij} = e^{-\frac{(X_j - a_{ij})^2}{2b_{ij}^2}} \quad \text{and} \quad Y = \frac{\sum_{i=1}^n u_i w_i}{\sum_{i=1}^n u_i} \quad (2)$$

A_{ij} is a fuzzy set for i -th rule and j -th input and w_i is a real number that represents a consequent part. In the present case, the membership function is defined as a Gaussian function, given by equation (2). The output of the issued neural-fuzzy network is given by equations (2), where $u_i = \mu_{i_1} \mu_{i_2} \cdots \mu_{i_j} \cdots \mu_{i_m}$. Now, let us define the set of parameters $z = [z_a, z_b, z_w]$, supposed to be adapted during the neural based minimization process, where $z_a = (a_{i_1}, \cdots, a_{i_j}, \cdots, a_{i_m})$, $z_b = (b_{i_1}, \cdots, b_{i_j}, \cdots, b_{i_m})$ and $z_w = (w_1, \cdots, w_i, \cdots, w_n)$. Let us also introduce the function $F(z, t)$, a quadratic function defined by equation (3), to be minimized during the aforementioned process, where t is the current time, $Y(t)$ is the output of the neural fuzzy network and $Y^d(t)$ is the desired output.

$$F(z, t) = f(\Delta_Y^2) \quad \text{with} \quad \Delta_Y^2 = (Y(t) - Y^d(t))^2 \quad (3)$$

In 1995, Godjevac show that if $F(z, t)$ is defined as it has been stated-above, then it is possible to use an iterative procedure to update the above-defined parameters in order to minimize the function $F(z, t)$ [9].

3 Nonholonomic Robot's Dynamics

Generally, the control of wheeled robots is performed by a follow of reference path and supposes to measure both the position and orientation with respect to a fixed frame. Let us consider a given trajectory C in the reference frame, and a point P attached to the robot chassis, at the mid-distance of the wheels, as illustrated in Fig.1. The state of the robot can be described by a triplet as $P(x, y, \theta)$, in which x and y are the robot's current coordinates, measured in the reference frame. θ is the angle formed by the robot's motion direction and the x -axis. For an unicycle-type mobile robot, the goal of the control strategy is to compute the velocities of each wheel in order to make the robot follow a given (e.g. desired) path. The given trajectory can be expressed as a time-dependent function $P(x^d(t), y^d(t), \theta^d(t))$, where $\theta^d(t)$ represents of the trajectory's curvature at each step time t , $x^d(t)$ is the desired x -axis coordinate and $y^d(t)$ is the desired y -axis coordinate of the robot.

The kinematical modeling of this wheeled robot (i.e. unicycle-type mobile robot) may be represented by the set of equations (4), where V_x and V_y represent the instantaneous horizontal and vertical velocities of the point P located at mid-distance of the actuated wheels respectively; V is the intensity of the longitudinal velocity and Ω the angular velocity of the robot [11]. Ω^{left} and Ω^{right} are the angular velocity of the "left" and "right" wheels, respectively, r is the wheels' radius and l is the distance between the two wheels. However, in the case of nonholonomic robots complying with the kinematical model, represented by equations (4), the achievement of the aforementioned goal is not a trivial problem because of nonlinear nature of the robot's dynamics sensitive as well to the given trajectory's shape as to the environment.

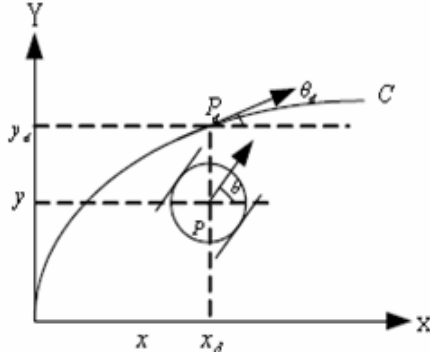


Fig. 1. Robot's coordinates described by a triplet as $P(x, y, \theta)$

$$V_x = V \cos(\theta), \quad V_y = V \sin(\theta), \quad \text{and} \quad \dot{\theta} = \Omega \quad (4)$$

$$V = \frac{r}{2}(\Omega^{right} + \Omega^{left}) \quad \text{and} \quad \Omega = \frac{r}{2l}(\Omega^{right} - \Omega^{left})$$

4 Proposed Control Strategy and Related Architecture

The proposed approach includes two neuro-controllers (one controlling the position and the other the orientation), allowing to track the desired trajectories. The goal of the two neural networks is to control the velocity of each wheel in order to minimize the position error, defined as the error between robot's current position and the desired position ($\varepsilon_x = x(t) - x^d(t)$ and $\varepsilon_y = y(t) - y^d(t)$), and the orientation error, defined as the error between the robot's current orientation and the desired orientation ($\varepsilon_\theta = \theta(t) - \theta^d(t)$).

The orientation control allows the robot to rotate in order to follow the target angle. Consequently, the related ANFIS based controller requires one input ($X_\theta(t) = \varepsilon_\theta$) which is the aforementioned orientation error and one output ($Y_\theta(t) = \Delta\Omega(t) = \Omega^{right}(t) - \Omega^{left}(t)$) defined as an angular velocities difference. $\Omega^{right}(t)$ and $\Omega^{left}(t)$ represent right wheel's and left wheel's current angular velocities, respectively. According to the Section 2, at each time step (e.g. iteration), the parameters $z_w^\theta = (w_1^\theta, \dots, w_i^\theta, \dots, w_n^\theta)$ are updated in order to minimize the function $F_\theta(t) = (\theta(t) - \theta^d(t))^2$.

The position ANFIS based controller requires two inputs ($X_{px}(t) = \varepsilon_x$ and $X_{py}(t) = \varepsilon_y$) which are defined on the basis of the position error and one output ($Y_p(t) = \Omega^{right}(t) = \Omega^{left}(t)$) which is also defined as an angular velocity. In the same way, at each iteration, the parameters $z_w^p = (w_1^p, \dots, w_i^p, \dots, w_n^p)$ are updated in order to minimize the function $F_p(t) = \sqrt{(x(t) - x^d(t))^2 + (y(t) - y^d(t))^2}$.

The trajectory's control strategy combines the above-described orientation's and position's ANFIS based controllers. Fig.2 gives the bloc-diagram of the suggested control strategy. In this case, the angular velocities of two wheels are given by the set of two equations (5). $\Omega_p^{right}(t)$ and $\Omega_p^{left}(t)$ are obtained from the position controller, and $\Delta\Omega$ is obtained from the orientation controller.

$$\Omega^{right}(t) = \Omega_p^{right}(t) \quad \text{and} \quad \Omega^{left}(t) = \Omega_p^{left}(t) + \Delta\Omega \quad (5)$$

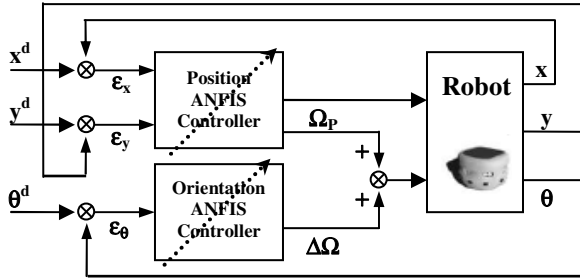


Fig. 2. Bloc-diagram of the trajectory's ANFIS based control strategy, including an orientation's dedicated neuron controller and a position's devoted neuron-controller

It is pertinent to note that in the most of already proposed neuro-controllers, the control task is performed in “generalization” phase (after accomplishing the training phase). In our approach, the trajectory's control (orientation and position control) is performed by adjusting (updating) the set of parameters z , minimizing the function $F(z, t)$. In other words, our control strategy uses the “learning” process (e.g. training phase) to control the robot's wheels. This way of doing may be interpreted as a kind of “reflexive” artificial cognitive mechanism (by opposition to “conscious” cognitive mechanism – see [12]) regulating the robot's rolling organs (wheels).

5 Implementation and Validation Results

5.1 Case Study and Simulation Results

The case study concerns the control of a wheeled robot moving from an initial position to a final (e.g. goal) position by passing a narrow path, according to a predefined trajectory. Fig.3 shows the case study's frame where, starting from the point “A”, the robot is supposed to reach the point “C” (final position) accordingly to the indicated trajectory. As shows Fig.3, the proposed example may be decomposed into three parts: firstly the robot moves from the point A toward the obstacles, secondly the robot follows a circle trajectory, and finally the robot goes towards the final position. The three above-mentioned parts could be characterized as follow:

- During the first part, the robot moves from initial position A to the obstacle with position's control only (the orientation remains unchanged). In this part, robot follows the vertical line $x^d(t) = 0.3$ without the orientation control ($\theta^d(t) = 0^\circ$). $y^d(t) = y^d(t-1) + \Delta y$ where, Δy is chosen according to both length L and duration T of the path's accomplishment.
- During the second part, firstly the robot turn around itself from $\theta^d(t) = 0^\circ$ to $\theta^d(t) = -180^\circ$ and secondly the robot use trajectory control to follow a circular arc. Finally, the robot rotates from $\theta^d(t) = -270^\circ$ to $\theta^d(t) = 0^\circ$.
- In the last part, as in the first part, the robot follows a vertical line ($x^d(t) = 0.0$) and passes the narrow path toward the final position (point "C").

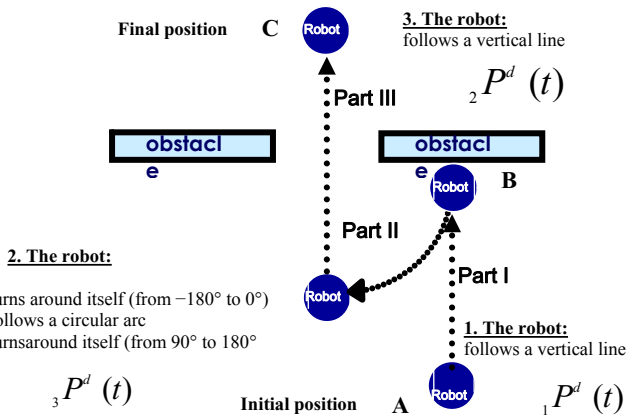


Fig. 3. Description of the “case study” frame and the desired (e.g. planned trajectory)

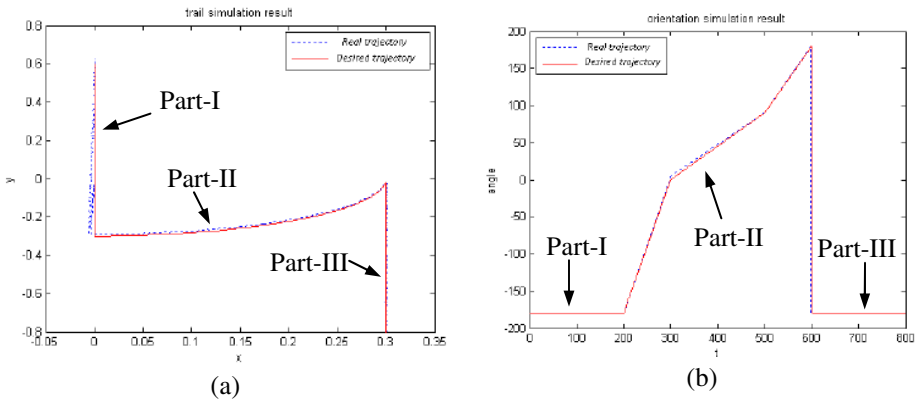


Fig. 4. Accomplished (dot line) and desired (continue line) positions (a) and orientation (b) relative to the three sub-trajectories defining the case study example

Simulation have been performed by using Webots software (www.cyberbotics.com) simulating a virtual robot within the above described case study's frame. The controller has been designed under MatLab software platform (www.mathworks.com). Fig.4 (Fig.4-a and Fig.4-b) shows the position and orientation of the robot during the simulation, respectively.

5.2 Implementation on Real Robot

The used robot is the robot kheperaIII with the korebotLE module (<http://www.k-team.com>) and “dsPIC 30F5011” microprocessor (managing all robot's devices through a I2C communication), equipped with two motors associated with incremental encoders, 9 infrared sensors and five ultrasonic sensors. In addition, this robot offers the possibility to connect a KoreBot board allowing increasing the computational abilities.

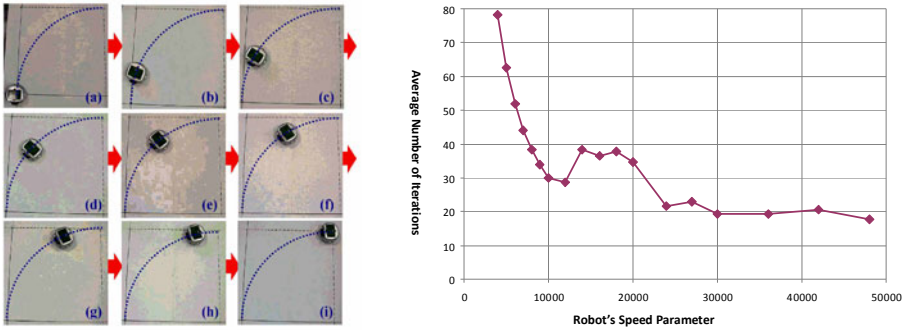


Fig. 5. Robot accomplishing a curvilinear trajectory, shaped in a 60-by-60 cm² 2-D frame (left pictures) and number of iterations versus the robot's speed for the same trajectory (right)

The above-presented control strategy has been designed using c language and implemented on the KoreBot. Both orientation (e.g. rotation) and position of the robot are computed by using an odometer based process. Experimental validations have been realized considering different kinds of trajectories (linear, circular, curvilinear, etc...). Fig.5 gives an example of experimental validation's results on accomplishing a curvilinear trajectory (a nonlinear trajectory where the robot's position and orientation change nonlinearly.), shaped in a 60-by-60 cm² 2-D frame. The starting point is $(x = 0, y = 0, \theta = 0^\circ)$ and the final destination is located at $(x = 60, y = 60, \theta = +90^\circ)$. As shows this figure through the 9 photographs, the robot follows correctly the desired trajectory. As it could be seen from the Fig.5, the considered trajectory is of same kind as accomplished in the “Part II” of the Fig.3, confirming the effective implementation of the proposed ANFIS based concept. On the other hand the right diagram of Fig.5 shows the weak number of iterations needed to reach the goal. These experimental results also demonstrate the effectiveness of the described neural architecture in satisfying the frame of real-time control requirements.

6 Conclusion

We have proposed a control strategy for nonholonomic robot based on Adaptive Neural Fuzzy Inference System. We have presented results relative to the control of a robot aiming to avoid an obstacle. The obtained results show the viability of the proposed machine-learning based approach in controlling as well the robot's position as its orientation. The first interest of our approach is that it is independent from the robot's kinematical model. The second interest is that it offers the possibility to design multi-level control, where the path planning and the trajectory computing are separated. Finally, contrary to other neuron-controller, the proposed approach uses the training process (continuously) to perform the control. This way of doing may be interpreted as kind of "unconscious" artificial cognitive mechanism where the "knowledge" (e.g. learning) based process operates in a "reflexive" way regulating the robot's rolling organs (wheels).

Further works will focus the generalization of such multi-level control strategy to the control of a robot's formation (e.g. a group of several Khepera III robots).

References

1. Samson, C.: Control of chained systems application to the path following and time varying point-stabilization. *IEEE Trans. on Automatic Control* 40(1), 64–77 (1995)
2. Kolmanovsky, I., McClamroch, N.H.: Developments in nonholonomic control problems. *IEEE Control Systems Magazine* 15, 20–36 (1995)
3. D'Andrea-Novel, B., Campion, G., Bastin, G.: Control of nonholonomic wheeled mobile robots by state feedback linearization. *Int. J. of Robotics Research* 14(6), 543–559 (1995)
4. De Luca, A., Di Benedetto, M.D.: Control of nonholonomic systems via dynamic compensation. In: *Workshop on System Structure and Control*, vol. 29(6), pp. 593–608 (1993)
5. Fliess, M., Levine, J., Martin, P.: Flatness and defect of non-linear systems: introductory theory and examples, *Internat. J. of Control* 61(6), 1327–1361 (1995)
6. Morin, P., Samon, C.: Practical stabilization of drift-less systems on Lie groups: the transverse function approach. *IEEE Trans on Automatic Control* 48(9), 1496–1508 (2003)
7. Barfoot, T.D., Clark, C.M.: Motion Planning for Formations of Mobile Robots. *Robotics and Autonomous Systems* 46(2), 65–78 (2004)
8. Jang, J.-S.R., Sun, C.T.: Neuro-fuzzy modeling and control. *Proc. of IEEE*, 378–406 (1995)
9. Godjevac, J.: A Learning Procedure for a Fuzzy System: Application to Obstacle Avoidance. In: *Proc. of the International Symposium on Fuzzy Logic*, pp. 142–148 (1995)
10. Pascal, M., Claude, S.: Motion Control of Wheeled Mobile Robots. In: Bruno, S., Oussama, K.S. (eds.) *Handbook of Robotics*, pp. 799–825. Springer, Heidelberg (2008)
11. Wang, T., Sabourin, C., Madani, K.: ANFIS controller for nonholonomic robots. In: *Proc. of Internat. Conf. on Neural Networks and Artificial Intelligence (ICNNAI 2010)*, Brest, Byelorussia, pp. 320–325 (2010); ISBN: 978-985-6329-79-4
12. Madani, K., Sabourin, C.: Multi-level cognitive machine-learning based concept for human-like "artificial" walking: application to autonomous stroll of humanoid robots. *Neurocomputing; Special Issue on Linking of phenomenological data and cognition* (in Press); doi: N° 10.1016/j.neucom.2010.07.021

A Multi-Objective Evolutionary Algorithm for Network Intrusion Detection Systems

J. Gómez¹, C. Gil², R. Baños², A.L. Márquez², F.G. Montoya³, and M.G. Montoya²

¹ Dpt. Languages and Computation, Univ. of Almería, Carretera Sacramento s/n, Cañada San Urbano, 04120 Almería, Spain
Tel/Fax (+34) 950214036/950015129
jgomez@ual.es

² Dpt. Computer Architecture and Electronics, Univ. of Almería, Carretera Sacramento s/n, Cañada San Urbano, 04120 Almería, Spain
{cgilm, rbanos, almarquez, dgil}@ual.es

³ Dpt. Rural Engineering, Univ. of Almería, Carretera de Sacramento s/n, Cañada San Urbano, 04120 Almería, Spain
pagilm@ual.es

Abstract. Attacks against computer systems are becoming more complex, making it necessary to develop new security systems continually, such as Intrusion Detection Systems (IDS) which provide security for computer systems by distinguishing between hostile and non-hostile activity. With the aim of minimizing the number of wrong decisions of a misuse (signature-based) IDS, an optimization strategy for automatic rule generation is presented. This optimizer is a Pareto-based multi-objective evolutionary algorithm included within a network IDS, which has been evaluated using a benchmark dataset. The results obtained show the advantages of using this multi-objective approach.

Keywords: computer security, network intrusion detection system, signature detection, Snort, evolutionary multi-objective optimization.

1 Introduction

An intrusion can be defined as the actions that attempt to compromise the integrity, confidentiality or availability of a resource [1]. An intrusion detection system (IDS) is a software designed to monitor the system with the aim of detecting and alerting about any hostile activity in a computer or network of computers.

A typical classification of IDS is based on whether they use a signature detection or anomaly detection paradigm. The majority of IDS are based on signature detection, which consists of analyzing the network traffic looking for patterns that match a library of known signatures. These signatures are composed of several elements that allow the traffic to be identified. One of the most popular open-source IDS that use signature detection is Snort [2], which uses a database of previous known attacks in its operation. The main advantage of this type of IDS is that they provide high detection accuracy with few false positives, but with the disadvantage that they are not able to detect new attacks other than those previously stored in the database. Some IDS

that use this strategy are Snort [2], Network Security Monitor [3], Network Intrusion Detection [4], etc. On the other hand, anomaly detection IDS have the ability to detect new attacks, but at the cost of increasing the number of false positives. In an initial phase the anomaly-based IDS is trained in order to obtain a normal profile of activity in the system. The incoming traffic is then processed in order to detect variations in comparison with the normal activity, in which case it will be considered as a suspicious activity. In previous work we have implemented an anomaly pre-processor that extends the functionality of Snort IDS, making it a hybrid IDS [5].

The aim of the paper is to extend the functionalities of a well-known IDS (Snort) by allowing the automatic generation of signatures (rules) to detect anomalous traffic. In particular, a Pareto-based multi-objective evolutionary algorithm (MOEA) is presented within the detection engine of Snort which will optimize the rule generation with the aim of minimizing two different objectives: non-detected hostile traffic and the non-hostile traffic erroneously considered as hostile. This MOEA, named MOEA-Snort, is included within the detection engine module and is designed to optimize the rules responsible for detecting new attacks.

2 General Operation of Signature-Based Network IDS

The general operation of signature-based network IDS (NIDS) consist of capturing packets from the network traffic (TCP, UDP, etc.) which are compared against a set of signatures (rules) of previous attacks to determine whether or not the traffic is considered anomalous. Thus, the IDS should determine if this traffic is normal or anomalous, which involves four different possible situations:

- True positive. Right, the IDS detects hostile traffic.
- False positive. Wrong, the IDS erroneously considers non-hostile traffic as an attack.
- True negative. Right, the IDS does not detect hostile traffic because it does not exist.
- False negative. Wrong, the IDS erroneously accepts hostile traffic.

As commented above, Snort [2] is a signature-based NIDS that allows the status of a network to be monitored analyzing the network traffic in order to detect any intrusion attempt. Snort implements a detection engine that can register, warn about, and respond to any attack previously stored in a database. Snort is available under GPL and runs under Windows and GNU/Linux. It is one of the most widely used IDS, and has a large number of predefined signatures which are continuously updated. As Figure 1 shows, the basic elements of its architecture are: the module of capture of traffic, which allows all the network packets to be captured; the decoder, which is responsible for creating data structures with the packets and identifying the network protocols; the pre-processors, which allow the system functionalities to be extended; the detection engine, which analyzes the packets according to the signatures; the file of signatures where the known attacks are defined for their detection; the detection plug-ins, which allow the functionality of the detection engine to be modified; and finally, the output plug-ins for determining which, how and where the alerts are saved (e.g. text file, database).

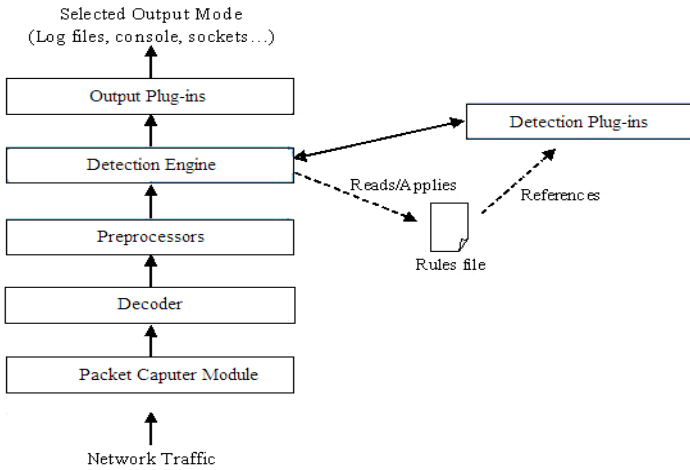


Fig. 1. Internal Scheme of Snort

3 MOEA-Snort: A New MOEA for Automatic Rule Generation

Evolutionary Algorithms (EA) [6] are optimization methods inspired by the principles of natural evolution that have been successfully applied to a large number of complex real optimization problems. An evolutionary algorithms works by maintaining a population of solutions which evolves during a number of iterations by applying certain operators, such as selection, mutation and crossover. A multi-objective evolutionary algorithm (MOEA) is an EA that simultaneously optimizes several objectives. Most efficient MOEA used the Pareto-optimization concept [7], which, instead of giving a scalar value to each solution, establishes relationships between solutions according to Pareto-dominance relations.

As commented above, the main disadvantage of signature detection IDS is the impossibility of detecting new attacks other than those previously stored in the database. Therefore, it would be advisable to extend the capability of the signature detection module in order to allow the automatic generation of rules that simultaneously minimize the number of false positives and false negatives. Recently, some authors have proposed using MOEA-based approaches for IDS [8,9]. The practical advantage of Pareto-based optimization algorithms is that they obtain not one, but a set (front) of non-dominated solutions, which can be used in a subsequent phase by the decision maker to select one of the solutions according to particular criteria. In the automatic rule generation of intrusion detection systems, this selection process can be carried out according to the number of false positives (non-hostile traffic considered as an attack), and false negatives (hostile traffic not detected). For instance, if the IDS were located in a financial institution it would be critical to reduce the number of non-detected attacks, i.e. solutions with fewer false negatives should be selected.

This paper proposes using a Pareto-based MOEA, named MOEA-Snort, which will be included within the detection engine module (see Figure 1) and is designed to optimize the rules responsible for detecting new attacks. The general scheme of this multi-objective evolutionary algorithm is shown in Figure 2.

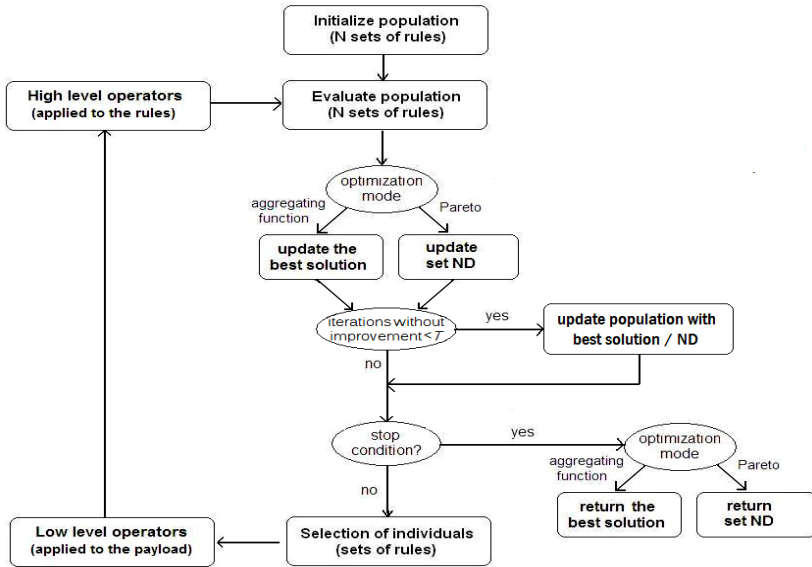


Fig. 2. MOEA-Snort: general operation

The general structure of MOEA-Snort is based on initializing a population of solutions (individuals) that evolve until the stop condition is fulfilled. Each individual is composed of a certain number of signatures (rules). In Snort, a signature is composed of the header (source address, destination address, ports, etc.) and its options (payload, metadata, etc.). The most important part of a signature is the payload that is the part of a package that identifies it. Therefore, MOEA-Snort will be responsible for updating the database of signatures in runtime according to incoming traffic. This task is performed using several operators that have been designed to improve rules dynamically with the aim of improving their ability to detect anomalous incoming traffic, and therefore reducing the number of false positives and false negatives. These operators are described in the Table 1 taking into account the following notation: RI^* and RJ^* are the sets of rules of the individuals I and J (respectively), RI_j and RI_k represent two existing rules of individual I , RI_w and RI_x represent new rules generated in individual I , and $p(RI_j)$ represents the payload of rule RI_j .

In each iteration, the individuals are evaluated according to Pareto-dominance relationships, and the best solutions are continually stored. A selection routine is invoked in order to determine the best individuals and rules within each individual. Furthermore, with the aim of improving the quality of the signatures, a set of operators is applied in order to improve them. When the stop condition is fulfilled, the best solution (single aggregate mode) or set of non-dominated solutions (Pareto-optimization) is returned. MOEA-Snort applies a backtracking procedure consisting of recovering the best solutions previously found whenever the quality of the solutions has not improved in the last T iterations (T is named the backtracking rate). The stop condition depends on the optimization mode. If single aggregate objective function is used, the stop condition is that the best obtained solution does not improve during the last 50

iterations, while if Pareto-optimization is used, the stop condition is that the archive of non-dominated solutions (ND) is not updated in the last 50 iterations. Therefore, MOEA-Snort is based on using the concept of Pareto-optimization where two fitness functions, representing the number of false negatives and false positives, respectively, are optimized, and a set of non-dominated solutions as a representative sample of the Pareto-optimal set.

Table 1. Operators

Operator	Definition
Division. The payload is divided into two parts ranging from 40% to 60% each one	$p(R_{tw}) \leftarrow \text{reduction}_{\text{left}} [p(R_{ij}), r]$ $p(R_{tk}) \leftarrow \text{reduction}_{\text{right}} [p(R_{ij}), \text{length}[p(R_{tw})]]$ $(0.4 * \text{length}[p(R_{ij})] \leq r \leq 0.6 * \text{length}[p(R_{ij})])$
Elimination. A bad rule is removed from an individual in order to reduce the number of rules	$R_{it} \leftarrow R_{it} - R_{ij}$
Inclusion. A good rule is removed if its payload is included in the payload of another good rule	if $\{p(R_{ij}) \subseteq p(R_{tk})\}$ then $\text{remove}(R_{ij}), j \neq k$
Migration. A good rule is migrated from one individual to another one	$R_{it} \leftarrow \text{copy}[R_{it}, R_{ij}]$
Reduction on the left. The first r bits of the payload are removed	$p(R_{ij}) \leftarrow \text{reduction}_{\text{left}} [p(R_{ij}), r]$
Reduction on the right. The last r bits of the payload are removed	$p(R_{ij}) \leftarrow \text{reduction}_{\text{right}} [p(R_{ij}), r]$
Union. The payloads of two good randomly selected rules are combined, and a new rule (w) is then generated	$p(R_{tw}) \leftarrow \text{union}[p(R_{ij}), p(R_{tk})], j \neq k$

4 Results and Discussion

The performance of MOEA-Snort first has been tested in the test network DARPA project [10] and then tested on the network at the University of Almeria obtaining in both cases the same results. The traffic of DARPA 1999 dataset has been used, which is free of attacks, in combination with a total of 173 attacks (on web servers and ftp) that are inserted into this traffic. For example, the attack *Wu-Ftpd File Globbing Heap Corruption Vulnerability* [11] has been inserted with the aim that MOEA-Snort creates the signature that could recognize this new attack. Obviously, this attack is unknown for MOEA-Snort.

When applying heuristic methods to optimization problems it is advisable to perform a sensitivity analysis, i.e. to determine to what extent the output of the model depends upon the inputs. In the following, two of the parameters used by MOEA-Snort have been analyzed: the population size and the backtracking rate. The population size has been set to 3, 5 and 10 individuals, while the backtracking rate has been set to 5 and 20 iterations. Figure 3 shows the non-dominated fronts obtained in the different executions of MOEA-Snort.

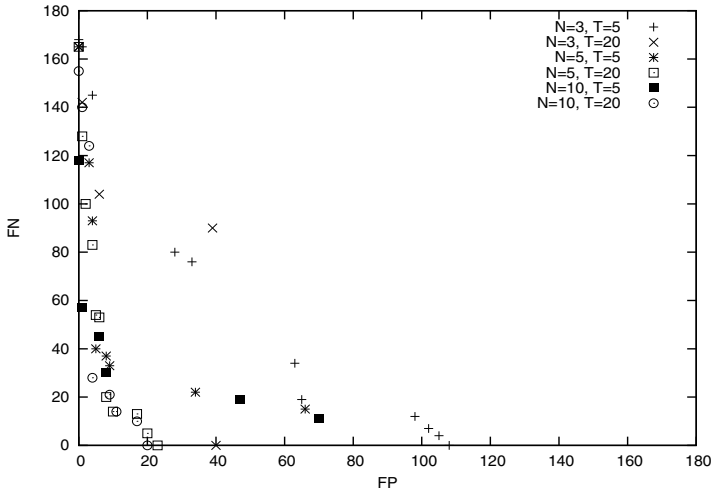


Fig. 3. Non-dominated fronts obtained using different population size and backtracking rate

Although the quality of the non-dominated fronts can be observed graphically, it is suitable to use performance measures to determine the relative quality of the fronts. The Coverage Rate metric [12] has been adopted in this work in order to compare the different executions of MOEA-Snort. Given two sets of solutions, X and X', the coverage rate function $CR(X, X')$ expresses the fraction of solutions of X' that are dominated by the solutions of X. The CR value equal to 1 indicates that all the solutions of X dominate the solutions of X'.

Table 2. Coverage rate of each parametric configuration

	<i>N</i> =3 <i>T</i> =5	<i>N</i> =3 <i>T</i> =20	<i>N</i> =5 <i>T</i> =5	<i>N</i> =5 <i>T</i> =20	<i>N</i> =10 <i>T</i> =5	<i>N</i> =10 <i>T</i> =20	AVG
<i>N</i> =3 <i>T</i> =5		0.2000	0.0000	0.0000	0.0000	0.0000	0.0400
<i>N</i> =3 <i>T</i> =20	0.8182		0.1250	0.0000	0.3333	0.0000	0.2553
<i>N</i> =5 <i>T</i> =5	0.5455	0.4000		0.1818	0.1667	0.1250	0.2837
<i>N</i> =5 <i>T</i> =20	1.0000	0.8000	0.7500		0.5000	0.5000	0.7100
<i>N</i> =10 <i>T</i> =5	0.7273	0.8000	0.6250	0.4545		0.3750	0.5963
<i>N</i> =10 <i>T</i> =20	1.0000	1.0000	0.8750	0.6364	0.66667		0.8356

Table 2 shows the results obtained when using *N*=3,5,10 individuals, a parameter of backtracking rate equal to *T*=5,20. The meaning of this table is that the value 0.3333 in row (*N*=3; *T*=20) and column (*N*=10; *T*=5) indicates that 33.33% of

solutions obtained when using ($N=10$; $T=5$) are dominated by at least one solution of the non-dominated front obtained by the execution using ($N=3$; $T=20$). If it is observed the last column, which represents the average value of all coverage rates, the best result is obtained by ($N=10$; $T=20$), dominating 83.56% of the other solutions. Thus, it is observed that the higher the population size and the backtracking rate, the better the quality of the non-dominated set (ND).

5 Conclusions

This paper presents MOEA-Snort, a multi-objective evolutionary algorithm to optimize automatic rule generation in order to detect anomalous traffic not stored in the database of attacks. MOEA-Snort uses two optimization modes, a single aggregate objective function and Pareto-optimization. When analyzing the traffic related to the data set of the DARPA project the results obtained using the single-aggregate objective function are very affected by the weights used in the objective function. Furthermore, the number of individuals in the population and the backtracking rate have a direct effect on the performance of the IDS, increasing the quality of the solutions when these values increase. Similar conclusions are obtained when using the Pareto-based optimization mode, with the additional advantage that it is able to obtain a wide set of solutions and so is very useful for the decision maker, while the aggregated optimization mode only provides a single solution. In both cases it is observed that the higher population size and the backtracking rate, a better quality of the solutions.

Acknowledgements

This work has been financed by the Excellence Project of Junta de Andalucía (P07-TIC02988), financed by the European Regional Development Fund (ERDF).

References

- [1] Heady, R., Luger, G., Maccabe, A., Servilla, M.: The architecture of a network level network intrusion detection system. Technical report CS90-20, Department of Computer Science, University of New Mexico (1990)
- [2] Beale, J., Baker, A., Caswell, B., Alder, R., Poore, M.: Snort 2.1 Intrusion detection, 2nd edn. (2004)
- [3] Heberlein, L.T.: Network Security Monitor (NSM) - Final Report. Lawrence Livermore National Laboratory, Davis (1995)
- [4] Lawrence Livermore National Laboratory: Network Intrusion Detector (NID). Overview. Computer Security Technology Center (1998)
- [5] Gómez, J., Gil, C., Padilla, N., Baños, R., Jiménez, C.: Design of a snort-based hybrid intrusion detection system. In: Omatu, S., Rocha, M.P., Bravo, J., Fernández, F., Corchado, E., Bustillo, A., Corchado, J.M. (eds.) IWANN 2009. LNCS, vol. 5518, pp. 515–522. Springer, Heidelberg (2009)
- [6] De Jong, K.A.: Evolutionary computation: a unified approach. MIT Press, Cambridge (2006)

- [7] Goldberg, D.: Genetic algorithms in search, optimization and machine learning. Addison Wesley, New York (1989)
- [8] Anchor, K.P., Zydallis, J.B., Gunsch, G.H., Lamont, G.B.: Extending the Computer Defense Immune System: network intrusion detection with a multiobjective evolutionary programming approach. In: Timmis, Bentley (eds.) First International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK, pp. 12–21 (2002)
- [9] Haag, C.R., Lamont, G.B., Williams, P.D., Peterson, G.L.: An artificial immune system-inspired multiobjective evolutionary algorithm with application to the detection of distributed computer network intrusions. In: Genetic And Evolutionary Computation Conference (GECCO), pp. 2717–2724 (2007)
- [10] DARPA, Intrusion detection evaluation data sets (2002), <http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/index.html>
- [11] Wu-Ftpd file globbing heap corruption vulnerability, CVE-2001-0550, <http://www.securityfocus.com/bid/3581/info>
- [12] Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the Strength Pareto approach. IEEE Transactions on Evolutionary Computation 3(4), 257–271 (1999)

A Cognitive Approach for Robots' Vision Using Unsupervised Learning and Visual Saliency

Dominik M. Ramík, Christophe Sabourin, and Kurosh Madani

Signals, Images, and Intelligent Systems Laboratory (LISSI / EA 3956), Paris Est University,
Senart Institute of Technology, Avenue Pierre Point, 77127 Lieusaint, France
{dominik.ramik,sabourin,madani}@u-pec.fr

Abstract. In this work we contribute to development of an online unsupervised technique allowing learning of objects from unlabeled images and their detection when seen again. We were inspired by early processing stages of human visual system and by existing work on human infants learning. We suggest a novel fast algorithm for detection of visually salient objects, which is employed to extract objects of interest from images for learning. We demonstrate how this can be used in along with state-of-the-art object recognition algorithms such as SURF and Viola-Jones framework to enable a machine to learn to re-detect previously seen objects in new conditions. We provide results of experiments done on a mobile robot in common office environment with multiple every-day objects.

Keywords: unsupervised learning, visual saliency, object recognition.

1 Introduction

Over the last five decades, robotics has been and continue to be subject to an ever increasing interest and has been the origin of numerous works and realizations. However, the performances of existing robots are still far from those of humans. In the 21st century, robots will be supposed to share with humans' living space (and vice-versa) and they won't be any more operated by skilled technicians. In fact, they will have to be self-sufficient enough to perform tasks in co-operation with their human users, who may have no a-priori technical skills. Consequently, future works in the robotic field should focus on the increasing of robots' autonomous abilities. In this context, one of the research interests of our laboratory (LISSI) is to design "cognitive" skill for autonomous robots, notably relating humanoid robots.

The term "cognition", derived from the Latin word "cognoscere", which means "to know" or "to recognize", refers to the ability for the processing of information applying knowledge. If the word "cognition" has been and continues to be used within quite a large number of different contexts, we consider in the present paper the term "cognition" as human-like functionality (behavior) of humanoid machines (robots) and their autonomy. In [1], we proposed a multi-level cognitive machine-learning based concept for human-like "artificial" walking. In that paper, we define two kind of cognitive functions: the "unconscious cognitive functions" (UCF: that we identify as "instinctive" cognition level handling reflexive abilities) and

“conscious cognitive functions” (CCF: that we distinguish as “intentional” cognition level handling thought-out abilities).

This paper focus on another fundamental skill of humanoid robot: a learning process based visual perception. The design of perceptual function is a major problem in robotics. Fully autonomous robots need perception to navigate in space and recognize objects and environment in which they evolve. But the question of how humans learn, represent, and recognize objects under a wide variety of viewing conditions presents a great challenge to both neurophysiology and cognitive research [2]. If we want an intelligent system to learn an unknown object from an unlabeled image, a clear need is the ability to select from the overwhelming flow of sensory information only the pertinent one. And it appears appropriate to draw inspiration from studies on human infants and robots learning by demonstration. Experiments in [3] show that it is the explicitness or exaggeration of an action that helps a child to understand, what is important in the actual context of learning. It may be generalized, that it is the saliency (in terms of motion, colors, etc.) that lets the pertinent information “stand-out” from the context [4]. We argue that in this context visual saliency may be helpful to enable unsupervised extraction and subsequent learning of a previously unknown object by a machine.

In this work, we present an approach for unsupervised real-time learning of objects from unlabeled images, and recognition of those objects when seen again. The visual architecture is given in Fig.1 and an overview of our system is summarized on Fig.2.

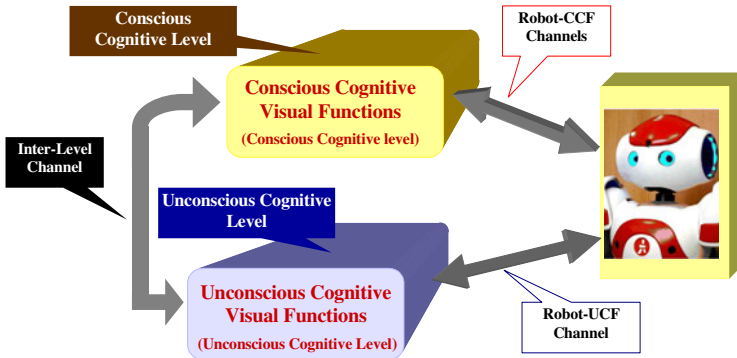


Fig. 1. Bloc diagram of Robot’s visual architecture

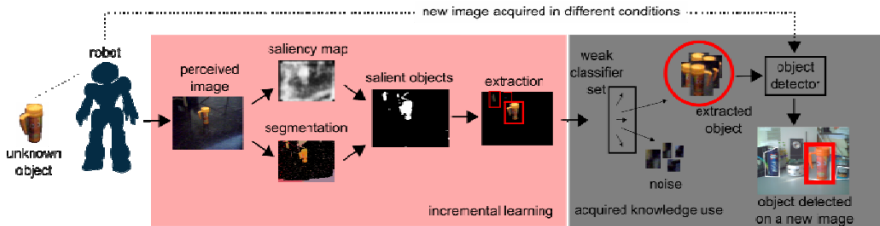


Fig. 2. An overview of the entire proposed system. An unknown object is incrementally learned and the acquired knowledge is provided to the object detector

Our approach is inspired by human vision system and by existing research on juvenile human (infants) learning process. The proposed approach extracts, first, objects of interest by means of visual saliency that we identify as unconscious cognitive visual functions (UCVF), and secondly categorizes those objects using such acquired data for learning. We consider this second process as conscious cognitive visual functions (CCVF). In order to implement our approach on real robots, we propose a fast salient object detection algorithm making use of photometric invariants. This serves to gather object samples for learning. To perform the recognition in runtime, we use two fast state-of-the-art object recognition methods: Speeded Up Robust Features [5] and the Viola-Jones detection framework [6]. To demonstrate capabilities of our approach, we apply it to a humanoid robot, enabling it to learn new objects in its surroundings without human intervention.

Section 2 of this paper presents the method of extraction of salient objects. The learning procedure is detailed in section 3. Section 4 describes experimental results. Conclusion and further work perspectives is presented in section 5.

2 Salient Object Detection and Extraction

Visual saliency is described as a perceptual quality that makes a region of image stand out relative to its surroundings and to capture attention of the observer [7]. The inspiration comes from clinical research on human vision system. In image processing, one of the first works to use visual saliency was [8], where Itti et al. use a biologically plausible approach based on a center-surround contrast calculation. Other common techniques include graph-based random walk, center-surround feature distances [9] or multi-scale contrast, center-surround histogram and color spatial distribution [10]. In [11] a biologically-motivated saliency detector is used along with an unsupervised grouping algorithm to group together images of similar objects. In [12] a purely bottom-up system with visual attention is presented, investigating the feasibility of unsupervised learning of objects from unlabeled images. Experiments are conducted by its authors on still images and on a mobile robot, where the capacity to learn landmark objects during navigation in an indoor environment is shown.

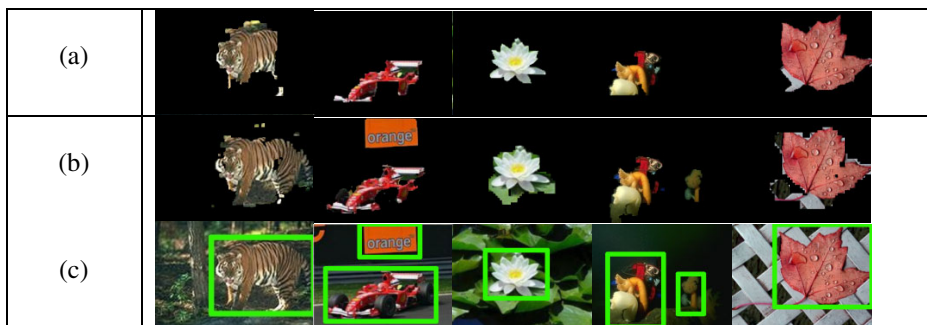


Fig. 3. Comparison of salient object detection algorithms: results of Liu et al. [10] (a) and those obtained using our approach (b). Results reported in (c) correspond to ground truth (e.g. taking into account multiple objects in the scene).

Our approach, bears some resemblance to the latter mentioned works (i.e.[11], [12]), however, by contrast to them, we demonstrate its capability of real-time learning and recognition of generic objects without supervision. In the proposed saliency computation algorithm, we are representing colors using a spherical interpretation of RGB color space (siRGB further on), with angular parameters (θ , φ) and intensity Ψ . This allows us to work with photometric invariants instead of pure RGB information. For works dealing with siRGB color space see [13] or [14]. We propose a visual saliency detector composed of two parts. One captures saliency in terms of colors and the other in frequency domain. Their resulting saliency maps are eventually merged together resulting in the final saliency map. Calculation of color saliency is done using two features: the intensity saliency and the chromatic saliency. With the saliency map computed, we can proceed to splitting it order to extract visually salient objects themselves from the image. Several final saliency map samples are shown on the bottom row of Fig.3.

3 Learning and Recognition

The both process learning and recognition are carried out on-line. When an agent (e.g. robot) takes images while it encounters a new object, saliency map of the image is calculated and salient objects are extracted and grouped incrementally as new images are acquired. Each image is processed into a number of fragments containing salient objects. Suppose that the object of interest is most of the time in field of view and the viewpoint of camera changes over time, or the object is presented in different conditions; we can assume that with increasing number of images acquired the object of interest will have high frequency of occurrence with respect to other fragments. Objects with too low frequency of occurrence in the sequence are rejected so only the important ones are learned (see Fig. 2). The following algorithm describes our learning workflow. In the first time, the algorithm classifies each found fragment, and in a second time, the learning process is updated (on-line learning).

```

acquire image
extract fragments by salient object detector
for each fragment F
    if(F is classified into one group)
        populate the group by F
    if(F is classified into multiple groups)
        populate by F the closest group by Euclidian distance of
        features
    if(F is not classified to any group)
        create a new group and place F inside
select the most populated group G
use fragments from G as learning samples for object detection
algorithm

```

3.1 Classifiers

To preserve the on-line and real time nature of learning, we have to group fragments incrementally on a per-image basis with low calculation efforts. For this task we

employ a combination of four weak classifiers $\{w_1, w_2, w_3, w_4\}$, each classifying a fragment as belonging or not belonging to a certain class. F denotes the currently processed fragment, G denotes an instance of the group in question.

Area: (7) separates fragments with too different areas. In experiments $t_{area} = 10$.

$$w_1 = \begin{cases} 1 & \text{if } c_{w_1} < t_{area} \\ 0 & \text{otherwise} \end{cases}; c_{w_1} = \frac{\max(G_{area}, F_{area})}{\min(G_{area}, F_{area})} \quad (1)$$

Aspect: the w_2 (8) classifier separates fragments, whose aspect ratios are too different to belong to the same object. In experiments, we set t_{aspect} to 0.3.

$$w_2 = \begin{cases} 1 & \text{if } c_{w_2} < t_{aspect} \\ 0 & \text{otherwise} \end{cases}; c_{w_2} = \left\| \log\left(\frac{G_{width}}{G_{height}}\right) - \log\left(\frac{F_{width}}{F_{height}}\right) \right\| \quad (2)$$

Chromaticity distribution: the w_3 (9) classifier separates fragments with clearly different chromaticity. It works over 2D normalized histograms of ϕ and θ component denoted by $G_{\phi\theta}$ and $F_{\phi\theta}$ respectively with L bins, calculating their intersection. We use $L = 32$ to avoid too sparse histogram and $t_{\phi\theta}$ equal to 0.35.

$$w_3 = \begin{cases} 1 & \text{if } c_{w_3} < t_{\phi\theta} \\ 0 & \text{otherwise} \end{cases}; c_{w_3} = \frac{\sum_{j=1}^{L-1} \sum_{k=1}^{L-1} \min(G_{\phi\theta}(j, k) - F_{\phi\theta}(j, k))}{L^2} \quad (3)$$

Texture uniformity: the w_4 (10) classifier separates fragments, whose texture is too different. We use the measure of texture uniformity calculated over the l channel of fragment. In (12), $p(z_i)$ where $i \in \{0, 1, 2, \dots, L-1\}$ is a normalized histogram of l channel of the fragment and L is the number of histogram bins. In experiments, we use 32 histogram bins to avoid too sparse histogram and value $t_{uniformity}$ of 0.02.

$$w_4 = \begin{cases} 1 & \text{if } c_{w_4} < t_{uniformity} \\ 0 & \text{otherwise} \end{cases}; c_{w_4} = \left\| \sum_{j=0}^{L-1} p_G^2(z_j) - \sum_{k=0}^{L-1} p_F^2(z_k) \right\| \quad (4)$$

A fragment belongs to a class if $\prod_{i=1}^n W_i$. A class may be populated only by one fragment per image, to prevent overpopulation by repeating patterns from the same image. If a fragment is not put into any class, a new one is created for it. If a fragment satisfies this equation for multiple classes, it is assigned to the one whose Euclidian distance is the smallest in terms of features measured by each classifier (i.e. c_{w_i}).

3.2 On-Line Learning

For detection of known objects, any suitable real time recognition algorithm can be used. For demonstration of our system, we have employed two recognition

algorithms. We will explain basics of their function and how they make use of data about objects acquired in form of groups of fragments. Speed-up Robust Features, or SURF, described in [5] is a well-established image detector technique based on matching interest points on the source image with interest points coming from the template. It provides robust detection to partial occlusions and perspective deformations. In our case we use the fragments acquired as matching templates. To speed-up the matching process, we work with key points pre-extracted from fragments. The second object recognition method, used in our work, is Viola-Jones detection framework [8]. Its principle is based on browsing sub-windows over the image and a cascade of classifiers, which determine whether the processed part of image belongs or not to a group of objects on which the classifier was trained. In this case, we use acquired fragments of an object as positive samples to learn the cascade of classifiers. As this method requires negative samples as well, we use the original images with the learned object replaced by a black rectangle. To be precise enough, the method needs up to several thousands of samples for learning. We achieved this number by applying random contrast changes, perspective deformations and rotation of learning fragments. Although Viola-Jones framework was originally designed to recognize a class of objects (i.e. human faces), rather than single instances, in our case we learn it in the way that it recognizes a class of only one object (i.e. the one found on learning fragments). It must be noticed that having the object detector learned, known objects can be detected directly from the input image when seen again.

4 Results

For experimental verification, we have collected ten common objects to be learned (an apple, a food-can, a beer-can, a book, a coke bottle, a khepera robot, a PC mouse, a mug, a pda, a shoe). We used humanoid robot Aldebaran Nao as an agent learning the objects. Images were taken with robots monocular color CMOS camera with maximum resolution of 640x480 pixels. To approach the reality as much as possible we have chosen objects with different surface properties (chromatic, achromatic, textured, smooth, reflective, ...) and put them in a wide variety of light conditions and visual contexts.

Table 1. Percentage of correct detections of object over testing image set using Viola-Jones detection framework

Viola-Jones	apple	beer	coke	khepera	mouse	mug	pda	shoe
% of correctly detected instances	98.0	88.2	77.3	60.0	76.0	89.1	80.0	81.8

Table 2. Percentage of correct detections of object over testing image set using

SURF	beans	beer	book	coke	mug	shoe
% of correctly detected instances	76.9	90.5	97.1	95.2	81.5	94.4

On Table 1, detection rates over testing dataset using a trained Viola-Jones detection framework are provided. Average detection rate is about 81.3%. It is well known that to learn properly the cascade of classifiers in this framework, a set of hundreds or few thousands of training samples are needed. Although our learning image sets contain normally up to few hundreds samples, we increase this number by automatically generating new samples with contrast and brightness modified and slightly rotated or deformed in perspective. Table 3 summarizes performance of SURF algorithm over the testing dataset. In case of this algorithm, the average detection rate was about 89.3%. Even such a fast algorithm as SURF is unable to match hundreds of templates in real time on a standard CPU. To preserve the real-time operation of detection, we pre-extract key points from each template in advance, match first in several parallel threads templates with the greatest number of key-points and stop this process when another image from camera arrives. This gives us opportunity to test up few tens template matches per frame. Further speed-up can be achieved using parallel computation power of CUDA-like architectures on modern GPUs. To demonstrate real-time abilities of our system, we have successfully run several experiments¹ with a robot required to learn, find and follow a moving object in an office environment. Images from a video from experiments are shown on Fig. 4 along with pictures from robot's camera detecting various objects from the testing set.

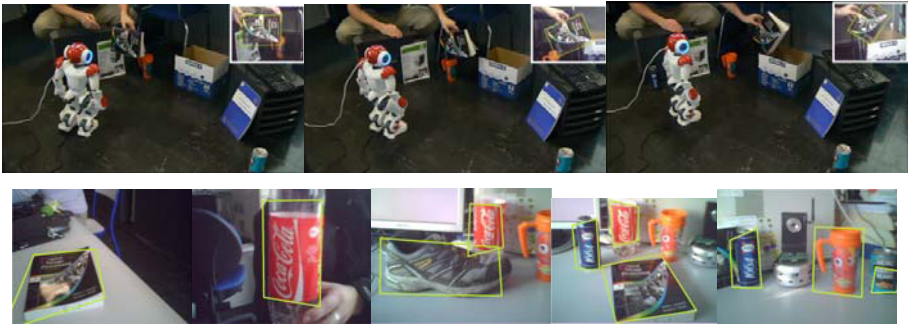


Fig. 4. Tracking a previously learned moving object (robot camera picture in upper right corner of each image) (a) and test's results relative to a set objects' detection (robot camera's issued pictures) (b)

5 Conclusion and Further Work

In this work we propose a real-time machine learning system allowing unsupervised learning of objects from unlabeled images. We suggest a novel algorithm for detection and learning of visually salient objects and demonstrate that this is a viable approach to machine learning from unlabeled images. The proposed system was successfully tested on a humanoid robot in common office environment.

¹ A video of this experiment can be found to the following address
<http://www.youtube.com/watch?v=xxz3wm3L1pE>

In future, other object detection algorithms, like receptive field co-occurrence histograms, could be adopted and different objects could be learned by algorithms that best suit their nature in a “mixture of experts” manner. An open question is, whether our technique, instead of learning individual objects, could be used for place learning and recognition. It would also be interesting to investigate, whether the saliency-based method could be applied for learning of other than visual data (eg. audio). Finally, our approach allows learning of new objects, but provides no semantic level clues about them. We are greatly interested in whether our system could be extended in semantic way by an interaction with existing information sources (e.g. the Internet) to anchor newly learned objects automatically into an existing ontology to finally understand their nature and be able to reason about them.

References

1. Madani, K., Sabourin, C.: Multi-level cognitive machine-learning based concept for human-like “artificial” walking: Application to autonomous stroll of humanoid robots. *Neurocomputing* (2010)
2. Bühlhoff, H., Wallraven, C., Giese, M.: Perceptual Robotic. In: Siciliano, B., Khatib, O. (eds.) *The Springer Handbook of Robotics*, Springer, Heidelberg (2007)
3. Brand, R.J., Baldwin, D.A., Ashburn, L.A.: Evidence for ‘motionese’: modifications in mothers’ infant-directed action. *Developmental Science* 5, 72–83 (2002)
4. Wolfe, J.M., Horowitz, T.S.: What attributes guide the deployment of visual attention and how do they do it. *Nature Reviews Neuroscience* 5, 495–501 (2004)
5. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
6. Viola, P., Jones, M.: Robust Real-time Object Detection. *International Journal of Computer Vision* 57(2), 137–154 (2002)
7. Achanta, R., Hemami, S., Estrada, F., Süsstrunk, S.: Frequency-tuned Salient Region Detection. In: *IEEE International Conference CVPR*, Miami Beach, Florida (2009)
8. Itti, L., Koch, C., Niebur, E.: A Model of Saliency-based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(11), 1254–1259 (1998)
9. Achanta, R., Estrada, F.J., Wils, P., Süsstrunk, S.: Salient Region Detection and Segmentation. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) *ICVS 2008*. LNCS, vol. 5008, pp. 66–75. Springer, Heidelberg (2008)
10. Liu, T., Zheng, N., Tang, X., Shum, H.: Learning to detect a salient object. In: *IEEE Computer Society Conference on Computer Vision*. IEEE Computer Society, Los Alamitos (2007)
11. Borba, G., Gamba, H., Marques, O., Mayron, L.: An Unsupervised Method for Clustering Images Based on Their Salient Regions of Interest. In: *Proceedings of the 14th ACM International Conference on Multimedia*, pp. 145–148. ACM, New York (2006)
12. Rutishauser, U., Walther, D., Koch, C., Perona, P.: Is bottom-up attention useful for object recognition? In: *Proceedings of the Conference on Computer Vision and Pattern Recognition, CVPR 2004*, vol. 2, pp. II-37–II-44. IEEE Computer Society, Los Alamitos (2004)
13. Mileva, Y., Bruhn, A., Weickert, J.: Illumination-Robust variational optical flow with photometric invariants, pp.152-162 (2007)
14. van de Weijer, J., Gevers, T.: Robust optical flow from photometric in- variants. In: *International Conference in Image Processing*, vol. 1 (2004)

Fusing Heterogeneous Data Sources Considering a Set of Equivalence Constraints

Manuel Martín-Merino

Universidad Pontificia de Salamanca
C/Compañía 5, 37002, Salamanca, Spain
mmartinmac@upsa.es

Abstract. Pattern Recognition algorithms depend strongly on the dissimilarity considered to evaluate the sample proximities. In real applications, several dissimilarities are available that may come from different object representations or data sources. Each dissimilarity provides usually complementary information about the problem. Therefore, they should be integrated in order to reflect accurately the object proximities.

In many applications, the user feedback or the a priori knowledge about the problem provide pairs of similar and dissimilar examples. In this paper, we address the problem of learning a linear combination of dissimilarities using side information in the form of equivalence constraints. The minimization of the error function is based on a quadratic optimization algorithm. A smoothing term is included that penalizes the complexity of the family of distances and avoids overfitting.

The experimental results suggest that the method proposed outperforms a standard metric learning algorithm and improves classification and clustering results based on a single dissimilarity and data source.

1 Introduction

Pattern recognition algorithms depend critically on the choice of a good dissimilarity [14]. However, in real applications a large number of dissimilarities are available coming from different object representations or data sources [12]. Choosing a good dissimilarity is a difficult task because they reflect often different features of the data [8]. So, instead of using a single dissimilarity it has been recommended in [7,8] to consider a linear combination of heterogeneous dissimilarities.

Several authors have proposed techniques to learn a linear combination of kernels (similarities) from the data [7,11,15]. These methods are designed for classification tasks and assume that the class labels are available for the training set. However, for certain applications such as Bioinformatics, domain experts provide only incomplete knowledge in the form of which pairs of proteins or genes are related [5]. This a priori information should be incorporated into semi-supervised clustering algorithms via equivalence constraints [1]. Thus, [14] proposed a distance metric learning algorithm that incorporates such similarity/dissimilarity information using a convex programming approach. The experimental results

show a significant improvement in clustering results. However, the algorithm is based on an iterative procedure that is computationally intensive particularly, for high dimensional applications. To avoid this problem, [16,13] presented more efficient algorithms to learn a Mahalanobis metric. However, these algorithms are not able to incorporate heterogeneous dissimilarities and rely on the use of the Mahalanobis distance that may not be appropriate for certain kind of applications.

Following the approach of [12], we consider that the integration of dissimilarities that reflect different features of the data should help to improve the performance of pattern recognition algorithms. To this aim, a linear combination of heterogeneous dissimilarities is learnt considering the relation between kernels and distances [9]. A learning algorithm is proposed to estimate the optimal weights considering only the similarity/dissimilarity constraints available. The method proposed is based on a convex quadratic optimization algorithm and incorporates a smoothing term that penalizes the complexity of the family of distances avoiding overfitting.

The algorithm has been evaluated considering several benchmark UCI datasets and two human complex cancer problems using the gene expression profiles. The empirical results suggest that the method proposed improves the clustering results obtained considering a single dissimilarity and a widely used metric learning algorithm.

This paper is organized as follows: Section 2 introduces the idealized metric considered in this paper, section 3 presents the algorithm proposed to learn a combination of dissimilarities from equivalence constraints. Section 4 illustrates the performance of the algorithm using several benchmark and real datasets. Finally, Section 5 gets conclusions and outlines future research trends.

2 The Idealized Dissimilarity

Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ be the set of input patterns. We are given side-information in the form of pairs that are considered similar or dissimilar for the application at hand. Let \mathcal{S} and \mathcal{D} be the subset of pairs of patterns known to be similar/dissimilar defined as:

$$\mathcal{S} = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i \text{ is similar to } \mathbf{x}_j\} \quad (1)$$

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i \text{ is dissimilar to } \mathbf{x}_j\} \quad (2)$$

Let $\{d_{ij}^l\}_{i,j=1}^n$ be the set of heterogeneous dissimilarities considered. Each dissimilarity can be embedded in a feature space via the empirical kernel map introduced in [10]. From now on, $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ will denote a kernel function [10] while K will refer to the kernel matrix defined as: $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) \forall i, j = 1, \dots, n$. The kernel function can be written as an inner product in feature space [10] $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ and therefore, it can be considered a similarity measure [13]. K_{ij}^l denotes the kernel matrix that represents the dissimilarity matrix $(d_{ij}^l)_{i,j=1}^n$ via the empirical kernel map.

The ideal similarity (kernel) should be defined such that it becomes large for similar patterns and small for dissimilar ones. Mathematically, the ideal kernel is defined as follows:

$$k^*(\mathbf{x}_i, \mathbf{x}_j) = K_{ij}^* = \begin{cases} \max_l \{K_{ij}^l\} & \text{If } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S} \\ \min_l \{K_{ij}^l\} & \text{If } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D} \end{cases} \quad (3)$$

The idealized kernel introduced in this paper is related to the one proposed by [2] for classification purposes: $k^*(\mathbf{x}_i, \mathbf{x}_j) = 1$ if $y_i = y_j$ and -1 otherwise, where y_i denotes the label of \mathbf{x}_i . However, there are two differences that are worth to mention. First, the ideal kernel proposed by [2] doesn't take into account the topology and distribution of the data, missing relevant information for the identification of groups in a semi-supervised setting. Second, this kernel can be considered an extreme case of the idealized kernel defined earlier and thus, more prone to overfitting.

Considering the relation between distances and kernels, the idealized distance can be written exclusively in terms of kernel evaluations [10].

3 Learning a Combination of Dissimilarities from Equivalence Constraints

In this section, we present a learning algorithm to estimate the optimal weights of a linear combination of kernels from a set of similarity or dissimilarity constraints.

Let $\{k_{ij}^l\}_{l=1}^M$ be the set of kernels obtained from a set of heterogeneous dissimilarities via the empirical kernel map introduced in [10]. If non-linear kernels with different parameter values are considered, we get a wider family of measures that includes non-linear transformations of the original dissimilarities. The kernel sought is defined as:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^M \beta_l k^l(\mathbf{x}_i, \mathbf{x}_j), \quad (4)$$

where the coefficients are constrained to be $\beta_l \geq 0$. This non-negative constraint on the weights helps to interpret the results and guarantees that provided all the individual kernels are positive semi-definite the combination of kernels is convex and positive semi-definite [11].

The optimization problem in the primal may be formulated as follows:

$$\min_{\beta, \xi} \frac{1}{2} \|\beta\|^2 + \frac{C_S}{N_S} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}} \xi_{ij} + \frac{C_D}{N_D} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}} \xi_{ij} \quad (5)$$

$$\text{s. t.} \quad \beta^T \mathbf{K}_{ij} \geq K_{ij}^* - \xi_{ij} \quad \forall (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S} \quad (6)$$

$$\beta^T \mathbf{K}_{ij} \leq K_{ij}^* + \xi_{ij} \quad \forall (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D} \quad (7)$$

$$\beta_l \geq 0 \quad \xi_{ij} \geq 0 \quad \forall l = 1, \dots, M \quad (8)$$

where the first term in equation (5) is a regularization term that penalizes the complexity of the family of distances, C_S and C_D are regularization parameters that give more relevance to the similarity or dissimilarity constraints. N_S , N_D are the number of pairs in \mathcal{S} and \mathcal{D} , $\mathbf{K}_{ij} = [K_{ij}^1, \dots, K_{ij}^M]^T$ and ξ_{ij} are the slack variables that allows for errors in the constraints.

To solve this constrained optimization problem the method of Lagrange Multipliers is used. Then, the dual problem becomes:

$$\max_{\alpha_{ij}, \gamma} \quad -\frac{1}{2} \sum_{\substack{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S} \\ (\mathbf{x}_k, \mathbf{x}_l) \in \mathcal{S}}} \alpha_{ij} \alpha_{kl} \mathbf{K}_{ij}^T \mathbf{K}_{kl} - \frac{1}{2} \sum_{\substack{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D} \\ (\mathbf{x}_k, \mathbf{x}_l) \in \mathcal{D}}} \alpha_{ij} \alpha_{kl} \mathbf{K}_{ij}^T \mathbf{K}_{kl} \quad (9)$$

$$+ \sum_{\substack{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S} \\ (\mathbf{x}_k, \mathbf{x}_l) \in \mathcal{D}}} \alpha_{ij} \alpha_{kl} \mathbf{K}_{ij}^T \mathbf{K}_{kl} - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}} \alpha_{ij} \gamma^T \mathbf{K}_{ij} - \frac{1}{2} \gamma^T \gamma \quad (10)$$

$$+ \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}} \alpha_{ij} \gamma^T \mathbf{K}_{ij} + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}} \alpha_{ij} K_{ij}^* - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}} \alpha_{ij} K_{ij}^*, \quad (11)$$

subject to:

$$0 \leq \alpha_{ij} \leq \begin{cases} \frac{C_S}{N_S} & \text{for } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S} \\ \frac{C_D}{N_D} & \text{for } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D} \end{cases} \quad (12)$$

$$\gamma_l \geq 0 \quad \forall l = 1, \dots, M, \quad (13)$$

where α_{ij} and γ_l are the Lagrange multipliers. This is a standard quadratic optimization problem similar to the one solved by the SVM. The computational burden does not depend on the dimensionality of the space and it avoids the problem of local minima.

Once the α_{ij} and γ_l are computed, the weights β_l can be obtained considering $\partial L / \partial \beta = 0$:

$$\beta = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}} \alpha_{ij} \mathbf{K}_{ij} - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}} \alpha_{ij} \mathbf{K}_{ij} + \gamma. \quad (14)$$

The weights β_l can be substituted in equation (4) to get the optimal combination of heterogeneous kernels. Next, a kernel k -means clustering algorithm [10] is run. Notice that the learning algorithm proposed may be applied together with any pattern recognition technique based on kernels or dissimilarities.

Several techniques are related to the one proposed here. In [14] it has been proposed an algorithm to learn a full or diagonal Mahalanobis metric from similarity information. The optimization algorithm is based on an iterative procedure that is more costly particularly for high dimensional problems. [1] and [6,13] have proposed more efficient algorithms to learn a Mahalanobis metric from equivalence constraints. The first one (Relevant Component Analysis), can only take into account similarity constraints. Both of them, rely solely on a Mahalanobis metric

that may fail to reflect appropriately the sample proximities for certain kind of applications. Hence, they are not able to integrate heterogeneous measures that convey complementary information.

4 Experimental Results

The algorithm proposed has been evaluated considering a wide range of practical problems. First, several clustering problems have been addressed. We have chosen problems with a broad range of signal to noise ratio (Var/Samp.), varying number of samples and classes. The first three problems correspond to benchmark datasets obtained from the UCI database: archive.ics.uci.edu/ml/datasets/. The last ones aim to the identification of complex human cancer samples using the gene expression profiles. They are available from bioinformatics2.pitt.edu. Next, we show some preliminary experiments for the prediction of protein subcellular location [7] considering a set of heterogeneous data sources. We use as a gold standard the annotation provided by the MIPS comprehensive yeast genome database (CYGD). CYGD assigns subcellular locations to 2138 yeast proteins. The primary input for the learning algorithm is a collection of seven kernel matrices obtained from different data sources. For a detailed description of the sources and kernels employed see [7].

All the datasets have been standardised subtracting the median and dividing by the inter-quantile range.

For high dimensional problems such as gene expression datasets, dimension reduction helps to improve significantly the clustering results. Therefore, for the algorithms based on a single dissimilarity we have considered different number of genes 280, 146, 101, 56 and 34 obtained by feature selection [8]. Genes have been ranked according to the method proposed by [3]. Then, we have chosen the subset that gives rise to the smallest error. Considering a larger number of genes or even the whole set of genes does not help to improve the clustering performance. Regarding the algorithm proposed to integrate several dissimilarities, we have considered all the dissimilarities obtained for the whole set of dimensions.

The similarity/dissimilarity constraints are obtained as in [14]. \mathcal{S} is generated by picking a random subset of all pairs of points sharing the same class label. The size is chosen such that the number of connected components is roughly 20% of the size of the original data set. \mathcal{D} is chosen in a similar way although the size in this case is less relevant.

Regarding the value of the parameters, the number of clusters is set up to the number of classes, C_S and C_D are regularization parameters and the optimal value is determined by cross-validation over the subset of labeled patterns. Finally, kernel k -means is restarted randomly 20 times and the errors reported are averages over 20 independent trials.

Clustering results have been evaluated considering two objective measures. The first one is the accuracy defined in [14]. One problem of the accuracy is that the expected value for two random partitions is not zero. Therefore, we have computed also the adjusted randindex defined in [4] that avoids this problem.

This index is also normalized between zero and one and larger values suggest better clustering.

Tables 1 and 2 show the accuracy and the adjusted randindex for the clustering algorithms evaluated. We have compared with a standard metric learning strategy proposed by [14], k -means clustering based on the Euclidean distance and k -means considering the best dissimilarity out of ten different measures. Both tables indicates which is the best distance for each case. Best results are shown in boldface.

Table 1. Accuracy for k -means clustering considering different dissimilarities. The results are averaged over twenty independent random subsets \mathcal{S} and \mathcal{D} .

Technique	Kernel	Wine	Ionosphere	Breast	Colon	Lymphoma
k -means (Euclidean)	linear	0.92	0.72	0.88	0.87	0.90
	pol. 3	0.87	0.73	0.88	0.88	0.90
k -means (Best diss.)	linear	0.94	0.88	0.90	0.88	0.94
	pol. 3	0.94	0.88	0.90	0.88	0.93
		χ^2	Maha.	Manha.	Corr./euclid.	χ^2
Comb. dissimilarities	linear	0.94	0.90	0.92	0.89	0.95
	pol. 3	0.96	0.89	0.92	0.90	0.92
Metric learning (Xing)	linear	0.87	0.74	0.85	0.87	0.90
	pol. 3	0.51	0.74	0.86	0.88	0.90

Table 2. Adjusted RandIndex for k -means clustering considering different dissimilarities. The results are averaged over twenty independent random subsets \mathcal{S} and \mathcal{D} .

Technique	Kernel	Wine	Ionosphere	Breast	Colon	Lymphoma
k -means (Euclidean)	linear	0.79	0.20	0.59	0.59	0.65
	pol. 3	0.67	0.21	0.60	0.59	0.65
k -means (Best diss.)	linear	0.82	0.58	0.66	0.59	0.77
	pol. 3	0.81	0.58	0.66	0.59	0.76
		χ^2	Maha.	Manha.	Corr./euclid.	χ^2
Comb. dissimilarities	linear	0.82	0.63	0.69	0.60	0.79
	pol. 3	0.85	0.60	0.69	0.63	0.73
Metric learning (Xing)	linear	0.68	0.23	0.50	0.54	0.66
	pol. 3	0.50	0.23	0.52	0.58	0.65

From the analysis of tables 1 and 2, the following conclusions can be drawn:

- The combination of dissimilarities improves significantly a standard metric learning algorithm for all the datasets considered. Our method is robust to overfitting and outperforms the algorithm proposed by Xing [14] in high dimensional datasets such as Colon cancer and Lymphoma. These datasets exhibit a high level of noise. We can explain this because the algorithm based on a combination of dissimilarities allows to integrate distances computed for several dimensions discarding the noise and reducing the errors.

- The combination of dissimilarities improves usually kernel k -means based solely on the best dissimilarity. This suggests that the integration of several dissimilarities allows to extract complementary information that may help to improve the performance. Besides, the algorithm proposed always achieves at least the same performance that k -means based on the best dissimilarity. Only for lymphoma and polynomial kernel we get worst results, probably because the value assigned to the regularization parameters overfit the data. We remark that the algorithm proposed, helps to overcome the problem of choosing the best dissimilarity, the kernel and the optimal dimension. This a quite complex and time consuming task for certain applications such as Bioinformatics.
- Finally, the combination of dissimilarities improves always the standard k -means clustering based on the Euclidean measure.
- Tables 1 and 2 show that the best distance depends on the data set considered. Moreover, we report that the performance of k -means depends strongly on the particular measure employed to evaluate the sample proximities.

Table 3. Accuracy of k -NN considering the best data sources and learning the optimal weights of a combination of heterogeneous data sources. Only five sources with non-null β_l are shown.

Source	Gen. Expre.	BLAST	Pfam	Hydropho.	Difussion	Combination	Lanckriet
Accuracy	73.30%	79.98%	82.48%	77.01%	80.16%	86.68%	88.66%
β_l	0.24	0.15	0.29	0.62	4.45	-	-

Regarding the identification of membrane protein classes, a linear combination of seven heterogeneous kernels (data sources) is learnt considering only similarity constraints. The size of \mathcal{S} is chosen such that the number of connected components is roughly 10% of the number of patterns. Once the kernel is learnt, a k -NN algorithm is run and the accuracy is estimated by ten-fold cross-validation. We have compared with k -NN based solely on a single data source and with the Lanckriet formalism [7]. Notice that the method proposed by Lanckriet is not able to work from similarity constraints only and needs the class labels. Besides, it is only applicable for SVM classifiers.

Table 3 shows that our algorithm improves k -NN based on a single kernel (source) by at least 4%. The β_l coefficients are larger for the diffusion and Hydrophobicity FFT kernels which is consistent with the analysis of [7] that suggests that diffusion kernels perform the best. Kernels removed correspond to redundant kernels and not to the less informative. Our method performs similarly to the algorithm proposed by Lanckriet Although we use only 10% of similarity constraints.

5 Conclusions

In this paper, we propose a semi-supervised algorithm to learn a combination of dissimilarities from equivalence constraints. The error function includes a penalty

term that controls the complexity of the family of distances considered and the optimization is based on a robust quadratic programming approach that does not suffer from the problem of local minima.

The experimental results suggest that the combination of dissimilarities improves almost always the performance of clustering algorithms based solely on a single dissimilarity. Besides, the algorithm proposed improves significantly a standard metric learning algorithm for all the datasets considered in this paper and is robust to overfitting.

References

1. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning a Mahalanobis Metric from Equivalence Constraints. *Journal of Machine Learning Research* 6, 937–965 (2005)
2. Cristianini, N., Kandola, J., Elisseeff, J., Shawe-Taylor, A.: On the kernel target alignment. *Journal of Machine Learning Research* 1, 1–31 (2002)
3. Dudoit, S., Fridlyand, J., Speed, T.P.: Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data. *Journal of the American Statistical Association* 97(457), 77–87 (2002)
4. Hubert, L., Arabie, P.: Comparing Partitions. *J. of Classification*, 193–218 (1985)
5. Huang, D., Pan, W.: Incorporating Biological Knowledge into Distance-Based Clustering Analysis of Microarray Gene Expression Data. *Bioinformatics* 22(10), 1259–1268 (2006)
6. Kwok, J.T., Tsang, I.W.: Learning with Idealized Kernels. In: *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 400–407 (2003)
7. Lanckriet, G., Cristianini, N., Bartlett, P., El Ghaoui, L., Jordan, M.: Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research* 3, 27–72 (2004)
8. Martín-Merino, M., Blanco, A., De Las Rivas, J.: Combining Dissimilarities in a Hyper Reproducing Kernel Hilbert Space for Complex Human Cancer Prediction. *Journal of Biomedicine and Biotechnology*, 1–9 (2009)
9. Pekalska, E., Paclik, P., Duin, R.: A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research* 2, 175–211 (2001)
10. Schölkopf, B., Smola, A.J.: *Learning with Kernels*. MIT Press, Cambridge (2001)
11. Soon Ong, C., Smola, A., Williamson, R.: Learning the kernel with hyperkernels. *Journal of Machine Learning Research* 6, 1043–1071 (2005)
12. Woznica, A., Kalousis, A., Hilario, M.: Learning to Combine Distances for Complex Representations. In: *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, USA, pp. 1031–1038 (2007)
13. Wu, G., Chang, E.Y., Panda, N.: Formulating distance functions via the kernel trick. In: *ACM SIGKDD*, Chicago, pp. 703–709 (2005)
14. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance Metric Learning, with Application to Clustering with Side-Information. In: *NIPS*, vol. 15, pp. 505–512. MIT Press, Cambridge (2003)
15. Xiong, H., Chen, X.-W.: Kernel-Based Distance Metric Learning for Microarray Data Classification. *BMC Bioinformatics* 7(299), 1–11 (2006)

A Novel Heuristic for Building Reduced-Set SVMs Using the Self-Organizing Map

Ajalmar R. Rocha Neto¹ and Guilherme A. Barreto²

¹ Federal Institute of Ceará, Campus of Maracanaú, Ceará, Brazil
ajalmar@gmail.com

² Department of Teleinformatics Engineering, Federal University of Ceará
Center of Technology, Campus of Pici, Fortaleza, Ceará, Brazil
guilherme@deti.ufc.br

Abstract. We introduce a novel heuristic based on the Kohonen's SOM, called Opposite Maps, for building reduced-set SVM classifiers. When applied to the standard SVM (trained with the SMO algorithm) and to the LS-SVM method, the corresponding reduced-set classifiers achieve equivalent (or superior) performances than standard full-set SVMs.

Keywords: SVMs, Least Squares SVMs, Self-Organizing Map, Reduced Set, ROC curve.

1 Introduction

A theoretical advantage of kernel-based machines concerns the production of sparse solutions [10]. By sparseness we mean that the induced classifier can be written in terms of a relatively small number of input examples, the so-called support vectors (SVs). In practice, however, it is observed that the application of different training approaches to the same kernel-based machine over identical training sets yield distinct sparseness [1], i.e. produce solutions with a greater number of SVs than are strictly necessary. To handle this issue, several Reduced Set (RS) methods have been proposed to alleviate this problem, either by eliminating less important SVs or by constructing a new (smaller) set of training examples, often with minimal impact on performance [2, 4, 13].

An alternative to standard SVM formulation is the Least Squares Support Vector Machine (LS-SVM) [12], which leads to solving linear KKT systems¹ in a least square sense. The solution follows directly from solving a linear equation system, instead of a QP optimization problem. On the one hand, it is in general easier and less computationally intensive to solve a linear system than a QP problem. On the other hand, the introduced modifications also result in loss of sparseness of the induced SVM. It is common to have all examples of the training data set belonging to the set of the SVs. To mitigate this drawback, several pruning methods have been proposed in order to improve the sparseness of the LS-SVM solution [3, 6, 11].

¹ Karush-Kuhn-Tucker systems.

In this paper, we introduce a novel heuristic based on the Self-Organizing Map (SOM) [5], called Opposite Maps, for building a reduced-set SVM classifier. When applied to the standard SVM and the LS-SVM approaches, the resulting reduced-set classifiers - called OM-SVM and OM-LSSVM, respectively - achieve equivalent (in some cases, superior) performances than standard full-set SVMs.

This paper is organized as follows. In Section 2 we review the fundamentals of the SVM and LS-SVM approaches, as well as the plain SOM algorithm. In Section 3 we describe the Opposite Maps Algorithm and the resulting OM-SVM and OM-LSSVM approaches. Simulations and results are shown in Section 4. The paper is concluded in Section 5.

2 Methods

The standard SVM - Consider a training data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^l$, so that $\mathbf{x}_i \in \mathbb{R}^p$ is an input vector and $y_i \in \{-1, +1\}$ are the corresponding class labels. For soft margin classification, the SVM primal problem is defined as

$$\min_{\mathbf{w}, \boldsymbol{\xi}} \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l \xi_i \right\} \tag{1}$$

subject to $y_i[(\mathbf{w}^T \mathbf{x}_i) + b] \geq 1 - \xi_i, \quad \xi_i \geq 0$

where $\{\xi_i\}_{i=1}^l$ are slack variables and $C \in \mathbb{R}$ is a cost parameter that controls the trade-off between allowing training errors and forcing rigid margins.

The solution of the problem in Eq. (1) is the saddle point of the following Lagrangean function:

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l [\alpha_i (y_i (\mathbf{x}_i^T \mathbf{w} + b) - 1 + \xi_i) + \beta_i \xi_i], \tag{2}$$

where $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^l$ and $\boldsymbol{\beta} = \{\beta_i\}_{i=1}^l$ are Lagrange multipliers. This Lagrangean must be minimized with respect to \mathbf{w} , b and ξ_i , as well as maximized with respect to α_i and β_i . For this purpose, we need to compute the following differentiations:

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}_i)}{\partial \mathbf{w}} = \mathbf{0}, \quad \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}_i)}{\partial b} = 0 \quad \text{and} \quad \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}_i)}{\partial \xi_i} = 0,$$

resulting on $\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i$, $\sum_{i=1}^l \alpha_i y_i = 0$ and $C = \alpha_i + \beta_i$, respectively. Introducing these expressions into Eq. (2), we present the SVM dual problem as

$$\max J(\boldsymbol{\alpha}) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j d_j \mathbf{x}_i^T \mathbf{x}_j, \quad \text{s.t.} \quad \sum_{i=1}^N \alpha_i y_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq C. \tag{3}$$

Once we have the values of the Lagrange multipliers, the output can be calculated based on the classification function described as

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i=1}^l \alpha_i y_i \mathbf{x}^T \mathbf{x}_i + b \right). \tag{4}$$

It is straightforward to use the *kernel trick* to generate non-linear versions of the standard linear SVM classifier. This procedure works by replacing the dot product $\mathbf{x}^T \mathbf{x}_i$ with the kernel function $k(\mathbf{x}, \mathbf{x}_i)$. A symmetric function $k(\mathbf{x}, \mathbf{x}_i)$ is a kernel if it fulfills Mercer's condition, i.e. the function $K(\cdot, \cdot)$ is (semi) positive definite. In this case, there is a mapping $\phi(\cdot)$ such that it is possible to write $K(\mathbf{x}, \mathbf{x}_i) = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}_i)$. The kernel represents a dot product on feature space into which the original vectors are mapped.

The LS-SVM Approach - The formulation of the primal problem for the LS-SVM [12] is given by

$$\min_{\mathbf{w}, \xi_i} \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \frac{1}{2} \sum_{i=1}^l \xi_i^2 \right\}, \quad \text{s.t.} \quad y_i[(\mathbf{w}^T \mathbf{x}_i) + b] = 1 - \xi_i, i = 1, \dots, l \quad (5)$$

where γ is a positive cost parameter similar to the parameter C in standard SVM formulation (see Eq. (1)). Unlike the standard SVM, in the LS-SVM framework the slack variables $\{\xi_i\}_{i=1}^l$ can assume negative values.

The Lagrangian function for the LS-SVM is then written as

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \frac{1}{2} \sum_{i=1}^l \xi_i^2 - \sum_{i=1}^l \alpha_i (y_i (\mathbf{x}_i^T \mathbf{w} + b) - 1 + \xi_i), \quad (6)$$

where $\{\alpha_i\}_{i=1}^l$ are the Lagrange multipliers. The conditions for optimality, similarly to the SVM problem, can be given by the partial derivatives

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{0} \Rightarrow \mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i, \quad \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial b} = 0 \Rightarrow \sum_{i=1}^l \alpha_i y_i = 0, \quad (7)$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial \alpha_i} = 0 \Rightarrow y_i (\mathbf{x}_i^T \mathbf{w} + b) - 1 + \xi_i = 0, \quad \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial \xi_i} = 0 \Rightarrow \alpha_i = \gamma \xi_i$$

Thus, one can formulate a linear system $\mathbf{A}\mathbf{x} = \mathbf{B}$ in order to represent this problem as

$$\left[\begin{array}{c|c} 0 & \mathbf{y}^T \\ \hline \mathbf{y} & \boldsymbol{\Omega} + \gamma^{-1} \mathbf{I} \end{array} \right] \left[\begin{array}{c} b \\ \boldsymbol{\alpha} \end{array} \right] = \left[\begin{array}{c} 0 \\ \mathbf{1} \end{array} \right] \quad (8)$$

where $\boldsymbol{\Omega} \in \mathbb{R}^{l \times l}$ is a matrix whose entries are given by $\Omega_{i,j} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j$, $i, j = 1, \dots, l$. In addition, $\mathbf{y} = [y_1 \ \dots \ y_l]^T$ and the symbol $\mathbf{1}$ denotes a vector of ones with dimension l . Using the kernel trick, we can write $\Omega_{i,j} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$. This linear equation system can be solved directly by $\mathbf{x} = \mathbf{A}^{-1} \mathbf{B}$, as long as the LS-SVM output can also be calculated by Eq. (4).

The Self-Organizing Map - Let us denote $\mathbf{m}_j(t) \in \mathbb{R}^p$ as the weight vector of the j -th unit in the map. After initializing all the weight vectors randomly or according to some heuristic, each iteration of the SOM algorithm involves two steps. First, for a given input vector $\mathbf{x}(t) \in \mathbb{R}^p$, we find the best-matching unit (BMU), $c(t)$, as follows

$$c(t) = \arg \min_{\forall j} \{ \|\mathbf{x}(t) - \mathbf{m}_j(t)\| \}. \quad (9)$$

where t denotes the iterations of the algorithm. Then, it is necessary to adjust the weight vectors of the BMU and its neighboring units:

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + \eta(t)h(c, j; t)[\mathbf{x}(t) - \mathbf{m}_j(t)], \quad (10)$$

where $0 < \eta(t) < 1$ is the learning rate and $h(c, j; t)$ is a Gaussian weighting function that limits the neighborhood of the BMU:

$$h(c, j; t) = \exp\left(-\frac{\|\mathbf{r}_c(t) - \mathbf{r}_j(t)\|^2}{2\sigma^2(t)}\right), \quad (11)$$

where $\mathbf{r}_j(t)$ and $\mathbf{r}_c(t)$ are, respectively, the positions of the j -th unit and the BMU in a predefined output array, and $\sigma(t) > 0$ defines the radius of the neighborhood function at time t . The parameters $\eta(t)$ and $\sigma(t)$ decay exponentially in time according to the following expressions: $\eta(t) = \eta_0 (\eta_T/\eta_0)^{(t/T)}$ and $\sigma(t) = \sigma_0 (\sigma_T/\sigma_0)^{(t/T)}$, where $\eta_0(\sigma_0)$ and $\eta_T(\sigma_T)$ are the initial and final values of $\eta(t)$ ($\sigma(t)$). The resulting ordered map approximately preserves the topology of the input data in the sense that adjacent input patterns are mapped into adjacent map units.

3 Opposite Maps

The Opposite Maps (OM) algorithm aims at building reduced-set SVMs for a binary classification problem. The steps required in the execution of the OM algorithm are formalized below.

STEP 1 - Split the available data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^l$ into two subsets:

$$\mathcal{D}^{(1)} = \{(\mathbf{x}_i, y_i) | y_i = +1\}, \quad i = 1, \dots, l_1 \quad (\text{for class 1}) \quad (12)$$

$$\mathcal{D}^{(2)} = \{(\mathbf{x}_i, y_i) | y_i = -1\}, \quad i = 1, \dots, l_2 \quad (\text{for class 2}) \quad (13)$$

where l_1 and l_2 are the cardinalities of the subsets $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$, respectively.

STEP 2 - Train a SOM network using the subset $\mathcal{D}^{(1)}$ and another SOM using the subset $\mathcal{D}^{(2)}$. Refer to the trained networks as SOM-1 and SOM-2.

STEP 3 - For each vector $\mathbf{x}_i \in \mathcal{D}^{(1)}$ find its corresponding BMU in SOM-1. Then, prune all *dead neurons*² in SOM-1. Repeat the same procedure for each vector $\mathbf{x}_i \in \mathcal{D}^{(2)}$: find the corresponding BMUs in SOM-2 and prune all the dead neurons. Refer to the pruned networks as PSOM-1 and PSOM-2.

STEP 4 - At this step the BMUs for the data subsets are searched within the set of prototypes of the *opposite map*.

Step 4.1 - For each $\mathbf{x}_i \in \mathcal{D}^{(1)}$ find its corresponding BMU in PSOM-2:

$$c_i^{(2)} = \arg \min_{\forall j} \|\mathbf{x}_i - \mathbf{w}_j^{(2)}\|, \quad i = 1, \dots, l_1, \quad (14)$$

where $\mathbf{w}_j^{(2)}$ is the j -th prototype vector in PSOM-2. Thus, $c_i^{(2)}$ denotes the index of the BMU in PSOM-2 for the i -th example in $\mathcal{D}^{(1)}$.

² Neurons which have never been selected as the BMU for any vector $\mathbf{x}_i \in \mathcal{D}^{(1)}$.

Step 4.2 - For each $\mathbf{x}_i \in \mathcal{D}^{(2)}$ find its corresponding BMU in PSOM-1:

$$c_i^{(1)} = \arg \min_{\forall j} \|\mathbf{x}_i - \mathbf{w}_j^{(1)}\|, \quad i = 1, \dots, l_2, \quad (15)$$

where $\mathbf{w}_j^{(1)}$ is the j -th prototype vector in PSOM-1. Thus, $c_i^{(1)}$ denotes the index of the BMU in PSOM-1 for the i -th example in $\mathcal{D}^{(2)}$.

STEP 5 - Let $\mathcal{C}^{(2)} = \{c_1^{(2)}, c_2^{(2)}, \dots, c_{l_2}^{(2)}\}$ be the index set of all BMUs found in Step 4.1, and $\mathcal{C}^{(1)} = \{c_1^{(1)}, c_2^{(1)}, \dots, c_{l_1}^{(1)}\}$ be the index set of all BMUs found in Step 4.2.

STEP 6 - At this step the reduced set of data vectors is formed.

Step 6.1 - For each PSOM-1 unit in $\mathcal{C}^{(1)}$ find its nearest neighbor among the data vectors $\mathbf{x}_i \in \mathcal{D}^{(1)}$. Let $\mathcal{X}^{(1)}$ be the subset of nearest neighbors for the PSOM-1 units in $\mathcal{C}^{(1)}$.

Step 6.2 - For each PSOM-2 unit in $\mathcal{C}^{(2)}$ find its nearest vector $\mathbf{x}_i \in \mathcal{D}^{(2)}$. Let $\mathcal{X}^{(2)}$ be the subset of nearest neighbors for the PSOM-2 units in $\mathcal{C}^{(2)}$. Then, the reduced set of data examples is given by $\mathcal{X}_{rs} = \mathcal{X}^{(1)} \cup \mathcal{X}^{(2)}$.

Opposite Maps for SVM and LS-SVM - In this paper, we use Platt’s Sequential Minimal Optimization (SMO) algorithm [7], which is usually much faster than standard numerical techniques used to solve the quadratic programming optimization problem required by training SVMs. By using the OM algorithm, it is possible to speed up the SMO algorithm even further. The main idea is to deliver to the SMO algorithm an “almost solved problem”, since for all data examples out of the reduced set (i.e. $\mathbf{x}_i \notin \mathcal{X}_{rs}$) their Lagrange multipliers are set to zero, the SMO algorithm is run only over the data examples belonging to the reduced set. This approach is henceforth called OM-SVM.

For the LS-SVM approach, we cannot set the Lagrange multipliers of the data examples out of the reduced set to zero, since the LS-SVM usually provide non-sparse solutions. However, the Lagrange multipliers associated with data examples located along the class border or within the overlapping region between the classes indeed assume higher values, i.e. $\alpha_i \gg 0$. These instances are the most likely to be the SVs.

Based on this property, we use the OM algorithm to build a modified version of the LS-SVM formulation shown in Eq. (8). Instead of building the original square matrix \mathbf{A} and inverting it to find \mathbf{x} , we build a *non-square* reduced matrix \mathbf{A}_{rs} using the data examples belonging to the reduced set \mathcal{X}_{rs} . Thus, since \mathbf{A}_{rs} is a non-square matrix, we solve for \mathbf{x} using the pseudoinverse method: $\mathbf{x} = \mathbf{A}^\dagger \mathbf{B}$, where $\mathbf{A}^\dagger = (\mathbf{A}_{rs}^T \mathbf{A}_{rs})^{-1} \mathbf{A}_{rs}^T$. This approach is henceforth called OM-LSSVM.

4 Simulations and Discussion

For all experiments to be described, 80% of the data examples were randomly selected for training purposes. The remaining 20% of the examples were used

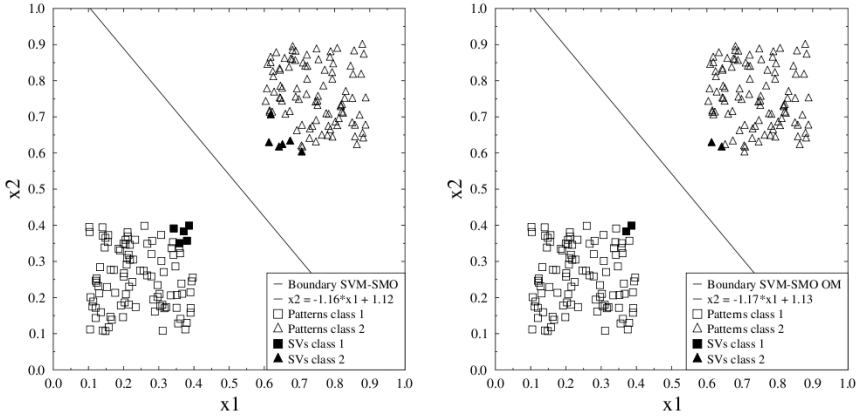


Fig. 1. (Left) Decision line and SVs for the standard SVM trained with the SMO algorithm. (Right) Decision line and SVs for the OM-SVM classifier.

for testing the classifiers’ generalization performances. All simulations were conducted using a standard 2-D SOM, hexagonal neighborhood, Gaussian neighborhood function, with random weight initialization. We simulated 10×10 SOMs, for 250 epochs with initial and final neighborhood radius (learning rate) of 5 (0.5) and 0.1 (0.01), respectively. For SVM-like classifiers we used linear kernels.

Initially, we have applied the OM algorithm to an artificial problem, consisting of a linearly separable two-dimensional data set. Data instances within each class are independent and uniformly distributed with the same within- and between-class variances. Results in Figure 1 indicate that the OM-SVM, using fewer SVs, produced a decision line equivalent to that of the standard SVM.

Tests with real-world benchmarking datasets were also carried out. We used two UCI datasets (Diabetes and Breast Cancer) and the vertebral column pathologies dataset described in [8], named henceforth VCP dataset. For this study, we transformed the original three-class VCP problem into a binary one by aggregating the two classes of pathologies, the disc hernia and spondylolisthesis, into a single one. The normal class remained unchanged.

The results for the SVM and OM-SVM classifiers are shown in Table 1. For the LS-SVM and OM-LSSVM classifiers they are shown in Table 2. We report performance metrics (mean value and standard deviation of the recognition rate) on testing set averaged over 50 independent runs. We also show the average number of SVs, as well as the values of the parameters C (SVM), γ (LS-SVM) and the tolerance. The values of the C and γ parameters shown in the tables were obtained through a greedy-like search procedure [9].

By analyzing Tables 1 and 2, one can conclude that the performances of the reduced-set classifiers (OM-SMV and OM-LSSVM) were equivalent to those achieved by the full-set classifiers. In some cases, as shown in Table 2 for the VCP and Pima Diabetes, the performances of the reduced-set classifiers were even better. It is worth mentioning that, as expected, the standard deviation

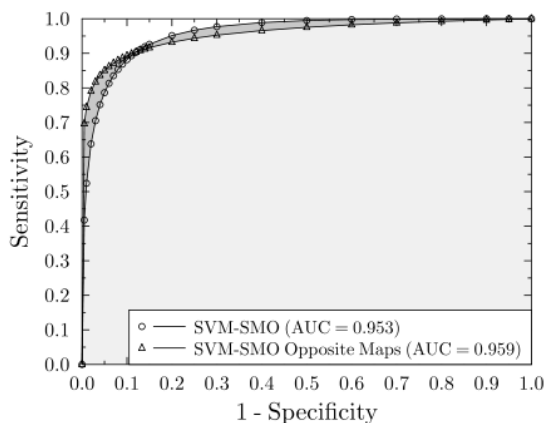
Table 1. Results for the SVM and OM-SVM classifiers

Data Set	Model	C	Tol.	Accuracy	Train. Size	Number of SVs
VCP	SVM	2.5	0.001	84.9 ± 3.9	248	87.2
VCP	OM-SVM	2.5	0.001	84.9 ± 4.8	248	70.1
Reduction =						19.6%
Breast Cancer	SVM	0.04	0.001	97.0 ± 1.4	546	61.5
Breast Cancer	OM-SVM	0.04	0.001	96.0 ± 1.9	546	46.3
Reduction =						24.7%
Pima Diabets	SVM	2.5	0.01	76.9 ± 2.7	614	321.9
Pima Diabets	OM-SVM	2.5	0.01	76.8 ± 3.3	614	292.6
Reduction =						9.1%

Table 2. Results for the LS-SVM and OM-LSSVM classifiers

Data Set	Model	γ	Accuracy	Train. Size	Number of SVs
VCP	LS-SVM	0.04	80.9 ± 4.8	248	248.0
VCP	OM-LSSVM	0.04	81.5 ± 5.7	248	113.1
Reduction =					54.4%
Breast Cancer	LS-SVM	0.04	97.1 ± 1.4	546	546.0
Breast Cancer	OM-LSSVM	0.04	94.4 ± 4.5	546	61.0
Reduction =					88.8%
Pima Diabets	LS-SVM	0.04	75.8 ± 2.7	614	614.0
Pima Diabets	OM-LSSVM	0.04	76.8 ± 3.5	614	428.5
Reduction =					30.2%

of the results for the reduced-set classifiers were always slightly higher than the full-set classifiers. The ROC curves of the SVM and OM-SVM classifiers are shown in Figure 2. By analyzing the AUC values, one can note again that their performances are equivalent for this dataset.

**Fig. 2.** ROC curves for SVM and OM-SVM classifiers for the VCP dataset

5 Conclusion

We have proposed a new algorithm, called Opposite Maps, for building reduced-set SVM-like classifiers. This algorithm is based on Kohonen's SOM and it can be applied to both standard SVM and LS-SVM classifiers. In both cases, the obtained results indicate that the OM algorithm works well, providing a reduced number of SVs while maintaining equivalent accuracy. Currently, we are evaluating the OM algorithm in multiclass problems and using different kernels.

References

1. D'Amato, L., Moreno, J.A., Mujica, R.: Reducing the complexity of kernel machines with neural growing gas in feature space. In: Lemaître, C., Reyes, C.A., González, J.A. (eds.) *IBERAMIA 2004*. LNCS (LNAI), vol. 3315, pp. 799–808. Springer, Heidelberg (2004)
2. Downs, T., Gates, K.E., Masters, A.: Exact simplification of support vector solutions. *Journal of Machine Learning Research* 2, 293–297 (2002)
3. Hoegaerts, L., Suykens, J.A.K., Vandewalle, J., De Moor, B.: A comparison of pruning algorithms for sparse least squares support vector machines. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) *ICONIP 2004*. LNCS, vol. 3316, pp. 22–25. Springer, Heidelberg (2004)
4. Hussain, A., Shahbudin, S., Husain, H., Samad, S.A., Tahir, N.M.: Reduced set support vector machines: Application for 2-dimensional datasets. In: *Proceedings of the Second International Conference on Signal Processing and Communication Systems (ICSPCS2008)* (2008)
5. Kohonen, T.K.: *Self-Organizing Maps*. Springer, Heidelberg (1997)
6. Li, Y., Lin, C., Zhang, W.: Letters: Improved sparse least-squares support vector machine classifiers. *Neurocomputing* 69, 1655–1658 (2006)
7. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge (1999)
8. Rocha Neto, A.R., Barreto, G.A.: On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis. *IEEE Latin America Transactions* 7(4), 487–496 (2009)
9. Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 3rd edn. Prentice-Hall, Englewood Cliffs (2009)
10. Steinwart, I.: Sparseness of support vector machines. *Journal of Machine Learning Research* 4, 1071–1105 (2003)
11. Suykens, J.A.K., Lukas, L., Vandewalle, J.: Sparse approximation using least squares support vector machines. In: *Proceedings of 2000 IEEE International Symposium on Circuits and Systems*, Geneva, Switzerland, pp. 757–760 (2000)
12. Suykens, J.A.K., Vandewalle, J.: Least squares support vector machine classifiers. *Neural Processing Letters* 9(3), 293–300 (1999)
13. Tang, B., Mazzoni, D.: Multiclass reduced-set support vector machines. In: *Proceedings of the 23rd international conference on Machine learning (ICML 2006)*, New York, NY, USA, pp. 921–928 (2006)

An Additive Decision Rules Classifier for Network Intrusion Detection

Tommaso Pani and Francisco de Toro*

CITIC-Dpt. of Signal Theory, Telematics and Communications
University of Granada, Spain
ftoro@ugr.es

Abstract. This paper presents an additive decision rules binary classifier applied for network intrusion detection. The classifier is optimized by a multiobjective evolutionary algorithm in order to maximize both the classification accuracy and the coverage level (percentage of items that are classified, in opposite to items unable to be classified). Preliminary results provides very good accuracy in detecting attacks which make this relatively simple classifier very suitable to be applied in the studied domain.

1 Introduction

With the increased complexity of security threats, such as malicious Internet worms, denial of service (DoS) attacks, and e-business application attacks, achieving efficient network intrusion security is critical to maintaining a high level of protection. The efficient design of intrusion detection systems (IDS) is essential for safeguarding organizations from costly and debilitating network breaches and for helping to ensure business continuity. An IDS is a program that analyzes what happens or has happened in a computer network and try to find indications that the computer has been misused. An IDS will typically monitor network traffic data passing through the network in order to generate an alert when an attack event is taking place. On the other hand, two different kinds of detection schemes can be applied to detect attacks in the data being monitored. Signature-based detection systems try to find attack signatures in the data monitored. Anomaly detection systems rely on the knowledge of what should be the normal behaviour of the monitored data to flag any deviation of the normal behaviour as an attack event, so they need a model of what a normal behaviour is. Machine learning algorithms [1] can be trained with labelled network traffic data so it can classify unknown traffic data captured from a computer network with certain accuracy. One type of these algorithms would consist on a binary classifier enabling to separate two different groups of observations: normal traffic and malicious traffic (containing some kind of attack).

One way to proceed of a signature-based intrusion detection system is to process network traffic data into data flows and use some features from the data flows to signature the current network traffic data so an adequate response may be triggered when an attack is occurring (see Fig. 1).

* Corresponding author.

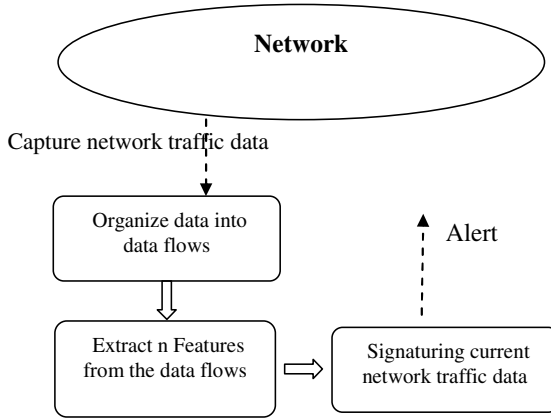


Fig. 1. A signature-based Intrusion Detection System

Evolutionary algorithms (EAs) [2,3] are stochastic optimization procedures which apply a transformation process (crossover and mutation operators), inspired by the species natural evolution, to a set (population) of coded solutions (individuals) to the problem. These procedures have been showing a great success [4,5] due to its special ability to explore large search spaces and capture multiple solutions in a single run.

In this work, an additive decision rules binary classifier is presented and applied to separate malicious network traffic from normal traffic. Such a classifier could be used as a signature-based intrusion detection system. The classifier is optimized in order to maximize the classification accuracy and the coverage level (percentage of items that are classified, in opposite to items unable to be classified).

This work is organized as follows: Section 2, the overall methodology presented in this paper is described. Thus, Section 3 shows some experimental work carried out by using labelled network traffic data provided by DARPA [6]. Finally, Section 4 is devoted to discuss main results and conclusions.

2 Materials and Methods

The general framework methodology proposed in this work is depicted in Fig. 2. First of all, network data (data packets) are organized into data flows. Then, n features - previously defined- characterizing the data flows are extracted to obtain an n -dimensional feature vector representing the network traffic in a given time window. A binary classifier is trained (optimized) to classify each feature vector as belonging to normal traffic or malicious traffic. The training of the classifier is formulated as a multiobjective optimization problem addressed by an evolutionary algorithm. The optimization problem consists on maximizing the classification accuracy and the coverage level achieved by the classifier.

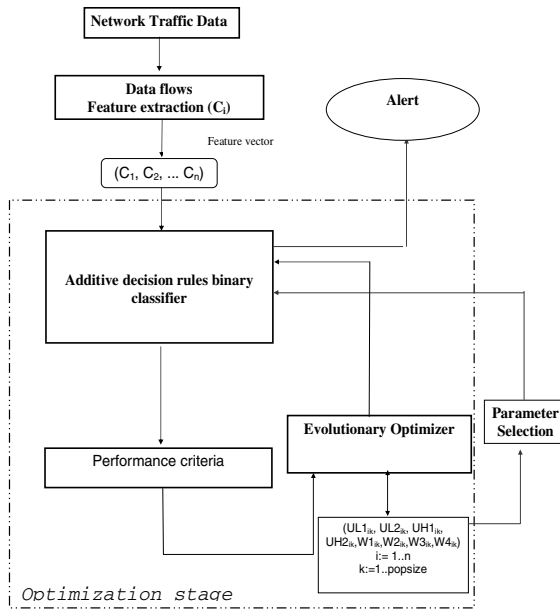


Fig. 2. Proposed methodology for detection of malicious network traffic

2.1 The Classifier

We propose a binary classifier which makes the classification decision according to a variable (DIAGNOSIS) that is modified with four types of different **if-then** rules. In each rule the value of a given feature C_i is compared with a threshold (e.g. $UL1_i$) and as a result DIAGNOSIS is increased or decreased a certain quantity (e.g. $W1_i$). The pseudocode of the classifier is as follows:

```

foreach item j
  foreach feature i from item j
    if  $C_i < UL1_i$  then DIAGNOSIS=DIAGNOSIS +  $W1_i$ 
    if  $C_i < UL2_i$  then DIAGNOSIS=DIAGNOSIS -  $W2_i$ 
    if  $C_i > UH1_i$  then DIAGNOSIS=DIAGNOSIS +  $W3_i$ 
    if  $C_i > UH2_i$  then DIAGNOSIS=DIAGNOSIS -  $W4_i$ 
  end
  if DIAGNOSIS > F1 then classify ítem j as A CLASS
  else if DIAGNOSIS < F2 then classify ítem j as B CLASS
  else let item j as UNCLASSIFIED
end

```

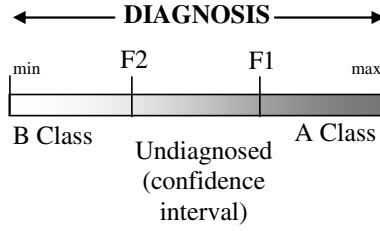


Fig. 3. Variable DIAGNOSIS used for the classification decision

An important matter in the proposed classifier is to choose the values F1 and F2. We propose to set those values according to the maximum and minimum values that DIAGNOSIS may take, in the following way:

$$F1 = \max(\text{DIAGNOSIS}) - \text{Cella}$$

$$F2 = \text{Cella} + \min(\text{DIAGNOSIS})$$

where,

$$\text{Cella} = \text{Dimmax} / 3, \text{ and}$$

$$\text{Dimmax} = \max(\text{DIAGNOSIS}) - \min(\text{DIAGNOSIS})$$

This way, three equally sized areas are taken into account for the classification decision (see Fig. 3).

2.2 The Evolutionary Optimizer

The chosen criteria to optimize the classifier performance are to maximize both the coverage level (CL) and the classification accuracy (CA) defined as:

$$CL = \frac{\text{Number of Classified Cases}}{\text{Total Number of Cases}} \tag{1}$$

$$CA = \frac{\text{Number of Correctly Classified Cases}}{\text{Total Number of Cases}} \tag{2}$$

Due to the fact that the optimization problem is multiobjective by nature, we use NSGA-2 [7], one of the state-of-the-art MOEAs (Multiobjective Optimization Evolutionary Algorithms) and widely used in the field. A possible solution (chromosome) is a real data vector containing 8 parameters, $UL_{1i}, UL_{2i}, UH_{1i}, UH_{2i}, W_{1i}, W_{2i}, W_{3i}, W_{4i}$, for each i -th feature considered for the classification.

3 Case of Study

For the purpose of testing the aforementioned classifier methodology, a network database for training purpose provided by DARPA [6] has been used. This database is

built with simulated network traffic data containing normal traffic data and 22 different kinds of computer attacks that fall in one of the following groups:

- DoS (Denial of Service): the attacker targets some computing or memory resource and makes it too busy or full to handle legitimate requests, or denies legitimate user access to that resource, for example SYN flood, ping of death, smurf, etc.
- R2U (Remote to User): the attacker exploits some vulnerability to gain unauthorized local access from a remote machine, for example guessing password.
- U2R (User to Root): the attacker has access to a normal user account (obtained legitimately or otherwise) and using this is able to gain root access by exploiting a vulnerability hole in the system, for example buffer overflow attacks.
- PROBE (Probing): attacker scans the network to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use this information to look for weak points, for example through port scan.

There are 41 features present in the data set. These 41 features are not all equally useful. Based on previous studies [8] and personal experience with this database we have selected the features shown in Table 1 to be used in our work.

Table 1. Features chosen to detect computer attack from normal network traffic data

Name of Feature	Description
count	# of connection made to the same host in a given interval of time.
Root shell	1 if root shell is obtained; 0 otherwise
dst_host_srv_serror_rate	% of connections to the same service that have SYN errors from a destination host
# of file creations	# of file creation operations
serror_rate	% of connections that have SYN errors
dst_host_same_src_port_rate	% of connections to same service ports from a destination host
guest login	1 if the login is a 'guest' login; 0 otherwise
# of file access	# of operations on access control files
destination bytes	# of bytes sent from the destination system to the host system
failed logins	# of failed login attempts
logged in	1 if successfully logged in; 0 otherwise

The classification result can fall into one of the following cases: (1) The algorithm classifies the traffic as malicious and the traffic is in fact malicious (True Positive, TP); (2) The algorithm classifies the traffic as normal and the traffic is in fact normal (True Negative, TN); (3) The algorithm classifies the traffic as malicious but the traffic is normal (False Positive, FP); (4) The algorithm classifies the traffic as normal but the traffic is malicious (False Negative, FN).

Within these cases, the confusion matrix can be build as:

	Normal Traffic	Attack Traffic
Classified as Normal	TN	FN
Classified as Attack	FP	TP

And five different measures can be considered:

- Classification Accuracy (CA): represents the ratio between that correctly classified traffic and the overall traffic.

$$CA = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- Producer accuracy for Attack Class (PAA):

$$PAA = \frac{TP}{FN + TP} \quad (2)$$

- Producer accuracy for Normal Class (PAN):

$$PAN = \frac{TN}{TN + FP} \quad (3)$$

- User accuracy for Attack Class (UAA):

$$UAA = \frac{TP}{FP + TP} \quad (4)$$

- User accuracy for Normal Class (UAN):

$$UAN = \frac{TN}{TN + FN} \quad (5)$$

The assessment of the performance of the presented methodology has two part: First of all an optimization stage where the parameters of the classifier and the F1 and F2 values are set. In this part, we have used two equally sized sets: one containing normal traffic and the other one containing the four different types of attacks (DoS, Probe, U2R and R2L respectively). In this part, the nature of the data (records) is known (each record is labeled as normal or attack); Secondly we use different sized test sets containing normal traffic and attack traffic to evaluate the classifier. This time the records are initially considered as unknown and the classifier must label them. Following, computed labels are compared to real labels and statistics about classification performance are obtained (see Table 3).

NSGA-2 ran during 100 iterations with 50 candidate solutions and with coverage level and classification accuracy as fitness criteria. After the optimization process the parameters in Table 6 were extracted.

Table 2. Parameters of the classifier obtained during optimization stage

Features	UH2	UL2	UH1	UL1	W4	W2	W3	W1
Root shell	0	0	1	1	0,03	0,03	0,86	0,96
# of file creations	9,65	39,00	73,33	92,11	0,20	0,80	0,31	0,01
Error_rate	0,11	0,55	0,58	0,84	0,68	0,20	0,81	0,82
Dst_host_same_src_port_rate	0,00	0,17	0,72	0,89	0,38	0,12	0,12	0,57
Dst_host_srv_error_rate	0,49	0,77	0,93	0,95	0,09	0,68	0,00	0,30
Guest login	0	0	0	1	0,76	0,40	0,96	0,02
Logged in	0	0	1	1	0,30	0,94	0,01	0,45
Destination bytes	770,69	1527,58	2366,85	8421,70	0,05	0,68	0,49	0,28
Failed logins	0,15	1,57	3,03	4,48	0,94	0,28	0,23	0,43
# of file access	0,47	2,32	3,50	6,00	0,78	0,40	0,34	0,97
Count	2689,10	4558,91	7638,02	9311,99	0,84	0,99	0,00	0,89

Table 3 shows the performance of our classifier with different number of records, while Figure 4 shows diagnosed value (DIAGNOSIS) for every record in a typical classification task when 90000 records are used.

Table 3. Performance of the classifier on different test sets with different number of records

#records	CL(%)	CA(%)	PAA(%)	PAN(%)	UAA(%)	UAN(%)
60000	99.60	96.89	99.64	73.13	96.97	95.87
90000	99.38	96.59	99.41	77.09	99.86	94.96
150000	99.03	87.78	91.46	75.16	92.67	71.95
200000	99.03	87.78	91.46	75.16	92.76	71.95

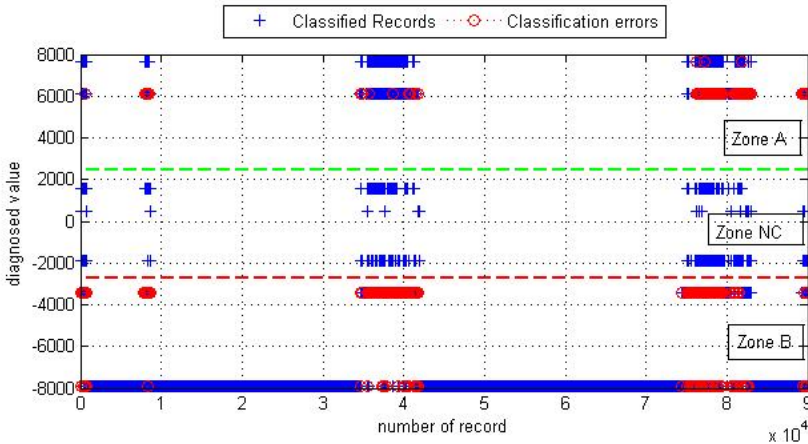


Fig. 4. Typical classification of a test set with 90000 records

First of all, it is important to mention that both classification accuracy and coverage level results were excellent considering that we are taking into account very large test data sets. Furthermore, the ability to detect attack (PAA and UAA

indicators) are always 90% in all test data sets which is very desirable in this particular type of application. Lastly, it is worthy to mention that most false positive and false negative were obtained within U2R and R2U classes.

4 Summary

This work proposes a decision rules classifier to be used for network intrusion detection. This classifier uses if-then rules containing thresholds and weights which are automatically adjusted with a multiobjective optimization evolutionary algorithm following two fitness criteria: classification accuracy and coverage level. The methodology has been preliminary applied to detect any of the attacks contained in DARPA database. Results show a great accuracy in detecting attacks (PAA and UAA indicators). As future work, the authors are intended to explore the performance of this methodology on other network traffic databases.

Acknowledgments. This work has been partially supported by Spanish MICINN under project TEC2008-06663-C03-02.

References

1. Sabhnani, M., Serpen, G.: Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context. In: Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications, pp. 209–215 (2003)
2. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Natural Computing Series. Springer, Heidelberg (2003)
3. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, New York (1989)
4. Li, J.P., Balazs, M.E., Parks, G.T., Clarkson, P.J.: A species conserving genetic algorithm for multimodal function optimisation. *Evolutionary Computation* 10(3), 207–234 (2002)
5. de Toro, F., Ortega, J., Ros, E., Mota, S., Paechter, B.: Parallel Processing and Evolutionary Algorithms for Multiobjective Optimisation. *Parallel Computing* 30(6), 721–739 (2004)
6. The, U.K.: Archive, Information and Computer Science, University of California, Irvine, “Kdd cup 1999, data set” (1999), <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
7. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multiobjective optimisation: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
8. Mukkamala, S., Sung, A.H., Abraham, A.: Modeling Intrusion Detection Systems Using Linear Genetic Programming Approach. *Journal Of Network And Computer Applications* 30(1), 1–3 (2007)

Multi-modal Opponent Behaviour Prognosis in E-Negotiations

Ioannis Papaioannou¹, Ioanna Roussaki^{2,*}, and Miltiades Anagnostou²

¹ Research and Academic Computer Technology Institute,
N. Kazantzaki Str, Rio, Patras, Greece
jpapai@cti.gr

² School of Electrical and Computer Engineering, National Technical University of Athens,
9 Heron Polytechniou Str, 157-73 Zographou, Athens, Greece
{ioanna.roussaki,miltos}@cn.ntua.gr

Abstract. Automated negotiations introduce a challenging research field that aims to enhance the performance and optimise several aspects of the electronic marketplace. This paper is concerned with the design and evaluation of negotiation strategies suitable for intelligent agents acting in Business-to-Consumer e-commerce environments. In order to minimize the cases where an agreement is not reached upon the expiration of its deadline, the client agent is enhanced with various function approximation techniques, which aim to predict the behaviour of the provider agent during the last negotiation rounds.

Keywords: Negotiating Agents, opponent behaviour prediction, MLP & RBF Neural Networks, Polynomial Approximators.

1 Introduction

The explosion of e-business has changed the way users interact when purchasing procedures and transactions take place [1]. Along with the revolution of e-trading systems, other technologies play an increasingly important role on the integration of the frameworks that are created. Towards this fact, intelligent software agents [2][3] are capable of performing sophisticated and challenging tasks to lead to environments with complexity that resembles the human-to-human procedures with regards to rationale and satisfaction from the overall negotiations' outcome [4].

The research area that has evicted out of the application of intelligent agents' technologies to e-business activities and specifically to e-negotiations is what is known as automated negotiations [5]. When building autonomous agents capable of negotiation, three main aspects need to be considered [5][6]: (i) the selection of issues under negotiation, (ii) the negotiation protocol and model, and (iii) the negotiation strategies that the agents will employ. This paper is mainly concerned with the third, and proposes an efficient approach for the second.

In such procedures, the intelligent agents aim to accomplish the objectives of their owners and address their initially designated or real-time requirements as efficiently

* Corresponding author.

as possible. As defined in [5], “Negotiation is a form of interaction in which a group of agents, with conflicting interests and a desire to cooperate try to come to a mutually acceptable agreement on the division of scarce resources”.

This paper is concerned with the evaluation of prognostic negotiation strategies adequate for intelligent agents acting in Business-to-Consumer e-commerce environments, addressing use cases where a Client negotiates with a Provider of a service or product. The proposed strategies are applicable in single-issue bilateral negotiations, where both agents have strict deadlines and are practically the mature extension and refinement of authors’ attempt to conclude to the best possible learning scheme for such kinds of problems [7]. Thus, the prediction mechanisms are implemented using cubic splines and polynomial approximators exploited by the client agents, complementing the authors’ work on MLP (Multi-layer Perceptron) and RBF (Radial Basis Function) neural network-enhanced prediction mechanisms [7], aiming to conclude on the best possible techniques for on-line negotiation procedures.

The rest of this paper is structured as follows. In Section 2, the formal problem statement is briefly presented avoiding repetition of authors’ previous publication [7], while the problem is extended to address multi-issue package-deal, parallel and sequential procedures [8]. In Section 3, a brief introduction of the employed techniques to predict the behaviour of the negotiating Provider Agent is provided, while their design properties are exposed. In Section 4, the proposed approximator-assisted negotiation strategies of the Client Agent are evaluated via extended experiments, and finally in the last Section the paper’s conclusions are drawn.

2 Formal Problem Statement

2.1 The Basic Problem Statement

In this section the negotiation framework designed and employed is briefly presented. This formal problem definition and justification has also been stated in detail in [7], so only the basics are repeated here in order to ensure the coherence of the paper.

The negotiation protocol employed [9], i.e. the rules that both agents should follow throughout the procedure has been described in full detail in the aforementioned authors’ work. For clarification the protocol employed is a modification of the single-issue alternating offers protocol [10], where each negotiation thread is initiated by the Client Agent (CA) in the Client-Provider model that is employed and formally initiated (with a real proposal) by the Provider Agent (PA). The procedure ends, when either a mutually accepted agreement is reached or one of the agents’ deadlines expires. At each step of the procedure, each agent should perform a proposal which is either an offer based on its strategy or a decision of agreement or disagreement.

At this point the formal description of the problem is appended: Suppose the agent that represents the Client is denoted by C and the one that represents the Provider is denoted by P . The objective of our problem for a Client Agent is to find a price p_{final} that lies above the current minimum acceptable price of the Provider. Thus, based on the selected protocol and model, the negotiation problem studied can be reduced to a decision problem that can formally be stated as follows [7]:

Given: (i) two negotiating parties: a Provider that offers a specific good (i.e., service or product) and a Client that is interested in this good's acquisition, (ii) the acceptable price interval $[p_{\min}^C, p_{\max}^C]$ for the Client, (iii) a deadline T_C up to which the Client must have completed the negotiation with the Provider, (iv) the final negotiation round index L_C for the Client, (v) the vector $P_l^P = \{p_l^P\}$, where $l = 2k$ and $k = 0, \dots, \left\lfloor \frac{L_C}{2} \right\rfloor - 1$, of the prices that were proposed by the Provider during the initial $L_C - 2$ negotiation rounds, and (vi) the price offer $p_{L_C-3}^C$ that was proposed by the Client during negotiation round $L_C - 3$, find a price p_{final} that should be proposed by the CA to the PA at the pre-final negotiation round $L_C - 1$, which maximizes the Client's overall utility function $U^C(p_{final})$, satisfies the known CA's price constraint (i.e. $p_{final} \in [p_{\min}^C, p_{\max}^C]$), and will get accepted by the PA at the last negotiation round L_C .

As already described, the client negotiates based on the fair Relative Tit-for-Tat (fRTFT) strategy until round $L_C - 1$ [7]. Then, not risking the negotiation to fail, the client makes use of the approximators' estimation $\overline{p_{L_C}^P}$ for the provider's next price offer. These tools, which will be further elaborated below are a key component capable of predicting the future offer of the opponent, leading the negotiation to an agreement and thus avoiding a highly possible waste of resources as the thread has reached its pre-final round. The approximator, based on the history of provider's price offers $P_l^P = \{p_l^P\}$ for $l = 2k$, $k = 0, \dots, \left\lfloor \frac{L_C}{2} \right\rfloor - 1$, attempts to produce a good $\overline{p_{L_C}^P}$ estimation, which at the same time ensures that the profit of the party that employs the mechanism is locally maximized (utility function). If $\overline{p_{L_C}^P} \leq p_{\max}^C$, then the client's last price offer sent to the provider is: $p_{L_C-1}^C = \overline{p_{L_C}^P}$. Otherwise, if $\overline{p_{L_C}^P} > p_{\max}^C$, then $p_{L_C-1}^C = p_{\max}^C$. If the approximator succeeds and $\overline{p_{L_C}^P} \geq p_{L_C}^P$, the provider will accept the offer upon the expiration of the client's negotiation deadline, thus leading to the desirable outcome. If the estimation is lower than the client's prior offer then $p_{L_C-1}^C = p_{L_C-3}^C$.

In this study, the PA adopts a strategy based on a time-dependent tactic, while the CA follows a negotiation strategy based on a behaviour-dependent tactic. This deterministic lightweight approach is selected as the studied framework is designed for PAs that are simultaneously engaged in multiple independent negotiation threads with different CAs, thus requiring reduced processing resources with regards to stochastic or behaviour-dependent alternatives.

The rest of the details for the purposes of this problem are mentioned in Sections 2 & 3 of [7] and are not repeated here to avoid redundancy. Besides, this paper focuses on comparing conventional approximation techniques (polynomials), which are increasingly employed in various research areas where time and resources are critical as it happens here, against neural networks which were proven notably useful in [7].

2.2 Extension to Support Multi-issue Procedures

Multi-issue negotiations refer to the case where agents negotiate over more than one issue. The main generic approaches that can be distinguished in support of multi-issue negotiations are the following [8]: *package-deal approach* where all the issues are settled together as a bundle, *sequential approach* where the issues are discussed one after another, and *simultaneous approach* where the issues are discussed in parallel.

In case of the package-deal approach, the application of the mechanism to agree on a specific price for the whole bundle is straightforward. On the other hand, the simultaneous approach is mainly used in cases where the issues under negotiation are disjoint [11]. It is obvious that the proposed lightweight mechanism can be applied on each parallel thread (for each issue) to provide the party that employs it with the maximal profit. Thus, in both the above cases the mechanism is adequate and can be applied without any modification.

However, the sequential approach requires a different perspective, as the issues are considered interdependent and each agreement affects the negotiation thread that follows [11]. In this case, we cannot provide optimal results, just by applying the mechanism as we should take into account these dependencies, which are absent from the above two approaches. Thus, in order to extend the framework to address multi-issue negotiations for interdependent issues, we adopt the following solution.

Let C denote the agent that represents the Client and P denote the agent representing the Provider. These two parties negotiate over issues x_i , $i = 1, \dots, n$ of the same service or product and $\bar{x} = \{x_i\}$, where \bar{x} is apparently the vector of the different issues. Let x_i^a (x_i^a max) express the lower (maximum) acceptable value for agent $a \in \{C, P\}$ concerning issue x_i . Thus, an offer is always rejected by agent a if $x_i \notin [x_i^a$ min, x_i^a max] for any issue. We may now introduce the utility function of the proposed framework as follows. Let $U^a(\bar{x}): [x_i^a$ min, x_i^a max] $\rightarrow [0,1]$ express the utility that agent a assigns to an offer expressed by values \bar{x} of the negotiation issues in the range of its acceptable values. We introduce the significance of issue x_i for agent a

as follows: $S_i^a = \frac{\partial}{\partial x_i} \{U^a(\bar{x})\}$. We now introduce the dependency degree of issue x_i

as follows: $DD_i = \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\partial x_i}{\partial x_j}$. The rationale of the proposed approach is to negotiate

over the entire set of issues sequentially starting from the issue with the lower dependency degree and the higher significance for agent a (in our case the Client Agent). Thus, the negotiation agenda is decided by the Client Agent as follows:

If $\bar{x}_{\min U}^C$ is the vector of issue values that minimizes the overall utility perceived by the Client, then the first issue x_i to be selected for negotiation is the one

maximizing quantity: $\frac{S_i^a}{DD_i} \Big|_{\bar{x}_{\min U}^C}$.

In the generic case where: (i) issues x_j , $j = 1, \dots, k$ have already been negotiated over during the previous k negotiation threads, and (ii) the agreement values of these issues are: x_j^* ($j = 1, \dots, k$), while (iii) the agreement values for issues x_j , $j = k + 1, \dots, n$ still remain to be negotiated over, then, the next issue x_i that will be selected for negotiation is the one maximizing quantity: $\frac{S_i^a}{DD_i^k} \Big|_{\overline{x_{\min U_k}^C}}$, where

$$DD_i^k = \sum_{\substack{l=1 \\ x_l \neq x_j^* (j=1, \dots, k)}}^{n-k} \frac{\partial x_i}{\partial x_l} \Big|_{x_j = x_j^* (j=1, \dots, k)} \quad \text{and} \quad \overline{x_{\min U_k}^C} \text{ is the vector of issue values where}$$

$x_j = x_j^*$ for $j = 1, \dots, k$ and x_j for $j = k + 1, \dots, n$ is set so that the overall utility perceived by the Client is minimized.

As already stated, the proposed approach delays the negotiation of the most interdependent and less significant issues as much as possible. These results are near optimal in the case where the negotiation issues do not depend on many of the other issues under negotiation (i.e. the dependency degree is not high).

The proposed approach to extend the single-issue oriented mechanism to avoid disagreements on the last round, towards multi-issue procedures is generic and applicable in any circumstances. The mechanism can be applied as is and provide multi-issue procedures with the results presented in the experiments section. Based on the last two subsections, the rest of the paper attempts to prove that the proposed approximation mechanisms are efficient and applicable in single-issue as well as multi-issue bilateral negotiation frameworks.

3 The Approximators Employed for Opponent Behaviour Prediction

As already mentioned, the purpose of this framework is to estimate the opponent's next offer in order to avoid unsuccessful negotiation threads, when those are very likely to happen, as the pre-final round has been reached at the time the approximators are used. Each of the studied approximator is coupled with the core negotiation strategy adopted by the CA based on the approach described in the previous section. The four approximators are examined independently of each other, as no cooperation among them is foreseen. As already stated, all approximators have been designed and configured in order to demonstrate optimal performance on predicting online the next round offer of PAs that have adopted arbitrary time-dependent negotiation strategies.

It is widely known that polynomials are by far the easiest functions to process and approximate [12]. There are three major classes of polynomial approximators: Interpolators, Least Squares, and Splines [12]. As interpolators do not offer the accuracy and convergence characteristics required in our case [13], we will use Least Squares and Splines approximators. The objective in the studied framework is to

approximate the function that corresponds to the set of datapoints, available in a form, observed in the course of the negotiation process.

Splines are a widely accepted and used method for function approximation that results in a smooth and continuous function approximating discrete datapoints [12]. Particularly popular is the cubic spline, mainly due to its continuous second derivative that ensures the C^2 continuity and also the uniqueness of the 3rd degree polynomials that perform the approximation and are easy to estimate.

On the other hand, the Least Squares (LS) approximation is widely used to estimate the numerical parameter values to fit a function to a set of data and to characterize the statistical properties of estimates [12]. It always provides a unique solution to the approximation problems, the complexity of which is relatively low, even for polynomials of high order [12]. In order to avoid the polynomial oscillation phenomenon [14], which is quite intense in high degree polynomials, we studied the performance of low degree polynomials (below 15). To heuristically determine the most appropriate degree for the polynomial approximator to be used by the CA to predict the behaviour of the PA, series of experiments were conducted that lead to the conclusion that the best fitting polynomial for our functions is that of a 7th degree.

With regards now to the Neural Networks (NNs) employed, which present a significantly suitable behaviour as presented in our previous work [7], what should be mentioned here for clarity and coherency is that a MLP with a single hidden layer of 3 log-sigmoid neurons and a linear output one and a RBF NN with 2.29 neurons in average [7] are the most suitable architectures for our problem.

4 Experiments and Evaluation

In this section, a thorough analysis of the experiments conducted is performed in two levels. More specifically, the time-dependent PA is coupled with a behaviour-dependent CA, while the performance of the proposed solution is measured for a wide range of PA strategies detailed below. The framework has been modelled using MATLAB, and after initial evaluation results, the negotiation framework described was implemented using the JADE v3.3 mobile agent platform, while open source java libraries have been exploited for the implementation of the approximators.

Extended experiments have been conducted to examine the effectiveness of coupling the MLP NN, the RBF NN, a cubic spline (CS) or a 7th degree polynomial (poly7) with the fRTFT imitative negotiation strategy to increase the success ratio of the negotiations. The main hypotheses used in the conducted experiments are as follows: (i) the Client Agent negotiates with a Provider Agent, following the rules of the framework described in Section 2, (ii) the two agents are unaware of the opponent's strategies, acceptable value ranges and deadlines, and aim to achieve the best possible result on behalf of their owners using as input only the opponent's history of offers, (iii) as the experiments' objective is to estimate the provider's last round offer, in all test cases the provider's deadline is never before the client's deadline (else the designed approximators would not be used at all), and (iv) the

following fixed parameter settings are used for all experiments conducted: $[p_{\min}^p, p_{\max}^p] = [0, 100]$, $p_{\min}^c = 0$, and $L_c = 100$. In this framework, three families of experiments have been conducted to compare the performance of the pure fRTFT strategy, the MLP-assisted fRTFT, the RBF-assisted fRTFT, the CS-assisted fRTFT and the poly7-assisted fRTFT.

The first family of conducted experiments monitors the negotiation result for Linear PA behaviour over varying upper acceptable price threshold for the CA (i.e., acceptable price intervals overlap for the negotiators ranges from 0% to 100%) and over varying PA deadlines. These experiments settings are: $[p_{\min}^p, p_{\max}^p] = [0, 100]$, $p_{\min}^c = 0$, $L_c = 100$, $\beta = 1$, $L_p \in [100, 200]$ and $p_{\max}^c \in [0, 100] \Rightarrow \text{PriceIntervalsOverlap} \in [0\%, 100\%]$, where L_p denotes the negotiation round index on behalf of the PA. The sample values for p_{\max}^c are 0:1:100 (all discrete values from 0 to 100 with a step of 1), while for L_p they are 100:1:200. It corresponds to PAs that are rather neutral and have plenty of time to spend for such procedures and to CAs that have various reservation prices' intervals, from extremely hard (maximum and minimum acceptable prices are very close) to substantially loose (maximum is much greater than the minimum acceptable price).

The second family records the negotiation result in case there is a full acceptable price interval overlap over varying concession rates (i.e., evaluating the system's performance for Conceder PA, when PA is willing to concede rapidly in the early phase of negotiation, or Boulware, when PA is willing to concede substantially only when its time deadline is approaching and Linear PA behaviour, when the convergence follows a linear manner) and over varying deadlines for the PA. These experiments' settings are: $[p_{\min}^p, p_{\max}^p] = [0, 100]$, $[p_{\min}^c, p_{\max}^c] = [0, 100]$, $L_c = 100$, $L_p \in [100, 200]$ and $\beta \in [0.1, 10]$. The sample values for β are 0.1:0.02:1 and 1.2:0.2:10, while for L_p they are 100:1:200. This case is present in many negotiation environments, where the acceptable prices are more or less fixed both for the PAs and the CAs, while the PAs retain a variety of alternatives with regards to the timeouts they have and the willingness to sell on higher or lower prices right from the beginning of the procedures leading to fast or slow convergence from their maximum offers to their minimum ones.

The third family of experiments performed investigates the case where CA's deadline is half than that of the PA, and records the negotiation result over varying concession rates (i.e., for Conceder, Boulware and Linear PA behaviour) and over varying upper acceptable price threshold for the CA. The specific experiments' settings are: $[p_{\min}^p, p_{\max}^p] = [0, 100]$, $p_{\min}^c = 0$, $L_c = 100$, $L_p = 200$, $\beta \in [0.1, 10]$ and $p_{\max}^c \in [0, 100] \Rightarrow \text{Price Intervals Overlap} \in [0\%, 100\%]$. The sample values for β are 0.1:0.02:1 and 1.2:0.2:10, while for p_{\max}^c they are 0:1:100. In these cases, the PAs have plenty of time to finish the procedure and every possible way to converge to their minimum, from extremely slow to substantially fast. On the other hand the CAs vary their acceptable price interval to cover all the possible combinations of overlap.

The results of the three families of experiments are depicted in Figure 1 (first experiment set), Figure 2 (second experiment set), and Figure 3 (third experiment set).

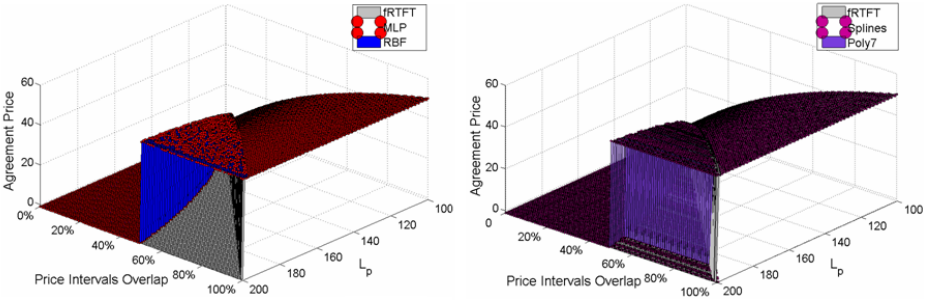


Fig. 1. Agreement price for Linear PA behaviour, $L_C = 100$, $L_P \in [100,200]$ and $p_{max}^C \in [0,100]$: (a) comparison of the pure fRTFT, the MLP-assisted fRTFT and the RBF-assisted fRTFT, (b) comparison of the pure fRTFT, the fRTFT assisted by the cubic spline approximator and the polynomial approximator

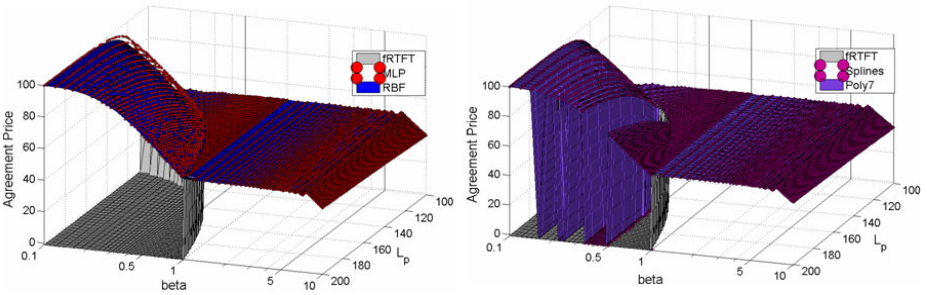


Fig. 2. Agreement price for full price interval overlap for $L_C = 100$, $L_P \in [100,200]$ and $\beta \in [0.1,10]$: (a) comparison of the pure fRTFT, the MLP-assisted fRTFT and the RBF-assisted fRTFT, (b) comparison of the pure fRTFT, the fRTFT assisted by the cubic spline approximator and the polynomial approximator

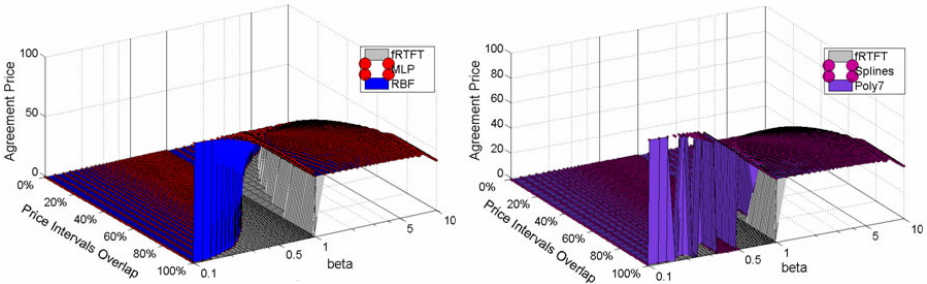


Fig. 3. Agreement price for $L_P = 200$, $p_{max}^C \in [0,100]$ and $\beta \in [0.1,10]$: (a) comparison of the pure fRTFT, the MLP-assisted fRTFT and the RBF-assisted fRTFT, (b) comparison of the pure fRTFT, the fRTFT assisted by the cubic spline approximator and the polynomial approximator

Table 1. Comparative results for the success ratios, estimation errors and temporal requirements of the five strategies (pure fRTTF, and fRTTF assisted by MLP NN, RBF NN, Cubic Spline approximator and polynomial approximator)

Experiments' Settings		$p_{\max}^c \in [0,100]$ $L_p \in [100,200]$ $\beta = 1$	$p_{\max}^c = 100$ $L_p \in [100,200]$ $\beta \in [0.1,10]$	$p_{\max}^c \in [0,100]$ $L_p = 200$ $\beta \in [0.1,10]$	Overall Results
Experiment Set Size		10201	9191	9191	28583
Feasible Agreements Number		7059	9191	5083	21333
Success Ratio (over feasible agreements)	fRTTF	72.30%	74.20%	71.67%	72.97%
	MLP	100.00%	100.00%	100.00%	100.00%
	RBF	99.82%	100.00%	100.00%	99.94%
	CubicSpline	84.25%	93.71%	92.45%	90.28%
	Poly-7	99.39%	91.13%	92.70%	94.24%
Successful Negotiations Increase	MLP	38.30%	34.77%	39.53%	37.04%
	RBF	38.05%	34.77%	39.53%	36.96%
	CubicSpline	16.52%	26.29%	28.99%	23.72%
	Poly-7	37.46%	22.82%	29.34%	29.14%
Mean Estimation Error	MLP	5.74%	2.79%	1.73%	3.50%
	RBF	1.32%	0.39%	0.14%	0.64%
	CubicSpline	4.91%	3.13%	2.89%	3.69%
	Poly-7	2.02%	1.08%	3.73%	2.27%
Standard Deviation of Error	MLP	3.93%	1.87%	1.12%	2.36%
	RBF	0.72%	0.12%	0.04%	0.31%
	CubicSpline	2.81%	2.02%	1.47%	2.13%
	Poly-7	1.03%	0.51%	1.89%	1.14%
Times the Approximators were used		5302	2476	5564	13342
Mean Overall Time Required (sec)	fRTTF	0.0095	0.0105	0.0082	0.0094
	MLP	0.3522	0.1889	0.4754	0.3727
	RBF	0.0849	0.0506	0.1122	0.0898
	CubicSpline	0.0260	0.0229	0.0320	0.0279
	Poly-7	0.0242	0.0190	0.0270	0.0244

In Table 1 comparative results for the three experiment families are illustrated with regards to the success ratio of the five strategies, summing up the results depicted in the aforementioned Figures. The results presented in the table indicate that all four assisted strategies clearly outperform the pure fRTTF in all three families of experiments, as they increase the success ratio up to 25.8% in absolute terms or up to 39.5% in relative terms. Overall results for the set of 28583 experiments conducted indicate that both NN-assisted strategies increase to ~100% the average acceptance ratio of the pure fRTTF that is ~73% considering only the cases of feasible agreements, while the CS-assisted and the poly7-assisted strategies increase this number to ~90% and ~94% respectively. With regards to the number of agreements, both NN-assisted strategies manage to increase it by almost 37% in average, which is highly significant in automated negotiation domains. The poly7-assisted strategy follows, presenting more than 29% increase, while the CS-assisted strategy manages

to increase the number of agreements of the pure fRTFT by approximately 24% in

average. Concerning the mean estimation error $E = \frac{1}{n} \sum_{i=1}^n \frac{|x_i - \bar{x}_i|}{x_i} = \frac{1}{n} \sum_{i=1}^n \frac{|e_i|}{x_i}$, the RBF-

assisted strategy outperforms the rest demonstrating only 0.64% error, while the MLP-, CS- and poly7-assisted strategies result in 3.50%, 3.69% and 2.27% error respectively. Similarly, with regards to the standard deviation (SD) of the estimation

error $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{|e_i|}{x_i} - E \right)^2}$, the RBF-assisted strategy outperforms the rest demonstrating

only 0.64% SD, while the MLP-, CS- and poly7-assisted strategies result in 2.36%, 2.13% and 1.14% SD respectively.

With regards to the processing/time resources required, the poly7 clearly outperforms the other three approximators in all three families of experiments. As presented in Table 1, the mean overall time required by the MLP, by the RBF or by the CS is approximately 15.3, 3.6 or 1.1 times higher respectively than the time required by the poly7.

Combining all experimental results obtained in this subsection aiming to identify the most appropriate approximator for the negotiation framework, it becomes evident that the RBF NN, not only achieves the lowest overall error over the estimated opponent's next offer, but (along with the MLP) it also maximises the number of successful negotiations, while the processing time it requires is quite low, i.e. less than 90 ms in average. The polynomial fitting is outperformed by the RBF due to the fact that the former is prone to fluctuations of the values it is required to fit. The four varying parameters of the time-dependent PA strategy cause the poly7 to fail in several circumstances, whereas the RBF approximator generalises far better being able to store more information monitored.

5 Conclusions

Autonomous agents are a powerful technology that may enhance the intelligence, sophistication and performance of various processes that take place in the e-marketplace. This paper presented a single-issue bilateral negotiation framework designed for self-interested agents that act in e-commerce environments, with extensibility to most multi-issue environments. In this framework, MLP and RBF NNs have been exploited, as well as cubic splines and least-square-based polynomial approximators aiming to predict the future offers of the negotiating Provider Agent. Extensive experiments were conducted over more than 28500 different settings in order to evaluate the designed negotiation strategies. These experiments indicate that the RBF NN is more suitable, as it maximises the number of successful negotiations, achieves the lowest overall error over the estimated opponent's next offer, while the processing resources required are quite low. The practical value of the results presented in this paper are: the establishment of e-marketplace environment where the percentage of agreements is maximised, while the profit of the clients is increased; protection of the negotiating parties' privacy, as the proposed solution requires no information other than the offers of the negotiators, and thus no private information is disclosed; online handling of single-instance negotiations that is the most popular and

lightweight type used in e-marketplaces, as no a-priori knowledge or training is required; easy implementation of the designed negotiating agents, as there are off-the-shelf libraries with the proposed approximators; achievement of high accuracy prediction and evaluation for arbitrary CA strategies; and finally requirement for minimal processing, storage, and network resources.

Acknowledgements

This work has in part been supported by the National Strategic Reference Framework (NSRF) (Regional Operational Programme – Western Greece) under the title “Advanced Systems and Services over Wireless and Mobile Networks” (number 312179).

References

1. Trastour, D., Bartolini, C., González-Castillo, J.: A Semantic Web approach to service description for matchmaking of services. In: *Int. Semantic Web Working Symposium*, Stanford, California, USA (2001)
2. Jennings, N., Sycara, K., Wooldridge, M.: A Roadmap of Agent Research and Development. *J. of Autonomous Agents and Multi-Agent Systems* 1(1), 7–38 (1998)
3. Silva, A.R., Romão, A., Deugo, D., da Silva, M.M.: Towards a Reference Model for Surveying Mobile Agent Systems. *J. of Autonomous Agents and Multi-Agent Systems* 4(3), 187–231 (2001)
4. Sierra, C.: Agent-Mediated Electronic Commerce. *J. of Autonomous Agents and Multi-Agent Systems* 9(3), 285–301 (2004)
5. Faratin, P., Sierra, C., Jennings, N.: Negotiation Decision Functions for Autonomous Agents. *Int. J. of Robotics and Autonomous Systems* 24(3-4), 159–182 (1998)
6. Jennings, N., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M.: Automated Negotiation: Prospects, Methods, and Challenges. *Int. J. of Group Decision and Negotiation* 10(2), 199–215 (2001)
7. Papaioannou, I., Roussaki, I., Anagnostou, M.: Neural networks against genetic algorithms for negotiating agent behaviour prediction. *Web Intelligence and Agent Systems* 6(2), 217–233 (2008)
8. Raiffa, H.: *The Art and Science of Negotiation*. Harvard University Press, Cambridge (1982)
9. Paurobally, S., Turner, P.J., Jennings, N.: Automating negotiation for m-services. *IEEE Trans. on Systems, Man and Cybernetics (Part A: Systems and Humans)* 33(6), 709–724 (2003)
10. Kraus, S.: *Strategic Negotiation in Multiagent Environments*. MIT Press, Cambridge (2001)
11. Fatima, S.S., Wooldridge, M., Jennings, N.R.: Multi-issue negotiation with deadlines. *J. of Artificial Intelligence Research* 27, 381–417 (2006)
12. Schatzman, M.: *Numerical Analysis: A Mathematical Introduction*. Oxford University Press, USA (2002)
13. Süli, E., Mayers, D.: *An Introduction to Numerical Analysis*, ch. 6. Cambridge University Press, UK (2003)
14. Milovanovic, G.V., Mitrinovic, D.S., Rassias, T.M.: *Topics in Polynomials: External Problems, Inequalities, Zeros*. World Scientific, Singapore (1994)

An AER to CAN Bridge for Spike-Based Robot Control

M. Dominguez-Morales¹, A. Jimenez-Fernandez¹, R. Paz¹, A. Linares-Barranco¹,
D. Cascado¹, J.L. Coronado², J.L. Muñoz², and G. Jimenez¹

¹ Robotic and Technology of Computers Lab. University of Seville, Spain
mdominguez@atc.us.es

² NEUROCOR Lab. Technical University of Cartagena, Spain

Abstract. Address-Event-Representation (AER) is a bio-inspired communication protocol between chips. A set of AER sensors (retina and cochleas), processors (convolvers, WTA, mappers, ...) and actuators can be found in the literature that have been specifically designed for mimicking the communication principle in the brain: spikes. The problem when developing complex robots based on AER (or spikes) is to command actuators (motors) directly with spikes. Commercial robots are usually based on commercial standards (CAN) that do not allow powering actuators directly with spikes. This paper presents a co-design FPGA and embedded computer system that implements a bridge between these two protocols: CAN and AER. The bridge has been analyzed under the Spanish project VULCANO¹ with an arm robot and a Shadow anthropomorphic hand.

Keywords: CAN, AER, spike-based, neuromorphic engineering, anthropomorphic robot, FPGA, VHDL, embedded computer.

1 Introduction

Controller Area Network (CAN) is an industrial protocol that minimizes the number of unrecovered transmission errors using a set of mechanism for detecting and signaling errors. This protocol is so efficient that it is usually used at automotive, robots and mobile robots for communicating sensors and actuator to a central processor. CAN is nowadays a typical interface for commercial robots, motors or even sensors. On the other hand Address-Event-Representation (AER) is a neuro-inspired communication protocol for transferring information between silicon neurons in different chips. Neuromorphic engineers work actively in developing sensors, processors and actuators that mimic the nervous system behavior.

Neuroinformatic aims to join together several field specialists (biologists, psychologists, engineers, physicists, chemists, and informatics) in order to develop auto-reconfigurable systems that mimic the human body and specially emulate the human brain. Neuromorphic engineers work in the study, design and development of neuro-inspired artifacts developed with artificial mechanisms, like VLSI chips for sensors [1][2][3], neuro-inspired processing, filtering or learning [4][5][6], neuro-inspired control-pattern-generators (CPG) [7], neuro-inspired robotics [8] and so on.

¹ This work has been supported by Spanish government grant VULCANO (TEC2009-10639-C04-02).

All these mechanisms share the way of producing, transforming and transferring the information: the spike-based representation of the information, like in a mammal neural system. A neuromorphic VLSI chip is designed in order to gather several thousands of silicon spike-based neurons. A problem arises when those neurons have to communicate outside the chip with other neurons of a different chip. Typically, neuromorphic engineers have adopted the so-called solution Address-Event-Representation (AER). AER was proposed by the Mead lab in 1991 [9] for communicating between neuromorphic chips with spikes (Figure 1). Each time a cell on a sender device generates a spike, it communicates with the array periphery and a digital word representing a code or address for that pixel is placed on the external inter-chip digital bus (the AER bus). In the receiver chip, spikes are directed to the pixels whose code or address was on the bus. In this way, cells with the same address in the emitter and receiver chips are virtually connected by streams of spikes. Cells that are more active access the bus more frequently than those less active. There is a growing community of AER protocol users for bio-inspired applications in vision, audition systems, and robot control, as demonstrated by the success in the last years of the AER group at the Neuromorphic Engineering Workshop series [7] and the CapoCaccia Cognitive Neuromorphic Engineering Workshop [10]. The goal of this community is to build large multichip and multi-layer hierarchically structured systems capable of performing massively-parallel data-driven processing in real time.

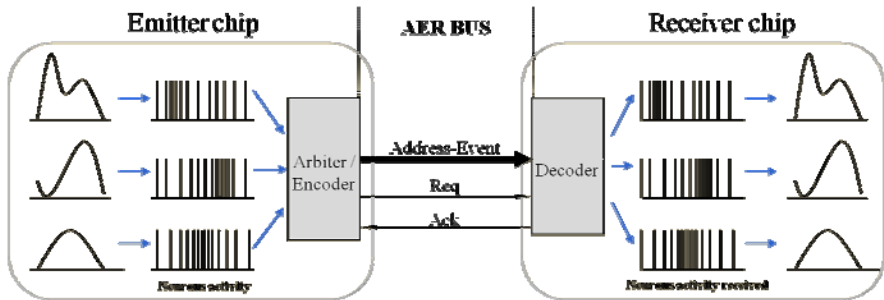


Fig. 1. Rate-coded AER inter-chip communication scheme

Furthermore, the application of these systems to real applications demonstrate the benefits of the spike-based representation, allowing extremely low latencies between spikes produced by a sensor and the first spike arriving to the actuator (in the order of few microseconds), as demonstrated in the EU project CAVIAR [11].

The implementation of spike-based Proportional-Integral-Derivative (PID) controller allows to complete the last stage of a neuromorphic system, by powering the equivalent of a muscle in a robot directly with spikes [8] (a DC motor).

Nevertheless, commercial robots are usually black-boxes consisting in a set of motors or actuators, and sensors with a common interface that allows communicating with a computer that is usually in charge of solving the cinematic and dynamic equations needed to implement tasks like grasping, tracking, manipulating an object, etc. One of these common interfaces is the Controller Area Network (CAN) bus. The CAN bus was developed by Robert Bosch GmbH company for communicating

messages in a distributed environment ensuring real-time capabilities and using only two wires. The number of nodes, the distance between them, the bus bandwidth, the error detection and correction mechanism, the arbitration policy and other aspects of the CAN protocol can be found in [12].

In this paper we present a hardware-software bridge between the AER bus and the CAN bus. The bridge has been implemented using the hardware platforms developed under Spanish Government granted VULCANO project. These platforms have been designed for covering the implementation of more complex spike based operations. VULCANO has the aim of developing and joining together a set of AER chips (retina sensors and filters) into a layer of image filtering. The result of this layer is fussed by a second layer implementing a set of brain-inspired algorithms for sensory fusion and visual-motor coordination. The implementation of this second layer is based on a co-design platform composed by a FPGA and an embedded computer (Toradex Colibri). This embedded computer provides a CAN interface. Arm and hand robots used in VULCANO offer CAN interfaces for motor control and sensors monitoring.

Section II offers a brief review of the most important characteristics of the CAN protocol. Section III explains the scenario in VULCANO project. Then, in section IV we describe the architecture of the AER-CAN bridge and we present a performance study.

2 Controller Area Network (CAN)

CAN physical layer establishes that the information is represented by the voltage difference between two wires. Logical bits are called dominant ('0') when the voltage difference between the two wires is high and recessive ('1') when they are similar. The network is composed by a set of nodes sharing a unique pair of wires (CAN bus). When several nodes write at the same time on the bus, if only one node writes a dominant bit, then the CAN bus will conserve the dominant bit, and if all the nodes write recessive bits, the CAN bus conserves the recessive value.

Each node connected to the CAN bus is always monitoring the bus, even when a node is transmitting. If several nodes are transmitting at the same time, a node can lose the CAN bus if it monitors a dominant bit when it is transmitting a recessive bit.

Distributed nodes are free to use the bus when a message is finished. Nodes do not need to have an arbiter selecting which node is able to transmit each time, because the protocol is auto-arbitrated thanks to the characteristic of dominant and recessive bits. All the messages are transmitted starting with a start of frame bit (SOF) and then, the identifier of the message. This identifier is used for arbitrating which node will use the CAN bus. When a node is writing its identifier on the CAN bus, if a recessive bit is written ('1') and a dominant bit ('0') appears on the bus, this node has lost the competition and it has to wait until the winner node finished its message transmission. Figure 2 bottom shows a competition example. A message includes protocol information, a variable number of data bytes from 0 to 8, followed by a 16-bit Cyclic Redundancy Code (CRC) for error detections (see Figure 2 top).

The medium access and transport protocol layers establish that when a node is transmitting a message, the rest of the nodes are checking the message. Therefore, in the CAN protocol non-transmitter nodes are responsible of the correctness and the

efficiency of message transmissions by monitoring all the bits of a message and collaborating in the ACK bit of a message when an error is found. Even more, the transmitter node is also monitoring the CAN bus during the transmission of its own message and it is also checking if each bit written in the bus is correctly read: if the transmitter node writes a recessive bit in the bus, it has to read a recessive bit during the rest of the time that this bit must appear on the bus.

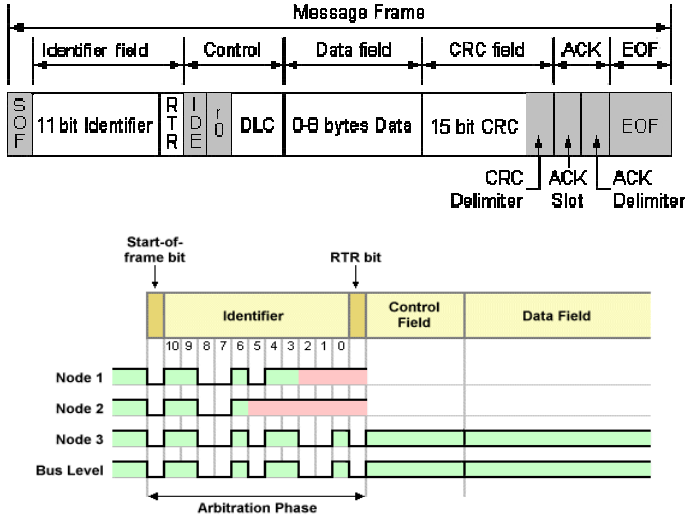


Fig. 2. CAN message (top) and priority arbitration example (bottom)

The maximum speed of the CAN standard is 1 Mega bit per second (Mbps) for up to 40 meters of cable distance between the furthest nodes. This implies one microsecond per bit. Every bit time is divided in several parts: synchronization, propagation (400 ns for 40 meters), segment one and segment two. Synchronization part is very small (usually one clock cycle of the CAN controller clock). The propagation part must be preserved in order to ensure the correct reception of the signal in both sides of a 40 meters cable. Finally segments 1 and 2 complete the bit time up to 1 microsecond. Receivers will check the state of the CAN signal between segment 1 and 2. These segments are equal to a setup and hold time of digital circuits.

CAN standard has several similarities with AER protocol:

- Both are event-based protocols. AER sensors or actuators produce either a stream of events that represents the sensor value, or its change, or the actuator intensity or change. CAN nodes codify in a unique message the sensor value or the command parameters.
- Both use addresses for identifying messages. When using AER, an emitter chip uses a set of addresses depending on the number of emitter neurons. In CAN, each node can work (transmit or receive) a set of messages with different identifiers. An identifier in CAN is not used for identifying neither an emitter nor a receiver; they are used to identify messages produced by sensors or received by actuators.

On the other hand there are several important differences:

- CAN is a serial and synchronous bus, while AER is typically parallel and asynchronous.
- CAN maximum speed is 1 Mbps. This is considerably low respect to AER that can work at peak rates higher than 25 Mega-events-per-second (400 Mbps for 16-bit buses)[4]. Nevertheless there are improvements to the CAN protocol that reach up to 16Mbps (CAN+) [13].
- CAN includes in the message transmission additional information for increasing the robustness. AER transmits the address of a neuron without any other complementary information.
- AER codify the information in frequency or by the number of repetitions of the same address in the bus, while CAN send a unique message with the command and value in up to 8 bytes.

For those applications with low speed requirements there is a reduced number of neurons or identifiers and scenarios with strong noise interferences, so it could be adequate to inherit the properties of the CAN bus in order to implement a serial and robust AER protocol. For example, the spike motor actuation and the motor sensor monitoring. An AER system is able to process visual information in a very fast way and with a low latency. The AER system can produce a stream of spikes for actuating into a set of muscles (or motors in robotic). Since the motor is a mechanical object, it has a huge delay in implementing the position or velocity orders, when compared to the AER processing system. Due to this limitation of the motors, it is justified to include an AER-CAN interface without losing performance.

3 Actuators and Sensors through CAN

VULCANO focuses on the AER for developing a sensory-motor system completely based on spikes. From sensing and filtering the visual information, developing AER retina and AER convolvers, to the adaptation and development of anthropomorphic robots (hand and arm) and their interfaces, through the development of high level algorithms for sensory fusion, visual-motor coordination and the cinematic and dynamic of the robots.

The robotic arm consists into a set of articulations based on commercial motors. Each of these motors is mounted with a controller that receives CAN messages with commands. These commands can be either a) new positions or degree for the articulation, with a fixed intensity and speed, or b) a request for the value of a sensor, like the position of the articulation, the intensity of the motor, the speed, ...

Existing AER based controllers for DC motors [8] requires to access directly to the motor and to receive directly the information of the sensors in order to adapt the frequency of spikes to be sent to the motor for modifying the position, speed or power of the motor. This kind of AER controller cannot be used in such a commercial platform because the robot will lose all the robustness and efficiency that the CAN is providing (apart from the warranty of the product). Therefore, in this case it is very

important and necessary to develop an AER-CAN interface able to receive streams of spikes from the last layer of the AER systems and convert it into CAN commands, and vice versa, to convert CAN sensory messages into a stream of spikes for the corresponding layer of the AER system.

4 CAN Bridge Architecture and Performance

Figure 5 shows a block diagram of a co-design prototype for implementing the sensory fusion, visual-motor coordination and kinematic/dynamic algorithms of robots in VULCANO. It consists in a Xilinx Spartan 3 400 FPGA connected to an embedded computer (Toradex Colibri with a Intel Xscale PXA270 under Windows CE) through Direct Memory Access interface (DMA). The FPGA is responsible for receiving, in the AER communication, all the sensing information coming from the cortex layer (AER retina + convolver) and spinal cord sensing (the robot sensors in AER format). This sensor information can be either 1) fused in the FPGA and stored, through DMA, in the computer DRAM, or 2) packed into a sequence of time-stamped events in the DRAM computer memory, through DMA. Then, the embedded computer will use this information for executing the sensory fusion and visual-motor coordination. In the first case, the computer program can work in a completely digital way, being the spike-based approach only resident in the FPGA. The computer receives digital values of the sensors, so a spike to digital conversion is done. In the second case, both the FPGA and the embedded computer are working with spikes, so the system takes all the advantages of a spike-based system, having very low latencies from sensors to actuators. DMA has a maximum bandwidth of 101'72 MBps between the Spartan 3 and the Toradex computer. This bandwidth allows transmitting packetized AER 16-bit events and their 16-bit timestamps at a regular rate of 25 Mevps that is the maximum throughput of the USB-AER hardware interface [14].

Since the sensory fusion algorithm is executed in the embedded computer and the CAN protocol is accessible through the Intel XScale board, we have decided to access the CAN bus through the embedded computer and offer the additional sensor information directly to the sensory fusion algorithm. On the other hand, the visual-motor coordination is also executed in the embedded computer, so it is faster and more adequate to send the CAN motor commands directly from the embedded computer.

Each node of the arm-robot consists in a Schunk motor and a PowerCube communication interface in CAN mode. There are six nodes connected through CAN to the Toradex embedded computer. For each message received by the PowerCube interface of a motor node, a command is executed in the node. During the execution of that command, an acknowledgment message (ACK) is sent back. If a new command is received while the previous one is being executed, the old one is aborted and the new one is executed. These events allow the system to correct trajectories at the same time they are improved.

We have analyzed the CAN traffic in order to extract the typical and maximum bandwidth of commands that is supported. As can be seen in figure 3, there is a

minimum Inter-Message-Interval (IMI) of $320\mu\text{s}$ that represents a very fast update time for motor commands compared to the response time of such a mechanical actuator. This time has been obtained when the commands are sent sequentially (a new command is not sent until the reception of the ACK of the previous one). A motor command of 6 data bytes requires $110\mu\text{s}$ to be transmitted and $120\mu\text{s}$ to receive and process it through PowerCube. Then it responds with an ACK message (2 data bytes) that requires $78\mu\text{s}$. Finally there is a $12\mu\text{s}$ pause before the next motor command appears in the CAN bus. With these data the performance of the system is a bandwidth of 3,125 K commands per second (Kcps). In Figure 4 (right), it is shown a performance graph considering different command lengths (from 1 to 6 bytes) and two different communication mechanisms: 1) Sequentially and 2) Interlaced (the sparse time between the command CAN message and the ACK is used to send a second command message to another motor, as represented in figure 4 left). The maximum bandwidth that can be extracted from the AER-CAN interface is 5Kcps.

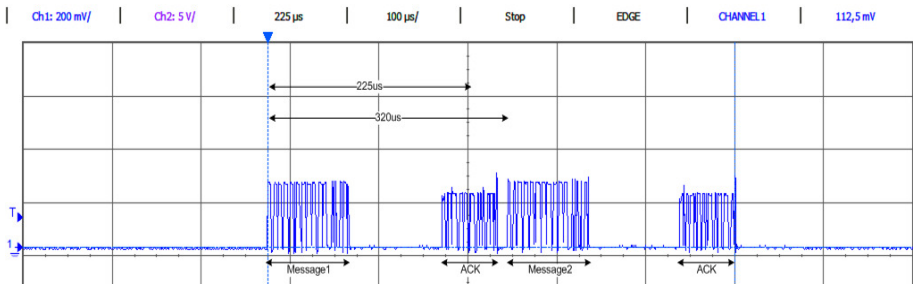


Fig. 3. Oscilloscope capture of CANRX over time during two CAN message transmissions (6 data bytes) for different motors. Acknowledge CAN messages are sent back (2 data bytes).

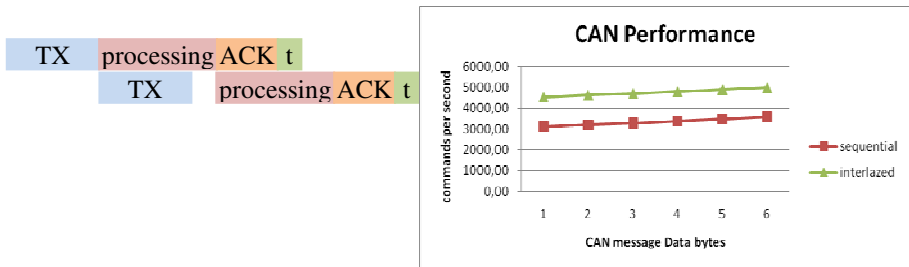
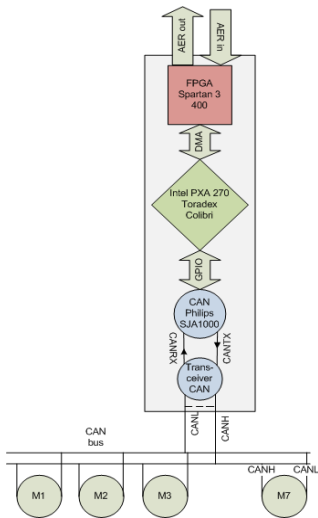
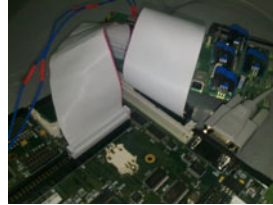


Fig. 4. Performance in Commands per second for sequential and interlaced CAN messages

If we suppose the 1) case with $320\mu\text{s}$ of IMI, as it is shown in figure 3, then it means that the FPGA can receive a considerable set of AER events (i.e., 320 events if we consider an Inter-Spike-Interval of $1\mu\text{s}$ or higher) in order to calculate the new updated position of each node of the arm before sending a new CAN message.



Toradex board with CAN interface and Colibri PXA270 connected to AER-Robot board.



Robotic arm used in VULCANO

Fig. 5. AER-CAN bridge block diagram and prototype boards used under VULCANO project

5 Conclusions

In this paper we present a bridge between AER and CAN buses that allow attaching commercial closed robots to neuro-inspired and spike-based sensory-processing systems developed under AER. The bridge is not able to transmit spikes directly to motors, but it is able to translate a stream of spikes from the AER system into an equivalent CAN message for modifying the degree, intensity and power of a fixed articulation in a robot without bandwidth problems. Furthermore, this AER-CAN bridge allows reading robot sensors and translating the information received into a stream of spikes for AER sensory fusion respect to visual information.

References

- [1] Lichtsteiner, P., Posch, C., Delbruck, T.: A 128×128 120dB 15 us Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal on Solid-State Circuits* 43(2), 566–576 (2008)
- [2] Chan, V., Liu, S.C., van Schaik, A.: AER EAR: A Matched Silicon Cochlea Pair with Address-Event-Representation Interface. *IEEE Trans. Circuits and Systems-I* 54(1), 48–59 (2007)
- [3] Costas-Santos, J., Serrano-Gotarredona, T., Serrano-Gotarredona, R., Linares-Barranco, B.: A Spatial Contrast Retina with On-chip Calibration for Neuromorphic Spike-Based AER Vision Systems. *IEEE Trans. Circuits and Systems-I* 54(7), 1444–1458 (2007)
- [4] Serrano-Gotarredona, R., et al.: A Neuromorphic Cortical-Layer Microchip for Spike-Based Event Processing Vision Systems. *IEEE T. Circuits Systems-I* 53(12), 2548–2566 (2006)
- [5] Haflliger, P.: Adaptive WTA with an Analog VLSI Neuromorphic Learning Chip. *IEEE Transactions on Neural Networks* 18(2), 551–572 (2007)

- [6] Indiveri, G., Chicca, E., Douglas, R.: A VLSI Array of Low-Power Spiking Neurons and Bistables Synapses with Spike-Timing Dependent Plasticity. *IEEE Transactions on Neural Networks* 17(1), 211–221 (2006)
- [7] Cohen, A., et al.: Report to the National Science Foundation: Workshop on Neuromorphic Engineering, Telluride, Colorado, USA (June–July 2004), <http://www.ini.unizh.ch/telluride>
- [8] CAN specification, <http://www.semiconductors.bosch.de/pdf/can2spec.pdf>
- [9] Sivilotti, M.: Wiring Considerations in analog VLSI Systems with Application to Field-Programmable Networks. Ph.D. Thesis, California Institute of Technology, Pasadena CA (1991)
- [10] The 2011 Cognitive Neuromorphic Engineering Workshop, <http://capocaccia.ethz.ch/capo/wiki/2011>
- [11] Jiménez-Fernández, A., Linares-Barranco, A., Paz-Vicente, R., Jimenez-Moreno, G., Berner, R.: Spike-Based Control Monitoring and Analysis With Address Event Representation. In: *Proceeding of the AICCSA-2009, Rabat, Morocco*, vol. 7, pp. 900–906 (2009)
- [12] Ziermann, T., Wildermann, S., Teich, J.: CAN+: A new backward-compatible Controller Area Network (CAN) protocol with up to 16× higher data rates. In: *DATE 2009*, pp. 1088–1093 (2009)
- [13] Linares-Barranco, A., Jiménez-Fernández, A., Paz-Vicente, R., Varona, S., Jiménez, G.: An AER-based actuator interface for controlling an anthropomorphic robotic hand. In: Mira, J., Álvarez, J.R. (eds.) *IWINAC 2007. LNCS*, vol. 4528, pp. 479–489. Springer, Heidelberg (2007)
- [14] Serrano-Gotarredona, R., Oster, M., Lichtsteiner, P., Linares-Barranco, A., Paz-Vicente, R., Gómez-Rodríguez, F., et al.: CAVIAR: A 45k-neuron, 5M-synapse AER Hardware Sensory-Processing-Learning-Actuating System for High-Speed Visual Object Recognition and Tracking. *IEEE Trans. on Neural Networks* 20(9), 1417–1438 (2009)

Neuromorphic Real-Time Objects Tracking Using Address Event Representation and Silicon Retina

F. Gómez- Rodríguez, L. Miró-Amarante, M. Rivas, G. Jimenez,
and F. Diaz-del-Rio

Robotics and Computer's Technology Lab.
University of Seville
Seville, Spain
gomezroz@us.es

Abstract. This paper presents a hierarchical neuromorphic system for tracking objects. We use AER (Address Event Representation) for transmitting and processing visual information provided by an asynchronous temporal contrast silicon retina. Two AER processing layers work in cascade for firstly detecting different objects, and secondly tracking them even with crossing trajectories. The output of the system offers not only the position of the tracked object but also the speed in pixels per second. The system is fully hardware implemented on FPGA (Spartan II 200), which is part of the USB-AER platform developed in part by authors. A 97.2% of the Spartan II is used for 128x128 pixels input resolution and 6 maximum objects recognition and tracking.

Keywords: object tracking, real-time, Address Event Representation, AER, neuromorphic, neuro-inspired, FPGA.

1 Introduction

This paper presents a neuromorphic [1] real time object tracking using visual information provided by a silicon retina¹; this information is processed and transmitted using Address Event Representation (AER) [2].

There are two previous works in the field of object tracking which also use AER for processing and transmitting the visual information. Our system has two important differences from these two previous approaches: first, we present a fully hardware system, described in VHDL and implemented in a FGPA, while in [3] and [4] a computer and a DSP were used for events computing, respectively. Second, our system uses only two events for getting the object position and with only two object positions it is possible to estimate the object velocity.

Followed in this section we review the *Address Event Representation* (AER) communication protocol; the hardware platform used, the *USB-AER Board* [5], developed by the Robotics and Computer's Architecture Lab; and the *Silicon Retina* [6] used, developed by the Institute of Neuroinformatics (INI) of the University of Zurich.

¹ Authors would like to thank to Tobias Delbruck and his group for their silicon retina, and to VULCANO project (TEC2009-10639-C04-02) for supporting this work.

Section 2 is devoted to system’s description; in section 3 we present some experiments to demonstrate the system’s capabilities; finally, in section 4 we present some conclusions.

1.1 The AER Communication Protocol

The AER protocol was proposed for neuro-inspired information transmission from one neuro-inspired chip to another. The basics of AER consist in assigning an address to each cell (neuron) in a chip. Cell activity is transmitted showing the cell’s address in a common bus; two flow control signals are commonly needed (REQ and ACK), to start and stop the transmission.

As a result, the activity of every cell will appear in the common bus. Usually the activity is frequency coded. In this way, if the cell’s activity is high, its address will appear in the bus more frequently than other with lower activity. Each address occurrence, in the bus, is known as an *event*. So, we can say that the activity of each cell is coded in events frequency. Since the AER bus multiplexes all the events in a common bus, an arbiter is needed in the transmitter. Fig. 1.a shows the organization of AER communication. To transmit each event, a simple handshake protocol is normally used (see Fig. 1.b).

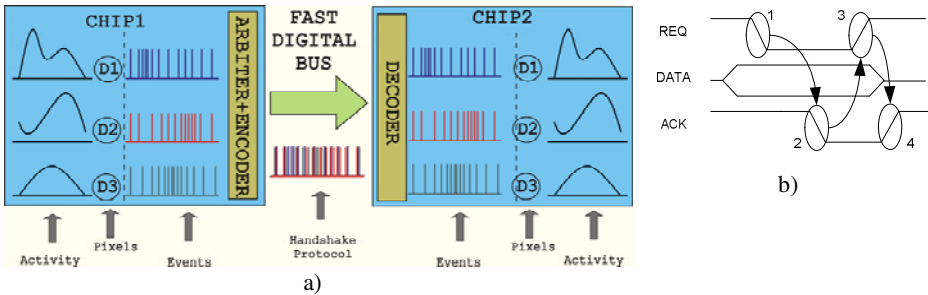


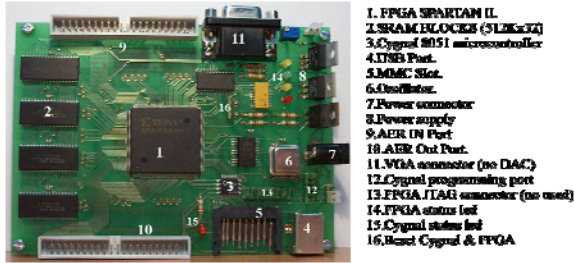
Fig. 1. AER protocol: a) AER transmission overview, b) AER Handshake protocol

1.2 USB-AER Board

The USB-AER board was developed by the Robotic and Computer’s Technology Lab of the University of Seville, to give support to AER-based systems developers. The USB-AER board was initially developed for AER-based systems testing and debugging. This board implements several modules for AER stream sequencing, monitoring, mapping (to change *on the fly* the address space); depending on the FPGA firmware.

The flexibility of USB-AER board allows implementing any others functionalities such as the one presented in this paper.

A picture of the USB-AER board is shown in Fig. 2. USB-AER board core is a Spartan-II 200 Xilinx FPGA, connected with a 12ns, 512Kwords of 32 bits SRAM memory bank. The board uses a Silicon Laboratories C8051F320 microcontroller to implement the USB and the MMC/SD interfaces.



1. FPGA SPARTAN II
2. SRAM/FLASH (SLK-77)
3. Cyrral 9051 microcontroller
4. USB Port
5. DMC Slot
6. Display
7. Power connector
8. Power supply
9. AER IN Port
10. AER Out Port
11. V/A sensor (no DAC)
12. Cyrral programming port
13. FPGA JTAG connector (no used)
14. FPGA status led
15. Cyrral status led
16. Reset Cyrral & FPGA

Fig. 2. USB-AER Board

1.3 The Asynchronous Temporal Contrast Silicon Retina

The silicon retina used was developed by Tobias Delbruck and his group at INI of University of Zurich [5]. Concretely, it is an asynchronous temporal contrast silicon retina, with an AER interface. So the retina output consists of an AER sequence that corresponds to the movement in the scene that the retina is “seeing”.

Fig. 3.a shows a real scene; and Fig. 3.b. a frame-like reconstruction of the AER sequence, dark dots represent pixels where positive contrast changes happened and light dots where negative changes happened.

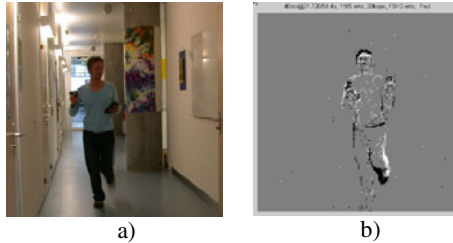


Fig. 3. Retina functionality: a) Scene, b) Retina Output (pictures extracted [6])

2 Real Time Object Tracking System

The system that we propose consists of several cells interconnected, like neurons that process the AER sequence. Every cell tries to track one object obtaining its position and estimating its velocity. We propose a new architecture where, instead of using a classical fully parallel cell architecture, we use cascade architecture.

2.1 Cascade Architecture

The biological neural system architecture consists in a fully parallel interconnected network distributed in layers; fig.4.a shows a scheme of this network.

Fig. 4.b shows the proposed architecture, in each layer the first cell receives the complete AER sequence and extracts, retains and processes several events; which and how many events are extracted depend on the application; the rest of the events are resent to the second cell in the layer. This procedure is repeated in the rest of the cells in a layer and in the rest of the layers.

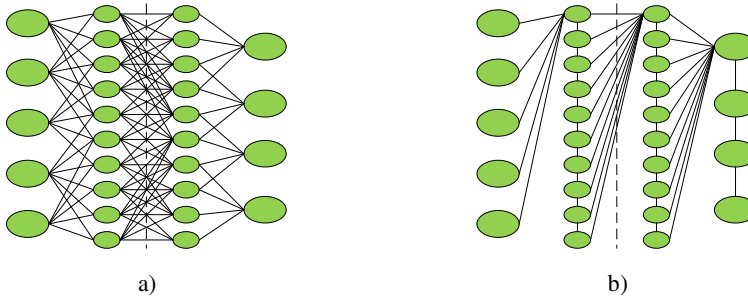


Fig. 4. Neural architectures: a) classical neural parallel network, b) cascade neural network

The most important advantage of this architecture is that each cell only processes the extracted events. Furthermore, this scheme presents an implicit inhibition mechanism, that is, the processed events by one cell are eliminated from the AER sequence and do not interfere with the computation. This scheme allows reducing the AER sequence complexity from one cell to the next.

In this system all cells use AER for information transmission, so each cell has 3 AER ports: one AER input and two AER outputs. The first output gives the result of events *computation* and the second one resends the refused events.

2.2 Objects Tracking Procedure

Following, we present the object tracking procedure used by the architecture presented above. We only use one layer, where each cell, called TrackCell, consists of two sub-cells: one is devoted to object's position determination, called CMCell, and the other, called VCell, is devoted to object's velocity estimation. Fig. 5.a shows these cells.

From now on, we suppose that the input is the visual information provided by the silicon retina, so events correspond to the movement in the scene.

The first cell in the layer receives all the events. Just after the first event is received, the CMCell only extracts and computes events from a small area around this first event (area of interest), resending the rest of the events to the second cell. If during a period of time CMCell does not receive enough events, typically 10 events, it will reset. On the contrary, if during this period of time CMCell receives enough events from the area of interest, it computes the object position as the mean value between the last positive event location and the last negative event location. After the object position computation, CMCell moves the center of the area of interest to the object position. This procedure is repeated after each event is received. If CMCell

receives events near the area of interest (a few pixels around) it will change the area size allowing adapting it to the object size. Fig. 5.b shows the CMCell state machine diagram.

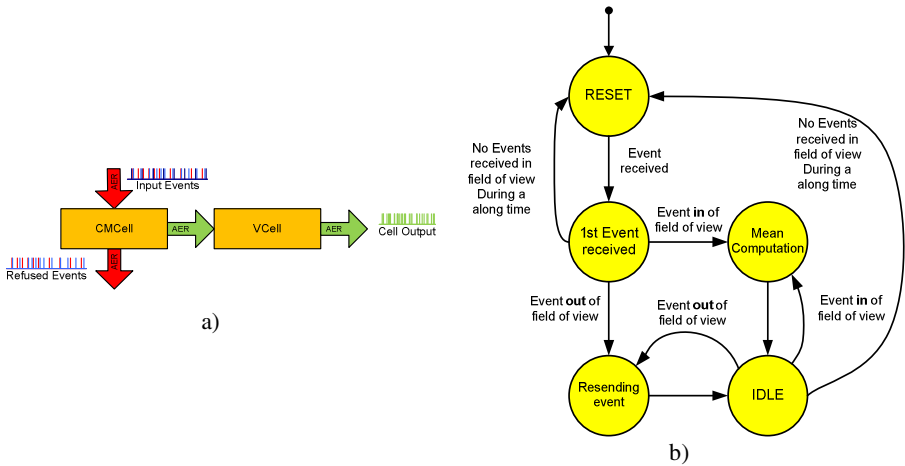


Fig. 5. a) CMCell+VCell and b) CMCell state machine diagram

The output of the CMCell consists of an AER sequence that encodes the object position. This information is used by VCell to estimate the object's velocity.

VCell takes the object position periodically and computes the velocity as the mean value of the last two object positions. Initially, the period used is 100ms, but it changes depending on the velocity computed. If the velocity is high the period will be reduced. On the contrary, if the velocity is low the period will be increased. The velocity is also transmitted using AER.

With this scheme it is possible to track as many objects as TrackCells can be synthesized in a FPGA in cascade. In our case, due to USB-AER board FPGA capacity only 6 objects can be tracked simultaneously.

The key point of this procedure is that events are processed as soon as they are received, without frame integration. Therefore the response time of each cell is very short; in fact it is the delay time (order of ns).

3 Experiments and Capabilities

In this section we present some experiments to demonstrate the system's capabilities and performance. The hardware configuration consists of a silicon retina and a USB-AER board.

In order to see what happened in the system the results are stored in the USB-AER SRAM and periodically they are downloaded to a PC and displayed on the screen.

3.1 Synthetic Stimulus

In these two first experiments we use a USB-AER board for AER sequence generating (no retina used). The USB-AER generates an AER sequence emulating the retina behavior but without noise and impreciseness. The AER sequence emulates a 4x4 pixels object describing a square. In the first experiment the object's velocity is 50 pixels/s and in the second 40000 pixels/s.

Fig. 6.a shows AER sequence fused with the system's output, both accumulated during 1s when the object's velocity is 50 pixels/s. The gray dots are the AER sequence reconstruction and the system's output are drawn in blue. The first value, near to the row, is the velocity modulus and the second is the VCell period. Fig.6.b shows the system's output when the object's velocity is 40000 pixels/s.

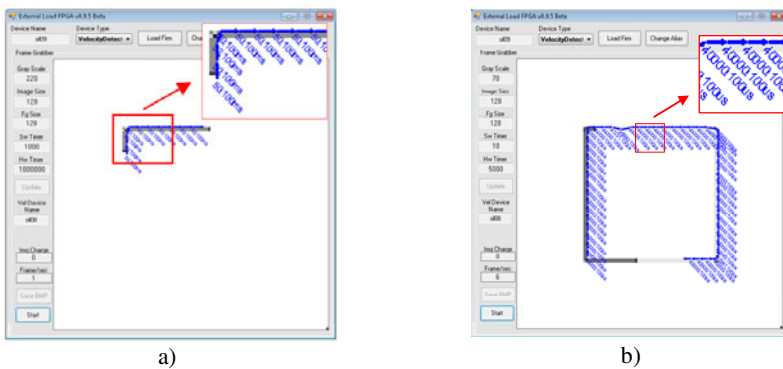


Fig. 6. System's output for synthetic stimulus fused with the reconstruction of the AER sequence: a) 50 pixels/s b) 40000 pixels/s

With this experiment we demonstrate the system's capability for determining the position and estimating the object velocity, even when the object velocity is very high.

3.2 Synthetic Objects

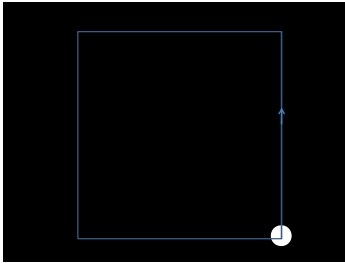
In these experiments we use a silicon retina observing a TFT monitor which is displaying objects moving in the scene at different location and velocities.

Circle at different velocities

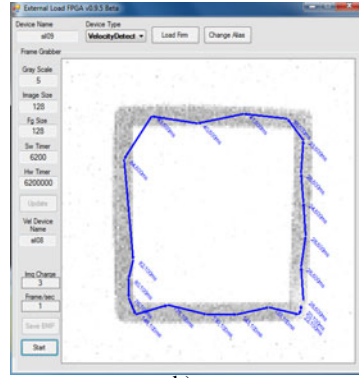
In this experiment a circle starts moving slowly (26.6 pixels/s) from the right bottom corner to the right up corner, it then moves faster to the left up corner (40 pixels/s), then much faster to the left bottom corner (80 pixels/s) and then fastest to the right bottom corner again (160 pixels/s). Fig 7.a shows a scheme of the scene.

Fig. 7.b shows the system's output fused with the reconstruction of the retina output, both accumulated during 6.2s. It can be observed how the system determines the object position and the velocity in every moment. The velocity estimation presents some discrepancies from the true object velocity (up to 25%). This error is due to the

retina imprecision and the speed of the estimation. In this system, real time is more important than accuracy. Furthermore, biological systems do not obtain a quantitative object velocity value, but a qualitative value. This system obtains a very good approximation to the real velocity.



a)



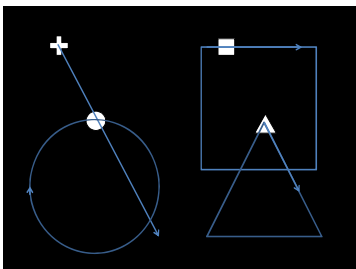
b)

Fig. 7. Second experiment: a) experiment configuration, b) system output fused with silicon retina output

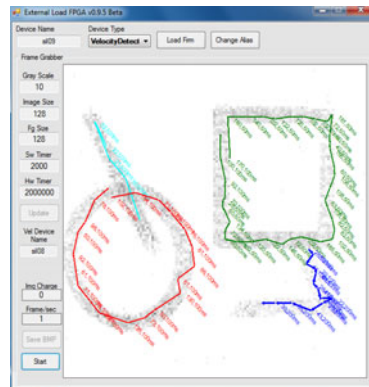
With this experiment we demonstrate the system’s capability for adapting to object velocity in real time, that is, the system can track objects even when the object’s velocity is changing.

Four objects at different velocities with crossing pathway

In this third experiment we present a more complex scene: 4 objects moving at different velocities with crossing pathways. Fig 8.a shows the experiment’s configuration.



a)



b)

Fig. 8. Third experiment: a) experiment configuration, b) system output fused with retina output

Fig.8.b shows the system's output fused with the retina's output reconstruction (accumulated during 2s). Each color represents the output of one TrackCell. Each TrackCell uses a different period for velocity estimation. Once again the object velocity estimation presents some discrepancies with the real velocities: the objects velocities are constant, and the system's output does not show constant velocities, the reason for that was explained above.

With this experiment we demonstrate the system capability for several objects tracking even when the pathways are crossing.

4 Conclusion

This paper presents a neuromorphic system for real-time object tracking using new cascade architecture. The system is fully hardware implemented, described in VHDL; each TrackCell (CMCell+VCell) block only requires 315 slices of a Spartan-II 200 Xilinx FPGA (13.4%). The response time is the minimum possible and depends on the sensor, typically the first object position is obtained after 10 events, and a new position after each new event. The maximum object speed for its tracking and speed prediction is 40000 pixels/s.

Results show that the system can determine the position and estimate the velocity of several objects simultaneously, even in difficult situations: velocity change and crossing pathways. And in ideal situations it can estimate the object velocity with 100% accuracy.

References

- [1] Indiveri, G., Liu, S.-C., Delbruck, T., Douglas, R.: Neuromorphic Systems. In: Squire, L.R. (ed.) *Encyclopedia of Neuroscience*, pp. 521–528. Elsevier Ltd., Amsterdam (2009)
- [2] Mahowald, M.: *VLSI Analogs of Neuronal Visual Processing*: Thesis (1992)
- [3] Delbruck, T., Lichtsteiner, P.: Fast sensory motor control based on event-based hybrid neuromorphic-procedural system. In: *2007 IEEE International Symposium on Circuits and Systems*, pp. 845–848 (May 2007)
- [4] Litzenberger, M., Posch, C., Bauer, D., Belbachir, A.N., Schon, P., Kohn, B., Garn, H.: Embedded Vision System for Real-Time Object Tracking using an Asynchronous Transient Vision Sensor. In: *2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*, pp. 173–178. IEEE, Los Alamitos (2006)
- [5] Paz, R., Gomez-Rodriguez, F., Rodriguez, M., Linares-Barranco, A., Jimenez, G., Civit, A.: Test infrastructure for address-event-representation communications. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) *IWANN 2005*. LNCS, vol. 3512, pp. 518–526. Springer, Heidelberg (2005)
- [6] Lichtsteiner, P., Posch, C., Delbruck, T.: A 128x128 120 dB 15 us Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits* 43, 566–576 (2008)
- [7] Lichtsteiner, P., Posch, C., Delbruck, T.: A 128 X 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. In: *IEEE International Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers*, pp. 2060–2069. IEEE, Los Alamitos (2006)

Performance Study of Software AER-Based Convolutions on a Parallel Supercomputer*

Rafael J. Montero-Gonzalez¹, Arturo Morgado-Estevez¹,
Alejandro Linares-Barranco², Bernabe Linares-Barranco³, Fernando Perez-Peña¹,
Jose Antonio Perez-Carrasco³, and Angel Jimenez-Fernandez²

¹ Applied Robotics Research Lab, Engineering School, University of Cadiz, Spain
C/Chile 1, 11002-Cádiz

{rafael.jesus.montero,arturo.morgado,fernandoperez.pena}@uca.es

² Robotic and Technology of Computers Lab, University of Seville, Spain
Av. Reina Mercedes s/n, 41012-Seville

{alinares,ajimenez}@atc.us.es

³ Institute of Microelectronics of Seville, IMSE-CNM-CSIC, Spain
Calle de los Descubrimientos, Pabellón Pza. de América, 41092-Seville

{jcarrasco,bernabe}@imse-cnm.csic.es

Abstract. This paper is based on the simulation of a convolution model for bio-inspired neuromorphic systems using the Address-Event-Representation (AER) philosophy and implemented in the supercomputer CRS of the University of Cadiz (UCA). In this work we improve the runtime of the simulation, by dividing an image into smaller parts before AER convolution and running each operation in a node of the cluster. This research involves a test cases design in which the optimal parameters are set to run the AER convolution in parallel processors. These cases consist on running the convolution taking an image divided in different number of parts, applying to each part a Sobel filter for edge detection, and based on the AER-TOOL simulator. Execution times are compared for all cases and the optimal configuration of the system is discussed. In general, CRS obtain better performances when the image is divided than for the whole image.

Keywords: AER, convolution, parallel processing, cluster, supercomputer, bio-inspired, AER simulator.

1 Introduction

Nowadays computer systems are increasing their performance looking for the solution of real-world problems using models inspired in biology. These systems, called bio-inspired systems, analyze the operation of parts of the body and try to implement it in a similar manner through electronic and/or computer systems.

Address-Event-Representation systems are composed of sets of cells typically distributed in a matrix that process the information spike by spike in a continuous

* This work has been supported by Spanish Government grant VULCANO (TEC2009-10639-C04-02) of the Minister of Science and Innovation.

way. The information or results of each cell is sent in a time multiplexed strategy using a digital bus, indicating which position is producing the event.

If we represent a black and white image as an array of cells where each pixel value is in gray scale, the white level would correspond to a frequency value determined by allocating the largest amplitude values, higher brightness values. The signal caused by each pixel is transformed into a train of pulses using PFM (pulse frequency modulation) [1].

Based on the interconnection of neurons present on human vision, the continuous state of transmission in a chip is transformed into a sequence of digital pulses (spikes) of a minimum size (of the order of ns) but with an interval between spikes of the order of hundreds of microseconds (us) or even milliseconds (ms). This interval allows time multiplexing of all the pulses generated by neurons into a common digital bus. Each neuron is identified with an address related to its position into the array. Every time a neuron emits a pulse, its address will appear in the output bus, along with a request signal, until acknowledge is received (handshake protocol). The receiver chip reads and decodes the direction of incoming events and issues pulses for the receiving neurons.

One of the operations performed by AER systems, applied to artificial vision, is the convolution. The first operations in the brain cortex consist of convolution for object edges detection, based on calculations of brightness gradients. In the design presented in [2], a system is described where a single convolution processor performs all operations for the whole image.

Based on this idea, and the divide and conquer premise, this paper is arguing that the division of the image into smaller parts before AER convolution processing in parallel will reduce the runtime. With this new design a convolution could be proposed where a multiprocessor system may perform operations in less time.

2 Methodology and Test Cases

The process of experimentation is to verify, through an exhaustive analysis, which would be the different runtimes of the convolution of an image. Each runtime will correspond to different divisions. All division convolutions are performed in parallel, instead of performing the convolution of the whole image.

We have used the Cluster of Research Support (CRS), part of the infrastructure of the UCA, for improving execution times of the simulation tool AER TOOL. In order to run this simulator on CRS we propose a new simulation model parameterized and adapted to running tests in parallel processors.

2.1 Supercomputer CRS (Cluster of Research Support)

The CRS is composed of 80 nodes. Each node has 2 Intel Xeon 5160 processors at 3 GHz with 1.33GHz Front Side Bus. Each processor is Dual Core, so we have 320 cores available. A total of 640GB of RAM memory, 2.4TB of scratch and Gigabit Ethernet communication architecture with HP Procurve switches allow to obtain a peak performance of 3.75 TFLOPS [3].

In terms of software features, to manage distributed work, Condor¹ tool is used. Condor is a job manager system that specializes in calculation-intensive tasks. The coding for the simulation was done using MATLAB and AER TOOL simulator [4] for MATLAB.

Developing this set of tests on a real physical implementation would be highly expensive. The supercomputer CRS provides the possibility of an AER simulation model implementation in parallel with acceptable runtimes, using the software installed and existing libraries.

2.2 Test Image and Successive Divisions

For this simulation we have designed an image in Adobe Photoshop CS, using gray scale, where the pixel having the darkest value will have a value close to 0 and the brightest will be close to 255. The GIF image size is 128x128 pixels of 0-255 gray levels.

The idea of dividing the original image and perform parallel convolution arises from trying to take advantage of distributed processing systems to expedite the process. This involves running a series of tests with different numbers of divisions.

Firstly, we have obtained the process runtimes of the convolution of the original image without divisions. Secondly the image has been divided into 4 parts (64x64 bits each), performing the convolution in a different processor. Then, the sequence has been repeated by 16 divisions (32x32 bits each). Next, using 256 divisions (8x8 bits each), and finally we have concluded with 1024 divisions (4x4 bits each). Conceptually, the operation would be as shown in Fig. 1.

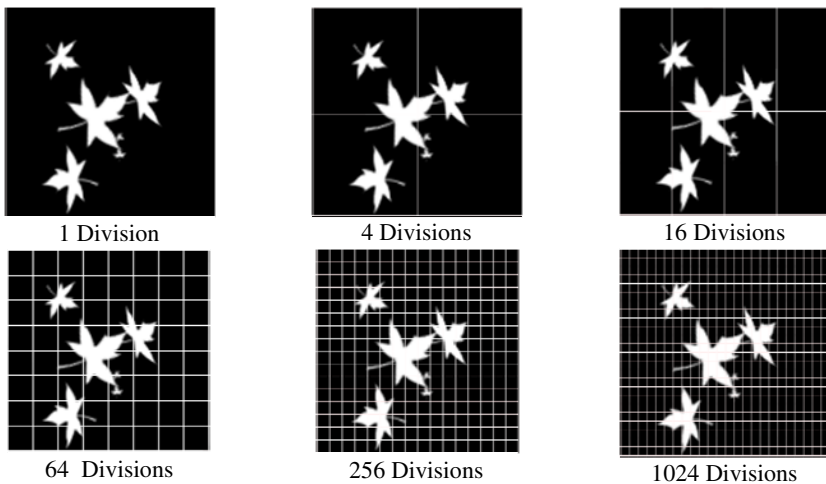


Fig. 1. Divisions of the original image to simulate

¹ <http://www.cs.wisc.edu/condor/>

Edge detection operation by convolution was performed at each division in a different node, estimating that for smaller image size the runtime will be reduced.

2.3 Topology Diagram Implementation

For this research, parametric model simulation software has been developed, whose test cases are specified by variable assignment.

Once the simulation variables are set, the system runs following the block diagram shown in Fig. 2.

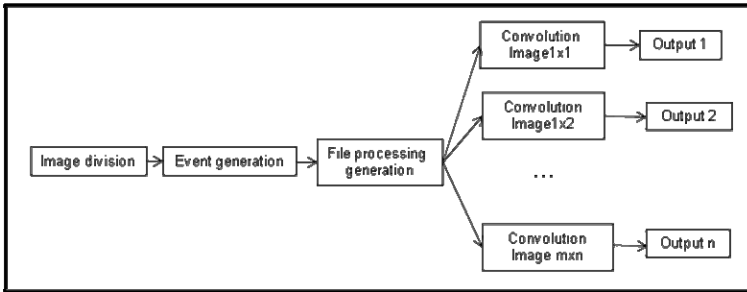


Fig. 2. Simulation block diagram

First, the division of the image is performed using specified parameters. Then, the Uniform method [2] was used for events generation algorithm. When applying this algorithm, a minimum time interval between consecutive events of 0.2 ms and a maximum of 400K events per frame are specified. The next step generates all files necessary for processing the AER TOOL in the CSR cluster. Then, the convolution filter is performed for each division on a different node. Finally, we got as many outputs as image divisions were generated, with the result of applying the operation.

For the convolution filter Sobel edge detection was used in horizontal averaging the diagonal values of a 3x3 size.

$$S = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad (1)$$

Parameters that have been considered for the study are:

- Number of cores: 4, 8, 16 and 32.
- Number of divisions of the image:
 - 1 image of 128x128 pixels.
 - 4 divisions of 64x64 pixels.
 - 16 divisions of 32x32 pixels.
 - 64 divisions of 16x16 pixels.
 - 256 divisions of 8x8 pixels.
 - 1024 divisions of 4x4 pixels.
- Convolution matrix: Sobel of 3x3.

Once we have recorded the runtimes of each stage, analyzed the graph generated and detected the highest peak on the surface, we can indicate the optimal design for the system.

3 Results and Discussion

CSR cluster is a shared computational resource at UCA. Execution times may depend on the cluster workload and users. A variation in the order of milliseconds has been detected. In order to minimize these undesirable situations we have selected a low workload day (Saturday) and a reduced number of nodes respect to the maximum available number of nodes in the cluster. Tests were performed 3 times and the averaged execution times are represented in tables 1- 4 and their respective figures.

The test took place on 9/10/2010 with a workload of 30% consumed by other 9 users running their own independent application of this test.

Processing time for each stage and the total can be seen in the following tables, expressing all the time in seconds for each number of nodes. Data movement time haven't considered because it's not possible to access using user privileges in cluster management.

Table 1 presents both the event generation and the convolution execution times for selected image divisions and using 4 nodes (16 cores) of the CSR. It can be observed that there is no significant difference for 1 or 4 divisions. Nevertheless, for 64 or 256 divisions, runtimes are doubled and a significant difference for 1024 divisions can be seen. However, when generating events it can be seen that the lower is the number of divisions, the higher is the execution time, except for 1024 divisions. In the case of parallel execution it can be seen that leaving the image on its original size and dividing it into 4 pieces of 64x64 has a significant time difference too. It can be also observed that there is a runtime increment for 64 image divisions. For the total runtime (table 5, 4 nodes column), the best execution times correspond to 64 divisions.

Table 2 presents corresponding runtime results when tasks are scheduled for 8 nodes of the cluster. Now it can be seen that runtimes are improved in general terms, but these results do not imply significant changes. For the image division task, the lowest execution time remains for 1 division. For the event generation task, the lowest result is obtained for 256 divisions. And for the convolution task, runtime is also the lowest for 4 divisions, like for 4 nodes.

In Table 3 runtime results correspond to the use of 16 nodes of the cluster. Image division task has similar results than for lower number of nodes. Event generation task runtime offers significant changes for 8x8 blocks (when divided into a total of 256 images), but their convolution runtimes do not produce improvements. In the parallel execution of convolutions, it is found that 64x64 divisions have reduced runtime. For 32x32 and 16x16 images runtime is very similar, but when you have 8x8 images runtime increases. This increment is due to the coordination of a large number of processors in the cluster that requires more data traffic between them, resulting in an overall implementation delay.

In Table 4 results are presented when 32 nodes of the cluster are used. Image division task runtime and event generation runtime show similar results to those presented for 16 nodes. Parallel convolution task runtimes are improved for 4, 8 and 16 divisions. Therefore increasing the number of nodes working in parallel does not imply runtimes reduction, but for larger number of divisions, runtimes also increase, starting at dawn when they are 64 divisions of 16x16 pixels and shooting when divisions reach the 1024 block of 4x4 pixels.

Table 1. Runtimes summary for 4 nodes

N. div	Tdiv img	Tgener. Even.	Runtime. paral
1	24 ms	413,9 s	5,6 s
4	26 ms	101,1 s	5,3 s
16	35 ms	59,1 s	7,3 s
64	40 ms	38,8 s	13,1 s
256	42 ms	20,7 s	37,6 s
1024	94 ms	25,7 s	135,5 s

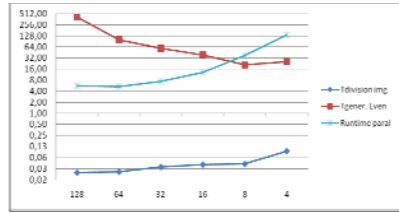


Table 2. Runtimes summary for 8 nodes

N. div	Tdiv img	Tgener. Even.	Runtime paral
1	25 ms	395,6 s	9,3 s
4	30 ms	109,8 s	6,6 s
16	31 ms	62,8 s	7,6 s
64	30 ms	38,8 s	17,0 s
256	43 ms	20,9 s	27,2 s
1024	95 ms	25,8 s	81,5 s

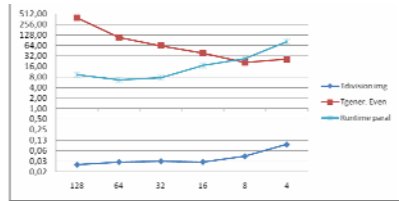


Table 3. Runtimes summary for 16 nodes

N. div	Tdiv img	Tgener. Even.	Runtime paral
1	26 ms	1199,2 s	8,3 s
4	26 ms	171,9 s	5,9 s
16	26 ms	108,9 s	17,4 s
64	28 ms	41,9 s	19,3 s
256	43 ms	22,7 s	34,8 s
1024	96 ms	31,3 s	76,8 s

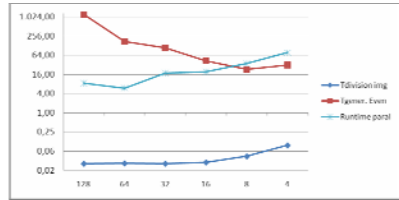
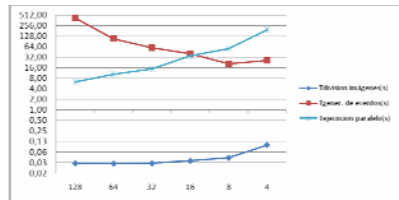


Table 4. Runtimes summary for 32 nodes

N. div	Tdiv img	Tgener. Even.	Runtime paral
1	30 ms	423,7 s	6,3 s
4	30 ms	107,7 s	10,5 s
16	30 ms	59,9 s	14,9 s
64	35 ms	41,0 s	34,8 s
256	43 ms	20,8 s	57,0 s
1024	100 ms	25,8 s	195,6 s



If we represent the total runtime with respect to the maximum number of nodes and the number of divisions, we get Table 5, noting the lowest total runtime shaded.

If instead of using the total runtime, we take the parallel runtime and we represent it in the same domain as Table 5, we obtain Table 6, noting the minimum runtime shaded.

It can be highlighted the case of 4 nodes and 4 divisions of 64x64 pixels blocks which have a faster execution, but not much different block sizes with the 32 or 128.

Table 5. Summary of total runtime as the number of divisions and the number of nodes

N div	4 nodes	8 nodes	16 nodes	32 nodes
1	420 s	405 s	1213 s	430 s
4	107 s	117 s	178 s	118 s
16	67 s	71 s	127 s	75 s
64	52 s	56 s	62 s	76 s
256	59 s	49 s	59 s	79 s
1024	165 s	111 s	113 s	225 s

Table 6. Summary of parallel runtime depending on the number of divisions and the number of nodes

N div	4 nodes	8 nodes	16 nodes	32 nodes
1	5,6 s	9,3 s	8,3 s	6,3 s
4	5,3 s	6,6 s	5,9 s	10,5 s
16	7,3 s	7,6 s	17,4 s	14,9 s
64	13,1 s	17,0 s	19,3 s	34,8 s
256	37,6 s	27,2 s	34,8 s	57,0 s
1024	135,5 s	81,5 s	76,8 s	195,6 s

4 Conclusions

In this work we have designed a test case set for AER convolution processing on a supercomputer, the CSR cluster of UCA, Cadiz, SPAIN. We have executed and compared all the test cases. If we rely on the data obtained we obtain the following conclusions:

- Referring to the data in Table 5, we can see that the total runtime minor by running a maximum of 8 nodes in parallel and 256 divisions of the image into blocks of 8x8 bits. This case is very similar to the case of a maximum of 8 nodes in parallel and 64 divisions of the image into blocks of 16x16 bits. Then, the two implementations would be valid for our system.
- If we look at the data in Table 6, in which only the parallel runtimes are shown, we see that the test case for a maximum of 4 nodes with 4 divisions of 64x64 bits of the image, obtained lower runtimes.
- If we consider that, in a hardware implementation the event generation time disappears when taking images directly from an acquisition event-based system (i.e. silicon retina), the best option is to have 4 nodes in parallel with 4 divisions of 64x64 bits.

This work represents the first steps on the execution of more complex AER system simulations on the cluster, which will improve considerably the performance of parameters adjustment of hierarchical AER systems where several convolution kernels work together in a multilayer system for more complex tasks as face recognition, etc, already illustrated in [7].

Acknowledgments. Andrés Yañez Escolano, member of Languages and Computing Systems Department at University of Cádiz, Spain, and Abelardo Belaustegui González, member of Computing System Area at University of Cádiz, Spain and main manager CRS.

References

1. Serrano-Gotarredona, T., Linares-Barranco, A.G., Andreou, B.: AER image filtering architecture for vision-processing systems. *Circuits and Systems I. IEEE Transactions on Fundamental Theory and Applications* 46, 1064–1071 (1999)
2. Linares-Barranco, A., Jimenez-Moreno, G., Linares-Barranco, B., Civit-Balcells, A.: On algorithmic rate-coded AER generation. *IEEE Transactions on Neural Networks* 17, 771–788 (2006)
3. Technical support in supercomputing, University of Cadiz, <http://supercomputacion.uca.es/>
4. Pérez-Carrasco, J.-A., Serrano-Gotarredona, C., Acha-Piñero, B., Serrano-Gotarredona, T., Linares-Barranco, B.: Advanced vision processing systems: Spike-based simulation and processing. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) *ACIVS 2009. LNCS*, vol. 5807, pp. 640–651. Springer, Heidelberg (2009)
5. Lujan-Martinez, C., Linares-Barranco, A., Rivas-Perez, M., Jimenez-Fernandez, A., Jimenez-Moreno, G., Civit-Balcells, A.: Spike processing on an embedded multi-task computer: Image reconstruction. In: *Fifth Workshop on Intelligent Solutions in Embedded Systems 2007*, pp. 15–26 (2007)
6. Camunas-Mesa, L., Acosta-Jimenez, A., Serrano-Gotarredona, T., Linares-Barranco, B.: Fully digital AER convolution chip for vision processing. In: *IEEE International Symposium on Circuits and Systems, ISCAS 2008*, pp. 652–655 (2008)
7. Perez-Carrasco, J.A., Acha, B., Serrano, C., Camunas-Mesa, L., Serrano-Gotarredona, T., Linares-Barranco, B.: Fast Vision Through Frameless Event-Based Sensing and Convolutional Processing: Application to Texture Recognition. *IEEE Transactions on Neural Networks* 21, 609–620 (2010)
8. Dominguez-Castro, R., Espejo, S., Rodriguez-Vazquez, A., Carmona, R.: A one-transistor-synapse strategy for electrically-programmable massively-parallel analog array processors. In: *2nd IEEE-CAS Region 8 Workshop on Analog and Mixed IC Design*, pp. 12–13, 117–122 (1997)

Frequency Analysis of a 64x64 Pixel Retinomorph System with AER Output to Estimate the Limits to Apply onto Specific Mechanical Environment*

Fernando Perez-Peña¹, Arturo Morgado-Estevez¹,
Alejandro Linares-Barranco², Gabriel Jimenez-Moreno²,
Jose Maria Rodriguez-Corral¹, and Rafael J. Montero-Gonzalez¹

¹ Applied Robotics Research Lab, University of Cadiz, C/Chile 1, 11002-Cadiz, Spain
{fernandoperez.pena, arturo.morgado, rafaeljesus.montero,
josemaria.rodriguez}@uca.es

² Robotic and Technology of Computers Lab, University of Seville, Av. Reina Mercedes s/n,
41012-Seville, Spain
{alinares, gajil}@atc.us.es

Abstract. The rods and cones of a human retina are constantly sensing and transmitting the light in the form of spikes to the cortex of the brain in order to reproduce an image in the brain. Delbruck's lab has designed and manufactured several generations of spike based image sensors that mimic the human retina. In this paper we present an exhaustive timing analysis of the Address-Event-Representation (AER) output of a 64x64 pixels silicon retinomorph system. Two different scenarios are presented in order to achieve the maximum frequency of light changes for a pixel sensor and the maximum frequency of requested directions on the output AER. Results obtained are 100 Hz and 1.66 MHz in each case respectively. We have tested the upper spin limit and found it to be approximately 6000rpm (revolutions per minute) and in some cases with high light contrast lost events do not exist.

Keywords: Bio-inspired, Spike, Retinomorph Systems, Address Event Representation.

1 Introduction

The human retina is made up of several layers. The first one is based on rods and cones that capture light. The following two additional layers of neurons are composed of different types of cells [1]. Next one layer is composed of bipolar, horizontal and amacrine cells. Each cell has specific skills; the horizontal cells implement a previous filter that has an inhibitory affect on the photoreceptors when light is shone onto them, the bipolar cells are responsible for the graded potentials generation. There are two different types of bipolar cells, ON cells and OFF cells that produce on and off graded potentials. The last type of cells of this layer are the amacrines, they connect

* This work was supported by the Spanish grant VULCANO (TEC2009-10639-C04-02).

distance bipolar cells with ganglion cells. The last layer of the retina is composed by ganglion cells. They are responsible for the action potentials or spikes generation. Information travels along the optic nerve that is composed by the axons of ganglion cells.

Delbruck’s silicon retina [2] consists of a set of 64x64 temporal differential light sensors that mimic the behavior of bipolar cells. There are several researchers from different labs and countries working on this type of bio-inspired and spike-based systems. They are called neuromorphic engineers. Telluride Neuromorphic Engineering Workshop and CapoCaccia Cognition Neuromorphic Engineering Workshops [3][4] are events where these researchers present and interchange ideas and results. This type of systems is called retinomorphic and they have a pixels structure. Each pixel should copy the behavior of one bipolar cell plus the sensing light.

This type of system was firstly proposed at 1988 by Mead and Mahowald [5] with an analog model of a pixel. But it was in 1996 when Kwabena Boahen presented two works [6][7] that established the basis for the silicon retinas and their communication protocol. After them, Culurciello [8] described a gray level retina with 80x60 pixels and a high level of response with AER output. The most important fact in all these works is the design of the spikes generator. Other way to design retinas, apart from AER, is with visual microprocessors based on the cellular neural network universal machine. A review of this type of designs could be find at [9].

In this paper we have used the 64x64 Delbruck retina developed under the EU project CAVIAR. This retina uses an Address Event Representation (AER) communication strategy (Fig. 1). If any pixel of the retina needs to communicate a spike, an encoder assigns a unique address to it and then this address will be put onto the bus using a handshake protocol. AER was proposed by Mead lab in 1991 [10] as an asynchronous communication protocol for inter neuromorphic chips transmissions. Two handshake lines of request (REQ) and acknowledge (ACK) are managing the communication.

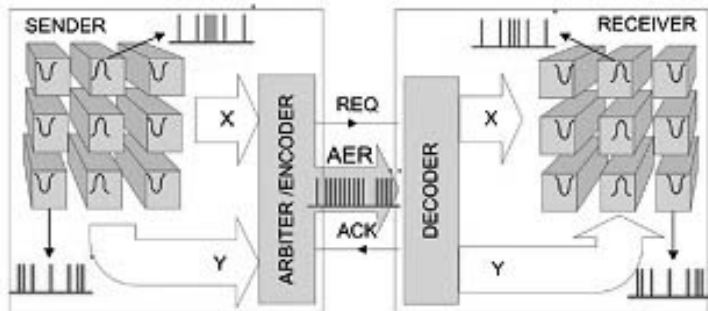


Fig. 1. Spiking pixels using AER communication between neuromorphic chips

2 64x64 AER Retina Chip

We used a silicon bio-inspired retina of 64x64 pixels (Fig. 3) designed by P. Lichtsteiner and T. Delbruck at Neuroinformatics Institute at Zurich [2]. This retina generates events corresponding to changes in log intensity, so static scenes do not produce any output. For this reason, each pixel has two outputs, ON and OFF events or two directions if we look through AER. The type of the event depends on the sign of the derivative of the light evolution respect to the time; if there is a positive change of light intensity within a configurable period of time a positive event should be transmitted. Consequently if it is a negative change of light intensity, then a negative event appears on the output AER bus. Due to the sign of the events, the address space used by this retina goes from 0 to 8191; although only 4096 addresses could be spiking. The frequency of output events is proportional to the light amplitude changes. The bigger the light intensity change, the more output events are produced. This frequency of output events can be adjusted through available bias, but in that case the activity of idle pixels is also increased. We have configured the retina in order to reduce the AER traffic of those pixels with no intensity changes, which could imply a decrease of the output frequency of spikes for a particular pixel whose intensity is changing.

At Delbruck's paper [2] there are several tests to characterize the retina but we need to know the behavior at the worst condition in order to use the retina at any industrial application (the aim of Spanish Project VULCANO). That kind of environment typically requires detecting and producing a decision taking and an action to really fast moving or rotating objects. It is very important to know exactly the maximum detected change of pixel light in the AER retina in order to determine the maximum frequency of rotation for a particular object. It is also important to know if there is any lost event at those frequencies. With those data it is possible to determine the best rpm observed for any kind of industrial machinery.

At CAVIAR project [11] a standard for the AER protocol was defined by Häfliger. This standard defines a 4-step asynchronous handshake protocol (Fig. 2 and Table 1).

Next section presents the experimental methodology to calculate exactly the times expressed at Häfliger standard. With those times and the number of pixels spiking at the same time, it is possible to determine the bandwidth limit of produced events and, if present, the percent of lost events.

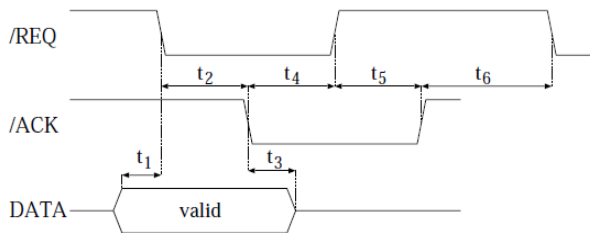


Fig. 2. Timing for a valid AER transmission with a 4 phase handshake protocol. Single sender and receiver [11].

Table 1. Timing requirements of a 4-step asynchronous handshake protocol (Fig.2)

Times	min	max	Avg
t_1	0s	∞	
t_2	0s	∞	≤ 700 ns
t_3	0s	∞	
t_4	0s	100 ns	
t_5	0s	100 ns	
t_6	0s	∞	

3 Experimental Methodology

In this section we present and describe two different methods in order to extract the bandwidth limit and the percent of lost events.

We have used several AER-tools for these experiments. We have used the jAER viewer and Matlab functions, available at the jAER wiki [12]. Furthermore, a logic analyzer from manufacturer Digiview (Model DVS3100) (Fig. 3) has been used.

3.1 Environment

In order to make the tests two different scenarios have been used.

The first one described in Fig. 3 is mainly composed of a mechanical drill. The reason to use this type of mechanical tool is because they provided a huge margin of spin frequency. This fact allows us to compare the spin frequency and the maximum frequency of one pixel, which is our first goal.

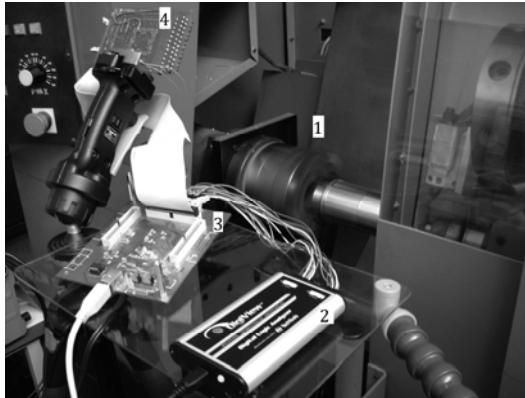


Fig. 3. Picture of the scenario prepared for the first test. Components are 1. Mechanical Drill, 2. Logic Analyzer, 3. Sequencer Monitor AER and 4. 64x64 pixels retina.

For the second test, we have taken advantage of the problem from the typical environment of a laboratory. The fluorescent tube makes all the objects at the room change their luminosity with the power network frequency (50 Hz at Spain). For us, that frequency is not visible, but the retina notice it, so it is possible to achieve that all

the pixel spiking focus the retina on the tubes. With this scenario, the logic analyzer will show the proper times of each spike and the Häfliger times could be resolved.

The Sequencer Monitor AER board called USB2AER, has been used in both experiments. This board consists of two main components, the Cypress FX2LP USB2.0 transceiver and a Xilinx Coolrunner 2 CPLD with 256 macroblocks of digital logic (XC2C256). It has three 16 bit word-parallel AER ports following a 4 phase handshake protocol (2 additional REQ and ACK signals). The power consumption is 60 mA while monitoring and sequencing and the sustained bandwidth of the board is 5 Meps (Mega Events per second) [13].

3.2 Maximum Spike Frequency

For this test we have assembled the retina to an USB2AER Monitor Sequencer Interface [13] that connects an AER bus to the computer sending packets of USB composed of sequences of Address-Events and timestamps that indicate the time instant when the event was coming from the AER retina.

In order to determine the frequency it is necessary to focus on a few pixels of the retina. To obtain this response of the retina we have stimulated it with a high range of frequency of the mechanical drill. The range goes from 0Hz to 100Hz (which is equivalent to a mechanical range from 0rpm to 6000rpm). The assembly appears at Fig. 3. The retina has been placed so the drill is stimulating just a few pixels of the retina. These pixels are producing output AER traffic which frequency depends on the drill spin frequency.

Using the Java application associated to the USB2AER monitor available at Sourceforge, called jAER viewer, we have captured a sequence of AER. Under Matlab we have extracted the most repetitive addresses and we have processed them in order to know which pixels are spiking. Additionally, all directions were ordered and the most repetitive direction was obtained. Also, we have looked for that direction inside the information from logic analyzer and studied the sampled frequency for that pixel for each spin frequency of the drill.

Note that, for high frequency luminosity changes, between two consecutive events of one pixel, all the other active pixel events should fit.

3.3 Maximum Frequency of Requested Directions

For the final aim of this test we cannot use the AER monitor board because its USB interface will limit the bandwidth peak of events to the size of the buffer and clock speed.

To determine this maximum frequency on the output AER bus of the retina it is necessary to illuminate all pixels in order to saturate the arbiter inside the retina that is managing the writing operation of events on the AER bus. So firstly we need to connect the retina to the AER monitor in order to check that the whole retina is illuminated and, therefore, all the pixels are producing events. Then the AER monitor is disconnected and the retina is connected to the logic analyzer with a jumper connecting the request and acknowledge signals. The aim is to calculate the time expressed by the standard of Häfliger.

Captured data by logic analyzer has been processed with Matlab in order to determine the minimum, medium and maximum inter-spikes-interval times.

4 Results and Discussions

The results obtained for the first testing scenario are shown in Fig. 4. It shows the evolution of spike frequency for the most repetitive direction calculated in front of the spin frequency of the drill expressed in rpm.

The graph shows that when the drill is quiet, there is no output frequency of events. So it can be said that the minimum frequency is 0 Hz. With increments at the spin frequency the spiking frequency level increases up to 100Hz which is the saturation level.

Using Matlab, we have fitted the values obtained within a linear polynomial regression. The coefficient of determination r-square of the regression is 0.7684. It shows a quite reliable approximation of the trend. The regression line is represented as a discontinuous line at the graph.

The nonlinear and non systematic behavior could be explained from two points of view: on the one hand is the fact that at those kind of industrial machinery with high rpm there is a process to stabilized the head and that could provoke some wrong values; in the other hand is that we have just choose the most repetitive pixel instead of an average of a few of them.

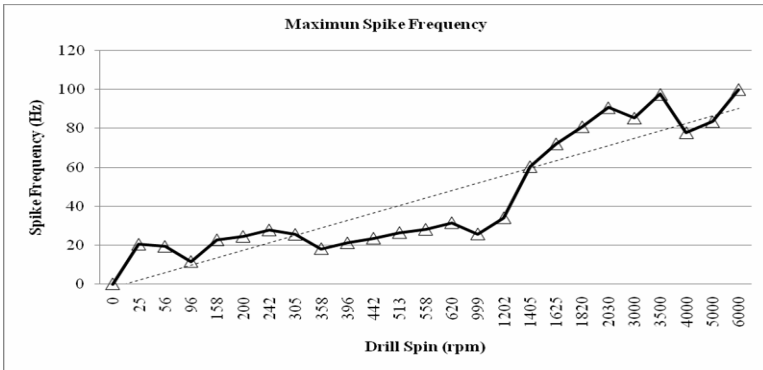


Fig. 4. Maximun spike frequency evolution for the spike frequency in front of the spin of the drill expressed at rpm

We have raised, step by step, the spin of the drill from 6000 rpm to 7000 rpm in order to test the retina. In this range, the target began to disappear from the retina view. This is the empiric limit for this retina.

Another result of this analysis should be highlighted: if the maximum frequency is 100 Hz, it is necessary to fit the 4096 addresses within this 10 ms (Fig. 5) in order to aim no miss events.

In both trials, the times by Häfliger standard have been obtained as it is shown in table 2:

Table 2. Comparative timing table from the obtained at trials and defined by Häfliger standard

Times	Häfliger	Laser Trial	Fluorescent Trial
t_1	(0 - ∞)	10 ns	200 ns
t_2	(0 - ∞)	60 ns	30 ns
t_3	(0 - ∞)	-	-
t_4	(0 - 100 ns)	990 ns	60 ns
t_5	(0 - 100 ns)	20 ns	60 ns
t_6	(0 - ∞)	370,16 μ s	470 ns

Note that t_3 is included in t_4 and it is impossible to be measured because the valid address is still on the bus until the next one arrives.

At the fluorescent trial we were looking for the maximum frequency of any requested address. That is the inverse of consecutive request times and the same as the sum of t_2 , t_4 , t_5 and t_6 that result 1.66 MHz (Fig.5).

If we join together the 10 ms obtained at the drill scenario between two consecutive events of the same pixel, that could be called time frame, and $t_2+t_4+t_5+t_6$ obtained on the tubes scenario between any two consecutive events, a maximum number of events could appear within these 10ms, as shown in Fig. 5.

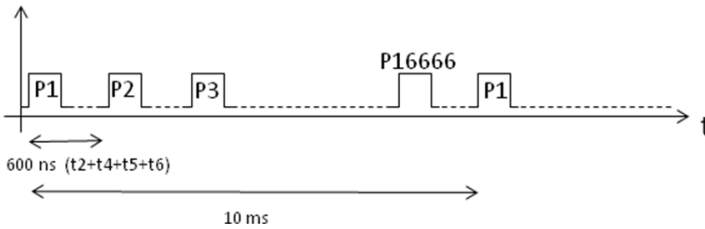


Fig. 5. Time Frame in the worst condition of luminosity change for the retinomorph system composed of a 64x64 pixels retina

Note that, with these timings, up to 16666 addresses could be placed within the time frame. If we had considered an address space of 4096 pixels, it would have confirmed no lost events. This situation has appeared in high luminosity change conditions, which shows the excellent behavior for an industrial application.

5 Conclusions

In this paper we have presented a study for the timing and limits of a retinomorph system composed of a 64x64 pixels retina in an industrial application. Two scenarios were assumed to test the retina; one to determine the maximum spike frequency and the other one for the maximum request frequency. With those results we have checked the upper limit of spin drill to be approximately 6000 rpm. Also, the results shown in Fig. 5 reveal that in the worst condition of luminosity change for our retina

there will be no lost events. Therefore, this AER retina can be used on industrial applications that do not require pixels changing at frequencies higher than 100Hz and that do not produce AER bandwidths higher than 1.66Meps for the present analog bias configuration of the AER retina.

References

1. Linsenmeier, R.A.: Retinal Bioengineering. In: He, B. (ed.) *Neural Engineering. Bioelectric Engineering*, vol. 3, pp. 421–484. Springer, US (2005)
2. Lichtsteiner, P., Delbruck, T.: A 64x64 AER logarithmic temporal derivative silicon retina. *Research in Microelectronics and Electronics* 2, 202–205 (2005)
3. The 2011 Cognitive Neuromorphic Engineering Workshop, <http://capocaccia.ethz.ch/capo/wiki/2011/>
4. Cohen, A., et al.: Report to the National Science Foundation: Workshop on Neuromorphic Engineering, Telluride, Colorado, USA (June-July 2004), <http://www.ini.unizh.ch/telluride>
5. Mead, C.A., Mahowald, M.A.: A Silicon Model of Early Visual Processing. *Neural Netw.* 1, 91–97 (1988)
6. Boahen, K.A.: A Retinomorph Vision System. *IEEE Micro.* 16, 30–39 (1996)
7. Boahen, K.A.: Retinomorph Vision Systems. I. Pixel Design. In: *IEEE International Symposium on Circuits and Systems, Suppl.*, Piscataway, NJ, United States, pp. 9–13 (1996)
8. Culurciello, E., Etienne-Cummings, R., Boahen, K.A.: A Biomorph Digital Image Sensor. *IEEE Journal of Solid-State Circuits* 38, 281–294 (2003)
9. Roska, T., Rodriguez-Vazquez, A.: Toward Visual Microprocessors. *Proceedings of the IEEE* 90, 1244–1257 (2002)
10. Sivilotti, M.: Wiring Considerations in Analog VLSI Systems with Application to Field-Programmable Networks. Ph.D. Thesis, California Institute of Technology, Pasadena CA (1991)
11. Serrano-Gotarredona, R., Oster, M., Lichtsteiner, P., et al.: CAVIAR: A 45k Neuron, 5M Synapse, 12G connects/s AER Hardware Sensory-Processing-Learning-Actuating System for High-Speed Visual Object Recognition and Tracking. *IEEE Trans. Neural Networks* 20, 1141–1438 (2009)
12. jAER Open Source Project, <http://jaer.wiki.sourceforge.net/>
13. Berner, R., Delbruck, T., Civit-Balcells, A., et al.: A 5 Meps \$100 USB2.0 Address-Event Monitor-Sequencer Interface. In: *IEEE International Symposium on Circuits and Systems, ISCAS*, New Orleans, LA, pp. 2451–2454 (2007)

An AER Spike-Processing Filter Simulator and Automatic VHDL Generator Based on Cellular Automata*

Manuel Rivas-Perez¹, A. Linares-Barranco¹, Francisco Gomez-Rodriguez¹,
A. Morgado², A. Civit¹, and G. Jimenez¹

¹ Robotic and Technology of Computers Lab. University of Seville. Spain

² Computer Architecture and Technology Area, University of Cadiz. Spain

mrvivas@atc.us.es

Abstract. Spike-based systems are neuro-inspired circuits implementations traditionally used for sensory systems or sensor signal processing. Address-Event-Representation (AER) is a neuromorphic communication protocol for transferring asynchronous events between VLSI spike-based chips. These neuro-inspired implementations allow developing complex, multilayer, multichip neuromorphic systems and have been used to design sensor chips, such as retinas and cochlea, processing chips, e.g. filters, and learning chips. Furthermore, Cellular Automata (CA) is a bio-inspired processing model for problem solving. This approach divides the processing synchronous cells which change their states at the same time in order to get the solution. This paper presents a software simulator able to gather several spike-based elements into the same workspace in order to test a CA architecture based on AER before a hardware implementation. Furthermore this simulator produces VHDL for testing the AER-CA into the FPGA of the USB-AER AER-tool.

Keywords: spiking neurons, address-event-representation, usb-aer, vhdl, fpga, image filtering, neuro-inspired, cellular automata.

1 Introduction

Cellular organization in biology has been an inspiration in several fields, such as the description and definition of Cellular Automata (CA). They are discrete models that consist of a regular grid of cells. Each cell has an internal state which changes into discrete steps and knows just one simple way to calculate the new internal state like a rudimentary automaton. Cellular activity is carried out simultaneously like it occurs in biology. Von Neumann referred to this system as a Cellular Space that is known currently as Cellular Automata [1].

The first self-reproducing CA, proposed by von Neumann consisted of a 2D grid of cells, and the self-reproducing structure was composed of several hundreds of elemental cells. Each cell presented 29 possible states [2]. The evolution rule was

* This work was supported by the Spanish grant VULCANO (TEC2009-10639-C04-02).

defined as a function of current state of the cell and its neighbors (up, down, right and left). Due to the high complexity of the model, von Neumann rule has never been implemented in hardware, but some partial implementations have been obtained [3].

A CA hardware implementation consists of a regular 2D array of cells. Each cell is connected to a neighborhood. The state of each cell is defined by a set of bits and varies longitudinally according to an evolution rule. This evolution rule should be the same for all the cells and it is a function of the current internal state of the cell and its neighborhood [1], so it does not depend on external stimulus. These neighbors are a fixed set of cells adjacent to the specified cell. A new generation is created every time the rule is applied to the whole grid. A global clock signal sets when the state of the cell is updated.

Address-Event- Representation (AER) is a spike-based representation technique for communicating asynchronous spikes between layers of silicon neurons or spike-processing cells of different chips. The spikes in AER are carried as addresses of neurons (called events) on a digital bus. This bio-inspired approach was proposed by the Mead lab in 1991 [4].

The spikes in AER are carried as addresses of sending or receiving cells on a digital bus. Time represents itself as the asynchronous occurrence of the event. An arbitration circuit ensures that neurons do not access the bus simultaneously. This AER circuit is usually built using self-timed asynchronous logic [5].

Every time a cell generates a spike, a digital word (address) which identifies the cell, is placed on an external bus. A receiver chip connected to the external bus receives the event and sends a spike to the corresponding cell. In this way, each cell from a sender chip is virtually connected to the respective cell in the receiver chip through a single time division multiplexed bus (See figure 1).

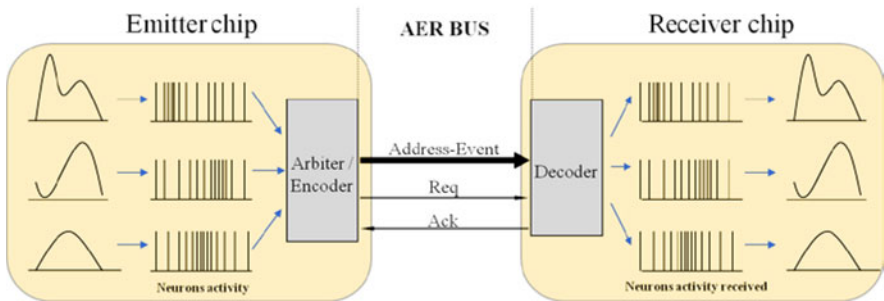


Fig. 1. AER inter-chip communication scheme

More active cells access the bus more frequently than less active ones, for example, if the AER information transmitted by a visual AER sensor is coded in gray levels, then the number of events transmitted by a pixel through the bus identifies the gray level of that pixel.

These AER chips with adequate AER interfaces allow the construction of multilayered, hierarchical, and scalable processing systems for visual processing (CAVIAR EU project [6]).

There is a world-wide community of AER protocol engineers and researchers for bio-inspired applications in vision and audition systems and robot control, as it is

demonstrated by the success in the last years of the AER group at the Neuromorphic Engineering Workshop series [7] and the CapoCaccia Cognitive Neuromorphic Engineering Workshop [8]. The goal of this community is to build large multi-chip and multi-layer hierarchically structured systems capable of performing massively-parallel data-driven processing in real time.

One of the first processing layers in the cortex consists of applying different kinds of convolution filters with different orientations and kernel sizes. Complex filtering processing based on AER convolution chips have been already implemented, which are based on Integrate and Fire (IF) neurons [9]. When an event is received, a convolution kernel is added to the neighborhood of the targeted neuron. When a neuron reaches a configurable threshold, a spike is produced and the neuron is reset. Bi-dimensional image convolution is defined mathematically by the following equation, being K an $n \times m$ convolution kernel matrix, X the input image and Y the convolved image.

$$\forall_{i,j} \rightarrow Y(i,j) = \sum_{a=-n/2}^{n/2} \sum_{b=-m/2}^{m/2} K(a,b) \cdot X(a+i,b+j)$$

Each convolved image pixel $Y(i,j)$ is defined by the corresponding input pixel $X(i,j)$ and weighted adjacent pixels, scaled by K coefficients. Therefore an input pixel $X(i,j)$ contributes to the value of the output pixel $Y(i,j)$ and their neighbors, multiplied by the corresponding kernel coefficients K .

Digital frame-based convolution processors implemented in FPGA or CPUs usually measure their performance by calculating the number of operations per second (MOPS). There is a comparative study between frame-based and spike-based convolution processors in [10]. In that work, a frame-based 3×3 kernel convolution processor in a Spartan-III FPGA that yielded 139 MOPS, were compared to a spike-based one that yielded 34.61 MOPS for the same kernel. Nevertheless, frame-based 11×11 kernel convolution processors decreased their performance to 23 MOPS, while the spike-based processors increased their performance to 163.51 MOPS. Therefore, spike-based convolution processors may achieve higher performances for the same hardware availability. This has to be thanked to the fully parallel processing allowed by AER or spike-based processing.

In a previous work [11] we presented an AER-CA 3×3 convolution processor for visual spike-processing running on a SPARTAN-II FPGA at 50MHz, able to yield up to 150 MOPS for 3×3 kernel sizes, which could imply a performance of up to 2 GOPS for a possible 11×11 kernel implementation. This work justified the potential of AER-CA implementations of spike-based processing.

Another interesting approach for frame-based neuro-inspired visual processing, and based on convolutions with high performances are the ConvNets [12][13]. ConvNets are based on convolutional neural networks and have been successfully used in many recognition and classification tasks including document recognition [14], object recognition [15][16][17], face detection [18] and robot navigation [19]. A ConvNet consists of multiple layers of filter banks followed by non-linearities and spatial pooling. Each layer takes as input the output of previous layer and by combining multiple features and pooling over space, extracts composite features over a larger input area. Once the parameters of a ConvNet are trained, the recognition operation is performed by a simple feed-forward pass. In [20] a hardware

implementation of a 7x7 kernel size convolver for ConvNets is presented. This system was synthesized for a Virtex 4 and it achieves up to 12GOPS with a 250 MHz clock that is equivalent to 2.4GOPS for 50MHz clock.

In order to study and develop the correct configuration of spike-based convolutional neural networks for a visual processing task based on CA and AER, it is very important to be able to make simulations evolving several AER-CA convolution processors in a network with different kernels.

In this paper we present an AER-CA simulator developed under C# for spiking convolutional neural networks that is able to generate VHDL for FPGA hardware implementation on USB-AER boards [21].

Next section introduces how AER and CA can work together for spike-based visual processing. Section 3 presents an AER simulator with an example for center of mass object detection. Finally the conclusions are presented in section 4.

2 AER Processing Based on Cellular Automata

AER neurons carry out an internal processing for every arriving spike and can produce an output spike or stream of spikes in response. AER chips develop hierarchical systems composed of layers of neurons like a brain. Results of one layer represent the input of the next layer or a feedback of a previous one. Furthermore, like in a biological neural system, several AER devices such as visual sensors (retina [22][25]), audio sensors (cochlea [23]), filters [24][27] and learning chips [26] have been developed, as well as a set of glue tools (AER tools[21]) which facilitate developing and debugging of these spike-based multi-layer hierarchical systems, like under the EU CAVIAR project [6].

The basic operation in visual processing is the mathematical convolution. In the previous section we introduced how a convolution is done using spike-based visual information. When a set of these spike-based convolution processors are connected in a network, more complex visual filtering can be implemented, like in the brain cortex.

The philosophy of AER systems is lightly different from CA but also similar in a certain sense. A CA is a cooperative system, whose evolution depends on the input, its neighborhood and the time. The state of the CA is able to evolve as many times as necessary with only one input stimulus that typically represents a change in one cell. A CA can implement several evolution rules in the same implementation. Evolution rules are also present in AER systems, but only between layers. The output of an AER convolution chip can be seen as the evolution of the input information. A feed-back connection and a dynamic kernel are necessary in an AER convolution processor chip in order to implement the evolution rules of a CA. This is not implemented nowadays. Therefore, evolution rule is found in AER systems between layers of chips, but not inside one chip.

Then, a CA implemented with AER should be able to evolve, and take advantage of this evolution process to improve present AER processors in one chip.

Let's suppose a set of spike cells connected in a 2D grid as a CA. Each cell of this grid is connected to its eight neighbors. A 3x3 kernel memory is visible by all the cells. An input spike can be received by any of the cells as an input stimulus. Every time a cell receives an input stimulus, this cell sends an internal stimulus to its eight neighbors and furthermore it processes the center of the kernel. When a cell receives a

stimulus from a neighbor, this cell processes the corresponding element of the 3×3 kernel depending on the source cell of this internal stimulus. The internal process implemented by each cell consists of adding the corresponding kernel value to the internal state. If the internal state is higher than a configurable threshold, this cell needs to communicate a new stimulus (a second generation internal stimulus). Depending on the convolution kernel and the threshold programmed, this necessity of communicating a new stimulus can reflect the detection of an edge in the input visual information, or any other detected feature of a first layer of visual processing.

In the case of an AER chip based on IF neurons, this behavior imply the production of an output spike (see figure 2). This output information can be the input of a second layer that will extract a new feature of the visual information.

In a CA, this situation can be seen as an evolution of the state of the CA. So the CA is ready to process a second layer of processing using the same set of cells thanks to the evolution.

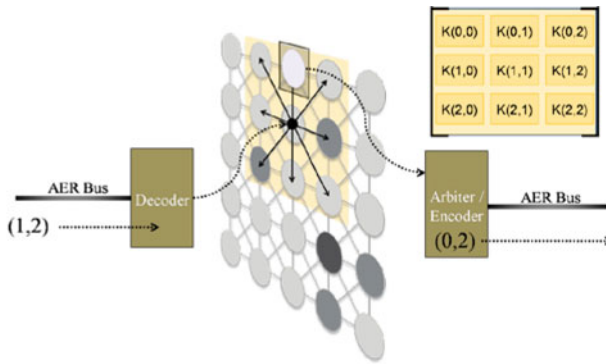


Fig. 2. AER-CA 3×3 kernel spike based convolution processor scheme

If we suppose that the first layer is processing edge detection (both vertical and horizontal), a second layer could be able to detect squares or rectangles if the second generation of internal stimulus is processed and transmitted between horizontal and/or vertical adjacent cells following the next rules:

1. When cell $C_{a,b}$ receives two different second generation internal stimulus from different adjacent cells between a configurable time window, it means that $C_{a,b}$ is the geometrical center between these two detected edges, so a new third generation of internal stimulus can be produced.
2. If the previous condition is not reached, the stimulus has to be retransmitted to the opposite adjacent cell in order to allow a future detection of the geometrical center by another cell.

By a correct configuration of these time windows and directions of retransmission of second, third, fourth ... generation stimulus between adjacent cells it is possible to detect any shape. Third, fourth... generations could be used to join basic shapes into complex ones in order to recognize faces, words, etc.

3 The C# AER Simulator

The simulator architecture is based on two main, separate but coordinated blocks: a graphical unit interface (GUI) that allows an easy interaction with the simulator, and the AER simulator. Figure 3 shows a block diagram example of AER scenario software architecture of the simulator and Figure 4 illustrates a captured screen of the GUI. Once the AER system to be simulated is designed, the user can easily construct the setup using the mouse and modifying some parameters through the GUI. This simulator is composed of several basic library blocks:

- Sequencer for AER traffic production from static bitmaps.
- Switch for easy AER traffic splitting and merging.
- Framegrabber for monitoring AER traffic histograms for a period of time.
- Prototype block for manipulating events. This block can be configured as a 3x3 AER-CA convolution based filter, or as a second generation propagation of stimulus for second, third... layer operations, as mentioned in the previous section.

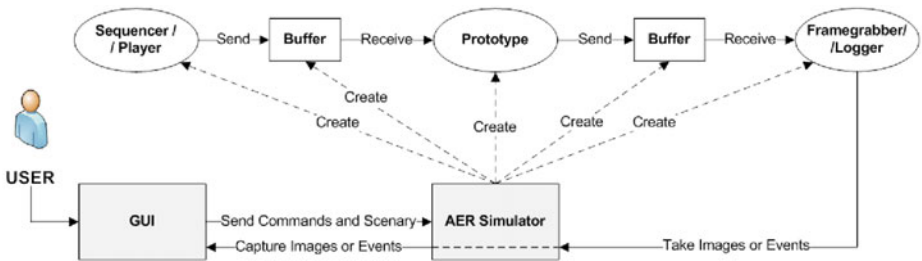


Fig. 3. AER Simulator block diagram

For simulating an AER-CA system with several layers that will be implemented in hardware with the same 2D array of cells but with different generations of stimulus transmissions, in the simulator, for simplicity, this is shown as several blocks connected in sequence.

Figure 4 shows a working example simulation where a sequencer is loaded with a bitmap that produces AER traffic only for those events whose addresses represent the circle. This AER traffic is sent to a 3x3 AER-CA convolver through a memory buffer. Those cells that produce output AER traffic represent the second generation stimulus. These stimuli are propagated through the perpendicular neighbors in order to look for a cell where several second generation stimuli arrive from different directions. This or these particular cells represent the center of mass of the object and produce the third generation of events. In figure 4, the simulation box number 1 is simulating the perpendicular propagation of AER, and it is producing output AER traffic of the center of mass, but it is also passing through the input events.

Inside the simulator, each box or component block of the GUI is able to receive or transmit AER from and to memory buffers. So the AER bus is implemented in the simulator as memory. The time information of the spikes is not represented with

timestamps in memory, but it is conserved depending on when the new AER event is stored on a buffer and read from it. Every box of the simulator is implemented as an object. Each object has its own class depending on the component (sequencer, Framegrabber, switch or prototype). Each object is executed with different and independent processes that communicate with each other through the memory buffers. A process associated with a box that produces events stores them on their corresponding output buffers. And a process associated with the receiver box will take the next event on the input buffer and it will execute necessary operations. When it is necessary this AER receiver process will generate an output event and then it will try to process the next input event. If a buffer is full, the process that is writing events on it is sent to sleep. If the buffer is empty, the process reading from it is sent to sleep.

GUI is periodically accessing the bitmaps stored in all the Framegrabber boxes and they are updated on the screen.

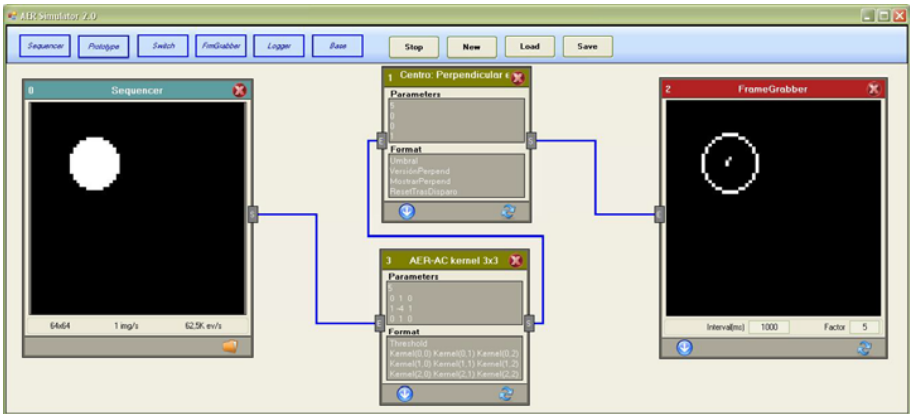


Fig. 4. AER-CA simulation of 2 generations for AER based center of mass detection

4 Conclusions

This paper presents an AER simulator for neuro-inspired AER based system simulations. The simulator has an easy GUI that allows fast simulations using sequencer and monitors of events for input and output AER traffic managing, and a configurable and expandable block that can implement 3x3 spike-based convolutions for image filtering inspired on Cellular Automata (AER-CA) with several generations of stimulus propagations. This philosophy allows not only detecting edges on an image, but also to find the center of mass of basic shapes. Third, fourth, generations could be used for object recognitions that are composed of basic shapes.

AER-CA hardware implementations have demonstrated for 3x3 kernel convolutions competitive performances of 150MOPS for 50MHz clocks for small FPGAs (Spartan II 200) that could be easily improved to more than 2GOPS for 200MHz clocks and 7x7 kernel sizes in higher capacity and faster FPGAs.

References

- [1] von Neumann, J.: The Theory of Self-reproducing Automata. In: Burks, A. (ed.). Univ. of Illinois Press, Urbana (1966)
- [2] Burks, A. (ed.): Essays on Cellular Automata. Univ. Illinois Press, U.S. (1970)
- [3] Pesavento, U.: An implementation of von Neumann's self-reproducing machine. *Artificial Life* 2, 337–354 (1995)
- [4] Sivilotti, M.: Wiring Considerations in analog VLSI Systems with Application to Field-Programmable Networks. Ph.D. Thesis, California Institute of Technology, Pasadena CA (1991)
- [5] Boahen, K.A.: Communicating Neuronal Ensembles between Neuromorphic Chips. In: *Neuromorphic Systems*, Kluwer Academic Publishers, Boston (1998)
- [6] Serrano-Gotarredona, R., Oster, M., Lichtsteiner, P., Linares-Barranco, A., Paz-Vicente, R., Gómez-Rodríguez, F., et al.: CAVIAR: A 45k-neuron, 5M-synapse AER Hardware Sensory-Processing-Learning-Actuating System for High-Speed Visual Object Recognition and Tracking. *IEEE Trans. on Neural Networks* 20(9), 1417–1438 (2009)
- [7] Cohen, A., et al.: Report to the National Science Foundation: Workshop on Neuromorphic Engineering, Telluride, Colorado, USA (June-July 2006), <http://www.ine-web.org>
- [8] The 2011 Cognitive Neuromorphic Engineering Workshop, <http://capocaccia.ethz.ch/capo/wiki/2011/>
- [9] Serrano-Gotarredona, R., et al.: A Neuromorphic Cortical-Layer Microchip for Spike-Based Event Processing Vision Systems. *IEEE T Circuits Systems-I* 53(12), 2548–2566 (2006)
- [10] Linares-Barranco, A., Paz, R., Gómez-Rodríguez, F., Jiménez, A., Rivas, M., Jiménez, G., Civit, A.: FPGA implementations comparison of neuro-cortical inspired convolution processors for spiking systems. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) *IWANN 2009*. LNCS, vol. 5517, pp. 97–105. Springer, Heidelberg (2009)
- [11] Rivas-Perez, M., Linares-Barranco, A., Cerda, J., Ferrando, N., Jimenez, G., Civit, A.: Visual Spike-based convolution processing with a Cellular Automata Architecture. In: *The 2010 International Joint Conference on Neural Networks, IJCNN* (2010), doi:10.1109/IJCNN.2010.5596924
- [12] Farabet, C., Poulet, C., Han, J.Y., LeCun, Y.: CNP: An FPGA-based Processor for Convolutional Networks. In: *International Conference on Field Programmable Logic and Applications, FPL* (2009)
- [13] Farrig, N., Mamalet, F., Roux, S., Yang, F., Paidavoine, M.: Design of a Real-Time Face Detection Parallel Architecture Using High-Level Synthesis. *EURASIP Journal on Embedded Systems* (2008), id 938256 doi:10.1155/2008/938256
- [14] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324
- [15] Huang, F.-J., LeCun, Y.: Large-scale learning with svm and convolutional nets for generic object categorization. In: *Proc. Computer Vision and Pattern Recognition Conference (CVPR 2006)*. IEEE, Los Alamitos (2006)
- [16] Ranzato, M., Huang, F., Boureau, Y., LeCun, Y.: Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: *Proc. Computer Vision and Pattern Recognition Conference (CVPR 2007)*. IEEE Press, Los Alamitos (2007)
- [17] Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: *Proc. International Conference on Computer Vision (ICCV 2009)*. IEEE, Los Alamitos (2009)

- [18] Osadchy, R., Miller, M., LeCun, Y.: Synergistic face detection and pose estimation with energy-based model. In: *Advances in Neural Information Processing Systems (NIPS 2004)*. MIT Press, Cambridge (2004)
- [19] Hadsell, R., Sermanet, P., Scoffier, M., Erkan, A., Kavackuoglu, K., Muller, U., LeCun, Y.: Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics* 26, 120–144
- [20] Farabet, C., Poulet, C., Han, J.Y., LeCun, Y.: CNP: an FPGA-based processor for Convolutional Networks. In: *International Conference on Field Programmable Logic and Applications (FPL)*, pp. 32–37, doi:10.1109/FPL.2009.5272559
- [21] Gomez-Rodriguez, F., Paz, R., Linares-Barranco, A., Rivas, M., et al.: AER tools for Communications and Debugging. In: *Proceedings of the IEEE ISCAS 2006*, Kos, Greece (May 2006)
- [22] Lichtsteiner, P., Posch, C., Delbruck, T.: A 128×128 120dB 15 us Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal on Solid-State Circuits* 43(2), 566–576 (2008)
- [23] Chan, V., Liu, S.C., van Schaik, A.: AER EAR: A Matched Silicon Cochlea Pair with Address-Event-Representation Interface. *IEEE Trans. Circuits and Systems-I* 54(1), 48–59 (2007)
- [24] Serrano-Gotarredona, R., et al.: A Neuromorphic Cortical-Layer Microchip for Spike-Based Event Processing Vision Systems. *IEEE T. Circuits Systems-I* 53(12), 2548–2566 (2006)
- [25] Costas-Santos, J., Serrano-Gotarredona, T., Serrano-Gotarredona, R., Linares-Barranco, B.: A Spatial Contrast Retina with On-chip Calibration for Neuromorphic Spike-Based AER Vision Systems. *IEEE Trans. Circuits and Systems-I* 54(7), 1444–1458 (2007)
- [26] Hafliker, P.: Adaptive WTA with an Analog VLSI Neuromorphic Learning Chip. *IEEE Transactions on Neural Networks* 18(2), 551–572 (2007)
- [27] Indiveri, G., Chicca, E., Douglas, R.: A VLSI Array of Low-Power Spiking Neurons and Bistables Synapses with Spike-Timing Dependent Plasticity. *IEEE Transactions on Neural Networks* 17(1), 211–221 (2006)

A Biologically Inspired Neural Network for Autonomous Underwater Vehicles

Francisco García-Córdova and Antonio Guerrero-González

Department of System Engineering and Automation
Polytechnic University of Cartagena (UPCT)
Campus Muralla del Mar, E-30202, Cartagena, Murcia, Spain
francisco.garcia@upct.es, antonio.guerrero@upct.es

Abstract. Autonomous underwater vehicles (AUVs) have great advantages for activities in deep oceans, and are expected as the attractive tool for near future underwater development or investigation. However, AUVs have various problems which should be solved for motion control, acquisition of sensors' information, behavioral decision, navigation without collision, self-localization and so on. This paper proposes an adaptive biologically inspired neural controller for trajectory tracking of AUVs in nonstationary environment. The kinematic adaptive neuro-controller is an unsupervised neural network, which is termed Self-Organization Direction Mapping Network (SODMN). The network uses an associative learning system to generate transformations between spatial coordinates and coordinates of propellers' velocity. The neurobiological inspired control architecture requires no knowledge of the geometry of the robot or of the quality, number, or configuration of the robot's sensors. The SODMN proposed in this paper represents a simplified way to understand in part the mechanisms that allow the brain to collect sensory input to control adaptive behaviours of autonomous navigation of the animals. The efficiency of the proposed neurobiological inspired controller for autonomous intelligent navigation was implemented on an underwater vehicle capable of operating during large periods of time for observation and monitoring tasks.

1 Introduction

With continuous advances in control, navigation, artificial intelligence, material science, computer, sensor and communication, autonomous underwater vehicles (AUVs) have become very attractive for various underwater tasks. The autonomy is one of the most critical issues in developing AUVs. The design, development, navigation, and control process of an AUV is a complex and expensive task. Various control architectures have been studied to help increase the autonomy of AUVs [\[1,2,3,4,5\]](#).

Trajectory generation with obstacle avoidance is a fundamentally important issue in robotics. Real-time collision-free trajectory generation becomes more difficult when robots are in a dynamic, unstructured environment. There are a lot

of studies on trajectory generation for robots using various approaches problem [2]. Some of the previous models [1,2,3] use global methods to search the possible paths in the workspace, which normally deal with static environment only and are computationally expensive when the environment is complex. Seshadri and Ghosh [1] proposed a new path planning model using an iterative approach. However this model is computationally complicated, particularly in a complex environment. Li and Bui [2] proposed a fluid model for robot path planning in a static environment. Oriolo et al. [3] proposed a model for real-time map building and navigation for a mobile robot, where a global path planning plus a local graph search algorithm and several cost functions are used.

Several neural network models [6,4,5] were proposed to generate real-time trajectories through learning. Ritter et al. [6] proposed a Kohonen's self-organizing mapping algorithm based neural network model to learn the transformation from Cartesian workspace to the robot manipulator joint space. Fujii et al. [4] proposed a multilayer reinforcement learning based model for path planning with a complicated collision avoidance algorithm. However, the generated trajectories using learning based approaches are not optimal, particularly during the initial learning phase.

Several papers [4,5,7,8] examine the application of neural network (NN) to the navigation and control of AUVs using a well-known backpropagation algorithm and its variants since it is not possible to accurately express the dynamics of an AUV as linear in the unknown parameters. Unfortunately, the backpropagation-based NN weight tuning is proven to have convergence and stability problems. Further, an offline learning phase, which is quite expensive, is required with the NN controllers [5].

In this paper, an unsupervised kinematic adaptive neuro-controller that can learn to guide AUVs towards a target located at an arbitrary location in a 3-D workspace is proposed. The underwater platform's movements are controlled by selecting the angular velocity of each propeller. The proposed kinematic adaptive neuro-controller requires no information about the robot's structure, is resistant to a variety of disturbances, and is based on existing neural networks of biological sensory-motor control [7]. The kinematic adaptive neuro-controller is a Self-Organization Direction Mapping Network (SODMN), and combines associative learning and Vector Associative Map (VAM) learning [8] to generate transformations between spatial coordinates and coordinates of propellers' velocity. The transformations are learned in an unsupervised training phase, during which the underwater robot moves as a result of randomly selected propellers' velocities. The robot learns the relationship between these velocities and the resulting incremental movements. The efficacy of the proposed kinematic adaptive neuro-controller is tested experimentally by an underwater vehicle capable of operating during large periods of time for observation and monitoring tasks.

This paper is organized as follows. We first describe (Section 2) the neural control system for AUVs using the proposed SODMN. Section 3 addresses experimental results with the proposed scheme for trajectory tracking control and

approach behavior over an underwater platform. Finally, in Section 4, discussion and conclusions based on experimental results are given.

2 Architecture of the Neural Control System

Figure 1 illustrates our proposed neural architecture. The trajectory tracking control without obstacles is implemented by the SODMN. The SODMN learns to control the robot through a sequence of spontaneously generated random movements. The random movements enable the neural network to learn the relationship between angular velocities applied at the propellers and the incremental displacement that ensues during a fixed time step. The proposed SODMN combines associative learning and Vector Associative Map (VAM) learning [8] to generate transformations between spatial coordinates and coordinates of propellers' velocity. The nature of the proposed kinematic adaptive neuro-controller is that continuously calculates a vectorial difference between desired and actual velocities, the underwater robot can move to arbitrary distances and angles even though during the initial training phase it has only sampled a small range of displacements. Furthermore, the online error-correcting properties of the proposed architecture endow the controller with many useful properties, such as the ability to reach targets in spite of drastic changes of robot's parameters or other perturbations.

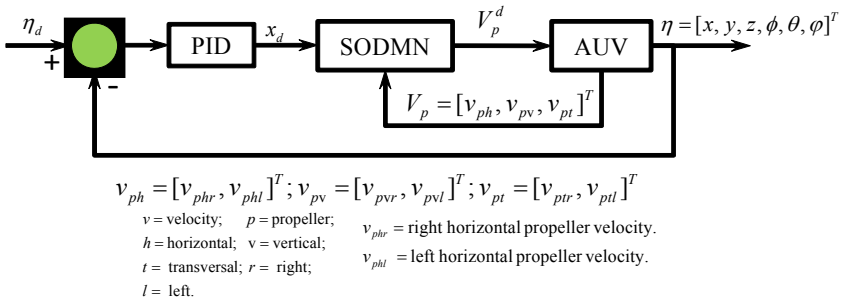


Fig. 1. Structure of adaptive biologically inspired neural controller for trajectory tracking of AUVs

For a dynamic positioning in the path tracking a PID controller was incorporated into the architecture of control system. It allows smooth the error signal in the reaching of objectives.

2.1 Self-Organization Direction Mapping Network (SODMN)

At a given set of angular velocities the differential relationship between underwater robot motions in spatial coordinates and angular velocities of propellers is expressed like a linear mapping. This mapping varies with the velocities of propellers. The transformation of spatial directions to propellers' angular velocities

is shown in Fig. 2. The tracking spatial error (e) is computed to get the desired spatial direction vector (\mathbf{x}_d) and the spatial direction vector (DVs). The DVs is transformed by the direction mapping network elements V_{ik} to corresponding motor direction vector (DVM). On the other hand, a set of tonically active inhibitory cells which receive broad-based inputs that determine the context of a motor action was implemented as a context field. The context field selects the V_{ik} elements based on the propellers' angular velocities configuration.

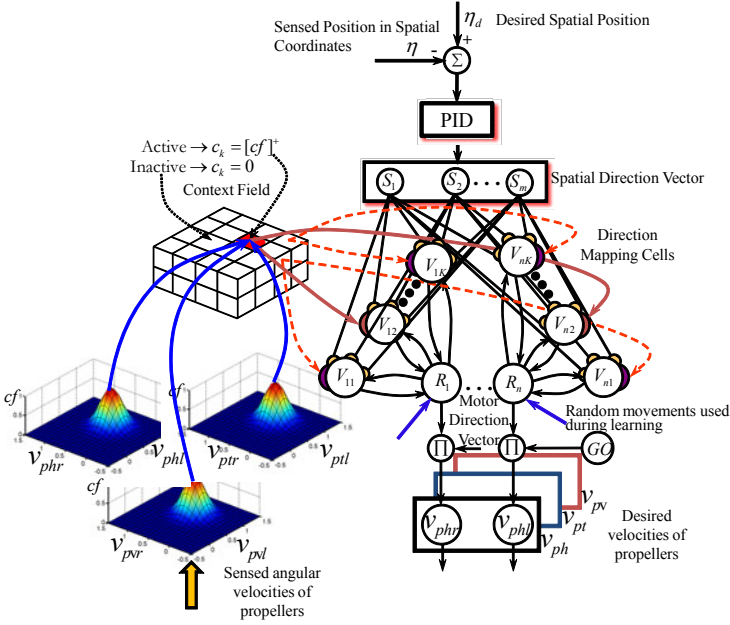


Fig. 2. Architecture of self-organization direction mapping network for autonomous robotic systems

A speed-control GO signal acts as a nonspecific multiplicative gate and control the movement's overall speed. The GO signal is a input from a decision center in the brain, and starts at zero before movement and then grows smoothly to a positive value as the movement develops. During the learning, sensed angular velocities of propellers are fed into the DVM and the GO signal is inactive.

Activities of cells of the DVs are represented in the neural network by quantities (S_1, S_2, \dots, S_m), while activities of the cells of the motor direction vector (DVM) are represented by quantities (R_1, R_2, \dots, R_n). The direction mapping is formed with a field of cells with activities V_{ik} . Each V_{ik} cell receives the complete set of spatial inputs S_j , $j = 1, \dots, m$, but connects to only one R_i cell (see Figure 2). The mechanism that is used to ensure weights converge to the correct linear mapping is similar to the VAM learning construction [9]. The direction mapping cells ($\mathbf{V} \in \mathbb{R}^{n \times k}$) compute a difference of activity between the spatial and motor direction vectors via feedback from DVM. During learning, this difference drives

the adjustment of the weights. During performance, the difference drives DVM activity to the value encoded in the learned mapping.

A context field cell pauses when it recognizes a particular velocity state (i.e., a velocity configuration) on its inputs, and thereby disinhibits its target cells. The target cells (direction mapping cells) are completely shut off when their context cells are inactive. This is shown in Fig. 2. Each context field cell projects to a set of direction mapping cells, one for each velocity vector component. Each velocity vector component has a set of direction mapping cells associated with it, one for each context. A cell is “on” for a compact region of the velocity space. It is assumed for simplicity that only one context field cell turns “on” at a time. In Figure 2, inactive cells in the context field are shown as white disks. The center context field cell is “on” when the angular velocities are in the center region of the velocity space, in this three degree-of-freedom example. The “on” context cell enables a subset of direction mapping cells through the inhibition variable c_k , while “off” context cells disable to the other subsets. When the k^{th} context cell is “off” or inactive (modeled as $c_k=0$), in its target cells, the entire input current to the soma is shunted away such that there remains only activity in the axon hillock, which decays to zero. When the k^{th} context cell is “on” or active, $c_k=1$, its target cells (V_{ik}) receive normal input.

The learning is obtained by decreasing weights in proportion to the product of the presynaptic and postsynaptic activities. Therefore, the learning rule can be obtained by using the gradient-descent algorithm. The training is done by generating random movements, and by using the resulting angular velocities and observed spatial velocities of the underwater robot as training vectors to the direction mapping network.

3 Experimental Results

The proposed biologically-inspired control system is implemented on a underwater robot from the UPCT (AUV-UPCT). The rebuilt vehicle was transferred to the UPCT by the Spanish Navy. Figure 3 shows the underwater platform and the interconnection scheme of hardware components from the AUV: Battery, CPU, inertial positioning systems, compass, propulsion systems, video capture, inclinometers, water intrusion detectors, monitoring station, and sonars. It consists of a pressure resistant body with 5 motor for propulsion and maneuverability. AUV-UPCT has a dimension of 1680 L \times 600 W \times 635 H (mm), a weight of 160 Kg, a maximum speed of 4 knots (48 V) and 2 knots (24 V), an operational depth of 300 mts, two vertical thrusters, two forward thrusters and one transversal thruster. The core of central controller system is a Kontron 986LCD-M/mITX motherboard. High-level control algorithms (SODMN) are written in VC++ and run with a sampling time of 10 ms on the central controller system.

The proposed neural network model is capable of generating optimal trajectory for underwater vehicles in an arbitrarily varying environment. The state space is the Cartesian workspace of underwater robot. The proposed model is the applied to a trajectory generation problem for a robot to track a target (O).

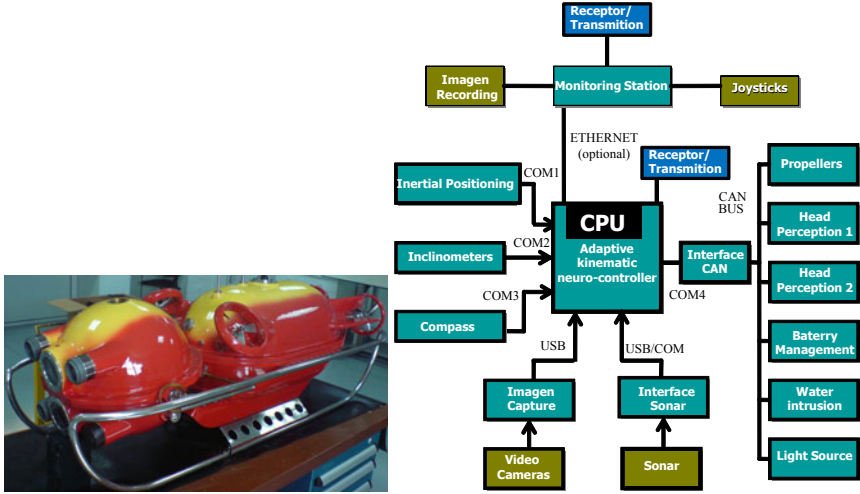


Fig. 3. Autonomous underwater vehicle from the UPCT (AUV-UPCT)

The SODMN assumes a context field structure of $100 \times 100 \times 10$ neuron. In a 3-D workspace without any obstacle, the traveling route of the target is shown in Fig. 4(a) as indicated by circles, with an initial position at $(x, y, z) = (0, 0, 4)$ m. The robot was immersed in a controlled pool with 15 m deep in the industrial area of Fuente Alamo, Murcia-Spain. Note that the depth was recorded as positive (z) and that the proposed SODMN responds to the real-time location of the targets with no prior knowledge of the varying environment. The underwater robot starts to move from position $(0, 0, 0)$ at a speed of 0.375 m/s. The generated trajectory of the robot is shown in Fig. 4(a) by boxes. Tracking errors of the adaptive controller system are shown in Fig. 4(b).

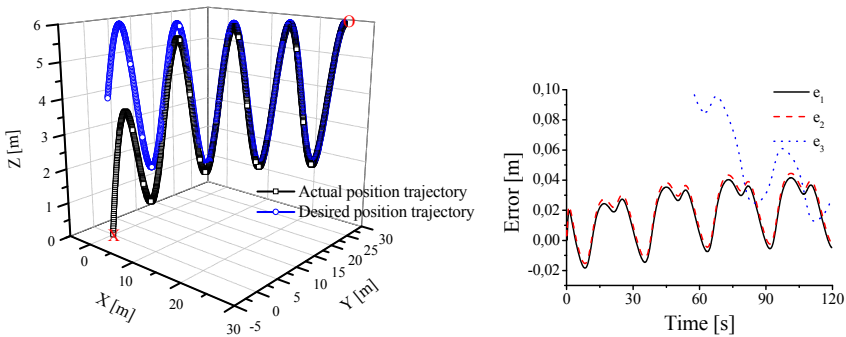


Fig. 4. Adaptive neuro-controller performance. (a) Tracking control of a desired trajectory. (b) Estimated tracking error.

4 Discussion

In present model, appropriate operations are learned in an unsupervised fashion through repeated action-perception cycles by recoding proprioceptive information related to the underwater robot. The resulting solution has two interesting properties: (a) the required transformation is executed accurately over a large part of the reaching space, although few velocities are actually learned; and (b) properties of single neurons and populations closely resemble those of neurons and populations in parietal and cortical regions [10]. The activity of the population of motor cortical cells which encode movement direction appears to represent the instantaneous velocity of movement [11]. In addition, the preferred directions of individual cells shifts with the movement origin, indicating that the directional coding of motor cortex may be influenced by velocity configuration (in the model is the context field) [12], as is necessary for a Jacobian-based mapping. Correspondence between layers of the network and brain regions can be made tentatively base on anatomical and physiological arguments [10,11]. The representation of DVs could be in posterior parietal cortex (PPC) [13]. Neurons in PPC exhibit activity patterns correlated with the spatial direction of movement [14]. A candidate region for participating in the direction mapping computation is the cerebellum [15]. Also, note that there are certain similarities between the nature of the context field cells in the underwater robot movement model and the Purkinje cells of the adaptive timing model. Both types of cells are tonically active and allow a response by “pausing” this tonic activity. Thus, the possibility that a context field type of function is performed by cerebellar cortex. The proposed direction mapping model also posits a learning site separate from the context field computation, which might be a cerebellar function. In the model, motor commands were emitted by a layer containing R_i neurons, which contribute to the movement by a displacement along a direction in velocity space. The individual influence of a command neuron is proportional to its discharge level.

5 Conclusions

In this paper, a biologically inspired neural network for the spatial reaching tracking has been developed. This neural network is implemented as a kinematic adaptive neuro-controller. The SODMN uses a context field for learning the direction mapping between spatial coordinates and coordinates of propellers’ velocity. The transformations are learned during an unsupervised training phase, during which the underwater robot moves as result of randomly selected angular velocities of propellers. It has the ability to adapt quickly for unknown states. The model algorithm is computationally efficient and the computational complexity linearly depends on the state space size of the neural network. The efficacy of the proposed neural network for reaching and tracking behaviors was tested experimentally by a underwater robot.

References

1. Seshadri, C., Ghosh, A.: Optimum path planning for robot manipulators amid static and dynamic obstacles. *IEEE Trans. Syst. Man, Cybern.* 23, 576–584 (1993)
2. Li, Z.X., Bui, T.D.: Robot path planning using fluid model. *J. Intell. Robot. Syst.* 21, 29–50 (1998)
3. Oriolo, G., Luca, A.D., Vendittelli, M.: WMR control via dynamic feedback linearization: Design, implementation and experimental validation. *IEEE Trans. Control. Syst. Technol.* 10, 835–852 (2002)
4. Fujii, T., Arai, Y., Asama, H., Endo, I.: Multilayered reinforcement learning for complicated collision avoidance problems. In: *Proceedings IEEE International Conference on Robotics and Automation*, Leuven, Belgium, vol. 3, pp. 2186–2191 (1998)
5. Carreras, M., Yuh, J., Batlle, J., Ridao, P.: A behavior-based scheme using reinforcement learning for autonomous underwater vehicles. *IEEE Journal Of Oceanic Engineering* 30, 416–427 (2005)
6. Ritter, H.J., Martinez, T.M., Schulten, K.J.: Topology-conserving maps for learning visuo-motor coordination. *Neural Networks* 2, 159–168 (1989)
7. García-Córdova, F.: A cortical network for control of voluntary movements in a robot finger. *Neurocomputing* 71, 374–391 (2007)
8. García-Córdova, F., Guerrero-González, A., Marín-García, F.: Design and implementation of an adaptive neuro-controller for trajectory tracking of nonholonomic wheeled mobile robots. In: Mira, J., Álvarez, J.R. (eds.) *IWINAC 2007*. LNCS, vol. 4528, pp. 459–468. Springer, Heidelberg (2007)
9. Gaudiano, P., Grossberg, S.: Vector associative maps: Unsupervised real-time error-based learning and control of movement trajectories. *Neural Networks* 4, 147–183 (1991)
10. Baraduc, P., Guigon, E., Burnod, Y.: Recording arm position to learn visuomotor transformations. *Cerebral Cortex* 11, 906–917 (2001)
11. Georgopoulos, A.P.: Neural coding of the direction of reaching and a comparison with saccadic eye movements. *Cold Spring Harbor Symposia in Quantitative Biology* 55, 849–859 (1990)
12. Caminiti, R., Johnson, P., Urbano, A.: Making arm movements within different parts of space: Dynamic aspects in the primate motor cortex. *Journal of Neuroscience* 10, 2039–2058 (1990)
13. Rondot, P., De-Recondo, J., Dumas, J.: Visuomotor ataxia. *Brain* 100, 355–376 (1976)
14. Lacquaniti, F., Guigon, E., Bianchi, L., Ferraina, S., Caminiti, R.: Representing spatial information for limb movement: Role of area 5 in the monkey. *Cerebral Cortex* 5, 391–409 (1995)
15. Fiala, J.C.: *Neural Network Models of Motor Timing and Coordination*. PhD thesis, Boston University (1996)

A Preliminary Study on the Use of Fuzzy Rough Set Based Feature Selection for Improving Evolutionary Instance Selection Algorithms

Joaquín Derrac¹, Chris Cornelis², Salvador García³, and Francisco Herrera¹

¹ Dept. of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology),
University of Granada, 18071 Granada, Spain

`jderrac@decsai.ugr.es`, `herrera@decsai.ugr.es`

² Dept. of Applied Mathematics and Computer Science. Ghent University,
Gent, Belgium

`chris.cornelis@ugent.be`

³ Dept. of Computer Science. University of Jaén, 23071 Jaén, Spain
`sglopez@ujaen.es`

Abstract. In recent years, the increasing interest in fuzzy rough set theory has allowed the definition of novel accurate methods for feature selection. Although their stand-alone application can lead to the construction of high quality classifiers, they can be improved even more if other preprocessing techniques, such as instance selection, are considered.

With the aim of enhancing the nearest neighbor classifier, we present a hybrid algorithm for instance and feature selection, where evolutionary search in the instances' space is combined with a fuzzy rough set based feature selection procedure. The preliminary results, contrasted through nonparametric statistical tests, suggest that our proposal can improve greatly the performance of the preprocessing techniques in isolation.

Keywords: Fuzzy Rough Sets, Evolutionary Algorithms, Instance Selection, Feature Selection, Nearest Neighbor Classifier.

1 Introduction

Data reduction is a data preprocessing task which can be applied to ease the problem of dealing with large amounts of data. Its main objective is to reduce the original data by selecting the most representative information. In this way, it is possible to avoid excessive storage and time complexity, improving the results obtained by any data mining application. The best known data reduction processes are Feature Selection (FS) [9], Feature Extraction, Attribute Discretization, Instance Generation and Instance Selection (IS) [8].

Recently, Rough Set Theory (RST) [10] has been employed to tackle the FS problem. This approach can be enhanced with the use of fuzzy logic, obtaining methods which offer a greater flexibility and better potential to produce high-quality feature subsets than the crisp ones [3]. On the other hand, Evolutionary

Algorithms [6] have been successfully applied in IS problems due to the possibility of defining it as a search problem [5]. Both fields can offer suitable tools for enhancing the performance of machine learning methods.

In this work, a new hybrid approach considering both fuzzy-RST based FS and evolutionary IS is presented, EIS-RFS (Evolutionary IS enhanced by RST-based FS). A steady-state Genetic Algorithm (GA) is used to conduct the search of instances, whereas features are selected by a fuzzy-RST based method. The features selected are considered within the framework of the GA, thus modifying the environment in which the instances are chosen. At the end of its application, EIS-RFS considers the best subsets found to build a reduced version of the training set, well suited to be used as a reference set for the 1 Nearest Neighbor classifier (1-NN). This method is compared with their main components in isolation and the 1-NN classifier without preprocessing. The results achieved are contrasted by using nonparametric statistical techniques, reinforcing the conclusions obtained.

The rest of this work is organized as follows. Section 2 gives some background about evolutionary IS and fuzzy RST based FS. Section 3 describes the main characteristics of IFS-RTS. Section 4 describes the experimental study performed and shows the results achieved. Finally, Section 5 summarizes our conclusions.

2 Background and Related Work

This section is focused on two topics: IS and FS as data reduction techniques (Section 2.1), and the use of fuzzy RST for FS (Section 2.2).

2.1 Instance and Feature Selection

The IS objective is to isolate the smallest set of instances which enable a data mining algorithm to predict the class of a instance with the same quality as the initial data set [8]. By minimizing the data set, it is possible to reduce the space complexity and decrease the computational cost of the data mining algorithms that will be applied, improving their generalization capabilities.

More specifically, IS can be defined as follows: Let $(\mathcal{X}, \mathcal{A})$ be an information system, where $\mathcal{X} = \{x_1, \dots, x_n\}$ and $\mathcal{A} = \{a_1, \dots, a_m\}$ are finite, non-empty sets of instances and features. Then, let us assume that there is a training set TR which consists of N instances, and a test set TS composed of T instances ($TR \cup TS = (\mathcal{X}, \mathcal{A})$). Let $S \subseteq TR$ be the subset of selected samples that resulted from the execution of an IS algorithm, then we classify a new pattern T from TS by a data mining algorithm acting over the instances of S .

In the data mining field many approaches of evolutionary IS have been developed [7,5]. The interest in this field was increased by the study performed by Cano et al. [2], where a complete study of the use of evolutionary algorithms in IS is made. They concluded that evolutionary algorithms outperform classical algorithms both in reduction rates and classification accuracy.

On the other hand, in FS the goal is to select the most appropriate subset of features from the initial data set. It aims to eliminate irrelevant and/or redundant features to obtain a simple and accurate classification system [9]. Starting from the definition given for IS, FS can be defined as follows: Let us assume \mathcal{A}, \mathcal{X} , TR and TS have been already defined. Let $B \subseteq \mathcal{A}$ be the subset of selected features that resulted from the execution of a FS algorithm over TR , then we classify a new pattern from TS by a data mining algorithm acting over TR , employing as a reference only the features selected in B .

As with IS methods, a great number of FS methods have been developed recently. Some complete surveys, analyzing both classical and advanced approaches to FS, can be found in the literature [9]. Some advanced methods, combining both evolutionary IS and FS, have been also developed [4].

2.2 Fuzzy RST for FS

In rough set analysis [10], each attribute a in \mathcal{A} corresponds to an $\mathcal{X} \rightarrow V_a$ mapping, in which V_a is the value set of a over \mathcal{X} . For every subset B of \mathcal{A} , the B -indiscernibility relation R_B is defined as

$$R_B = \{(x, y) \in \mathcal{X}^2 \text{ and } (\forall a \in B)(a(x) = a(y))\} \tag{1}$$

Therefore, R_B is an equivalence relation. Its equivalence classes $[x]_{R_B}$ can be used to approximate concepts, in other words, subsets of the universe \mathcal{X} . Given $A \subseteq \mathcal{X}$, its lower and upper approximation with respect to R_B are defined by

$$R_B \downarrow A = \{x \in \mathcal{X} | [x]_{R_B} \subseteq A\} \text{ and } R_B \uparrow A = \{x \in \mathcal{X} | [x]_{R_B} \cap A \neq \emptyset\} \tag{2}$$

A *decision system* $(\mathcal{X}, \mathcal{A} \cup \{d\})$ is a special kind of information system, used in the context of classification, in which d ($d \notin \mathcal{A}$) is a designated attribute called the decision attribute. Its equivalence classes $[x]_{R_d}$ are called decision classes. Given $B \subseteq \mathcal{A}$, the B -positive region POS_B contains those objects from X for which the values of B allow to predict the decision class unequivocally:

$$POS_B = \bigcup_{x \in X} R_B \downarrow [x]_{R_d} \tag{3}$$

Indeed, if $x \in POS_B$, it means that whenever an instance has the same values as x for the attributes in B , it will also belong to the same decision class as x . The predictive ability with respect to d of the attributes in B is then measured by the following value (degree of dependency of d on B):

$$\gamma_B = \frac{|POS_B|}{|\mathcal{X}|} \tag{4}$$

Instead of using a crisp equivalence relation R to represent objects' indiscernibility, we can also measure it by means of a fuzzy relation R . Typically, we assume that R is at least a fuzzy tolerance relation (reflexive and symmetric).

Assuming that for a qualitative attribute a , the classical way of discerning objects is used, that is, $R_a(x, y) = 1$ if $a(x) = a(y)$ and $R_a(x, y) = 0$ otherwise, we can define, for any subset B of \mathcal{A} , the fuzzy B-indiscernibility relation by

$$R_B(x, y) = \mathcal{T}(R_a(x, y)), a \in B \tag{5}$$

in which \mathcal{T} represents a t-norm. It can be seen that if only qualitative attributes are used, then the traditional concept of B-indiscernibility relation is recovered.

For the lower and upper approximation of a fuzzy set A in X by means of a fuzzy tolerance relation R , the formulas defined in (2) are paraphrased (given the Lukasiewicz implicator $\mathcal{I}(x, y) = \min(1, 1 - x + y)$ and the minimum t-norm $\mathcal{T}(x, y) = \min(x, y), x, y \in [0, 1]$) to define $R \downarrow A$ and $R \uparrow A$, for all y in X , by

$$(R \downarrow A)(Y) = \inf_{x \in X} \mathcal{I}(R(x, y), A(x)) \quad (R \uparrow A)(Y) = \sup_{x \in X} \mathcal{T}(R(x, y), A(x)) \tag{6}$$

Using fuzzy B-indiscernibility relations, the fuzzy B-positive region is defined by

$$POS_B(y) = \left(\bigcup_{x \in X} R_B \downarrow [\mathcal{X}_{R_d}] \right) (y) \tag{7}$$

Once we have fixed the fuzzy positive region, we can define an increasing $[0, 1]$ -valued measure to gauge the degree of dependency of a subset of features on another subset of features. For FS it is useful to phrase this in terms of the dependency of the decision feature on a subset of the conditional features:

$$\gamma_b = \frac{|POS_B|}{|POS_{\mathcal{A}}|} \tag{8}$$

3 EIS-RFS: Instance and Feature Selection Enhanced by Rough Set Theory

This section is devoted to describe EIS-RFS. Section 3.1 describes the steady-state GA employed for performing IS and the fuzzy RST based FS method. Section 3.2 shows the full model combining both techniques.

3.1 Basic Techniques of EIS-RFS

The IS component of EIS-RFS is guided by a steady-state GA where only two offspring are produced in each generation. Binary codification, binary tournament selection procedure, a two point crossover operator and a bit-flip mutation operator are considered. Concerning the fitness function, it should pursue both reduction and accuracy objectives. To do so, we will follow the proposal given in [2], where Cano et al. defined *AccRate* as the accuracy achieved by a 1-NN classifier when classifying the entire training set, using the currently selected subset as a reference and using leave-one-out as validation scheme, *RedRate* as the

reduction rate achieved over the currently selected instances, and a real-valued weighting factor, α , to adjust the strength of each term in the resulting fitness value. Equation 9 defines it, where J is an IS chromosome to be evaluated.

$$Fitness(J) = \alpha \cdot AccRate(J) + (1 - \alpha) \cdot RedRate(J) \quad (9)$$

Following the recommendations given in [2], EIS-RFS will employ a value $\alpha = 0.5$, which should offer an adequate trade-off between accuracy and reduction.

The fuzzy RST based FS method is taken from [3], where a hill climbing heuristic (quickreduct heuristic) is used for searching iteratively subsets of features maximizing the gamma measure (Equation 8). The similarity measure selected for quantitative values is

$$R_a(x, y) = \max \left(\min \left(\left(\frac{a(y) - a(x) + \sigma_a}{\sigma_a}, \frac{a(x) - a(y) + \sigma_a}{\sigma_a} \right), 0 \right) \right) \quad (10)$$

where x and y are two different instances belonging to the training set, and σ_a denotes the standard deviation of a . For nominal attributes we use the Value Difference Metric (VDM) [11], where two values are closer if they have more similar classifications (that is, more similar correlations with the output classes).

3.2 Hybrid Model for Simultaneous IS and FS

Once the two basic tools considered for performing IS and FS have been defined, the hybrid model which composes our approach can be described. Basically, it can be described as a steady-state GA for IS where, every time a fixed number of evaluations has been spent, a fuzzy-RST based FS procedure is applied to modify the features taken in consideration during the search.

1. **Initialization:** IS chromosomes are initialized randomly. Two different subsets are considered for the selection of the initial subset of features: The full set and the subset of features selected by the RST based FS method using as input the whole training set. The best performing subset is selected as the global subset of features of EIS-RFS.
2. **New IS generation:** An IS generation is carried out using the steady-state GA scheme. Note that, when evaluating a new chromosome, the 1-NN classifier used in the fitness function will only consider the selected features in the global subset of features of EIS-RFS.
3. **Update Features:** If the Stabilize phase has not been activated yet (see below), the procedure of update of features is called every time *UpdateFS* evaluations has been spent by the steady-state GA. This procedure consists of using the RST-based FS filter method, considering as input instance set the current best chromosome of the population. The new subset of features obtained is tested by applying it to a 1-NN classifier (considering as reference set only the current best subset of instances). If this subset performs better than the former, it is accepted as the global subset of features of EIS-RFS.

4. **Stabilize phase:** Changes in the current subset of features are only considered if the search is not near its end. Therefore, if the number of evaluations spent is higher than $\beta \cdot NEvaluations$, the stabilize stage is activated and no further changes in the subset of features selected are considered. This mechanism allows EIS-RFS to easily converge for hard problems, where the final subset of features is fixed before the end of the search. It allows EIS-RFS to focus its last efforts in optimizing the subsets of instances selected, performing a final refinement of the solutions achieved.
5. **Termination criterion:** The search process ends if EIS-RFS has spent $NEvaluations$. Otherwise, a new cycle of the algorithm begins.
6. **Output:** When the fixed number of evaluations runs out, the best chromosome of the population (a subset of instances) and the current global subset of selected features are returned as the output of EIS-RFS.

The resulting subsets of instances and features define a pruned version of the original training set. This set can be used as reference set by a 1-NN classifier to perform a faster and more accurate classification of new test instances.

4 Experimental Framework and Results

This section describes the experimental study conducted in this contribution. Data sets, comparison methods and parameters are detailed in Section 4.1. Results are shown and discussed in Section 4.2.

4.1 Experimental Framework

In our experiments, we have used 20 data sets taken from the KEEL-Datasets [1] repository. Table 1 shows their main characteristics. For each data set, it is shown the number of instances, features and classes of the problem described. We have used a ten fold cross-validation (10-fcv) procedure as validation scheme.

As comparison methods, we have selected the techniques considered in the construction of EIS-RFS (a Steady-State GA for IS (IS-SSGA) and the fuzzy-RST based FS method (FS-RST)). The preprocessed data sets obtained as result of the application of them have been used as reference sets for a 1-NN, estimating accuracy. Furthermore, we have considered the inclusion of the 1-NN classifier as a baseline (using the full training set as reference). The parameters of each method are shown in Table 2.

Finally, we will employ the well-known Wilcoxon Signed-Ranks test for contrasting the accuracy results achieved. Further information about this test and other statistical procedures specifically designed for use in the field of Machine Learning can be found at the SCI2S thematic public website on *Statistical Inference in Computational Intelligence and Data Mining* [2].

¹ <http://www.keel.es/datasets.php>

² <http://sci2s.ugr.es/sicidm/>

Table 1. UCI Data sets used in our experiments

Data set	Instances	Features	Classes	Data set	Instances	Features	Classes
Australian	690	14	2	Housevotes	435	16	2
Balance	625	4	3	Iris	150	4	3
Bupa	345	6	2	Mammographic	961	5	2
Cleveland	303	13	5	Newthyroid	215	5	3
Contraceptive	1473	9	3	Pima	768	8	2
Ecoli	336	7	8	Sonar	208	60	2
German	1000	20	2	Tic-tac-toe	958	9	2
Glass	214	9	7	Wine	178	13	3
Hayes-roth	160	4	3	Wisconsin	699	9	2
Hepatitis	155	19	2	Zoo	101	16	7

Table 2. Parameter specification for the algorithms tested in the experimentation

Algorithm Parameters	
EIS-RFS	NEvaluations: 10000, Pop. size: 50, Cross. prob.: 1.0, Mutat. prob.: 0.005 per bit, α : 0.5 <i>MaxGamma</i> : 1.0, UpdateFS: 100, β : 0.75
IS-SSGA	NEvaluations: 10000, Pop. size: 50, Cross. prob.: 1.0, Mutat. prob.: 0.005 per bit, α : 0.5
FS-RST	<i>MaxGamma</i> : 1.0
1-NN	-

4.2 Results and Analysis

Table 3 shows the results measured by accuracy, ratio of reduction over instances (Reduction (IS)) and ratio of reduction over features (Reduction (FS)) in test data. For each data set, the best result in accuracy is highlighted in **bold**.

As can be seen in the table, EIS-RFS achieves the best average accuracy result in test phase. For contrasting this fact, we have carried out a Wilcoxon Signed-Ranks test, the results of which are summarized in Table 4.

Reading Tables 3 and 4, we can make the following analysis:

- In accuracy, EIS-RFS outperforms the rest of algorithms on 14 of 20 data sets, and has the best average result. This superiority is remarked as significant by the results of the Wilcoxon Signed-Ranks test, the results of which show that EIS-RFS outperforms all the comparison methods with a level of significance $\alpha = 0.01$. This is a strong result, which supports the fact that EIS-RFS clearly outperforms all the other techniques in accuracy.
- Concerning reduction in instances' space, EIS-RFS achieves slightly better rates than IS-SSGA. Therefore, our approach is able to effectively reduce the training set without harming the accuracy of the 1-NN rule. Moreover, its accuracy is even increased, exceeding the IS-SSGA method. Concerning reduction in features' space, it shows a similar behavior as FS-RST, although the features selected are different (which is one of the reasons of the accuracy improvement of EIS-RFS over FS-RST).

Table 3. Results obtained

Measure	Accuracy				Reduction (IS)		Reduction (FS)	
	Data set	EIS-RFS	IS-SSGA	FS-RST	1-NN	EIS-RFS	IS-SSGA	EIS-RFS
Australian	85.66	85.65	81.45	81.45	0.8872	0.8799	0.1571	0.0000
Balance	85.92	86.40	79.04	79.04	0.8464	0.8686	0.0000	0.0000
Bupa	65.72	61.14	62.51	61.08	0.8502	0.8644	0.0000	0.1274
Cleveland	55.16	52.82	52.51	53.14	0.9014	0.9171	0.0462	0.3908
Contraceptive	45.42	44.54	42.63	42.77	0.7637	0.7530	0.0667	0.0360
Ecoli	82.14	80.38	76.58	80.70	0.8882	0.9077	0.1286	0.2286
German	70.80	70.40	67.90	70.50	0.8014	0.7914	0.2350	0.1450
Glass	67.35	67.10	74.50	73.61	0.8718	0.8791	0.0444	0.0168
Hayes-roth	80.86	69.15	76.07	35.70	0.8544	0.8384	0.2500	0.1000
Hepatitis	82.58	79.33	79.50	82.04	0.9262	0.9226	0.5368	0.4263
Housevotes	94.48	93.79	90.78	91.24	0.9387	0.9410	0.3500	0.0188
Iris	96.00	94.67	93.33	93.33	0.9511	0.9481	0.1250	0.0000
Mammographic	80.65	79.50	75.76	76.38	0.8322	0.8229	0.0000	0.3396
Newthyroid	96.77	98.16	97.23	97.23	0.9473	0.9571	0.0600	0.0000
Pima	74.80	72.26	70.33	70.33	0.7911	0.8187	0.0000	0.0000
Sonar	80.76	75.45	81.69	85.55	0.8899	0.8595	0.2900	0.7183
Tic-tac-toe	78.29	78.71	73.07	73.07	0.8655	0.7917	0.0000	0.0000
Wine	97.19	92.68	95.49	95.52	0.9451	0.9538	0.3308	0.5231
Wisconsin	96.42	96.13	95.57	95.57	0.9103	0.9027	0.0444	0.0000
Zoo	96.39	94.22	96.50	92.81	0.8634	0.8714	0.2125	0.2750
Average	80.67	78.63	78.12	76.55	0.8763	0.8745	0.1439	0.1673

Table 4. Wilcoxon Signed-Ranks Test results

Comparison	R^+	R^-	P-value
EIS-RFS vs IS-SSGA	188	22	0.0010
EIS-RFS vs FS-RST	183	27	0.0023
EIS-RFS vs 1-NN	174	36	0.0083

These results confirm the benefits of the hybridization of fuzzy-RST based FS and evolutionary IS techniques as a whole, highlighting EIS-RFS as a suitable preprocessing method for greatly reducing the size of the training set (>87%, on average) and enhancing the accuracy of the 1-NN classifier.

5 Conclusions

In this work we have presented EIS-RFS, a novel approach for simultaneous IS and fuzzy-RST FS. This approach includes the features selected by a fuzzy-RST based FS method inside a evolutionary IS search process, thus combining the benefits of both techniques into a stand-alone, yet accurate, procedure.

The experimental results show that our approach improves the accuracy achieved by the considered techniques in isolation, whereas reduction rates are maintained. Nonparametric statistical procedures confirm that EIS-RFS can be considered as a suitable tool for optimizing the 1-NN classifier.

Acknowledgements

This work was supported by Project TIN2008-06681-C06-01. J. Derrac holds a FPU scholarship from Spanish Ministry of Education. Chris Cornelis would like to thank the Research Foundation - Flanders for funding his research.

References

1. Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing* (2010) (in press)
2. Cano, J.R., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. *IEEE Transactions on Evolutionary Computation* 7(6), 561–575 (2003)
3. Cornelis, C., Jensen, R., Hurtado, G., Slezak, D.: Attribute selection with fuzzy decision reducts. *Information Sciences* 180, 209–224 (2010)
4. Derrac, J., García, S., Herrera, F.: IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule. *Pattern Recognition* 43(6), 2082–2105 (2010)
5. Derrac, J., García, S., Herrera, F.: A survey on evolutionary instance selection and generation. *International Journal of Applied Metaheuristic Computing* 1(1), 60–92 (2010)
6. Eiben, A.E., Smith, J.E.: *Introduction to Evolutionary Computing*. Natural Computing. Springer, Heidelberg (2003)
7. García, S., Cano, J.R., Herrera, F.: A memetic algorithm for evolutionary prototype selection: A scaling up approach. *Pattern Recognition* 41(8), 2693–2709 (2008)
8. Liu, H., Motoda, H. (eds.): *Instance Selection and Construction for Data Mining*. The Springer International Series in Engineering and Computer Science. Springer, Heidelberg (2001)
9. Liu, H., Motoda, H. (eds.): *Computational Methods of Feature Selection*. Chapman & Hall/Crc Data Mining and Knowledge Discovery Series. Chapman & Hall/CRC, Boca Raton (2007)
10. Pawlak, Z., Skowron, A.: Rudiments of rough sets. *Information Sciences* 177, 3–27 (2007)
11. Wilson, D.R., Martinez, T.: Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research* 6, 1–34 (1997)

Forecasting Based on Short Time Series Using ANNs and Grey Theory – Some Basic Comparisons

Jelena Milojković¹, Vančo Litovski¹, Octavio Nieto-Taladriz²,
and Slobodan Bojanić²

¹ University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva
14, 18000 Niš, Serbia

² Universidad Politécnica de Madrid, ETSIT, Avenida Complutense nº 30,
28040 Madrid, Spain

{jelena,vanco}@elfak.ni.ac.rs,
{nieto,slobodan}@die.upm.es

Abstract. Two modern forecasting methods based on short time series are compared. Results obtained by use of artificial neural nets (ANNs), are contrasted to the ones produced by use of the so called grey theory or Grey Model (GM). Specifically, the Feed-Forward Accommodated for Prediction (FFAP) and the Time Controlled Recurrent (TCR) ANNs are used along with the GM(1,1) algorithm for one- and two-steps-ahead forecasting of various quantities (electricity loads, number of fixed telephones lines, obsolete computers, etc). Advantages of the ANN concept are observed.

Keywords: ANNs, Grey theory, prediction, forecasting, short time series.

1 Introduction

Two ANN structures convenient for prediction based on short time series were introduced in our recent study [1]. Namely, the FFAP and the TCR structure were successfully implemented to predict the quantity of obsolete computers in the USA. These ideas were later applied for forecasting electricity loads [2] and microelectronic development [3], [4]. Important collorary of these results was the experience that the claim “One major criticism about the NN model is that it demands a great deal of training data and relatively long training period for robust generalization” [5] may be easily denied. ANNs with small numbers of hidden neurons were used and the forecasts were produced based on reduced sets of data.

In [1], [2], [3], [4], the comparisons were given for some other forecasting methods but no discussion of the Grey theory [6], [7] and, accordingly, no adequate comparisons, were published.

Here we compare the forecasting results obtained by FFAP and TCR ANNs, with the Grey Model GM(1,1). One- and two-steps-ahead forecasting of various quantities based on reduced information will be demonstrated. A set of examples of implementation taken from a variety of fields will be presented preceded by short introductory information for all concepts.

2 Statement of the Problem

A time series is a number of observations that are taken consecutively in time. A time series that can be predicted precisely is called deterministic, while a time series that has future elements which can be partly determined using previous values is said to be stochastic, where the exact value cannot be predicted [8]. We are here addressing only deterministic type of time series.

Consider a scalar time series denoted by y^i , $i=1,2, \dots n$. It represents a set of observables of an unknown function, $\hat{y} = \hat{f}(t)$ taken at equidistant time instants separated by the interval Δt i.e. $t^{i+1}=t^i+\Delta t$. k -step ahead forecasting means to find such a function $f(\cdot)$ that will perform the mapping

$$y^{n+k} = f(t^{n+k}) = \hat{y}^{n+k} + \varepsilon, \quad (1)$$

The prediction of a time series is synonymous with modeling of the hidden underlying physical process responsible for its generation [9]. This is the reason of the difficulty of the task. There have been many attempts to find solution to the problem. One claims that the method based on the so called Grey Theory is favorable alternative to all ideas including the non-classic methods based on use of artificial neural networks.

It is our intention here to study the results of our implementation of ANNs and of the basic GM(1,1) model in order to get information about potential advantages and disadvantages of these concepts.

3 ANN Application to Forecasting

In the past decades ANNs have emerged as a technology with a great promise for identifying and modeling data patterns that are not easily discernible by traditional methods. A common feature, however, of the existing application was that they ask for a relatively long time series to become effective. Typically it should be not shorter than 50 data points [10]. This is due to the fact that they all were looking for periodicity within the data.

Having that in mind we were forced to look for alternative method to get efficient forecasting. These were reported in [1,2,3,4] and may be characterized as follows: use of simple ANNs (only several neurons in the hidden layer, between one and ten), and use of short time series. There are two reasons for the later. First, in many cases, some of which will be shown here, the observation were taken only recently so that there are no more data available. Second, only the most recent data are of prime importance in the behavior of many natural and social phenomena. In the next, we will briefly describe our ANN solution as given in [1,2,3,4].

The time series under consideration is in fact a set of samples given per year while the values are expressed in numbers. To make the forecasting problem numerically feasible we performed transformation (reduction) of both the time variable and the response. The time was reduced by t_0-1 so that

$$t=t^*-(t_0-1) . \tag{2}$$

Having in mind that t^* stands for the year, this reduction gives the value of 1 to the year (t_0) related to the first sample. The samples are normalized in the following way

$$y=y^* / M \tag{3}$$

where y^* stands for the current value of the target function, and M is a constant appropriately selected for the problem at hand.

According to [11], putting $t_0=1991$, after normalization, we get Table 1 as the set of observables representing the quantities of obsolete computers in the USA in millions of cubic feet. These data were used for developing our ANN based methods in [1].

Table 1. The complete available set of data after normalization

t	1	2	3	4	5	6	7	8	9
$f(t)$	7.03	8.67	10.0	9.33	9.85	10.18	12.54	14.76	18.4

The first eight samples were used as training data while the last one i.e. $t=9$ and $f(t)=18.4$, was compared with the obtained predictions in order to validate the new methods. In the following the two solutions considered most successful in [1] will be shortly described.

3.1 Implementation of Time Controlled Recurrent ANNs

As often stated in the literature [9], a promising architecture for forecasting is the time-delayed recurrent ANN. An architecture that is recurrent and time delayed and, in the same time, insists on the time variable to control the predicted value is depicted in Fig. 1. We will refer to this architecture as the time-controlled recurrent (TCR). Here the network is learning the set consisting of the following pairs $\{[t^i, f(t^{i-j}), j=0, \dots, q]; [f(t^{i+j}), j=1, \dots, k]\}$, $i=q, \dots, n-k$, so that:

$$\{y^{i+k}, \dots, y^{i+j}, \dots, y^{i+1}\} = f(t^i, y^i, y^{i-1}, \dots, y^{i-q}), \quad i=q, \dots, n-k, \tag{4}$$

where k denotes the number of intervals in future and q being the number of old values presented to the network. Note that the learning procedure here was implemented exactly in the same way as in [12]. For this case we got $f(9)=17.2114$ for a network with ten hidden neurons.

3.2 Feed Forward ANNs Accommodated to Prediction

Our idea was here to force the neural network to learn the same mapping several times simultaneously but delayed in time. In that way, we suppose the previous responses of the function will have larger influence on the $f(t)$ mapping than in the TCR network.

The architecture is depicted in Fig. 2. There is one input terminal that, in our case, is t^i . The $Output_3$ terminal or the future terminal, in our case, is to be forced to approximate y^{i+1} . In cases where multiple-step prediction is planned, $Output_3$ may be seen as a k -vector. $Output_2$ should represent the present value i.e. y^i . Finally, $Output_1$ should learn the past value i.e. y^{i-1} . Again, if one wants to control the mapping by a set of previous values $Output_1$ may be seen as a q -vector. We will refer to this architecture as the feed forward accommodated for prediction (FFAP).

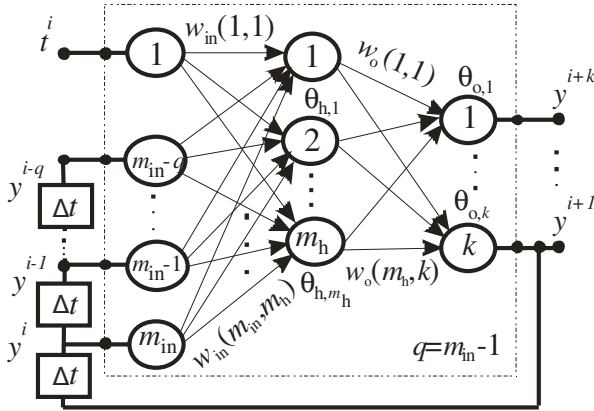


Fig. 1. Time controlled recurrent ANN

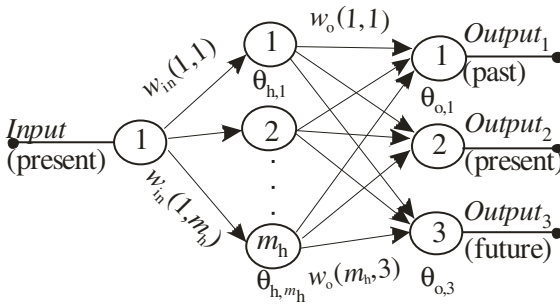


Fig. 2. ANN structure for the feed forward accommodated for prediction

The functionality of the network in our example is

$$\{y^{i+k}, \dots, y^{i+1}, y^i, y^{i-1}, \dots, y^{i-q}\} = f(t^i) \quad i = q, \dots, n-k, \tag{5}$$

where $Output_1 = \{y^{i-1}, \dots, y^{i-q}\}$ and $Output_3 = \{y^{i+1}, \dots, y^{i+k}\}$.

For this architecture, using $q=2$ and $k=1$, we got $f(9) = 18.2274$ for a network with four hidden and four output neurons.

4 GM(n,m) Model

In grey systems theory, GM(n,m) denotes a grey model, where n is the order of the difference equation and m is the number of variables. The most popular version is the GM(1,1) model, pronounced as “Grey Model First Order One Variable” thanks to its computational efficiency [13]. The GM(1,1) model can only be used in positive data sequences.

In order to smooth the randomness, the primitive data obtained from the system to form the GM(1,1) is subjected to an operator, named Accumulating Generation Operator (AGO). A differential equation (i.e. GM(1,1)) is solved to obtain the k-step ahead predicted value of the system. Finally, using the predicted value, the Inverse AGO (IAGO) is applied to find the predicted values of original data. In the next, a short version of the development of the GM(1,1) model will be given and its implementation to the series depicted in Table 1 will be performed.

Consider a non-negative time sequence X (as depicted in Fig. 3a.)

$$X = \{x(1), x(2), \dots, x(n)\}, \quad n \geq 4. \tag{6}$$

When this sequence is subjected to the AGO, the following is obtained

$$Y = \{y(1), y(2), \dots, y(n)\}, \quad n \geq 4, \tag{7}$$

where

$$y(j) = \sum_{i=1}^j x(i) \quad j = 1,2,3,\dots,n. \tag{8}$$

Y is monotonically increasing as shown in Fig. 3b.

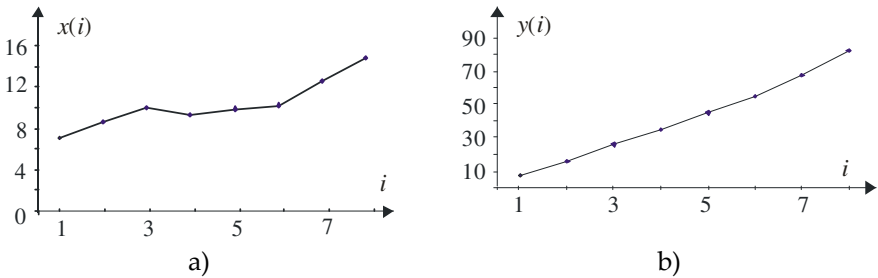


Fig. 3. a) Quantities of obsolete computers in the USA and b) the accumulated data set

The generated mean sequence Z of Y is defined as:

$$Z = \{z(1), z(2), \dots, z(n)\}, \tag{9}$$

where z(j) is the mean value of adjacent data, i.e.

$$z(j) = 0.5 \cdot [y(j) + y(j-1)], \quad j = 2, 3, \dots, n. \tag{10}$$

The whitening equation is as follows:

$$\frac{dy}{dt} + a \cdot y(t) = b \tag{11a}$$

It is related to “the least square estimate sequence of the grey difference equation of GM(1,1)” [7]:

$$x(j)+a \cdot z(j)=b \tag{11b}$$

Above $[a \ b]^T$ can be found from:

$$(\mathbf{B}^T \mathbf{B})[a \ b]^T = \mathbf{B}^T \mathbf{R} \tag{12}$$

where

$$\mathbf{R} = [x(2) \ x(3) \ \dots \ x(n)]^T \tag{13}$$

$$\mathbf{B} = \begin{bmatrix} -z(2) & -z(3) & \dots & -z(n) \\ 1 & 1 & \dots & 1 \end{bmatrix}^T \tag{14}$$

According to (11a), the solution of $y(t)$ at time j :

$$y_p(j+1) = [x(1) - b/a] \cdot e^{-aj} + b/a \tag{15}$$

To obtain the predicted value of the primitive data at time $(j+1)$, the Inverse Accumulating Generation Operation (IAGO) is used to establish the following grey model

$$x_p(j+1) = y_p(j+1) - y_p(j) = [x(1) - b/a] \cdot (1 - e^{-a}) \cdot e^{-aj} \tag{16}$$

and the predicted value of the primitive data at time $(j+k)$:

$$x_p(j+k) = [x(1) - b/a] \cdot (1 - e^{-a}) \cdot e^{-a(j+k-1)} \tag{17}$$

Table 2. Overview of the solutions. $\hat{f}(9) = 18.4$

Solution type	No. of hidden neurons	No. of output neurons	$f(9)$	error %
TCR	10	1	17.2114	6.46
FFAP	4	4	18.2274	0.938
GM(1,1)			14.8872	19.56

Implementation of this procedure to the example from Table 1, leads to the result of $f(9)=x_p(9)=14.887$ as shown in the fourth row of Table 2. Note the solution of (12) was strongly dependent on the number of significant figures used for computations. The result above was obtained with nine and more significant figures.

The results obtained so far are summarized in Table 2. It is obvious that both TCR and FFAP ANNs exhibit incomparably better results than the GM(1,1) model. To verify this conclusion we implemented the TCR and FFAP networks to various problems from the literature where the GM(1,1) model was exercised.

5 Comparison of the Results

In [14] the problem of power demand forecast in Taiwan was considered, based on data for the period 1985-1998. The results given in Table 3 and Table 4 are in GW.

In [15] the predictions were made based on the number of fixed telephone subscribers in China data from 1989 to 2004. The results given in Table 3 and Table 4 are in 100 millions of telephones.

Finally, in [16] electric power load in certain region of Hebei province in China was forecasted based on the period 1986-1996. The results given in Table 3 and Table 4 are in GW.

The results of implementation of the TCR and FFAP ANNs together with the ones taken from literature are summarized in Table 3 and Table 4. All ANN solutions have five neurons in the hidden layer.

By inspection of the two tables we may find out that in the prediction for the first year, both ANN solutions are favorably compared to all three results given in references [14], [15], [16]. In the case of the two steps ahead prediction, the situation is slightly different. The ANN solutions are advantageous in two out of three cases.

Table 3. One step ahead prediction

Ref.	Expected	Value			Error %		
		GM(1,1)	TCR	FFAP	GM(1,1)	TCR	FFAP
[14]	131.725	137.469	126.716	131.489	-4.36	3.80	0.18
[15]	31086.8	55797.8	33983.8	33626.5	-17.95	-9.31	-8.17
[16]	2965	3098	2863.79	2842.23	-4.485	3.41	4.14

Table 4. Two steps ahead prediction

Ref.	Expected	Value			Error %		
		GM(1,1)	TCR	FFAP	GM(1,1)	TCR	FFAP
[14]	142.4	147.23	144.095	146.37	3.39	-1.19	-2.78
[15]	36666.6	55385.0	36812.3	39610.4	17.82	0.39	-8.03
[16]	3012	3122	2815.74	2796.06	3.652	6.52	7.17

6 Conclusion

A set of solutions for the forecasting problem in cases where short time series are available was investigated. Two architectures previously proposed by the authors were favorably compared to the GM(1,1) model. In addition ill-conditioning was experienced while solving equation (12) which is crucial for the GM(1,1) model.

It is important to have in mind that most of the ANN based solutions were obtained with networks having only five neurons in the hidden layer. That is opposed to the claims that ANN solutions are complex and need large amount of training data [5].

References

1. Milojković, J., Litovski, V.: Comparison of Some ANN Based Forecasting Methods Implemented on Short Time Series. In: 9th Symp. on Neural Network Applications in Electrical Eng., NEUREL-2008, Belgrade, pp. 179–179 (2008)
2. Milojković, J., Litovski, V.: Dynamic Short-Term Forecasting of Electricity Load Using Feed-Forward ANNs. *Int. J. of Eng. Intelligent Systems for Electrical Eng. and Communication* 17(1), 38–48 (2009)
3. Milojković, J., Litovski, V.: Short Term Forecasting in Electronics. *Int. J. of Electronics* 98(2), 161–172 (2011)
4. Milojković, J., Bojanić, S., Litovski, V.: On Prediction in Microelectronics. In: 27th Int. Conf. on Microelectronics, Niš, Serbia, pp. 275–278 (2010)
5. Jo, T.C.: The effect of virtual term generation on the neural based approaches to time series prediction. In: Proc. of the IEEE 4th Conf. on Control and Automation, Montreal, Canada, vol. 3, pp. 516–520 (2003)
6. Deng, J.L.: *Grey Control Systems*. The Publishing House of Hua Zhong Institute, Wuhan, China (1985) (in Chinese)
7. Deng, J.: Introduction to grey system theory. *The Journal of Grey System* 1(1), 1–24 (1989)
8. Hussain, A.: Physical time-series prediction using second order pipelined recurrent neural network. In: Proc. of the 2002 IEEE Int. Conf. on Artificial Intelligence Systems (ICAIS 2002), Divnomorskoe, Russia, pp. 219–223 (2002)
9. Haykin, S.: *Neural Networks, A Comprehensive Foundation*. Macmillan College Publ. Co., New York (1994)
10. Zhang, B.G., Patuwo, E., Hu, M.Y.: Forecasting with artificial neural networks: The state of the art. *Int. J. of Forecasting* 14(1), 35–62 (1998)
11. Matthews, H.S., et al.: *Disposition and End-of-Life Options for Personal Computers*, Green Design Initiative Technical Report #97-10, Carnegie Mellon University (1997)
12. Bernieri, A., D'Apuzzo, M., Sansone, L., Savastano, M.: A neural network approach for identification and fault diagnosis on dynamic systems. *IEEE Transactions on Instrumentation and Measurements* 43, 867–873 (1994)
13. Hsu, C.-C., Chen, C.-Y.: Applications of improved grey prediction model for power demand forecasting. *Energy Conversion and Management* 44, 2241–2249 (2003)
14. Hsu, C.-C., Chen, C.-Y.: Applications of improved grey prediction model for power demand forecasting. *Energy Conversion and Management* 44(14), 2241–2249 (2003)
15. Li, G.-D., Yamaguchi, X., Nagai, M.: A high precision prediction model using hybrid Grey dynamic Model. *Int. J. Learning and Change* 3(1), 92–109 (2008)
16. Cui, H.-R.: A long-term electrical power load forecasting model based on grey feed-back modification. In: Proc. of the 7th Int. Conf. on Machine Learning and Cybernetics, Kunming, Yinan, China, pp. 2198–2201 (2008)

Short-Term Wind Power Forecast Based on Cluster Analysis and Artificial Neural Networks

Javier Lorenzo¹, Juan Méndez², Modesto Castrillón¹, and Daniel Hernández¹

¹ Universidad de Las Palmas de Gran Canaria
Instituto Universitario SIANI

Campus Universitario de Tafira- 35017 Las Palmas - Spain

² Universidad de Las Palmas de Gran Canaria
Departamento de Informática y Sistemas

Campus Universitario de Tafira- 35017 Las Palmas - Spain

Abstract. In this paper an architecture for an estimator of short-term wind farm power is proposed. The estimator is made up of a Linear Machine classifier and a set of k Multilayer Perceptrons, training each one for a specific subspace of the input space. The splitting of the input dataset into the k clusters is done using a k-means technique, obtaining the equivalent Linear Machine classifier from the cluster centroids. In order to assess the accuracy of the proposed estimator, some experiments will be carried out with actual data of wind speed and power of an experimental wind farm. We also compute the output of an ideal wind turbine to enrich the dataset and estimate the performance of the estimator on one isolated turbine.

1 Introduction

Nowadays there exists a wide consensus about the global warming [6]. The implication of the human activity in this climatological phenomenon has not been proved because there exists a cyclic component too. However, it is clear that the burning of fossil fuel increases the concentration of greenhouse gases that accelerates the warming effect, hence the growing interest in the use of renewable energy sources that will reduce the emission of them. Among the available alternatives, wind is a promising one, showing a steady expansion.

Unfortunately, the wind is not constant and it can be considered as a chaotic system whose predictability is limited. This fact along with the increase in installed power capacity have made that in many countries research groups have been granted to develop forecasting systems [4,17].

Depending on the forecast horizon, models can be divided into very short-term, short-term and long-term models. In each country, the Transmission System Operator has to deal with the management of the electric system in the different control and planning levels and also with the power production schedules in power plants. So the very short-term and short-term forecasting of wind power production becomes essential [1].

The autoregressive (AR) based models such as ARMA, ARX and Box-Jenkins methods have been used historically for very short-term wind forecasting up to few hours ahead [1]. Artificial neural networks (ANN) have been also used for wind or power forecast due to their ability of dealing with non linearities unlike AR models. Mohandes et al. [14] presented a comparison between AR model and neural networks for wind speed prediction and conclude that the ANN model outperforms the AR model in both one and several days horizon. Another comparison between regression and ANN models was presented by Li et al. [11] using as input the speed and direction of the wind measured in two meteorological towers. They found that Multilayer Perceptron ANN model outperforms the best regression model, which is a 3rd degree polynomial. More recent works have also confirmed the validity of ANN models for power forecasting [13,10].

Some researchers have proposed hybrid models. Ramirez-Rosado et al. [16] compared two systems for wind power forecast: FORECAS and SGP. In FORECAS the Power Curve Model (PCM) of each turbine was obtained with a Multilayer Perceptron neural network whose inputs were numerical weather prediction forecast values. The output of the system was obtained from a neural network that combines the PCM and AR outputs. On the other hand, SGP system is made up of 12 different models: nine neural networks used to forecast in the very short-term horizon and three additional models based on Elman and Modular neural networks. Thus, depending on the forecast horizon, the models were selected to forecast the hourly mean of electric power generated. Kusiak and Li [9] presented an approach of several power predictor which were the results of a previous clustering analysis. In their work, instead of using a combination of NN for each horizon, they customize a NN for each subspace of the input domain.

The aim of this paper is to present a hybrid architecture for short-term wind power forecast. The proposed approach differs from one presented in [9] in two aspects. On the one hand, the estimation is done based on data that are easily available in a wind farm, namely, wind speed and generated power. On the other hand, a short-term power prediction scenario is considered instead of a very short-term scenario. To evaluate the validity of the proposed architecture some tests are carried out with actual data from an experimental wind farm in the North of Spain.

The paper is organized as follows: in section 2 the estimator proposed in this work is explained. Experiments are shown in section 3 and in section 4 conclusions and further works are presented.

2 Proposed Hybrid Estimator

Some works have demonstrated the validity of Multilayer Perceptrons (MLP) as function approximators of scalar continuous functions [2,8]. The aim is to find a mapping between a set of samples (\mathbf{x}_i, y_i) where $y_i = f(\mathbf{x}_i)$, being f the unknown scalar underlying function: $f : R^n \rightarrow R$. The set of all samples made up the training dataset, $D = \{(\mathbf{x}_i, y_i)\}$, under the assumption that they are independently and identically distributed.

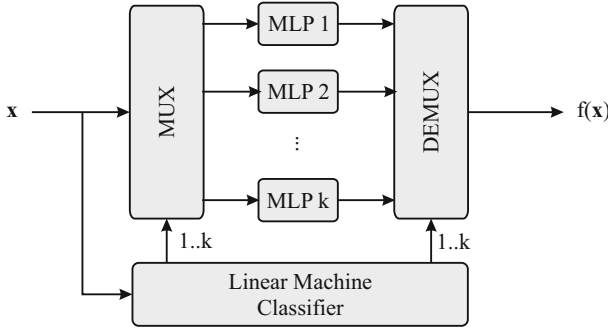


Fig. 1. Architecture of the proposed combined estimator

The set of nonlinear weight equations that relates the neural network’s adjustable parameters to the data can be obtained with optimization-based techniques [12][7][5]. These methods minimize the error function, normally the Root-mean-square error (RMSE), over the whole training dataset. If the underlying function f is too complex due to the nature of the problem as in short-term wind power prediction, the complexity of the MLP must be increased to approximate better f from the training dataset. Consequently the optimization methods can get trapped in undesirable local minima.

Instead of using the original training set, the proposed architecture is based on a splitting of the input space into subspaces under the hypothesis that customized models in each subspaces yield higher prediction accuracy. Unsupervised learning methods are designed to split a set of samples into different subsets (clusters) where the samples of each cluster are similar among them and different from the samples of other clusters. Thus the initial training dataset D is divided into k non-overlapping subsets, $D = \cup_{j=1}^k D^j$, using the k-means clustering method [3], where each subset D^j is characterized by its centroid \mathbf{Z}^j .

The clustering process is equivalent to define k classes in the input space so when the power must be estimated for an unseen sample, it must be firstly classified to feed the corresponding estimator. This classification is done with a Linear Machine classifier $d(\mathbf{x})$,

$$d(\mathbf{x}) = \max_{j=1..k} \{d^j(\mathbf{x})\} \tag{1}$$

where $d^j(\mathbf{x})$ is the linear discriminant function associated to each cluster

$$d^j(\mathbf{x}) = \omega_0^j + \sum_{l=1}^n \omega_l^j x_l^j \tag{2}$$

whose weights ω_l^j are computed from the centroids \mathbf{Z}^j as

$$\omega_0^j = -\frac{1}{2} \|\mathbf{Z}^j\|^2 \tag{3}$$

$$\omega_l^j = \mathbf{Z}_l^j; \quad l = 1 \dots n \tag{4}$$

where \mathbf{Z}_l^j is the l -th centroid element and n is the problem dimension (lag).

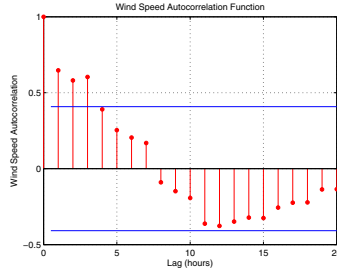


Fig. 2. Autocorrelation of the wind speed

In the wind power forecast problem, the samples \mathbf{x} are wind speed or power measured in n intervals previous to current time t . The function of interest is the estimated power at a horizon h , $\hat{p}(t+h)$. For each cluster D^j , a MLP is trained getting k estimators customized for each subspace of the input domain. The architecture of the estimator is shown in Figure 1. When the power for a new sample must be estimated, a first classification stage is carried out by the Linear Machine to decide which MLP will be in charge of the estimation.

3 Experiments

Experiments were made with actual wind speed and wind farm power data from Sotavento Galicia project. The wind speed series comprises from August 5th, 2009 until February 4th, 2010 with time steps of 10 minutes. Data were preprocessed to obtain mean hourly wind speed which yield a total of 4416 values. The data are divided into the training dataset (2/3) and test dataset (1/3).

The training dataset is used to split the input space into k clusters using the k-means algorithm, training a MLP for each cluster. After that, a Linear Machine Classifier is obtained which assigns every test sample to one of the k MLP. The output of the estimator for each test sample is the output of the MLP selected by the classifier (Figure 1).

Due to the random initialization of the k-means clustering method, 200 runs were done and the one with the lowest value for the sum of within-cluster point-to-centroid distances for the whole clusters is considered. Also to minimize the effect of the random initialization of the MLP weights, we provide the mean obtained from 25 training trials in order to reduce the uncertainty of the results.

As it is not possible to access to the power produced by only one turbine, the output of an ideal wind turbine whose transfer function has 5 and 12.5 m/sec cut-off values was included in the experiments. So, four different scenarios were considered in the experiments depending on the predicted variable and the inputs that feed the estimator:

Table 1. Number of neurons in the hidden layer of the MLP

	Prediction Horizon					
	1 hour	2 hours	3 hours	4 hours	5 hours	6 hours
<i>Scenario A</i>	4	7	6	6	9	6
<i>Scenario B</i>	8	8	7	8	6	8
<i>Scenario C</i>	2	2	1	2	2	2
<i>Scenario D</i>	2	1	4	4	6	4

Scenario A. The predicted variable is the wind farm power computed from the wind speeds, $\hat{p}(t+h) = f(v(t), v(t-1), v(t-2), \dots, v(t-n+1))$

Scenario B. The predicted variable is the wind farm power computed from previous wind farm power values, $\hat{p}(t+h) = f(p(t), p(t-1), p(t-2), \dots, p(t-n+1))$

Scenario C. The predicted variable is the ideal turbine output computed from the wind speeds, $\hat{p}_{vesta}(t+h) = f(v(t), v(t-1), v(t-2), \dots, v(t-n+1))$

Scenario D. The predicted variable is the ideal turbine output computed from previous ideal turbine outputs, $\hat{p}_{vesta}(t+h) = f(p_{vesta}(t), p_{vesta}(t-1), \dots, p_{vesta}(t-n+1))$

Instead of using the RMSE as measure to assess the accuracy of the results, the improvement over a reference model is carried out using,

$$Imp_{RMSE} = \frac{RMSE_{reference\ model} - RMSE_{proposed\ model}}{RMSE_{reference\ model}} 100\% \quad (5)$$

As reference model, the one proposed by Nielsen [15] is used which is an extension over the pure persistence model that also includes long-term information as a linear expression: $\hat{y}(t+k) = b + ay(t)$. Coefficients of the enhanced persistence model for wind farm power are: $a = 0.9487$ and $b = 37.5692$; and for ideal turbine output: $a = 0.8947$ and $b = 0.0281$.

An important parameter of the MLP topology for the experiments is the number of inputs n , that is, the lag time for the power prediction. To find it, the autocorrelation of the wind speed is obtained (Figure 2) and it is found that beyond 4 hours the influence is negligible so a lag time of 4 hours is considered. Bayesian regularization [5] was used to train the network. It uses as goal function a combination of squared errors and weights, and then determines the correct combination so as to produce a network that generalizes well.

The other parameters that define the topology of the network are the number of hidden layers and neurons in each layer. A one hidden layer topology is considered with hyperbolic tangent sigmoid activation functions in the hidden layer and linear activation function in the output layer. To decide the number of neurons, 25 training trails for 1 to 16 neurons in the hidden layer for each scenario and horizon were carried out. After that, an ANOVA statistical test was performed to find the number of neurons that yields better performance and the results are shown in Table 1.

Table 2. Imp_{RMSE} for different number of clusters and horizons (in hours) for the *Scenario A*

Prediction horizon	k = 1	k = 2	k = 3	k = 4	k = 5	k = 6	k = 7	k = 8
$h = 1$	-92%	-78%	-67%	-74%	-69%	-66%	-65%	-67%
$h = 2$	-44%	-43%	-40%	-34%	-40%	-32%	-29%	-33%
$h = 3$	-25%	-25%	-26%	-23%	-24%	-18%	-18%	-21%
$h = 4$	-16%	-17%	-19%	-20%	-14%	-11%	-11%	-12%
$h = 5$	-9%	-15%	-15%	-13%	-8%	-8%	-6%	-6%
$h = 6$	-5%	-9%	-10%	-9%	-5%	-5%	-3%	-2%

Table 3. Imp_{RMSE} for different number of clusters and horizons (in hours) for the *Scenario B*

Prediction horizon	k = 1	k = 2	k = 3	k = 4	k = 5	k = 6	k = 7	k = 8
$h = 1$	83%	93%	91%	93%	93%	90%	91%	0%
$h = 2$	84%	91%	89%	90%	92%	84%	88%	0%
$h = 3$	72%	91%	88%	90%	90%	90%	92%	91%
$h = 4$	70%	74%	70%	76%	76%	77%	79%	79%
$h = 5$	54%	66%	64%	73%	73%	73%	72%	72%
$h = 6$	52%	63%	62%	70%	70%	70%	70%	66%

Now the results of every experiment are analyzed. Table 2 shows the results obtained for the *Scenario A* where the wind speed of the previous 4 hours are used as input, $\hat{p}(t+h) = f(v(t), v(t-1), v(t-2), v(t-3))$. In this scenario the MLP model performance is lower than the reference model for all the configurations. Analyzing the effect of the cluster stage in the performance of the estimator it can be observed that the performance increases for almost all the horizons with a maximum in $k = 7$. With more clusters the performance decreases because the obtained MLP are overspecialized and the generalization capability of the estimator is worse.

Results for *Scenario B* are shown in Table 3 where the 4 previous hourly mean wind farm power values are used as predictor variables, $\hat{p}(t+h) = f(p(t), p(t-1), p(t-2), p(t-3))$. Unlike *Scenario A*, in this scenario the estimator surpasses the reference model giving better accuracy for all the horizons and number of clusters. The splitting of the input space gives better performance because for all the columns the results are higher than those of the column labeled as $k = 1$.

Tables 4 and 5 show the results considering an ideal turbine. When the power is estimated from previous wind speed (Table 4), the splitting of the input space with clustering analysis improves slightly the results obtained with the initial training dataset (column $k = 1$). On the other hand, the use of cluster analysis for the *Scenario D* makes slightly worse the results. However in these two scenarios where the output of an ideal turbine was used, the difference between the best and worst estimation is not so noticeable as in scenarios where actual data are used.

Table 4. Imp_{RMSE} for different number of clusters and horizons (in hours) for the *Scenario C*

Prediction horizon	k = 1	k = 2	k = 3	k = 4	k = 5	k = 6	k = 7	k = 8
$h = 1$	-8%	-7%	-10%	-8%	-11%	-11%	-10%	-15%
$h = 2$	-6%	-6%	-7%	-7%	-4%	-4%	-3%	-4%
$h = 3$	1%	0%	0%	-1%	-1%	-1%	-1%	-2%
$h = 4$	-1%	-2%	-1%	-3%	-3%	-2%	-3%	-3%
$h = 5$	-2%	-1%	-2%	-2%	-2%	-2%	-3%	-2%
$h = 6$	-2%	-3%	-1%	-1%	-1%	-3%	-2%	-2%

Table 5. Imp_{RMSE} for different number of clusters and horizons (in hours) for the *Scenario D*

Prediction horizon	k = 1	k = 2	k = 3	k = 4	k = 5	k = 6	k = 7	k = 8
$h = 1$	1%	0%	0%	0%	0%	0%	-1%	-2%
$h = 2$	4%	4%	4%	4%	4%	3%	3%	1%
$h = 3$	7%	6%	5%	6%	6%	5%	5%	4%
$h = 4$	8%	7%	6%	7%	6%	7%	6%	6%
$h = 5$	9%	8%	8%	8%	7%	8%	7%	7%
$h = 6$	10%	9%	8%	9%	9%	8%	8%	8%

4 Conclusions

This paper proposes a hybrid estimator for power prediction. The estimator is composed of a Linear Machine and a set of customized MLP. The Linear Machine classifies the samples into one of several subsets which has been previously obtained with a clustering analysis. The proposal has been tested on actual data from a wind farm and data from an ideal wind turbine, both for different prediction horizons.

From the obtained results it can be concluded that for the actual data, which are of interest to industry, the proposed estimator increases the estimation accuracy compared to a single MLP. On the other hand, for the data of an ideal wind turbine the proposed estimator performance is similar to a single MLP. An interesting conclusion is that the wind farm power prediction is better done when power is used as predicting variable instead of wind speed. Another fact that the experiments has brought up and that it is in consonance with the nature of the persistence model, it is that as the horizon goes farther the proposed estimator yields better performance.

Acknowledgements

This work has been partially supported by the Canary Islands government through projects SolSubC200801000137 and ProID2010-0062, and by the Spanish government and FEDER through the project TIN2008-06068.

References

1. Costa, A., Crespo, A., Navarro, J., Lizcano, G., Madsen, H., Feitosa, E.: A review on the young history of the wind power short-term prediction. *Renewable and Sustainable Energy Reviews* 12(6), 1725–1744 (2008)
2. Cybenko, G.: Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)* 2(4), 303–314 (1989)
3. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. Wiley Interscience, Hoboken (2000)
4. Focken, U., Lange, M., Monnich, K., Waldl, H., Beyer, H., Luig, A.: Short-term prediction of the aggregated power output of wind farms - a statistical analysis of the reduction of the prediction error by spatial smoothing. *Journal of Wind Engineering and Industrial Aerodynamics* 90, 231–246 (2002)
5. Foresee, F., Hagan, M.: Gauss-newton approximation to bayesian regularization. In: *Proceedings of the 1997 International Joint Conference on Neural Networks* (1997)
6. Hansen, J., Ruedy, R., Sato, M., Lo, K.: Global surface temperature change. *Reviews of Geophysics* 48, 1–29 (2010)
7. Haykin, S.: *Neural Networks; A comprehensive Foundation*, 1st edn. Macmillan, New York (1994)
8. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 359–366 (1989), <http://portal.acm.org/citation.cfm?id=70405.70408>
9. Kusiak, A., Li, W.: Short-term prediction of wind power with clustering approach. *Renewable Energy* 35, 2362–2369 (2010)
10. Kusiak, A., Zheng, H., Song, Z.: Wind farm power prediction: A data-mining approach. *Wind Energy* 12, 275–293 (2009)
11. Li, S., Wunsch, D.C., Ohair, E.A., Giesselmann, M.G.: Using neural networks to estimate wind turbine power generation. *IEEE Transactions on Energy Conversion* 16(3), 276–282 (2001)
12. Marquardt, D.: An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics* 11, 431–441 (1963)
13. Méndez, J., Lorenzo, J., Hernández, M.: Experiments and reference models in training neural networks for short-term wind power forecasting in electricity markets. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) *IWANN 2009*. LNCS, vol. 5517, pp. 1288–1295. Springer, Heidelberg (2009)
14. Mohandes, M.A., Rehman, S., Halawani, T.O.: A neural networks approach for wind speed prediction. *Renewable Energy* 13(3), 345–354 (1998)
15. Nielsen, T.S., Joensen, A., Madsen, H., Landberg, L., Giebel, G.: A new reference for wind power forecasting. *Wind Energy* 1(1), 29–34 (1998)
16. Ramirez-Rosado, I.J., Fernandez-Jimenez, L.A., Monteiro, C., Sousa, J., Bessa, R.: Comparison of two new short-term wind-power forecasting systems. *Renewable Energy* 34(7), 1848–1854 (2009)
17. Sánchez, I.: Short-term prediction of wind energy production. *International Journal of Forecasting* 22, 43–56 (2006)

Back Propagation with Balanced MSE Cost Function and Nearest Neighbor Editing for Handling Class Overlap and Class Imbalance*

R. Alejo¹, J.M. Sotoca², V. García², and R.M. Valdovinos³

¹ Tecnológico de Estudios Superiores de Jocotitlán
Carretera Toluca-Atlacomulco KM. 44.8, Col. Ejido de San Juan y San Agustín, 50700
Jocotitlán (Mexico)

² Institute of New Imaging Technologies, Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana (Spain)

³ Centro Universitario UAEM Valle de Chalco, Universidad Autónoma del Estado de México
Hermenegildo Galena No.3, Col. Ma. Isabel, 56615 Valle de Chalco (Mexico)

Abstract. The class imbalance problem has been considered a critical factor for designing and constructing the supervised classifiers. In the case of artificial neural networks, this complexity negatively affects the generalization process on under-represented classes. However, it has also been observed that the decrease in the performance attainable of standard learners is not directly caused by the class imbalance, but is also related with other difficulties, such as overlapping. In this work, a new empirical study for handling class overlap and class imbalance on multi-class problem is described. In order to solve this problem, we propose the joint use of editing techniques and a modified MSE cost function for MLP. This analysis was made on a remote sensing data. The experimental results demonstrate the consistency and validity of the combined strategy here proposed.

Keywords: Multi-class imbalance, Overlapping, backpropagation, cost function, editing techniques.

1 Introduction

Class imbalance constitutes one of the problems that has recently received most attention in research areas such as Machine Learning, Pattern Recognition and Data Mining. The class imbalance occurs when some classes heavily outnumber other classes. In the area of the artificial neural networks (NN) has been observed that the class imbalance problem causes important losses in the generalization capacity when the minority classes [1,2] are learned, because these are often biased towards the majority class. This issue can be found in real-world applications from Government, Industry and Academic or Scientific Area [3,4,5,6].

* Partially supported by the Spanish Ministry of Education and Science under grants CSD2007-00018, TIN2009-14205-C04-04, and by Fundacio Caixa Castello-Bancaixa under grants P1-1B2009-04 and P1-1B2009-45; SDMAIA-010 of the TESJO and 2933/2010 from the UAEM.

Research on this topic can be roughly classified into three categories: assigning different classification error costs [7], resampling the original training set, either by over-sampling the minority class and/or under-sampling the majority class until the classes are approximately equally represented [8,9], and internally biasing the discrimination-based process so as to compensate for the class imbalance [10,11].

Recently, several works have pointed out that there does not exist a direct correlation between class imbalance and the loss of performance. These studies suggest that the class imbalance is not a problem by itself, but the degradation of performance is also related to other factors, such as the degree of overlapping between classes [12,13,14].

In this paper, we propose to combine two strategies for addressing the class overlap and the class imbalance for the classification of remote sensing data. The problem is of great relevance since very few approaches to deal with this challenge. In order to face such a problem, this work focus on the joint use of editing techniques and a modification in the Mean Square Error (MSE) cost function for a Multilayer Perceptron (MLP). This approach can be considered a two-stage method. Firstly, we remove noisy and borderline samples of the majority classes by application of editing techniques. Secondly, the edited data set is used for training a MLP with a modified MSE cost function, which overcomes the class imbalance problem.

2 Methodology

2.1 A Balanced MSE Cost Function for Backpropagation Algorithm

In the multilayer perceptron neural network (MLP) the training by Backpropagation algorithm is based on minimization of a cost function. One of the most popular used cost functions is the mean-square error (MSE) between the desired d_{zi} and the actual y_{zi} outputs for each class $i = 1, 2 \dots J$,

$$E_i(U) = \frac{1}{N} \sum_{z=1}^{n_i} (d_{zi} - y_{zi})^2, \quad (1)$$

where $N = \sum_i^J n_i$ is the total training samples and n_i is the size of class i .

For a two-class problem ($J = 2$) the mean square error function can be expressed as,

$$E(U) = \sum_{i=1}^J E_i = E_1(U) + E_2(U). \quad (2)$$

If $n_1 \ll n_2$ then $E_1(U) \ll E_2(U)$ and $\|\nabla E_1(U)\| \ll \|\nabla E_2(U)\|$, consequently $\nabla E(U) \approx \nabla E_2(U)$, which means that $-\nabla E(U)$.

To obtain a balanced MSE cost function, we introduce a parameter (γ) that balance the contributions of the MSE,

$$E(U) = \sum_{i=1}^J \gamma(i)E_i = \gamma(1)E_1(U) + \gamma(2)E_2(U) \quad (3)$$

where $\gamma(1)\|\nabla E_1(U)\| \approx \gamma(2)\|\nabla E_2(U)\|$ avoiding that the minority class be ignored in the learning process. In this work, the parameter γ is defined as

$$\gamma(i) = \|\nabla E_{max}(U)\|/\|\nabla E_i(U)\|, \quad (4)$$

where $\|\nabla E_{max}(U)\|$ corresponds to the largest majority class. When γ is included in the training process, the data probability distribution is altered [11]. However, this parameter (Eq. 4) reduces the impact in the data distribution probability because the cost function value is diminished gradually. In this way, the class imbalance problem is reduced in early iterations, and later $\gamma(J)$ reduces its effect on the data distribution probability.

2.2 Editing Techniques

The editing techniques have been proposed to remove noise prototypes and possible overlap among classes from the training set. The aim is improve the classifier accuracy by producing smooth decision boundaries. One the most popular editing schemes is based on the well-know k -NN rule, which is mainly used for classification. However, this rule only take into account the distances to a number of close neighbors. Alternative concepts of neighborhood have been proposed to consider the neighbors of a sample in terms of proximity and spatial distribution (Surrounding Neighborhood).

The editing techniques was used to remove noisy samples of the majority classes but keeping all the positive examples. This task allows to improve the learning mechanics of the MLP. In next paragraphs we describe briefly basic concepts about editing algorithms.

Wilson Editing. Wilson [15] developed the Edited Nearest Neighbor (ENN) algorithm in which S starts out the same as Training Set (TS), and then each instance in S is removed if it does not agree with the majority of its k nearest neighbors (with $k=3$, typically). This eliminates noisy instances as well as close border cases producing smoother decision boundaries. Algorithmically, the ENN scheme can be expressed as follows:

1. Let $S = X$.
2. For each x_i in X do:
 - Discard x_i from S if it is misclassified using the k -NN rule with prototypes in $X - \{x_i\}$.

Editing Via Surrounding Approaches. The Nearest Centroid Neighborhood (NCN) [16] refers to a concept in which neighborhood is defined taking into account the proximity of prototypes to a given input sample and maintaining their symmetrical distribution around it. The k -Nearest Centroid Neighborhood rule(k -NCN) [17] has been proved to overcome the traditional k -NN classifier in many practical situations. The NCN Editing (NCNE) approach corresponds to slight modification of the original work of Wilson and basically consists of using the error estimated by the k -NCN classification rule.

Proximity graph editing scheme is based on the concepts of Gabriel Graph (GG) and Relative Neighborhood Graph (RNG) [18]. The method applies Wilson's editing algorithm [15] using proximity graphs (GG or RNG) for each sample instead of the Euclidean distance.

The Gabriel Graph Editing (GGE) and Relative Neighborhood Graph Editing (RNGE) can be summarized as follows: after computing the graph neighborhood of every sample in the original training set, discard those samples that are misclassified by their graph neighbors (instead of their k nearest neighbors).

These editing techniques provide some advantages in comparison to conventional methods. GGE and RNGE get some kind of information about prototypes close enough but homogeneously distributed around a given sample, which can be specially interesting to detect outliers close to the inter-class or decision boundaries. A more detailed description of GGE and RNGE can be found in [19].

2.3 Random Under-Sampling

Random under-sampling aims at balancing the data set through the random removal of negative examples. Despite its simplicity, it has empirically been shown to be one of the most effective resampling methods.

In this work, the random under-sampling is used to compare with the editing techniques, also, it was not hired to balance the training set.

3 Experimental Set-Up

In this part, a comparative was carried out among the strategies previously described to validate the methodology exposed in Section 2. The database used corresponds to remote-sensing task which is basically a multi-classification problem.

Experiments were conducted as follows:

Data set: A large data set, the Cayo data (4 bands, 11 classes) which corresponds to a spectral image with reference to a particular region in the Gulf of Mexico was employed in the experiments. The data set was transformed into five-class problems (MCayo) by joining the samples of several classes. The fourth column in Table 1 indicates the original classes that have been joined to shape the new classes. For instance, the samples of classes 1, 3, 6, 7, and 10 were combined to form the class C-01 and the original classes 2, 4 and 5 were left as C-02, C-04 and C-05, respectively.

Partitions: A stratified 10-fold cross-validation was employed.

Under-sampling strategies: random under-sampling (RUS), nearest centroid neighborhood editing (NCNE), Wilson's editing (ENN), relative neighborhood graph editing (RNGE) and Gabriel graph editing (GGE) were employed. All these techniques were applied over the majority classes. In the case of ENN and NCNE, the value of k has been set to 15 and 13, respectively.

Classifiers: We use a MLP with and without balanced cost function (Cost-MLP). Each one was trained with the back-propagation algorithm in batch mode. The following parameter settings were used: a learning rate $\eta = 0.1$ and one hidden layer with seven neurons.

Performance metrics: Overall accuracy, accuracy by class and the geometric mean of accuracies measured separately on each class were used. These measures can be easily derived from a $m \times m$ matrix confusion as that given in Table 2. Thus overall accuracy is computed as $Accuracy = \sum_{i=1}^m n_{ii}/N$, where N is the total number of samples, $Accuracy\ by\ class = n_{ii}/n_{i+}$ and the geometric mean as $g-mean = (\prod_{i=1}^m n_{ii}/n_{i+})^{\frac{1}{m}}$.

Table 1. Number of training and testing samples in each class

New Classes	Training	Test	Original Classes	%
C-01	2689	299	1,3,6,7,10	49.64
C-02	264	29	2	4.87
C-03	2055	228	8,9,11	37.93
C-04	290	32	4	5.35
C-05	120	13	5	2.21

Both MLP as Cost-MLP classifiers were trained using each original and pre-processed training data set by different editing techniques.

Table 2. Confusion matrix for a multi-class problem

Predicted Classes	Real Classes				total (n_{i+})
	1	2	...	m	
1	n_{11}	n_{12}	...	n_{1m}	n_{1+}
2	n_{21}	n_{22}	...	n_{2m}	n_{2+}
⋮	⋮	⋮		⋮	⋮
m	n_{m1}	n_{m2}	...	n_{mm}	n_{m+}
total (n_{+j})	n_{+1}	n_{+2}	...	n_{+m}	N

4 Results and Discussion

Several experiments with MCayo database were developed in the experimental process. 'Cost-MLP' denotes the balanced MSE cost function with MLP and 'TS edited' is the training set edited.

Table 3 shows the overall accuracy and the $g-mean$ obtained with the approaches previously described. We can observe that the classification accuracy is high and the $g-mean$ is low. So, the minority samples are misclassified while the samples of majority classes are well identified. When the original data set is classified with Cost-MLP both performance measures are improved.

On other hand, when the editing techniques are employed, the $g-mean$ is improved than the original training data set (without preprocessing). The classification results obtained from the joint application of editing techniques and Cost-MLP outperform the $g-mean$ with respect to apply the two techniques separately.

We observe that the RUS algorithm, although shows a slightly improvement, the editing techniques appears as the best strategies. Analyzing the percentage of reduction, higher values are obtained for the editing techniques that obtain better *g-mean*.

Table 3. Experimental results by editing the majority classes

MLP	Original	ENN	NCNE	RNGE	GGE	RUS
Accuracy	83.27(1.20)	85.50(1.43)	84.65(2.00)	84.96(1.29)	85.55(1.34)	84.95(1.46)
<i>g-mean</i>	00.00(0.00)	43.18(27.27)	65.19(12.15)	22.85(26.84)	47.06(25.94)	37.43(26.01)
Cost-MLP	Original	ENN	NCNE	RNGE	GGE	RUS
Accuracy	86.40(1.06)	83.37(2.32)	82.60(2.57)	84.64(1.84)	83.59(2.27)	86.25(1.21)
<i>g-mean</i>	69.80(3.14)	81.14(4.26)	82.26(3.73)	77.10(5.51)	82.05(4.42)	71.97(3.90)
% reduction	00.00(0.00)	25.00(0.46)	33.00(0.55)	19.00(0.19)	28.00(0.33)	33.00(0.08)

In Table 4 we can see the results of editing techniques and Cost-MLP for each class. The two first columns indicate the strategy applied and the number class. The third column we show the proportion of class elements in relation with the total samples ($ratio = n_i/N$, where n_i is the elements number of class i and N the total samples in the TS). The fourth column is the classification accuracy and the last one shows the classes with the level of confusion which is greater than 10% (the percentage of confusion appears in brackets).

Table 4. Performance on each class with the Cost-MLP

	Class	Ratio	Accuracy	% confusion (> 10 %)
Original	C-01	0.49	88.32	
	C-02	0.05	51.67	C-01 (48.26)
	C-03	0.38	93.69	
	C-04	0.06	61.76	C-01 (34.25)
	C-05	0.02	63.91	C-01 (34.14)
ENN	C-01	0.49	76.50	
	C-02	0.05	88.50	C-01 (11.16)
	C-03	0.38	93.37	
	C-04	0.06	74.61	C-01 (15.91)
	C-05	0.02	76.62	C-01 (21.73)
NCNE	C-01	0.49	74.87	
	C-02	0.05	91.40	
	C-03	0.38	92.53	
	C-04	0.06	76.91	C-01 (15.85)
	C-05	0.02	79.25	C-01 (19.17)
RNGE	C-01	0.49	81.53	
	C-02	0.05	75.39	C-01 (24.37)
	C-03	0.38	93.23	
	C-04	0.06	67.86	C-01 (24.61)
	C-05	0.02	72.93	C-01 (25.26)
GGE	C-01	0.49	76.59	
	C-02	0.05	91.60	
	C-03	0.38	93.34	
	C-04	0.06	74.88	C-01 (17.53)
	C-05	0.02	77.59	C-01 (20.30)
RUS	C-01	0.49	87.35	
	C-02	0.05	55.02	C-01 (44.98)
	C-03	0.38	93.57	
	C-04	0.06	62.68	C-01 (33.39)
	C-05	0.02	70.08	C-01 (28.27)

In table 4, we can observe that the editing techniques reduce the confusion level among classes. For example, Class 2 is overlapped with the class 1. When it is applied the editing techniques, the confusion level is diminished. In this case, the NCNE obtains a better performance.

5 Conclusion

In this paper, we analyze how to deal the class overlap and class imbalance in a multi-classification problem. The goal was study the performance of these two techniques combined: editing techniques joint balanced MSE cost function with MLP.

The experiments show that the benefits associated to inclusion a balanced MSE cost function in the training process. However, this is not enough for reducing the overlapping among classes. For that, using the edition strategies, we can reduce the overlapping problem increasing the prediction of the minority classes. In this paper, the use of both techniques is the best option to reduce the classification error and solve these kind of problems.

Future works will be addressed to investigate the potential of these editing methods applied in the hidden space of the neural network. This involves working in the space of the hidden layer and not in the feature space, such as commonly happens with the Wilson's editing and its variants.

References

1. Anand, R., Mehrotra, K., Mohan, C., Ranka, S.: An improved algorithm for neural network classification of imbalanced training sets. *IEEE Transactions on Neural Networks* 4, 962–969 (1993)
2. Ou, G., Murphey, Y.L.: Multi-class pattern classification using neural networks. *Pattern Recognition* 40(1), 4–18 (2007)
3. Al-Haddad, L., Morris, C.W., Boddy, L.: Training radial basis function neural networks: effects of training set size and imbalanced training sets. *Journal of Microbiological Methods* 43(1), 33–44 (2000)
4. He, H., Garcia, E.A.: Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering In Knowledge and Data Engineering* 21(9), 1263–1284 (2009)
5. Huang, Y.M., Hung, C.M., Jiau, H.C.: Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem. *Nonlinear Analysis: Real World Applications* 7(4), 720–747 (2006)
6. Mazurowski, M.A., Habas, P.A., Zurada, J.M., Lo, J.Y., Baker, J.A., Tourassi, G.D.: Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks* 21(2-3), 427–436 (2008)
7. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*. 18, 63–77 (2006)
8. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16, 321–357 (2002)
9. García, S., Herrera, F.: Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evolutionary Computation* 17, 275–306 (2009)

10. Anand, R., Mehrotra, K., Mohan, C., Ranka, S.: Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks* 6(1), 117–124 (1995)
11. Bruzzone, L., Serpico, S.B.: Classification of imbalanced remote-sensing data by neural networks. *Pattern Recognition Letters* 18, 1323–1328 (1997)
12. García, V., Mollineda, R.A., Sánchez, J.S.: On the k-nn performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis and Applications* 11(3), 269–280 (2008)
13. Prati, R.C., Batista, G.E.A.P.A., Monard, M.C.: Class imbalances *versus* class overlapping: An analysis of a learning system behavior. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) *MICAI 2004. LNCS (LNAI)*, vol. 2972, pp. 312–321. Springer, Heidelberg (2004)
14. Visa, S., Ralescu, A.: Learning imbalanced and overlapping classes using fuzzy sets. In: *Workshop on Learning from Imbalanced Datasets (ICML 2003)*, pp. 91–104 (2003)
15. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics* 2(4), 408–420 (1972)
16. Chaudhuri, B.B.: A new definition of neighborhood of a point in multi-dimensional space. *Pattern Recognition Letters* 17(1), 11–17 (1996)
17. Sánchez, J.S., Pla, F., Ferri, F.J.: On the use of neighbourhood-based non-parametric classifiers. *Pattern Recognition Letters* 18(11-13), 1179–1186 (1997)
18. Jaromczyk, J.W., Toussaint, G.T.: Relative neighborhood graphs and their relatives. *Proceedings of the IEEE* 80(9), 1502–1517 (1992)
19. Sánchez, J.S., Pla, F., Ferri, F.J.: Prototype selection for the nearest neighbour rule through proximity graphs. *Pattern Recognition Letters* 18(6), 507–513 (1997)

Combination of GA and ANN to High Accuracy of Polarimetric SAR Data Classification

Ataollah Haddadi G. and Mahmoodreza Sahebi

Faculty of Geomatics, K.N. Toosi University of Technology, No. 1346, Mirdamad Cross,
ValiAsr St., Tehran, Iran, Postal Code: 1996715433

ata.haddadi@gmail.com,

sahebi@kntu.ac.ir

Abstract. In this paper, a combination of artificial neural network (ANN) and genetic algorithm (GA) has been proposed as a method to obtain a high accuracy in classification of polarimetric SAR data. First we extracted 57 features based on decomposition algorithms and then the best features among inputted features by use of GA-ANN were selected. The classification results of a data set, composed of different land cover elements, exhibited higher accuracy than maximum likelihood and Wishart classifier; moreover the input features were decreased to small numbers which contain sufficient information for classification of data set.

Keywords: Land covers classification, POLSAR data, genetic algorithm, neural network, decomposition algorithm.

1 Introduction

One of the remote sensing analyses is land cover classification and producing thematic map. This map is more useful for many applications like monitoring of environmental issue, change detection etc. [1]. Most of the remote sensing data appropriate to land cover production, but POLSAR data due to the possess of SAR sensor benefits, such as cloud penetration independence of sun light against of visible data and contain more information in regard to the conventional SAR sensors, are more suitable for the land cover classification [2].

In order to analyze and classify the POLSAR data, many methods are used in both supervised and unsupervised algorithms. Supervised algorithms in contrast to unsupervised algorithms need training data set [1], [3]; moreover different decomposition algorithms have been proposed for interpretation and feature production of classification algorithms [2]. Beside decomposition algorithms, some of the researchers proposed parameters that may contain information which are useful to analysis of POLSAR data [4], [5]. All kinds of these features and parameters can be used as input of classification algorithms; however generally none of algorithms are able to make use of all the information available from POLSAR data; furthermore, there are some assumptions in all of the decomposition algorithms which limit the technique to specific targets detection or applications. In turn, each algorithm may represent certain specific features. For instance, circular and various linear polarimetric features are used in the self-organizing map algorithm [6], while entropy, anisotropy and alpha angle are applied in the unsupervised

algorithm based on dominant scattering in different zones of H/A/Alpha space [3],[7], [8].And the degree of polarization and co-polarization phase difference was used in unsupervised algorithms based on defined thresholds applied for class separation [9].

Statistical algorithms like maximum likelihood are not suitable for classification of POLSAR data too, because they depend on a-priority assumption, distance measure in the multidimensional space, unless the features would be separable by the algorithm [10], on the contrary there is no assumption and probability density function of the data in ANN as well there is no restriction to the dimension of the input features [11]. Due to these advantageous, many researches applied ANN extensively in the various filed of remote sensing particularly in image classification [12], [13], [14].

In this paper, we focus on finding the best features among the input features of POLSAR data to achieve the most accurate classified image. The combination of genetic algorithm and artificial neural network in other applications such as feature selection of IKONOS images were applied [15], but there is no research activity in feature selection of polarimetric SAR data.

2 Features

Classification of derived features from POLSAR data can be obtained by two different procedures: a) directly extracted features from the scattering matrix and b) extracted features from decomposed or analyzed polarimetric data. In this paper, we extracted certain features, in both categories, because there is no accurate information about the targets. Therefore, many features are extracted, and then the most suitable ones are considered as input features of classification algorithm, while all others are neglected. We found the most important decomposition algorithms which researchers had reported valid results of POLSAR classification data based on them. Table 1 illustrates the extracted features in different category. Eager researchers can find more information about these features in related paper mainly at [2].

Table 1. Extracted feature based on decomposition algorithm

Feature category	Object feature	Number of feature
Coherency matrix	Element of coherency matrix	9
Covariance matrix	Element of covariance matrix	9
Circular covariance matrix	Element of circular covariance matrix	9
Total scatter power	TP	1
Polarization Ratio	co-polarization ratio (r_co) cross-polarization ratio (r_cross)	2
eigenvalues and eigenvectors	Entropy Anisotropy23 Lambda The biggest value of eigenvalue (Lambda) Alpha angle Beta angle Delta angle Gamma angle	8

Table 1. (continued)

Feature category	Object feature	Number of feature
Polarization Phase Difference(PPD)	co-polarized phase difference ($\phi_{hh} - \phi_{vv}$) cross-polarized phase difference ($\phi_{hh} - \phi_{hv}$) cross-polarized phase difference ($\phi_{vv} - \phi_{hv}$)	3
Correlation coefficient	correlation coefficient of two co-polarization signals (Chh_vv) cross correlations with the cross polarization signal (Chh_hv) cross correlations with the cross polarization signal (Cvv_hv) circular correlation coefficient RR-LL (CCC)	4
Krogager decomposition	Deplane component: Kd Helix component: Kh Sphere component: Ks	3
Freeman and Durdan decomposition	Power scatter of Double bounce scattering: Pd Power scatter Single bounce scattering: Ps Power scatter Volume scattering: Pv	3
Other feature from decomposition algorithm	Polarization Asymmetry (PA) single bounce eignvalues relative difference (SERD) double bounce eignvalues relative difference (DERD) Anisotropy ₁₂ Radar vegetation index: RVI Pedestal height: Ph	6
Total number of features		57

All mentioned parameters are not necessarily independent; but some of them will enhance the different targets and should be considered in the classification algorithms. For this purpose the input dimension has to be decreased, ignore some features that do not contribute significantly in the classification accuracy, so that the remain features have as much information as possible. In addition, minimum numbers of features do not need more training data and training process to run the algorithm.

3 Combination of ANN and Genetic Algorithm

Neural networks are made up of organized nodes in the different layers with full connections between successive layers, called input, hidden and output layers, which receive process and present the final result [16].

In this paper the weight vectors weremodified using the training set by back-propagation algorithm. In this algorithm the error function is estimated by equation (1).

$$E^p = \frac{1}{2} \sum_{o=1}^{N_o} (d_o^p - y_o^p)^2 \tag{1}$$

Where d_o^p and y_o^p are desired output and current response of neurons o respectively and N_o is the number of neurons in the output layer. After that the weight vectors were updated by the following equation:

$$\Delta w_{jk}(t+1) = \gamma \left(-\frac{\partial E}{\partial w_{i,j}} \right) y_j^p + \alpha \Delta w_{jk}(t) \tag{2}$$

Where t indicates the presentation number and α is a constant which determines the effect of previous weight change and γ is learning rate which is between zero to one [16].

Neural network by itself cannot determine the best feature; however it can classify the data at any features as input of ANN. Genetic algorithm can supply neural network to find the best features. We use a simple binary chain, chromosome include 57 genes, each gene equivalent to each feature. The value of gens is one or zero, which one means the feature is selected in ANN. 50 initial population were procreated randomly in the beginning. The fitness function in genetic algorithm is a function to assess of chromosome performance. We defined it as the classification accuracy in terms of Kappa index, equation (3).

$$\text{Kappa} = \frac{N \sum_{i=1}^r x_{ii} - \sum_{i=1}^r (x_{i+} \cdot x_{+i})}{N^2 - \sum_{i=1}^r (x_{i+} \cdot x_{+i})} \tag{3}$$

Figure 1 illustrates the different steps of proposed algorithm. Firstly a generated chromosome transfer to selected features as input of ANN, and then ANN classify the data and the accuracy of classified image as kappa index is calculated.

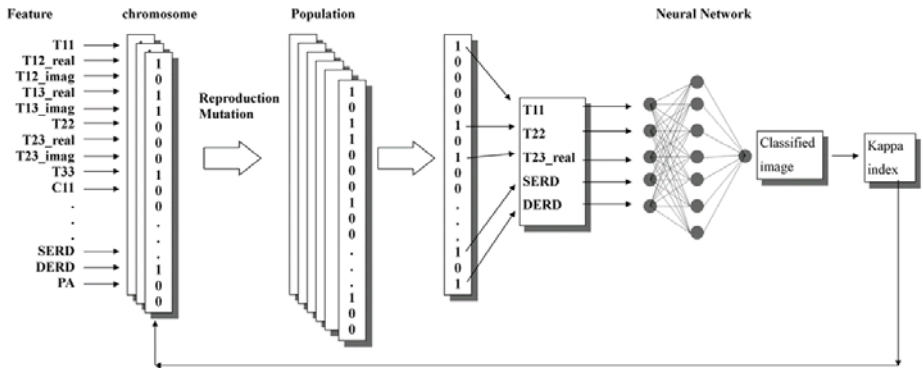


Fig. 1. The different steps of proposed algorithm include genetic algorithm and neural network

After assessment of initial population, half of the chromosomes with lower fitness values perish, and new generation is reproduced by use of remaining chromosomes. Reproduction includes crossover and mutation to evolve GA. In crossover two parents are randomly selected from the survived chromosomes and the offsprings are generated by the crossover operation and in mutation small number of all genes in the

population were selected and reversed value of genes is performed [17]. In this paper crossover was carried out by equation (4) and the 3% of all genes was considered as mutation value.

$$\begin{aligned}
 offspring_1 &= M * P_1 + ((-1) * M + 1) * P_2 \\
 offspring_2 &= M * P_2 + ((-1) * M + 1) * P_1
 \end{aligned}
 \tag{4}$$

Where m is a binary chain generate randomly, P_1 and P_2 are two chromosomes from survived individuals and $offspring_1$ and $offspring_2$ are two new chromosomes in new generation. The number of generated offsprings in this study is the same as that of extinguished chromosomes and the stop condition is reached when the 100 repetitions or convergence of GA to one fitness function.

4 Experimental Result

In order to assess the proposed algorithm, a polarimetric L- band SAR data over San Francisco, acquired by AIRSAR sensor was used.

Figure 2 shows this image in the false color composite of Pauli elements as well as the true color composite. Four different land covers were included in the scene which are forest, vegetation, ocean and urban. For training of ANN, and calculation of classification accuracy, ground truth and check data are collected from the image.

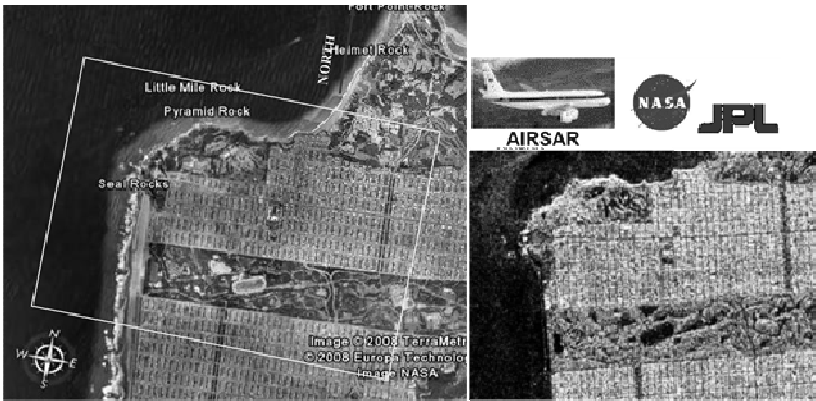


Fig. 2. Study site showing visible image (left) and polarimetric image [HH+VV, HH-VV, 2HV] as RGB channel (right)

The algorithm halted after 100 generation. Based on the proposed algorithm, the best extracted features are fifth element of coherency matrix, first element of Krogager decomposition, first and third element of Freeman decomposition, entropy, DERD and SERD. The classified image based on the best extracted features illustrated in figure 3 and the confusion matrix is presented in Table 2.

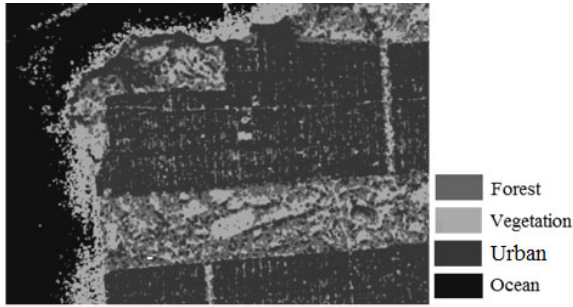


Fig. 3. Result of the proposed algorithm based on the best extracted features

Table 2. Confusion matrix of the proposed algorithm

	Urban	Forest	Vegetation	Ocean	Produce Acc.	User Acc.
Urban	251	65	0	0	79.4	83.7
Forest	49	214	0	0	81.14	76.7
Vegetation	0	0	296	13	95.8	98.7
Ocean	0	0	4	287	98.6	95.7
Overall accuracy: 88.90						
Kappa coefficient: 0.8519						

The value of 0.8519 for Kappa index shows the high performance of the proposed algorithm and both user and produce accuracy of all classes are high. It is noticeable that some pixels are remained unlabeled due to existing noise in the image. Additionally, because of existence of forest pixels in urban area, the user and produce accuracy of forest and urban classes are slightly lower than other classes.

4.1 Discussion and Comparisons

We compare the result of proposed algorithm with maximum likelihood classifier (Richard and Jia, 2006) and Wishart algorithm (lee et al., 1994). The covariance elements were considered as input features of these algorithms. Table 3 shows the result of maximum likelihood classifier and Wishart algorithm along with the result of proposed algorithm.

Table 3. The result of maximum likelihood, wishart classifier, and the proposed algorithm

algorithm	Overall Acc. (%)	Kappa index
Wishart algorithm	73.18	66.31
Maximum likelihood classifier	68.62	58.13
Proposed algorithm	88.90	85.19

According to the table 3 the accuracy of Wishart algorithm is higher than maximum likelihood classifier. Moreover, the overall accuracy of the proposed algorithm is higher than both Wishart and maximum likelihood algorithm.

5 Conclusions

We presented an algorithm to find the best features of POLSAR data as input of the classification algorithm. The proposed algorithm was consisted of genetic algorithm and artificial neural network. At first features extracted by use of various decomposition algorithms and then the most sensitive polarimetric features were determined and the image classified. It is concluded that the large amount of information does not necessarily produce better results. Additionally the result of proposed algorithm was higher than maximum likelihood and Wishart classifier.

References

1. Richards, J.A., Jia, X.: Remote sensing digital image analysis an introduction, ch. 1, 4th edn. Springer, Heidelberg (2006)
2. Lee, J.S., Pottier, E.: Polarimetric radar imaging, pp. 1–28, 53–85, 179– 224. CRC Press, Boca Raton (2009)
3. Lee, J.S., Grunes, M.R., Ainsworth, T.L., Du, L.J., Schuler, D.L.: Unsupervised classification using polarimetric decomposition and the complex wishart classifier. *IEEE Trans. Geosci. Remote Sens.* 37(5), 2249–2258 (1999)
4. Kim, Y., Van Zyl, J.J.: On the relationship between polarimetric parameters. In: Proceedings of IEEE 2000 International Geoscience and Remote Sensing Symposium, IGARSS 2000, vol. 3, pp. 1298–1300 (2000a)
5. Kim, Y., Van Zyl, J.J.: Overview of polarimetric interferometry. In: IEEE Aerospace Conference Proceedings, vol. 3, pp. 231–236 (2000b)
6. Burini, A., Putignano, C., Del Frate, F., Del Greco, M., Schiavon, G., Solimini, D.: A neural approach to unsupervised classification of very-high resolution polarimetric SAR data. In: IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2007, pp. 4164–4166 (2007)
7. Cloude, S.R., Pottier, E.: An entropy based classification scheme for land applications of polarimetric SAR. *IEEE Trans. Geosci. Remote Sens.* 35(1) (1997)
8. Famil, L.F., Pottier, E., Lee, J.S.: Unsupervised classification of multi-frequency and fully polarimetric SAR images based on the H/A/Alpha-wishart classifier. *IEEE Trans. Geosci. Remote Sens.* 39(11), 2332–2342 (2001)
9. Chang, G., Oh, Y.: Polarimetric SAR image classification based on the degree of polarization and co-polarized phase-difference statistics. In: Proceedings of Asia-Pacific Microwave Conf., APMC 2007, pp. 1–4 (2007)
10. Ersahin, K., Scheuchl, B., Cumming, I.: Incorporating Texture Information into Polarimetric Radar Classification Using Neural Networks. In: *Geosci. Remote Sens. Symposium, IGARSS 2004*, vol. 1, pp. 560–563 (2004)
11. Taner, M.T., Images, R.S.: kohonen'sself-organizing networks with "conscience", Kohonen'sself-organizing maps and their use in interpolation (1997)

12. Hara, Y., Atkins, R.G., Shin, R.T., Kong, J.A., Yueh, S.H., Kwok, R.: Application of neural networks for sea ice classification in polarimetric SAR images, radar image classification. *IEEE Trans. Geosci. Remote Sens.* 33(3), 740–748 (1995)
13. Baraldi, A., Parmiggiani, F.: A neural network for unsupervised categorization of multi valued input patterns: an application to satellite image clustering. *IEEE Trans. Geosci. Remote Sens.* 33(2), 305–316 (1995)
14. Shah, S.K., Gandhi, V.: Image classification based on textural features using artificial neural network (ANN). *IE (I) Journal-ET* 84 (2004)
15. Van Coillie, F.M.B., Verbeke, L.P.C., Wulf, R.R.D.: Feature selection by genetic algorithms in object-based classification of Ikonos imagery for forest mapping in Flanders, Belgium. *Remote Sensing of Environment* 110, 476–487 (2007)
16. Krose, B., Smagt, P.V.D.: An introduction to neural networks, 8th edn. University of Amsterdam (1996)
17. Sivanandam, S.N., Deepa, S.N.: *Introduction to Genetic Algorithms*. Springer, Heidelberg (2008)

Gradient Descent Optimization for Routing in Multistage Interconnection Networks

Mehran Ghaziasgar¹ and Armin Tavakoli Naeini²

¹ Department of Computer Science
Islamic Azad University – Majlesi Branch, Isfahan, Iran
gghmrm@yahoo.com, m.ghaziasgar@iaumajlesi.ac.ir

² McGill University – School of Computer Science
armin.tavakolinaeini@mail.mcgill.ca

Abstract. The design and development of interconnection networks is a critical issue in the field of communication switches, routers and multiprocessors. Here we consider the problem of routing in multistage interconnection networks (MIN). In our previously reported research we proposed a method to find the best path in a graph. In this paper, we use our algorithm which is based on neural network for routing in MINs. In this algorithm, we define a suitable energy function; the minimum of this function correspond to a valid route. By using gradient descent method, the energy is minimized at the convergence of neural network. Simulation results show that this method finds a valid path between source and destination and because neurons act in parallel, the performance is comparable with other approaches.

Keywords: Neural Network, Multistage Interconnection Networks, Learning Rate, Upper Trigger Point.

1 Introduction

One of the important applications of neural network is to solve optimization problems. Optimization problems are usually computationally expensive if an exhaustive search is to be performed comparing all possible solutions to find the best one. Therefore, it is important to find an acceptable solution more quickly. In many cases, a good solution found quickly is better than the best solution found too late to be of use. A model that is commonly used to solve optimization problems is called Hopfield network, introduced by J. Hopfield [1, 2]. One of the most important features of this model is that Hopfield network can be easily implemented in hardware, therefore neuron computations is done in parallel and the solution is found more quickly.

In this paper, we consider the problem of routing in multistage interconnection networks (MIN). MIN is a network of switches organized in multi levels; each switch in a level could have connections to the switches in the next level. MIN should connect the inputs of the network to the desired outputs. The design and development of interconnection networks is a critical issue in the field of communication switches,

routers and multiprocessors. You can find a comprehensive text of interconnection networks in [3]. In order for routing in a MIN, we use our method introduced to find an optimum path in a graph [4]. A neural network to solve the problem is introduced. Usually parameters of a learning algorithm are achieved by trial and error but here we analyze the behavior of neural network to suggest appropriate values for parameters. Designing neural network based on trigger points improves the behavior of the network [5]. So in the early algorithm we developed, upper trigger point (UTP) and lower trigger point (LTP) for neurons being on and off were suggested. I mean by UTP a point that if neuron's value is equal or greater than it, that neuron is considered to be on. And LTP a point that if neuron's value is equal or lower than it, that neuron is considered to be off. Simulation results show that running our network based on UTP and LTP gives ambiguity in the convergence of the algorithm, for example it was possible the a neuron reaches LTP but in the rest of the iterations it's value is added until reaching UTP. Therefore, we revised our method only based on UTP.

We have simulated this approach to see the number of iterations in which the network will converge to a reasonable state, because the neural network acts like a parallel computer, we can use this method to speed up routing. Our method is compared to other approaches for routing in MINs, and simulation results show that based on some standard measures our solution acts better and the performance is comparable with other methods.

2 Basic Concepts

In this section, we will briefly describe the gradient descent method that is used to minimize energy in neural networks, and then the basic structure of a multistage interconnection network will be discussed. In [6] different methods of learning algorithms and in [3] interconnection networks have been described widely.

2.1 Gradient Descent Method

In order to solve an optimization problem an energy function based on the constraints of a problem is defined. After the definition, gradient descent method can be used to minimize the energy. Assume the energy is a function of n variables u_1 to u_n :

$$E(\mathbf{u}) = E(u_1, u_2, \dots, u_n). \quad (1)$$

In which E is energy function. The gradient vector of E is as follows:

$$\frac{\partial E}{\partial \mathbf{u}} = \left(\frac{\partial E}{\partial u_1}, \frac{\partial E}{\partial u_2}, \dots, \frac{\partial E}{\partial u_n} \right). \quad (2)$$

In order to converge to minimum of E , if each component of gradient vector is positive, the variable of that component is decreased. Moreover, if each component of gradient vector is negative the variable of that component is increased.

It means that we should decrease a variable in which the gradient component of that variable is positive. Alternatively, we should increase a variable in which the gradient component of that variable is negative. Using this method the following recursive equation, to change the variables is created:

$$\mathbf{u}_{i+1} = \mathbf{u}_i - \alpha \frac{\partial E}{\partial \mathbf{u}_i} \Rightarrow \Delta \mathbf{u}_i = -\alpha \frac{\partial E}{\partial \mathbf{u}_i}. \tag{3}$$

In the above equation α is a positive parameter that is called the learning rate. \mathbf{u}_i shows the value of variables in the i th iteration of executing the algorithm. The algorithm is started using some initial value, \mathbf{u}_0 , and will continue step by step; using the values in step i , the values in step $i+1$ is found. The algorithm will be continued until desired constrains are met.

2.2 Multistage Interconnection Networks

A multistage interconnection network is composed of some $a \times b$ switches. Each $a \times b$ switch has a input lines, (the number of inputs to the switch equals to a) and b output lines. In a MIN these switches are organized in different levels, a switch in a level could (but not necessarily) have a connection link to each of the switches in the next level. The inputs of the network are the inputs of the first stage, and the outputs of the network are the outputs of the last stage.

The switch can connect each of its inputs to the desired outputs as long as there is no conflict in using the links that connect switches together. In Fig. 1, you can see a four stage MIN with nine inputs and nine outputs in which the inputs and outputs are numbered from zero to eight. In this figure the input 0, is connected to the output 8. Routing in a MIN should be in a way that no conflict occurs in the links between switches. For example, two input-output pairs should not use the same link.

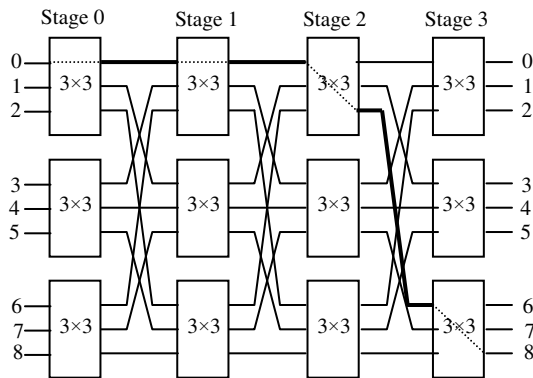


Fig. 1. A four stage MIN

3 Introducing the Solution

The structure of a MIN is closely related to neural networks. We consider a neuron for each switch in a MIN. In addition, the connections between neurons are the links of the MIN.

3.1 The Energy Function of Neural Network

To define an energy function in a K stage network the stages are numbered from zero to $K-1$. We have one neuron for each of the nodes in our multistage network, and in each stage neurons are numbered from zero to n_k-1 in which n_k is the number of neurons in k th stage. Using these definitions, the j th neuron in k th stage is named u_{kj} . $u_{(k-1)i}$ is connected to u_{kj} and the weight of this connection is w_{kij} .

Now we should take suitable values for w_{kij} ; if a route in the network uses a connection, or the connection does not exist in the structure of the MIN, we will set the value of one as the weight of that link. Nevertheless, if a connection is ready to be use of, we will set a small value lower than 0.5 as the weight of that link. The following formula satisfies the above constraints for the connection weights:

$$w_{kij} = \begin{cases} 1 & \text{if the link is occupied or do not exist} \\ (j/2n_k) & \text{otherwise.} \end{cases} \tag{4}$$

Each path in neural network uses $K-1$ links and in each of the stages, one neuron is selected for routing. If the value of a neuron, u_{kj} , is always a real number between zero and one, when the value of a neuron is zero it is off, and when the value of a neuron is one that neuron is on. At the convergence of neural network, at most, one neuron should be on in each stage of the network that shows the node to be used for routing in that stage. When no path exists from starting to ending nodes all neurons in a stage that cannot find a node for routing should be off.

A valid path should satisfy the following constrains:

1. There is no conflict in using connection links.
2. In each stage, at most one neuron is on.

Consider the following energy in which β is a positive parameter:

$$E = \sum_{k=1}^{(K-1)} \sum_{i=0}^{(n_{k+1}-1)} \sum_{j=0}^{(n_k-1)} w_{kij} u_{(k-1)i} u_{kj} + \beta \sum_{k=1}^{K-2} \left(\left(\sum_{i=0}^{n_k-1} u_{ki} \right) - 1 \right)^2 \tag{5}$$

The minimum of the first term in equation (5) occurs when the connection link between neurons $u_{(k-1)i}$ and u_{kj} , is free for use, or one of the neurons is off. The latter scenario is when the link is occupied and should not be used for the route. Therefore, if a route is valid the first term in equation (5) will be very close to zero, In other

words $w_{kij}u_{(k-1)i}u_{kj}$ shows the cost of using the neurons for a route and the summation of all of these combinations corresponds to the first constraint of the problem.

The contribution of the second term in equation (5) is zero when the sum of neurons in each stage is one. If in each layer one neuron is on, this term will be zero. Therefore, second term in equation (5) corresponds to the second constraint of the problem. By the above discussion, a valid path results a minimum energy.

4 Analyzing the Behavior of Network to Find Proper Parameters

The neural network will start its evolution with each neuron having an initial value, and a pair of input-output is given to the network to find a path for that pair. The input of the pair is the input of a switch in the first stage, the neuron corresponds to this switch should be included in the final route. The value of this neuron is one at start of the evolution, and all other neurons in the first stage are zero. These values do not change in different steps of the algorithm. The same scenario is true for the neuron corresponds to the output of the given pair. All other neurons that are not in the first or last stage are given the value of $1/n_k$. This value would suggest that initially the probability of choosing each neuron is equal, and sum of neurons in each stage is one. In the first part of Table 1 typical initial values for the network in Fig. 1 are shown.

It is clear that the maximum weight of a connection is equal to one. We can analyze the evolution of network using the method in [4] to find proper values for learning parameters. Note that the equations in [4] are based on the maximum weight of connections; so the notation M , used for maximum weight, here should be replaced by one:

$$\alpha \leq \frac{1}{2\beta n_k} \text{ and } \alpha \leq \frac{u_{kj}}{2}. \tag{6}$$

$$1/2 < \beta < 1. \tag{7}$$

In order to have quicker convergence α should be chosen as large as possible. Equation (6), propose limits for α , naturally the minimum of these limits should be as a learning rate in each iteration. By choosing α in this way, the value of neurons remain positive and the sum of neurons in a layer remain less than one, therefore the value of a neuron is always between zero and one.

4.1 Upper Trigger Point of a Neuron

Let a^+ be the upper trigger point of a neuron. If the value of a neuron is greater than a^+ it is assumed to be on. a^+ is found based on the maximum weight of a connection ready to be used for a route, according to equation (4) this weight is equal to the following formula in which N is the maximum number of neurons in a layer:

$$Max(w_{kij}) = N - 1/2N \quad N \geq 1. \tag{8}$$

The following equation can be used for the trigger point [4]:

$$a^+ = \frac{2\beta - (N - 1/N)}{2\beta}. \tag{9}$$

The value in the above equation can be used as a down limit for upper trigger point.

5 Simulation Results

In part (a) of Table 1 initial values for the network in Fig. 1 are shown, in which we have to go from the first input to the 8th output. Network evolution after 50 and 100 iterations are shown in Table 1. If β is equal to 0.9 According to equation (9) upper trigger point of a neuron is 0.63. The first switches in the second and third stages have reached the desired values and would be selected for the route. After the routing is done the corresponding weights are changed. Now consider we want to find a route from the first input to the last output, network evolution is shown in Table 2, this time the second switch in the second stage is included in the path.

In order to speed up the convergence, if only one neuron in a layer is greater than or equal to upper trigger point that neuron is more likely to be the desired node and is chosen for the path. When in all layers a node is selected, the algorithm will stop. If the algorithm does not finish in reasonable iterations we assume there is no route from input to output, the maximum number of iterations we agreed in our simulations is 100.

Consider a message set of M input-output pairs is given to the network for routing. One of the measures to evaluate the performance of a method is that how many pairs could be routed without conflict and how many should be queued. In order to route maximum pairs we have to use an exhaustive search. If all the pairs can not go through the network, a subset of the messages with maximum length will be selected for routing. It is clear that implementing exhaustive search to find the best way is very expensive but it is a measure to evaluate other methods [7, 8].

Table 1. Neuron changes

Stage 0		Stage 1		Stage 2		Stage 3	
1.00		0.33		0.33		0.00	
0.00		0.33		0.33		0.00	
0.00		0.33		0.33		1.00	
a. Initial values							
k=0	k=1	k=2	k=3	k=0	k=1	k=2	k=3
1.00	0.97	0.79	0.00	1.00	1.00	0.81	0.00
0.00	0.02	0.02	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	0.00	0.00	0.00	1.00
b. Neuron values after 50 iterations				c. Neuron values after 100 iterations			

Table 2. Neuron changes to go from the first input to the first output

Stage 0	Stage 1	Stage 2	Stage 3
1.00	0.00	1.00	1.00
0.00	0.91	0.00	0.00
0.00	0.00	0.00	0.00
Neuron values after 100 iterations			

The first column of Table 3 shows the message set length, and the second column shows the average number of messages that can be routed in the random multistage interconnection network used in [7]. I mean by random that there is no special pattern to connect switches in different stages of the network. These numbers are found by generating all possible sets of length M . Message sets with two common inputs or two common outputs are not included [7].

The third column of Table 3 shows the average number of messages that is routed by the method in [7]. To find these numbers 1000 random pairs are generated for each row of the table. The fourth column of Table 3 shows our method, and in Fig. 2 these numbers are compared. As can be seen in this figure our method acts better when we increase the length of a message set. We enhanced our method to work only based on UTP, so the results are found faster but there is no considerable change in the average number of routed messages.

Table 3. The average number of routed messages

M	E_{es}	E_{nn}	E
1	1.00	1.00	1.00
2	1.86	1.87	1.81
3	2.52	2.51	2.41
4	3.08	2.98	2.95
5	3.53	3.24	3.30
6	3.89	3.45	3.62
7	4.16	3.66	3.86
8	4.33	3.78	4.06

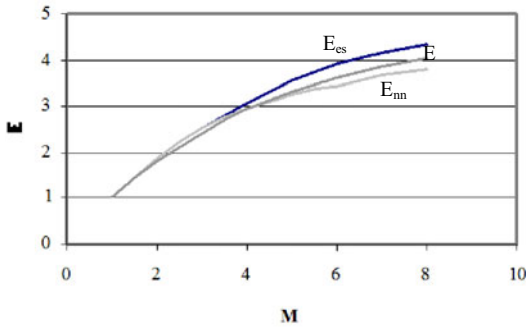


Fig. 2. Comparing the average number of routed messages

6 Conclusion

Multi stage interconnection networks are common structures used to act as a connecting interface between off-the-shelf components, so the design of high performance MINs is a critical design issue. To address this application we proposed a neural network method for routing in a multistage interconnection network. Different parameters of the algorithm were analyzed, and simulation results show that this method can find a valid route for an input-output pair. This kind of approach is flexible and can be utilized and enhanced used to solve other optimization problems.

In order to speed up the simulation and eliminating the ambiguity to end up the run, we only use the value of UTP to find the selected neuron, because neuron computations can be done in parallel the neural network is useful to speed up finding the solution, moreover using Hopfield model hardware implementation of the network is possible.

References

1. Hopfield, J.: Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Nat. Acad. Sci. USA* 81, 3088–3092 (1984)
2. Hopfield, J., Tank, D.: Neural's computation of decisions in optimization problems. *Biol. Cybern.* 52, 141–152 (1985)
3. Duato, J., Yalamanchili, S., Ni, L.: *Interconnection Networks: An Engineering Approach*. IEEE Computer Society, Los Alamitos (1997)
4. Ghaziasgar, M., Tavakoli Naeini, A.: *Neural Network for Routing in a Directed and Weighted Graph*, ISDA, vol. 1, pp. 631–636. IEEE Computer Society, Los Alamitos (2008)
5. Xia, G., Tang, Z., Li, Y., Wang, J.: A binary Hopfield neural network with hysteresis for large crossbar packet-switches. *Neurocomputing* 67, 417–425 (2005)
6. Kekman, S.: *Learning and Soft Computing Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. MIT press, Massachusetts Institute of Technology (2001)
7. Goudreau, M.W., Giles, C.L.: Neural network routing for random multistage interconnection networks. *Neurals Information Processing System* 4, 722–729 (1992)
8. Goudreau, W., Lee Giles, C.: Routing in optical multisatge interconnection networks: a neural network solution. *Journal of Light Way Technology* 13(6), 1111–1115 (1995)

The Command Control of a Two-Degree-of-Freedom Platform by Hand Gesture Moment Invariants

Chih-Lyang Hwang¹ and Chen-Han Yang²

¹ Department of Electrical Engineering, National Taiwan University of Science and Technology, Taiwan, R.O.C.

² Department of Electrical Engineering, Tamkang University, Taiwan, R.O.C.
chih-lyang_hwang@hotmail.com, blaze_0207@hotmail.com

Abstract. In this paper, the functional commands based on hand gesture are designed by the method of moment, which is invariant to the translation, rotation and scale of a hand gesture. After the transform of the original image with a hand gesture into the YCrCb coordinate, the segmentation of the skin-like object is obtained by the thresholds of Cr and Cb. Then the dilation and median filtering and the area constraint are employed to obtain an acceptable binary image. Various feature vectors corresponding to different processed hand gestures are applied to train the input weight matrix and layer weight matrix of a probability neural network for classification. Furthermore different lighting conditions verify the robustness of the image processing and classification. Finally, eight hand gestures are designed as the commands for the motion control of a 2 DOFs platform. The experiment confirms the effectiveness of the proposed method.

Keywords: Hand gesture, Image processing, Moment invariant, Probability neural network, Classification, Motion control.

1 Introduction

According to the survey of the database of IEEE/IET Electronic Library (IEL), there have 21 journal papers and 320 conference papers related with the keyword “hand gesture” from January 2009 to January 2011. Most of these papers (e.g., about 150 papers) discussed the recognition of hand gesture through skin color segmentation and different hand shapes including finger tips and finger roots. Parts of them investigated the dynamic trajectory tracking and recognition in 2D or 3D coordinate. Some papers were the application to human-computer interface (HCI), e.g., electronics consumer, game, intelligent space, the control interface of mobile robot. HCI is emerged as a new field with the motivation to bridge the communication gaps among the humans (or hand gestures) and computers (or embedded vision system).

There was a survey paper about the visual interpretation and recognition of hand gesture [1]. 3D hand models offered a way of more elaborate modeling of hand gestures but lead to computational hurdles that had not been overcome given the real-time requirements of HCI. Appearance-based models led to computationally efficient approaches that worked well under constrained situations but seemed to lack

the generality desirable for HCI. Besides the above survey paper, the relative papers about the recognition of hand gesture were introduced as follows. The paper developed by Rashid et al. [2] provided interaction through gesture and posture recognition for alphabets and numbers. The two-layered Bayesian network developed by Roh et al. [3] provided robust recognition of hand gestures, as it compensated for an incorrectly recognized hand posture and its location via the preceding and following information. To recognize hand postures, an algorithm based on skin color segmentation and geometric invariant feature had been used. The optical flow tracking approach and direction encoding were adopted to identify dynamic gestures. The implementation of high reliable algorithm on DSP board kept the system robust and efficient [4]. In addition, a YCrCb active model for the hand segmentation and then hand feature extracted by calculating the seven moments were accomplished by Tan et al. [5]. In [6], a skin-color model to adapt different users' skin color was used. Then the use of density distribution feature and the shape features with respect to different distance were employed to recognize hand gestures. A six degree-of-freedom virtual mouse based on hand gesture was presented by an inexpensive USB webcam. The hand tracking and gesture recognition framework included the motion, skin color and finger information of the hand [7]. According to the temporal characteristic and the spatial characteristic of video sequence, a recognition method of sign language spatio-temporal appearance modeling was introduced for the vision-based multi-features classifier of Chinese sign language recognition [8].

In this paper, the functional commands based on hand gesture are designed by the moment method, possessing the invariants of translation, rotation and scale. At beginning, the original image with a hand gesture is transformed into the color space of YCrCb. The skin-like segmentation is obtained by the thresholds of Cr and Cb in a specific region; then the dilation and median filtering and the shape selection are applied to obtain an acceptable binary image. After that various feature vectors related to processed hand gestures are used to train the input weight matrix and layer weight matrix of a probability neural network for classification. Moreover different lighting conditions are applied to verify the robustness of image processing and classification. Finally, eight hand gestures are designed as the commands for the motion control of a 2 DOFs platform.

2 Experimental Setup and Task Description

Two important parts constitute the overall system: (i) an embedded vision system (EVS) for the capture of an image, the image processing and classification of a hand gesture, (ii) an embedded system for the execution of assigned command and the feedback of the complete execution of the assigned command. The EVS includes an embedded system of PICO820 of 1.6 GHz and a webcam of VX5000. Another embedded system RB-100 of 1GHz. The 2 DOFs platform driven by two servo motors is applied for the motion control of the pitch and yaw directions of the head of a humanoid robot.

In this paper, the following tasks are designed and accomplished: (i) the image of a hand gesture, which is in the distance between 80 and 160 cm from a webcam, is first captured by the embedded vision system; (ii) the image processing based on skin

color segmentation and connected component labeling are obtained; (iii) a method of moment invariant is employed to calculate the feature vectors of various hand gestures; (iv) after the achievement of these feature vectors, the training of probability neural network is applied to classify 8 hand gestures; (v) then, the design of various commands for different classes of hand gesture is considered; (vi) finally, the experiment for the motion control of a 2 DOFs platform is executed to confirm the usefulness of the proposed control system.

3 Eight Hand Gestures and Their Possible Position

At beginning, 8 hand gestures in Fig. 1 are selected for the design of various commands. According to their specific feature vectors, they must be easily separated in different lighting conditions or in the presence of noise to obtain a robust recognition of hand gesture. Based on our test of the webcam, the recognizable distance for the resolution 1024x768 is about 3 meters. However, for the reduction of the image processing time the resolution of 320x240 is suggested to capture a hand gesture in the range of 80~160 cm from the webcam. In addition, the areas and perimeters for 8 hand gestures in this range are investigated. Then according to the lower and upper bounds of these areas or perimeters, i.e., 370 and 2800 pixels or 70 and 300 pixels, the miss-recognition of a hand gesture can be avoided or the noise removal can be fast.

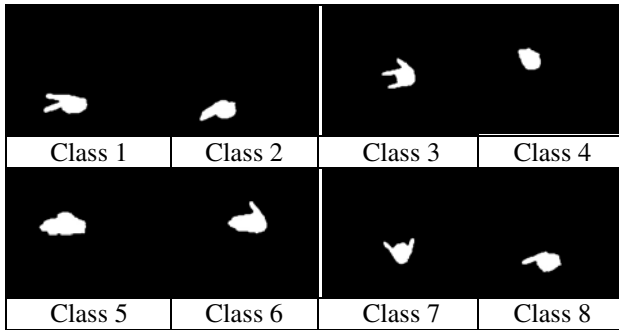


Fig. 1. The 8 hand gestures are chosen for the design of various commands

4 Image Processing for Classification

In the beginning, the flow chart of the image processing is addressed in Fig. 2, containing the following six parts: (i) image inquiry, (ii) the coordinate transform of RGB into YCrCb, (iii) the separation of YCrCb into three components, (iv) the thresholds of Cr and Cb for the skin color segmentation, (v) merging the above three components and then obtaining the binary image, (vi) the noise removal including dilation, median filtering and shape selection.

According to our test of skin color and previous studies [5], [6], the segmentation of skin color using YCrCb color space is better than that of HSV. Because Y

component is affected by the illumination, it is discarded. On other hand, the Cr and Cb components of skin color are segmented by the following threshold:

$$F = \begin{cases} 1, & \text{if } C_r \in [133,147] \text{ and } C_b \in [122,137] \\ 0, & \text{others} \end{cases} \quad (1)$$

where F denotes a binary pixel. Based the above threshold (1), the original image with skin color in Fig. 3(a) becomes a binary image as shown in Fig. 3(b). Because the result of Fig. 3(b) is still not satisfactory for the extraction of feature of hand gesture, a dilation and then median filtering are employed to obtain an acceptable binary image for the calculation of moment (see Fig. 3(c)). The reason to first use a dilation operation is that the segmentation of skin color in a far distance is often small and therefore it is not suitable for the use of an erosion operation.

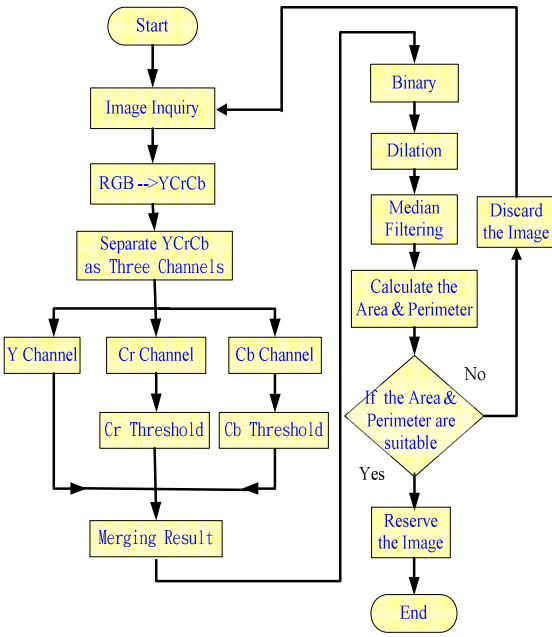
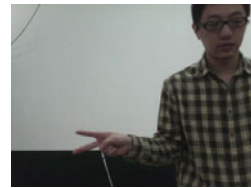


Fig. 2. The flowchart of image processing



(a) Original image.



(b) Binary image.



(c) Dilation & median filtering.

Fig. 3. Image processing

Then after using the improved recognition of a hand gesture by area constraint, the moment of order $(p+q)$ for the processed image (i.e., a pure binary image of a hand gesture) is calculated as follows:

$$m_{pq} = \sum_{i=1}^M \sum_{j=1}^N i^p j^q f(i, j) . \quad (2)$$

Then the central moment of order $(p+q)$ is given as follows:

$$U_{pq} = \sum_{i=1}^M \sum_{j=1}^N (i - \bar{i})^p (j - \bar{j})^q f(i, j) \quad (3)$$

where $\bar{i} = m_{10}/m_{00}$, $\bar{j} = m_{01}/m_{00}$. The normalization of central moment is defined as follows:

$$\eta_{pq} = U_{pq} / U_{00}^\lambda \tag{4}$$

where $\lambda = 0.5(p + q) + 1$. Then 7 moment characteristics are calculated as follows:

$$\begin{aligned} \varphi_1 &= \eta_{20} + \eta_{02} \\ \varphi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \varphi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \varphi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \varphi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] \\ &\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + 3\eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] \\ \varphi_6 &= (\eta_{20} - \eta_{02}) \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\ \varphi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] \\ &\quad - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] \end{aligned} \tag{5}$$

5 Probability Neural Network for the Classification

It is first assumed that a classification problem possesses K classes, e. g., C_1, C_2, \dots, C_K . The related rule for the classification is determined by m -dimensional feature vectors: $X = (X_1, X_2, \dots, X_m)$. That is, in these m -dimensional samples space, the probability density function of each class, i.e., $f_1(X), f_2(X), \dots, f_K(X)$, is function of these feature vectors. The decision formulation of Bayesian classification is then expressed as follows:

$$h_i c_i f_i(X) > h_j c_j f_j(X), \quad \forall j \neq i \tag{6}$$

where f_i denotes the i th probability density function, c_i is the value of cost function for misclassification of the i th class, h_i denotes the prior probability of the i th class. Theoretically, we can use the formula (6) to deal with the problem of classification. However, it is difficult to obtain the probability density function (PDF) of trained data in advance. Therefore, we must assume that the trained data satisfy the specific condition of PDF (e.g., normal distribution); then the collected data is employed to train the parameters of these PDFs. In summary, the architecture of PNN is depicted in Fig. 4, where $P = [\bar{\varphi}_1 \ \bar{\varphi}_2 \ \bar{\varphi}_3 \ \bar{\varphi}_4]^T \in \mathfrak{R}^N$ is the normalized feature vector (i.e., $\bar{\varphi}_i = \varphi_i / (\varphi_{i,\max} - \varphi_{i,\min}), i = 1, 2, 3, 4$) the output $a^2 = [C_1 \ C_2 \ \dots \ C_8]^T$, the number of hidden layer weights, i.e., $Q=30$, $IW^{1,1}$ and $LW^{1,2}$ respectively denote the input weight matrix and layer weight matrix; $\|dist\|$ denotes the Euclidean norm distance, n^i , $i=1,2$, is the activation signal of the i th layer, and $a^1 = e^{-(n^i)^2}$ is the Gaussian distribution function.

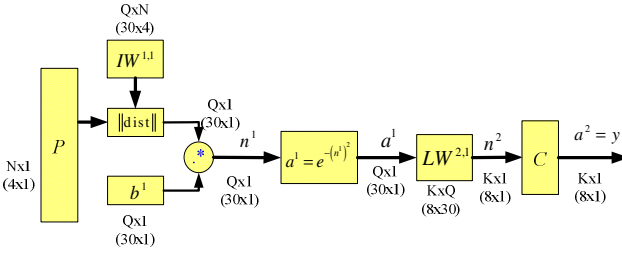


Fig. 4. Architecture of PNN

Table 1. Recognition rate in the turn-on of 8 lamps

Class \ Result	C1	C2	C3	C4	C5	C6	C7	C8
C1	100	0	0	0	0	0	0	0
C2	0	100	0	0	0	0	0	0
C3	0	0	100	0	0	0	0	0
C4	0	0	0	100	0	0	0	0
C5	0	0	0	0	100	0	0	0
C6	0	0	0	0	0	100	0	0
C7	0	0	0	0	0	0	100	0
C8	0	0	0	0	0	0	0	100
Accuracy(%)	100	100	100	100	100	100	100	100

Table 2. Recognition rate in the turn-on of 4 lamps

Class \ Result	C1	C2	C3	C4	C5	C6	C7	C8
C1	93	0	0	0	0	0	0	0
C2	7	100	0	0	0	0	0	0
C3	0	0	100	0	0	0	0	0
C4	0	0	0	97	0	0	0	0
C5	0	0	0	2	100	0	0	0
C6	0	0	0	1	0	100	0	0
C7	0	0	0	0	0	0	100	0
C8	0	0	0	0	0	0	0	100
Accuracy(%)	93	100	100	97	100	100	100	100

Table 3. Recognition rate in the turn-on of 2 lamps

Class \ Result	C1	C2	C3	C4	C5	C6	C7	C8
C1	58	0	0	0	0	0	0	0
C2	20	47	0	0	0	0	0	0
C3	0	0	83	0	0	0	0	0
C4	0	0	2	95	0	0	0	0
C5	0	0	1	5	100	0	0	0
C6	0	0	0	0	0	100	0	0
C7	0	0	14	0	0	0	100	0
C8	22	53	0	0	0	0	0	100
Accuracy(%)	58	47	83	95	100	100	100	100

From the result of Tables 1, 2 and 3, the following results are concluded. (i) classes 5, 6, 7 and 8 are absolutely acceptable for different lighting conditions because the moment characteristics of these four classes separate enough, (ii) classes 3 and 4 are acceptable for different lighting conditions, and (iii) classes 1 and 2 are poor for weak lighting condition because the moment characteristics of three classes 1, 2 and 8 do not separate enough, (iv) in fact, the recognition rate is acceptable until only 2 lamps are lighted.

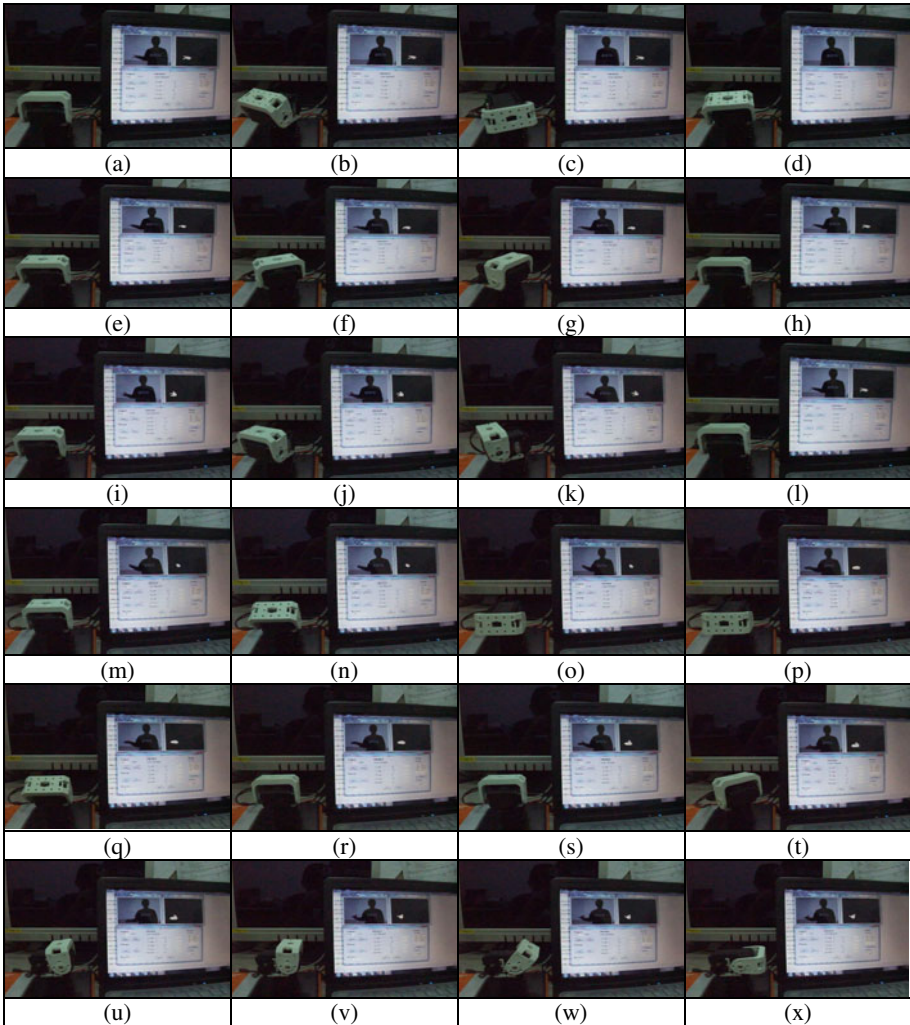


Fig. 5. Experimental results for various gesture commands: (a)~(e): the 1st command and then the 8th command; (f)~(h): twice the 2nd commands and then the 8th command; (i)~(l) twice the 3rd commands and then the 8th command; (m)~(o): twice the 4th commands; (p)~(r) : twice the 5th commands; (s)~(u) : twice the 6th commands; and (v)~(x): twice the 7th commands.

6 Experiment Results and Discussions

The design of 8 commands for a 2 DOFs platform is described as follows: (1) C1: Start search in the yaw and pitch directions, (2) C2: Increase 30 degree of motor 2 in the yaw direction (turn right), (3) C3: Decrease 30 degree of motor 2 in the yaw direction (turn left), (4) C4: Increase 40 degree of motor 1 in the pitch direction (tilt down), (5) C5: Decrease 40 degree of motor 1 in the pitch direction (tilt up), (6) C6: Increase 50 degree of motor 2 in the yaw direction (turn right), (7) C7: Increase 50 degree of motor 1 in the pitch direction (tilt down), (8) C8: Let the motor back to the center and then stop.

The corresponding experimental results using the designed commands are shown in Fig. 5, which is satisfactory as expected. In addition, the total process time for image processing, classification, motion control of 2 DOFs platform, and a feedback command for the next image inquiry, is about 160ms (or between 105 and 215ms).

7 Conclusions

Because the moment method possesses the invariants of translation, rotation and scale, it is suitable for the recognition of a hand gesture. However, the value of moment invariant is sensitive to noise; it must be eliminated before the calculation of its moment. In this paper, the shape selection based on the area and perimeter of ROI (region of interest), which is function of distance, is not exact to remove the unnecessary skin-like objects. Because the relative magnitudes of four components of feature vector for these 8 hand gesture are very large, the normalization of the input for the PNN must be employed to improve the successful recognition rate. According to the experimental motion control of a 2 DOFs platform, a PNN for the classification of the planned hand gestures is indeed successful.

Our further researches include (i) the recognition of hand gesture in the incandescent lighting condition to improve the generalization or robustness, (ii) the extra consideration of feature vector (e.g., circularity, rectangularity) to improve the successful recognition rate in a poor lighting condition, (iii) the recognition of the dynamic trajectory of a hand gesture, (iv) the application to the obstacle avoidance of a humanoid robot.

Acknowledgement. The authors want to thank the financial support from the project of NSC-99-2221 -E-032-066-MY3 of Taiwan.

References

1. Pavlovic, V.I., Sharma, R., Huang, T.S.: Visual Interpretation of Hand Gestures for Human-Computer Interaction: A review. *IEEE Trans. Pattern Analysis and Machine Intelligence* 19(7), 677–695 (1997)
2. Rashid, O., Al-Hamadi, A., Michaelis, B.: A Framework for the Integration of Gesture and Posture Recognition using HMM and SVM. In: *IEEE Int. Conf., Intelligent Computing and Intelligent Systems*, Shanghai, China, pp. 572–577 (2009)

3. Roh, M.C., Huh, S.J., Lee, S.W.: A Virtual Mouse Interface Based on Two-Layered Bayesian Network. In: IEEE Workshop on Applications of Computer Vision (WACV), Snowbird, UT, pp. 1–5 (2009)
4. Fei, Q., Li, X., Wang, T., Zhang, X., Liu, G.: Real-time hand gesture recognition system based on Q6455 DSP board. In: IEEE Global Congress on Intelligent Systems, pp. 139–144 (2009)
5. Tan, W., Wu, C., Zhao, S., Chen, S.: Hand Extraction Using Geometric Moments Based on Active Skin Color Model. In: IEEE Int. Conf., Intelligent Computing and Intelligent Systems, Shanghai, China, pp. 468–471 (2009)
6. Zhang, B., Yun, R., Weng, C.: Application of Improved Gesture Recognition Algorithm Based-on Density Distribution Feature. In: IEEE International Conference on Computer Application and System Modeling, Taiyuan, China, vol. 2, pp. 233–237 (2010)
7. Wang, X., Qin, K.: A Six-Degree-of-Freedom Virtual Mouse Based on Hand Gestures. In: IEEE International Conference on Electrical and Control Engineering, Wuhan, China, pp. 257–260 (2010)
8. Yang, Q.: Chinese Sign Language Recognition Based on Video Sequence Appearance Modeling. In: IEEE 5th Conference on Industrial Electronics and Application, Taichung, Taiwan, pp. 1537–1543 (2010)

Network Intrusion Prevention by Using Hierarchical Self-Organizing Maps and Probability-Based Labeling

Andres Ortiz¹, Julio Ortega², Antonio F. Díaz², and Alberto Prieto²

¹ Communications Engineering Department. University of Malaga.
29004 Malaga, Spain

² Department of Computer Architecture and Technology. University of Granada.
18060 Granada, Spain

aortiz@ic.uma.es, julio@atc.ugr.es,
afdiaz@atc.ugr.es, aprieto@ugr.es

Abstract. Nowadays, the growth of the computer networks and the expansion of the Internet have made the security to be a critical issue. In fact, many proposals for Intrusion Detection/Prevention Systems (IDS/IPS) have been proposed. These proposals try to avoid that corrupt or anomalous traffic reaches the user application or the operating system. Nevertheless, most of the IDS/IPS proposals only distinguish between normal traffic and anomalous traffic that can be suspected to be a potential attack. In this paper, we present a IDS/IPS approach based on Growing Hierarchical Self-Organizing Maps (GHSOM) which can not only differentiate between normal and anomalous traffic but also identify different known attacks. The proposed system has been trained and tested using the well-known DARPA/NSL-KDD datasets and the results obtained are promising since we can detect over 99,4% of the normal traffic and over 99,2 % of attacker traffic. Moreover, the system can be trained on-line by using the probability labeling method presented on this paper.

Keywords: IDS, IPS, Attack Classification, Self-Organizing Maps, Growing Self-Organizing Maps, SOM relabeling, clustering.

1 Introduction

The interest in computer network security has increased in recent years, as the trend to on-line services available through the Internet have exposed a lot of sensitive information to intruders and attackers [1]. Although there are several methods to protect the information, there is not any infallible encryption method and the encryption/decryption process can impose a high overhead in high speed networks that use the TCP/IP protocol stack. On the other hand, the complexity of the newer attacks make necessary the use of elaborate techniques such as pattern classification or artificial intelligence techniques for successfully detecting an attack or just to differentiate among normal and abnormal traffic. IDS and IPS systems are active systems which implement a protection by continuously monitoring the network. They calculate some traffic features to be able to classify it, to detect abnormal behaviors and to react according to some predefined rules. There are two design approaches to

IDS/IPS systems [2]. The first one consists on looking for patterns corresponding with known signatures of intrusions. The second one searches for abnormal patterns by more complex features, to discover not only intrusions but also potential intrusions.

Several neural and machine learning techniques have been used for implementing IDS/IPS systems [2-6]. In [2], perceptron-like neural networks are used for traffic classification as well as fuzzy classifiers to improve the decision step. In [3, 4], Self-Organizing Maps (SOMs) [5] are applied to implement unsupervised clustering of the data instances in order to classify the traffic anomalies according to several known attack. In order to improve the classification task either by distinguishing the normal traffic from anomalies or by classifying the different attacks with high accuracy, hierarchical SOM have been used in works. Although the proposal on [3] tries to overcome some of the difficulties on the static structure of classic SOMs splitting the SOM into three smaller SOMs, the size of these maps is still static.

An alternative to avoid the limitations of the classical SOM is the Growing Hierarchical SOM (GHSOM) [9]. It presents a dynamic hierarchical structure composed of several layers, with several SOMs in each layer. The number of SOMs in each layer and their respective sizes are determined in the GHSOM training process. In this paper, GHSOM is used for both, anomaly detection and attack classification. Moreover, a probability-based mechanism applied to the previously trained structure is used to label the units when the GHSOM is applied to new data instances.

After this introduction, the remainder of this paper is organized as follows. Section 2 shows the features of the NSL-KDD dataset that, as in most of the IDS/IPS works, has been used to evaluate our system. Section 3 describes our proposals for data preprocessing and for applying GHSOM to attack classification. In Section 4, the results obtained are shown, and finally, Section 5 provides the conclusions of the paper and our future work.

2 Attack Classification with GHSOM

The first step in IDS and IPS implementation is the extraction of some significant features from the captured traffic. The features on the NSL-KDD dataset [12] (the benchmark set we have used in this work) are split into three classes: basic features, content-based features, and traffic features. The main reason for having these three groups of features is that detecting and identifying some attacks requires the use of more than just one feature class. For instance, time-based features are necessary to detect some attacks, as some statistics should be calculated over a certain time period. Thus, the first step we perform consists of parsing the dataset in order to extract the features from the text files and to build the vectors which comprise the feature space.

This feature space is composed of vectors belonging to \mathbb{R}^{41} which contains the connection traffic features. Examples of these features are the duration of the connection, the protocol type, or the number of *user-to-root* attempts. Nevertheless, the feature selection for network-based IDS is not straightforward. This way, there are

works [14, 15, 16] which use multivariate techniques such as PCA [14] or LDA [15]. The use of multivariate techniques try to obtain the components with the highest variability, supposing the rest are unimportant or just noise. Although good results are given in [14, 15, 16] other works have shown that the full set of features outperforms feature selection with PCA/LDA [17]. Moreover, feature reduction has to be applied to each input vector since the most discriminant component depends on the specific attack. Thus, in [12] a SOM classifier for attack detection is presented without reducing the feature space. However, the authors only use a selection of 28 features from the 41 available in the NSL-KDD dataset, and the influence of each feature is figured out through the *U-Matrix* of each component belonging to the input (feature) space. Instead, in this paper we propose the use of a dynamic structure such as GHSOM and the full set of features.

2.1 SOMs and GHSOMs

SOM is a very useful tool for discovering structures and similarities in high dimensional data, organizing the information and visualizing it in a 2D or 3D way. Detailed descriptions of SOM can be found elsewhere [5]. Nevertheless, SOM is not able to figure out the inherent hierarchical structure of data [9], and, at the same time, the performance of SOM depends on the size of the map which has to be set in advance. Thus, GHSOM [9] is a hierarchical and non-fixed structure developed to overcome the main limitations of classical SOM. The structure of GHSOM consists of multiple layers composed by several independent SOMs. Hence, during the learning process, the number of the SOMs on each layer and the size of each SOM are determined by minimizing the quantization error. Thus an adaptive growing process is accomplished by using two parameters τ_1 and τ_2 that respectively control the breadth of each map (horizontal direction) and the growing of the hierarchy (vertical direction). Therefore, these two parameters are the only parameters that have to be set in advance.

In order to determine how far the GHSOM grows [9], the quantization error of each unit is calculated according to the Equation 1, where C_i is the set of input vectors mapped to the *i-th* unit, \mathbf{x}_j is the *j-th* input vector belonging to C_i , and \mathbf{w}_i is the weight associated to the *i-th* unit.

$$qe_i = \sum_{\mathbf{x}_j \in C_i} \|\mathbf{w}_i - \mathbf{x}_j\| \quad (1)$$

Initially, all the input vectors belong to C_0 . This means all the inputs are used to compute the initial quantization error, qe_0 . Then, the quantization errors qe_i for each neuron are calculated. Thus, whenever $qe_i < \tau_2 < qe_0$, the *i-th* neuron is expanded in a new map on the next level of the hierarchy. Each new map is trained as an independent SOM, and the calculation of its BMU (*Best Matching Unit*) is done by using the Euclidean distance. Once the new map is trained, the quantization error of each neuron on this map is computed. Then, the mean quantization error MQE_m of the

new map can be determined. Whenever $MQE_m < \tau_f \cdot qe_u$ (qe_u is the quantization error of the unit u on the upper layer), the map stops growing. The process has been schematized in Figure 1 and it is explained with detail in [9].

To apply GHSOM to IDS/IPS, it is necessary to encode some qualitative features included in the NSL-KDD dataset due to the numeric nature of the GHSOM input vectors. For example, this is the case of *protocol*, *service* and *flag* features. The encoding of these features is performed by assigning numeric values in order to keep a high enough distance among them for the effectiveness of the SOM classifier. Specifically, the number 1 and other prime numbers with distances higher than 6 among them have been chosen. This coding has been chosen in order to increase the distance among different features. For example, in the component *protocol* in the input vectors, TCP is encoded as 1, UDP as 7, and ICMP as 17. In order to avoid that some features have more influence than others, the input vectors have been normalized by subtracting the mean and dividing by the standard deviation (zero mean and unity variance). Thus, each dimension takes a value between 0 and 1, and the feature space belongs to \mathbb{R}^{41} .

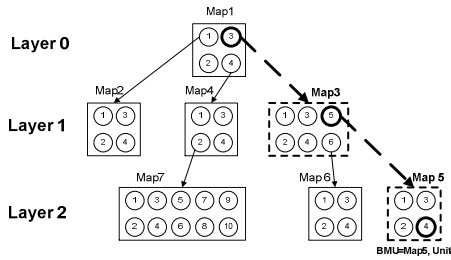


Fig. 1. BMU calculation on the GHSOM hierarchy

2.2 BMU Calculation for GHSOM

In order to calculate the BMU in the GHSOM, we have to go through all the hierarchy to determine the winning unit and the map to which it belongs. Thus, an iterative algorithm has been developed as shown in Figure 1, where an example of BMU calculation on a three-level GHSOM hierarchy is considered. After computing the distances between an input pattern and the weight vectors of the map in layer 0, the minimum of these distances is determined. Once, the winning neuron on map 1 is found, since other map could be grown from this winning neuron, we have to check whether the winning neuron is a parent unit. This can be accomplished with the parent vectors resulting from the GHSOM training process. If a new map arose from the winning neuron, the BMU on this new map is calculated. This process is repeated until a BMU with no growing map is found. Thus, the BMU in the GHSOM is identified inside a map in a layer of the hierarchy (for example, map 5 and unit 4).

2.3 GHSOM Training and Relabeling

The GHSOM structure has been trained by using 10% of the training samples provided by the NSL-KDD dataset. Then, the system has been tested by using the rest

of the training patterns. As it is commented in Section 2.2, we have used the full set of features. After several tests using different values for τ_1 (to control the breadth of the map) and τ_1 (to control the depth), we have selected $\tau_1=0.6$ and $\tau_1=10^{-5}$. This could make the GHSOM to grow more than necessary, leaving some of the units unlabeled. Thus, the number of neurons on the output map surrounding the winning neuron for training data is increased. When an input pattern similar to one of the training patterns is presented to the GHSOM, the winning neuron can be labeled or unlabeled. If the winning neuron is unlabeled, a probability-based scheme is used in order to determine the label of that neuron. More specifically, the winning neuron is labeled (or relabeled) with the more repeated label in its neighborhood by using a probability calculated according to (7).

$$P_u = \frac{M_{\sigma(u,\epsilon)}(u)}{n} \tag{2}$$

In this expression, P_u is the probability for the winning unit u to be successfully relabeled, where $M_{\sigma(u,\epsilon)}(u)$ is the label that appears more frequently in the Gaussian neighborhood $\sigma(u,\epsilon)$, of the winning neuron, u , and n is the number of neurons belonging to the neighborhood of u , $\sigma(u,\epsilon)$. In this equation, parameter ϵ noted the width of the neighborhood of the winning neuron.

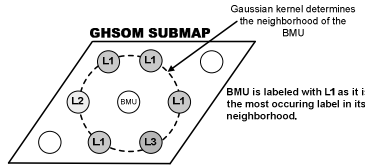


Fig. 2. Example of the relabeling process (white units are unlabeled)

In Figure 2, an example of the relabeling process is shown. In this Figure, the BMU that has not been initially labeled, is labeled by using $\epsilon=1$ to establish its neighborhood. In this neighborhood, we found four units labeled as L1, one unit labeled as L2 and one unit labeled as L3. Then, P_w , the probability for successful relabeling for this BMU is $4/6=0.66$ (66%) ($M_{\sigma(u,\epsilon)}(u)=4, n=6$).

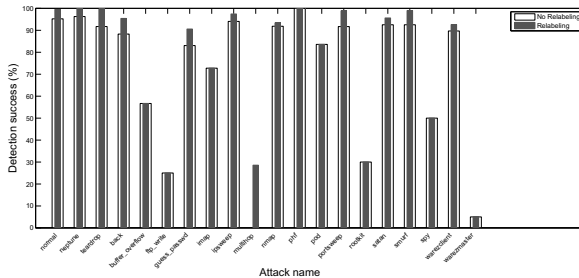


Fig. 3. Detection success with (black bar) and without (white bar) unit relabeling

3 Experimental Results

In this section, we present the experimental results obtained with the NSL-KDD dataset [13] and the GHSOM classifier described in Section 2. In Figure 3, the detection success for each type of attack included in the NSL-KDD dataset is shown. As this figure shows, the probability-based labeling process performed with new data (black bar in Figure 2) increases the detection success for most attacks. Moreover some attacks such as *multihop*, are not detected before the unit relabeling process.

In order to show the effectiveness of the relabeling process due to the associated probability, the ROC (receiver operating characteristic) curves are shown in Figure 4. They constitute an effective alternative to evaluate the performance of a classifier.

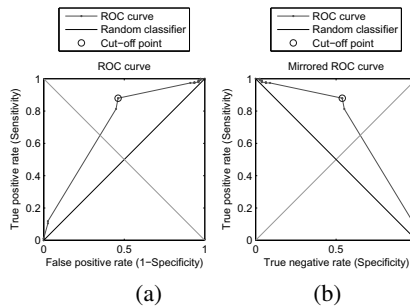


Fig. 4. ROC curves for the relabeling process

Figure 4.a shows the ROC curve (false positive rate) and Figure 4.b the mirrored ROC curve (true negative rate). Regarding a measure of performance derived from these curves, we computed the *Area Under ROC Curve* (AUC). The use of AUC makes the interpretation of the results from the ROC curve easier. Thus, a perfect classifier will provide an $AUC=1.0$ whereas in a random classifier $AUC=0.5$. In the graphs of Figure 4, the cut point determines the best performance the classifier can provide. The AUC computed from our ROC curves is 0.71. Hence, the AUC is statistically greater than 0.5 which denotes a fair behavior of the relabeling process.

In Figure 5.a, we present the detection success rate per attack type. As can be seen, in all cases, the relabeling method increases the classification performance as well as the detection success rate. Regarding to *User to Root* (U2R) attacks, the performance is worse than that obtained for other attacks. Nevertheless, the number of U2R training patterns on the NSL-KDD is significantly less than for other attacks [14].

Moreover, Figure 5.b summarizes the performance of our proposal when detecting normal/abnormal traffic. As shown in this figure, 99.6% of normal traffic patterns and 99.2% of the attack patterns have been correctly classified.

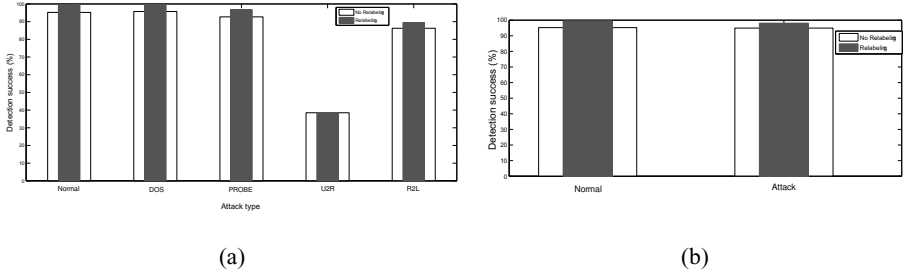


Fig. 5. (a) Detection Success rate for (a) different types of attacks and (b) for normal/attack traffic. Detection success with unit relabeling (black bar) and without unit relabeling (white bar).

In Table 1, testing results for previous proposed IDSs based on SOM and GHSOM are extracted from [18]. As it is shown in this table, our GHSOM with relabeling probabilities (RL-GHSOM in Table 1) reaches a high rate of detected attacks and clearly outperforms the false positive rate provided by other similar proposals.

Table 1. Basic features of individual TCP connections

IDS implementation	Detected attacks (%)	False Positive (%)
RL-GHSOM	99.68	0.02
GHSOM	99.99	3.72
K-Map	99.63	0.34
SOM	97.31	0.04

4 Conclusions and Future Directions

In this paper we present a network intrusion prevention approach that takes advantage of the discriminating properties of the GHSOM. Moreover, instead of applying any feature selection technique over the dataset, the full set of data features has been used. This circumstance has required to let the GHSOM grow more than it should be necessary and to devise a labeling (or relabeling) process for the BMUs that uses a probability for relabeling success. Acceptable results have been obtained from an analysis of the effectiveness of the relabeling process which has been done by using the ROC curves. The results obtained for the proposed IPS are promising, since it can detect 99.6% of the normal traffic patterns and 99.2% of the abnormal ones.

As future work, we will consider a real-time implementation of the IPS. This could be feasible as we avoid the need for principal component analysis. With such kind of implementation, as normal and abnormal behaviors could be accurately detected on line, it would be possible to perform some complementary activities, such as IP blocking, in real time in order to improve the quality of the network intrusion prevention.

Acknowledgments. This work was supported by project SAF2010-20558 (Ministerio de Educación).

References

1. Ghosh, J., Wanken, J., Charron, F.: Detecting anomalous and unknown intrusions against programs. In: Proceedings of the Annual Computer Security Applications Conference (1998)
2. Hoffman, A., Schimitz, C., Sick, B.: Intrusion Detection in Computer networks with Neural and Fuzzy classifiers. In: Kaynak, O., Alpaydin, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714, Springer, Heidelberg (2003)
3. Lichodziejewski, P., Zincir-Heywood, N., Heywood, M.: Host Based Intrusion Detection Using Self-Organizing Maps. In: Proceedings of the IEEE International Joint Conference on Neural Networks (2002)
4. Zhang, C., Jiang, J., Kamel, M.: Intrusion Detection using hierarchical neural networks. *Pattern Recognition Letters* 26, 779–791 (2005)
5. Kohonen, T.: *Self-Organizing Maps*, 3rd edn. Springer, Heidelberg (2001)
6. Fisch, D., Hofmann, A., Sick, B.: On the versatility of radial basis function neural networks: A case study in the field of intrusion detection. *Inf. Sci.* 180(12), 2421–2439 (2010)
7. Rauber, A., Merkl, D., Dittenbach, M.: The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High-Dimensional Data. *IEEE Transactions on Neural Network* 13(6) (2002)
8. Oh, H., Doh, I., Chae, K.: Attack Classification based on data mining technique and its application for reliable medical sensor communication. *International Journal Of Science and Applications* 6(3), 20–32 (2009)
9. The NSL-KDD dataset, <http://iscx.ca/NSL-KDD/>
10. Lakhina, S., Joseph, S., Verma, B.: Feature Reduction using Principal Component Analysis for Effective Anomaly-Based Intrusion Detection on NSL-KDD. *International Journal on Engineering Science and Technology* 2(6), 1790–1799 (2010)
11. Datti, R., Verma, B.: Feature Reduction for Intrusion Detection Using Linear Discriminant Analysis. *International Journal on Engineering Science and Technology* 2(4), 1072–1078 (2010)
12. Zargar, G.R., Kabiri, P.: Selection of Effective Network Parameters in Attacks for Intrusion Detection. In: *IEEE International Conference on Data Mining* (2010)
13. Mukkamala, S., Sung, A.H.: Feature Ranking and Selection for Intrusion Detection Systems Using Support Vector Machines. In: *Proceedings of the Second Digital Forensic Research Workshop* (2002)
14. Palomo, E.J., Domínguez, E., Luque, R.M., Muñoz, J.: Network security using growing hierarchical self-organizing maps. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 130–139. Springer, Heidelberg (2009)
15. Datti, R., Verma, B.: Feature Reduction for Intrusion Detection Using Linear Discriminant Analysis. *International Journal on Engineering Science and Technology* 2(4), 1072–1078 (2010)
16. Zargar, G.R., Kabiri, P.: Selection of Effective Network Parameters in Attacks for Intrusion Detection. In: *IEEE International Conference on Data Mining* (2010)
17. Mukkamala, S., Sung, A.H.: Feature Ranking and Selection for Intrusion Detection Systems Using Support Vector Machines. In: *Proceedings of the Second Digital Forensic Research Workshop* (2002)
18. Palomo, E.J., Domínguez, E., Luque, R.M., Muñoz, J.: Network Security Using Growing Hierarchical Self-Organizing Maps. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 130–139. Springer, Heidelberg (2009)

Human/Robot Interface for Voice Teleoperation of a Robotic Platform

L. Gallardo-Estrella and A. Poncela

Departamento de Tecnología Electrónica, E.T.S.I. Telecomunicación
Universidad de Málaga, Campus de Teatinos, 29071, Málaga
apg@dte.uma.es

Abstract. Speech is the most natural way of human communication. If the interaction between humans and machines is accomplished through voice, humans will feel more comfortable. Thus, this paper presents a Human/Robot Interface to teleoperate a robot by means of voice commands. To that purpose, an acoustic model in Spanish has been developed to recognize voice commands with Julius. The model is user dependent and has been suited to the proposed set of commands to achieve a better recognition rate. One of the advantages of the proposed speech recognition mechanism is that it can be easily adapted to a new list of commands. A robot has been successfully teleoperated with voice. Results about the recognition rate are promising in using the proposed Human/Robot Interface for voice teleoperation.

1 Introduction

Remote interaction with mobile robots is often referred to as teleoperation. It implies that the human and the robot are separated spatially, requiring communication between the robot and the human. This communication is commonly achieved through a Human-Robot Interface (HRI).

The human operator can communicate with the robot through conventional control inputs, such as a mouse [2], a keyboard [19] or joysticks [11]. There are HRI for teleoperation that use more sophisticated devices, such as gloves [8] or muscular activity sensors [6]. However, verbal communication is the primary and most natural way of human communication [5]. Nowadays it is possible to interact with robots via speech, which is a communication mechanism for humans. Voice has been previously used to interact with a tour-guide robot [3], to control an intelligent wheelchair [15,7] and to interact in a rehabilitation system [1]. Examples of teleoperation of a robotic platform can also be reported [13]. It must be pointed out that the integration of voice-control into HRI would make humans more comfortable during the interaction. Thus, in this work, speech is the primary form of interaction between the human and the machine.

A HRI for teleoperation of a robotic platform is presented in this paper. The interaction between the user and the robot can be achieved via a mouse, a keyboard and the user's voice. Nevertheless, the main form of interaction will be the user's voice, since natural spoken language is the most user-friendly means of

interacting with machines and from the human standpoint, spoken interactions are easier than others, relying on natural ways of communication [3].

The interface as a whole is one of the contributions, as well as the speech recognition mechanism, which has been designed to teleoperate the robot through a set of commands in Spanish. The HRI is integrated into a robot teleoperation system. Thus, this system is firstly presented in Section 2. Then, the interface is analyzed in Section 3, paying special attention to the design of the speech recognition. Results are described in Section 4. Finally, conclusions and future work are presented in Section 5.

2 System Architecture

The proposed HRI has been integrated into a robot system to effectively achieve its teleoperation. The structure of the system is presented in Fig. 1. The style of the *Distributed and Layered Architecture* (DLA) is used to support this structure [18].

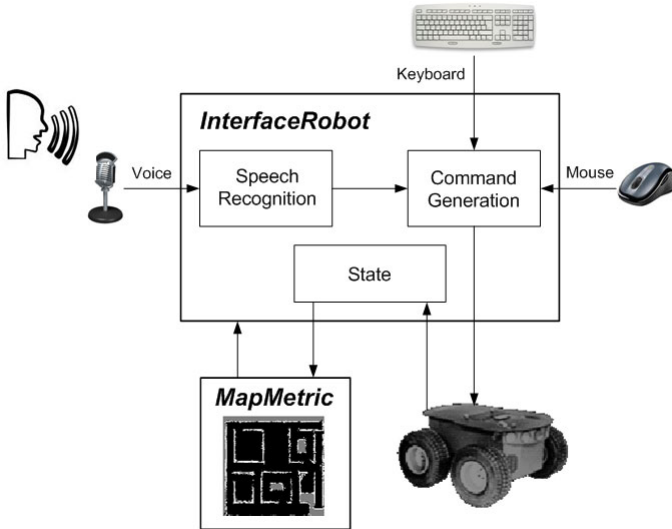


Fig. 1. System architecture for robot teleoperation

The system works under Linux. It is made up of two different modules which allow the robot to be teleoperated:

1. *InterfaceRobot*. It is described in detail in Section 3. It is the HRI to teleoperate the robot. The interface receives the user's input in the form of a voice command, mouse command or keyboard command, being the voice command the principal. In response to these commands, the interface interacts with the robot to apply its motion commands and capture its state.

It also shows the geometrical representation of the environment, which it is built in the module *MapMetric*. As illustrated in Fig. 1, *InterafaceRobot* is composed of several interconnected submodules:

- State. It captures odometrical data and sonar readings from the robotic platform. These data are made available to build the geometrical model of the environment with the module *MapMetric*.
 - Speech Recognition. It involves the user’s voice processing entity. Its output is the command to be sent to the robotic platform. It is further described in Section 3.2.
 - Command Generation. It can receive orders from three different sources: keyboard, mouse and the most important, voice saved through a microphone. Its output is the command to be sent to the robot mobile platform.
2. *MapMetric*. It builds a geometrical map of the environment using occupancy grids [4]. The metric map will allow the user of the HRI to progressively increase the knowledge of the robot environment while teleoperating.

3 Description of the Interface

3.1 Characteristics

Three aspects must be considered when designing a HRI: i) effectiveness; ii) usability; and iii) aesthetics. These three aspects have been made concrete in a set of criteria, which have been applied to the proposed HRI [14][16][17]: i) consistency; ii) effective use of colour; iii) Fitt’s law; iv) state visibility; v) known language; vi) recognize instead of remember; and vii) minimalistic and aesthetic design.

The interface consists of a form where all the components are arranged, as it is shown in Fig. 2. These elements are found in the proposed HRI:

- Menu bar. It holds an ordered list of all the actions which can be run.
- Toolbar. It gives a rapid access to the set of commands which can be sent to teleoperate the robot.
- State area. It shows information about the robot state (sonar readings and odometrical data), the geometrical representation of the environment and the state of the speech engine.
- Control area. Components related to commands sending and voice control are located in this area.

3.2 Speech Recognition

Since the proposed HRI is run under Linux, a speech recognition software for Linux is needed. In this work, Julius was used to develop a voice control mechanism to teleoperate the robot [10]. Julius is an open-source large vocabulary continuous speech recognition (LVCSR) engine. It combines high recognition

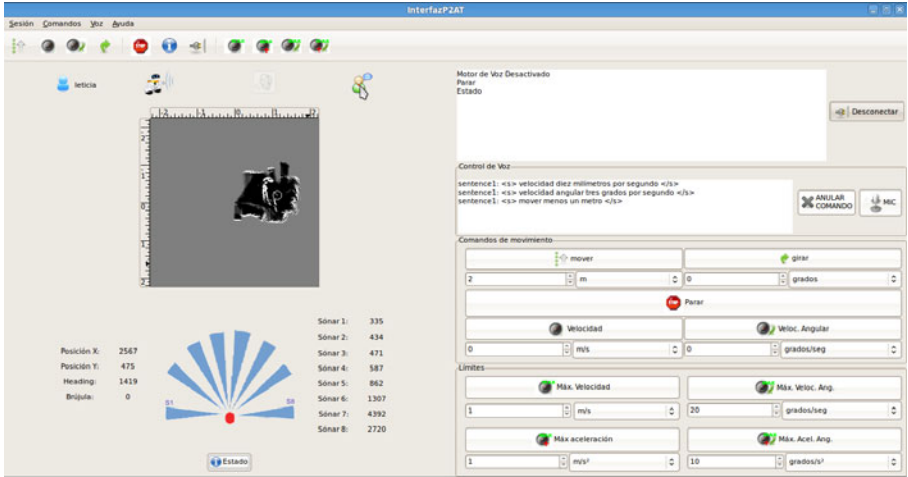


Fig. 2. Human/Robot interface for voice teleoperation

rate coupled with high speed speech recognition, working at almost real time. Thus, it becomes suitable for human-robot tasks.

Julius needs an acoustic model and a language model (or grammar) to be run. It can only be found, for free, acoustic models for Japanese and English. Although Julius was originally developed for Japanese, it has been successfully applied for several languages, such as English, French, Mandarin Chinese, Thai, Estonian, Slovenian and Korean [9]. However, since Julius computes the most likely sentence for a given input, the recognition accuracy largely depends on the model. Thus, it would be desirable to have an specific model for a particular language, Spanish in this work.

The acoustic model which has been developed for the proposed HRI is: i) valid for Spanish users from Spain, since people from Latin America speak with distinct accent to people from Spain [12]; ii) specific for the proposed set of commands; and iii) particular for each different user. The acoustic model has been developed with the standard Hidden Markov Model Toolkit (HTK).

The grammar consists of a set of selected commands which can be said, in Spanish, to teleoperate the robot. They are presented in Table 1. There are four types of commands: i) connection, to connect or disconnect from the robot; ii) movement, to move or to stop the robot; iii) state, to receive information about robot sensors; and iv) parameter, to shape the robot navigation. Furthermore, from a speech recognition point of view, commands are divided in two groups: short commands and long commands. Short commands are composed of one and unique word. Long commands, however, consist of a sentence, a list of ordered words. For these commands, one, two or three words are needed to identify the command. The command identifier is followed by a Spanish spoken number. The last part of each sentence is the unit of the spoken number.

Table 1. Spanish voice commands

Short Commands (Command)		
Name	Type	Description
Conectar	Connection	Connect to the robot
Desconectar	Connection	Disconnect from the robot
Parar	Movement	Stop the robot
Estado	State	Read the robot state (sensors, position and orientation)
Long Commands (Command + Number + Unit)		
Name	Type	Description
Mover	Movement	Move linearly a distance
Girar	Movement	Rotate a number of degrees
Velocidad	Movement	Move linearly at a specified velocity
Velocidad angular	Movement	Rotate at a specified angular velocity
Máxima velocidad	Parameter	Maximum translational velocity
Máxima velocidad angular	Parameter	Maximum rotational velocity
Máxima aceleración	Parameter	Maximum translational acceleration
Máxima aceleración angular	Parameter	Maximum rotational acceleration

Recognition is based on the generated acoustic model, which is obtained after a training stage. To achieve a better recognition rate than in the case of using a generic Spanish acoustic model, each user employs a particular one. To that purpose, the HRI provides a Training Assistant to generate the user dependent acoustic model. For each speaker, the training phase starts saving 72 sentences (802 words), where most of the words appear in the grammar. Then, the HTK tool generates the user dependent acoustic model. Mel frequency cepstral coefficients (MFCC) have been chosen as the feature vector since they are usually used in speech recognition applications.

It is necessary to point out that there have not been formulated any explicit consideration about the quality of the microphone used to perform both the training and recognition. If a good recognition rate is wanted, it is just mandatory to use the same microphone for training and for recognition, independently of its quality. It must also be realized that when an user has not generated his/her acoustic model, the HRI will not allow that user to teleoperate the robot with his/her voice. Therefore, the robot could only be teleoperated with the mouse and the keyboard in such a case.

4 Results

The system has been tested in a Pioneer P2AT equipped with eight frontal sonar sensors. Linux Ubuntu 9.10 has been used as the operating system. The robot has been simulated with MobileSim. Furthermore, due to the features of the DLA architecture, the system can be easily adapted to another platform (real or simulated), e.g., a wheelchair, even with new sensors as a laser range finder. This section presents the results after voice teleoperating the robotic platform with success.

Tests have been performed by two users, one female and one male. Each user trained the system to generate their personal acoustic model. Then, they sent, with their voice, 100 sentences with 678 words. These sentences included the 12 possible commands to be sent to the robot (Table II). Long commands were purposefully made up with different numeric values and different units.

The designed speech recognition has been evaluated with the most commonly measure, the Word Recognition Rate (WRR). Let N be the number of total words. Let S , D and I also be, respectively, the number of substituted words, the number of deleted words and the number of inserted words. Then, the WRR is defined as:

$$WRR = 1 - \frac{S + D + I}{N} \quad (1)$$

Although the WRR represents the recognition rate of the system at word level, sentences can be formed of several words. Thus, for the proposed command-based system, it is necessary to measure the correct number of sentences. Therefore, a new metric has been evaluated, the Command Success Rate (CSR), that is, the number of correct recognized sentences among the total number of sentences. Let TS be the total number of sentences. Let RS also be the number of recognized sentences. The CSR is given by:

$$CSR = \frac{RS}{TS} \quad (2)$$

WRR and CSR have been calculated for both users in four different cases. Each case corresponds with various groups of considered MFCC's features when obtaining the acoustic model of an user:

Case 1. MFCC_0_D_A_39. It is a 39-dimensional vector: static MFCC coefficients (13), delta coefficients (13) and acceleration coefficients (13).

Case 2. MFCC_0_D_N_Z_25. The vector is 25-dimensional: static MFCC coefficients (13), delta coefficients without their absolute energy (12) and cepstral mean normalization.

Case 3. MFCC_0_D_Z_26. The feature vector is 26-dimensional: static MFCC coefficients (13), delta coefficients (13) and cepstral mean normalization.

Case 4. MFCC_0_D_26. As in the previous case the vector is 26-dimensional: static MFCC coefficients (13) and delta coefficients (13).

Results in these four cases for both users are presented in Table III. The WRR is above the 98,8% in all cases and the CSR is above the 95% in all cases. These rates are much better than others obtained in command-based speech recognition systems applied to Robotics, as in [7], where the authors report a CSR of only a 83,4% with a commercial software. It is believed that such successful results are obtained due to the user dependent acoustic model for a determined set of commands. A huge effort has been applied in designing the speech recognition through the calculation of particular acoustic models, resulting in a very high WRR , near the 99%. It must be realized that the CSR is very high as well, even when many of the sentences the robot must recognize are a combination of the command identifier, any possible numeric value and different units.

Table 2. Results: *WRR* and *CSR*

User 1: female				
Test	<i>MFCC_0_D_A_39</i>	<i>MFCC_0_D_N_Z_25</i>	<i>MFCC_0_D_Z_26</i>	<i>MFCC_0_D_26</i>
<i>WRR</i>	98,97%	100%	99,71%	98,97%
<i>CSR</i>	97%	100%	98%	95%
User 2: male				
Test	<i>MFCC_0_D_A_39</i>	<i>MFCC_0_D_N_Z_25</i>	<i>MFCC_0_D_Z_26</i>	<i>MFCC_0_D_26</i>
<i>WRR</i>	98,82%	99,7%	99,7%	99,11%
<i>CSR</i>	96%	98%	98%	96%

5 Conclusions and Future Work

This paper has presented a Human/Robot Interface for command-based voice teleoperation of a robotic platform. The HRI is a contribution of the paper, as well as the designed speech recognition mechanism. An user and command dependent acoustic model in Spanish have been developed to recognize voice commands with Julius. The HRI has been integrated into a robot teleoperation system. Tests have been carried out with a Pioneer P2AT, which has been successfully teleoperated with the proposed voice commands. They have yielded a very high *WRR* and *CSR*, above 98,8% and 95%, respectively. The proposed set of commands to teleoperate the robot is fixed. However, one of the advantages of the proposed speech recognition is its easy adaptation to new grammars. Future work will focus on introducing new grammars and teleoperating new platforms as a wheelchair. Also, the combination of speech recognition and text-to-speech will be explored to achieve a most natural human/robot interaction.

Acknowledgment

This work has been partially supported by the Junta de Andalucía (Spain) under Project SIAD (P08-TIC-03991).

References

1. Barkana, D.E., Das, J., Wang, F., Groomes, T.E., Sarkar, N.: Incorporating verbal feedback into a robot-assisted rehabilitation system. *Robotica* (2010)
2. Courreges, F., Edkie, A., Poisson, G., Vieyres, P.: Ergonomic mouse based interface for 3d orientation control of a tele-sonography robot. In: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009), St. Louis, USA, pp. 61–66 (2009)
3. Drygajlo, A., Prodanov, P.J., Ramel, G., Meisser, M., Siegwart, R.: On developing a voice-enabled interface for interactive tour-guide robots. *Advanced Robotics* 17, 599–616 (7)
4. Elfes, A.: Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation* 3, 249–265 (1987)

5. Ferre, M., Macias-Guarasa, J., Aracil, R., Barrientos, A.: Voice command generation for teleoperated robot systems. In: Proceedings of the IEEE ROMAN 1998, Takamatsu, Japan (1998)
6. Fukuda, O., Tsuji, T., Kaneko, M., Otsuka, A.: A human-assisting manipulator teleoperated by emg signals and arm motions. *IEEE Transactions on Robotics and Automation* 19(2), 210–222 (2003)
7. Galindo, C., González, J., Fernández-Madrugal, J.A.: Control architecture for human-robot integration application to a robotic wheelchair. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* 36(5), 1053–1067 (2006)
8. Harada, T., Sato, T., Mori, T.: Human motion tracking system based on skeleton and surface integration model using pressure sensors distribution bed. In: Proceedings of the Workshop of Human Motion (HUMO 2000), Austin, USA, pp. 99–106 (2000)
9. Lee, A., Kawahara, T.: Recent development of open-source speech recognition engine julius. In: Proceedings of the 2009 APSIPA Annual Summit and Conference, Sapporo, Japan, pp. 131–137 (2009)
10. Lee, A., Kawahara, T., Shikano, K.: Julius – an open source real-time large vocabulary recognition system. In: Proceedings of the Interspeech 2001, Aalborg, Denmark, pp. 1691–1694 (2001)
11. Mahony, R., Schill, F., Corke, P., Oh, Y.S.: A new framework for force feedback teleoperation of robotic vehicles based on optical flow. In: Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009), Kobe, Japan, pp. 1079–1085 (2009)
12. Mayorga, P., Martín-Olguín, O.J., Hernández, A.M., Flores, J.I.: Gaussian components optimization for a robot controlled by speech commands in mexican spanish. In: Proceedings of the Fourth Congress of Electronics, Robotics and Automotive Mechanics (CERMA 2007), Cuernavaca, Mexico, pp. 324–329 (2007)
13. Medicherla, H., Sekmen, A.: Human-robot interaction via voice-controllable intelligent user interface. *Robotica* 25, 521–527 (2007)
14. Nielsen, J.: Usability inspection methods. John Wiley & Sons, New York (1994)
15. Nishimori, M., Saitoh, T., Konishi, R.: Voice controlled intelligent wheelchair. In: Proceedings of the 2007 SICE Annual Conference, Takamatsu, Japan, pp. 336–340 (2007)
16. Shneiderman, B.: Designing the user interface. Addison-Wesley, Massachusetts (1997)
17. Spolsky, J.: User interface design for programmers. Apress, New York (2001)
18. Urdiales, C., Bandera, A., Pérez, E.J., Poncela, A., Sandoval, F.: Hierarchical planning in a mobile robot for map learning and navigation, pp. 2165–2188. *Physica Verlag, Heidelberg* (2003)
19. Zhang, L., Huang, Q., Liu, Q., Liu, T., Li, D., Lu, Y.: A teleoperation system for a humanoid robot with multiple information feedback and operational modes. In: Proceedings of the 2005 IEEE International Conference on Robotics and Biomimetics (ROBIO), Hong Kong, pp. 290–294 (2005)

Graph Laplacian for Semi-supervised Feature Selection in Regression Problems

Gauthier Doquire* and Michel Verleysen

Université catholique de Louvain, ICTEAM - Machine Learning Group
Place du Levant, 3, 1348 Louvain-la-Neuve, Belgium
{gauthier.doquire,michel.verleysen}@uclouvain.be
<http://www.ucl.ac.be/mlg>

Abstract. Feature selection is fundamental in many data mining or machine learning applications. Most of the algorithms proposed for this task make the assumption that the data are either supervised or unsupervised, while in practice supervised and unsupervised samples are often simultaneously available. Semi-supervised feature selection is thus needed, and has been studied quite intensively these past few years almost exclusively for classification problems. In this paper, a supervised then a semi-supervised feature selection algorithms specially designed for regression problems are presented. Both are based on the Laplacian Score, a quantity recently introduced in the unsupervised framework. Experimental evidences show the efficiency of the two algorithms.

Keywords: Feature selection, semi-supervised learning, Graph Laplacian.

1 Introduction

Feature selection is an important task for many applications involving the mining of high dimensionnal datasets. Indeed, many features are often either redundant or totally uninformative and can harm learning algorithms, making them prone to overfitting [1]. Moreover, the elimination of such useless features is generally benefical both for the learning time and the interpretation of models.

Traditionally, feature selection algorithms are said to be *supervised*, in the sense that they assume the knowledge of the output (a class label for classification problems and a continuous value for regression ones) associated with each training sample [2]. On the other hand, *unsupervised* feature selection methods have also been developped, whose most obvious example is simply the evaluation of each feature variance. More complex unsupervised algorithms include for example an approach using feature similarity [3] or a graph Laplacian based ranking [4] which will be desribed later as it is the base of this paper.

Halfway between those two situations, a more realistic assumption is that, in many real-world problems, unsupervised samples are often easy to obtain and

* Gauthier Doquire is funded by a Belgian FRiA grant.

thus numerous, while only a few labeled samples are typically available. This limitation is mainly due to the cost associated to the obtention of the desired outputs (human expertise, destructive test...). These considerations naturally led to the development of *semi-supervised* learning, in which the few available information about the output is used to improve learning algorithms based on the unsupervised part only [5,6].

In this context, many feature selection algorithms have been proposed recently. Among others, Zhao et Liu proposed an approach using spectral analysis [7] while Quinzan et al. introduced an algorithm based on feature clustering, conditional mutual information and conditional entropy [8]. These two works as well as the very large majority of semi-supervised feature selection algorithms are basically designed to handle classification problems, while, to the best of our knowledge, almost no work has been done to develop algorithms specific to regression problems.

This paper first introduces a supervised feature selection algorithms, which is then used to achieve semi-supervised feature selection. Both are specifically designed to handle continuous outputs. They extend the unsupervised concepts in [4]. Within the unsupervised framework, this algorithm scores features according to their locality preserving power. Roughly speaking, good features have close values for close samples and thus preserve the local structure. In this work, the idea is extended by using distance information between the output of supervised samples.

The rest of the paper is organized as follows. Section 2 briefly presents the original unsupervised Laplacian Score. Section 3 presents the supervised feature selection criterion. Section 4 introduces the semi-supervised algorithm which combines in a simple way the information from supervised and unsupervised samples. Experimental evidences of its efficiency are presented in Section 5. Eventually, Section 6 gives some concluding remarks and directions for future work.

2 Laplacian Score

This section briefly presents the Laplacian Score, as introduced by He et al. [4] for unsupervised feature selection. As already discussed, the method selects features according to their locality preserving power.

Consider a dataset X . Let f_{r_i} denote the r^{th} feature of the i^{th} sample ($i = 1 \dots m$), \mathbf{x}_i the i^{th} data point and \mathbf{f}_r the r^{th} feature. A proximity graph with m nodes is built, which contains an edge between node i and node j if the corresponding points \mathbf{x}_i and \mathbf{x}_j are close, i.e. if \mathbf{x}_i is among the k nearest neighbors of \mathbf{x}_j or conversely. Throughout this paper, the proximity measure used to compute the nearest neighbors of a point is always the Euclidean distance.

From the proximity graph, a matrix S^{uns} is built by setting

$$S_{i,j}^{uns} = \begin{cases} e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{t}} & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are close} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where t is a suitable constant. $D^{uns} = \text{diag}(S^{uns}\mathbf{1})$, with $\mathbf{1} = [1 \dots 1]^T$, is defined, as well as the graph Laplacian $L^{uns} = D^{uns} - S^{uns}$ [9].

The mean (weighted by the local density of data points) of each feature \mathbf{f}_r is then removed: the new features are called $\tilde{\mathbf{f}}_r = \mathbf{f}_r - \frac{\mathbf{f}_r^T D^{uns} \mathbf{1}}{\mathbf{1}^T D^{uns} \mathbf{1}} \mathbf{1}$. This is done to prevent a non-zero constant vector such as $\mathbf{1}$ to be assigned a zero Laplacian score as such a feature obviously does not contain any information.

Eventually the Laplacian score of each feature \mathbf{f}_r is computed as

$$L_r = \frac{\tilde{\mathbf{f}}_r^T L^{uns} \tilde{\mathbf{f}}_r}{\tilde{\mathbf{f}}_r^T D^{uns} \tilde{\mathbf{f}}_r} \tag{2}$$

and features are ranked according to this score, in increasing order.

As a convincing justification of this criterion for feature selection, one can notice that

$$L_r = \frac{\sum_{i,j} (f_{ri} - f_{rj})^2 S_{ij}^{uns}}{\text{Var}(\mathbf{f}_r)}. \tag{3}$$

The numerator thus penalizes features belonging to close samples and however having very different values. $\text{Var}(\mathbf{f}_r)$ can be seen as the estimated weighted variance of feature r where the D matrix models the importance of the data points. Features with a high variance are thus preferred, as they are expected to have a higher discriminative power. More details can be found in [4].

3 Supervised Laplacian Score

3.1 Definitions

A formalism similar to the one described in the previous section can also be derived for *supervised* feature selection.

Consider again the training set X containing m samples \mathbf{x}_i described by n features. In case of a supervised regression problem, an output vector $Y = [y_1 \dots y_m] \in \mathbb{R}^m$ is also given. Under the assumption that the output Y is generated by a continuous and smooth enough function of X , it is natural to expect close samples \mathbf{x}_i and \mathbf{x}_j to have close output values y_i and y_j . Consequently, good features are expected to have close values for data points whose outputs are close too.

Define the matrix S^{sup} as:

$$S_{i,j}^{sup} = \begin{cases} e^{-\frac{(y_i - y_j)^2}{t}} & \text{if } y_i \text{ and } y_j \text{ are close} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

and $D^{sup} = \text{diag}(S^{sup}\mathbf{1})$, $L^{sup} = D^{sup} - S^{sup}$, $\tilde{\mathbf{f}}_r = \mathbf{f}_r - \frac{\mathbf{f}_r^T D^{sup} \mathbf{1}}{\mathbf{1}^T D^{sup} \mathbf{1}} \mathbf{1}$. Criterion (2) can again be used to rank features by computing a so-called Supervised Laplacian Score (SLS):

$$SLS_r = \frac{\tilde{\mathbf{f}}_r^T L^{sup} \tilde{\mathbf{f}}_r}{\tilde{\mathbf{f}}_r^T D^{sup} \tilde{\mathbf{f}}_r} \tag{5}$$

Here again, in Equation (4), two points are considered to be close if one is among the k nearest neighbors of the other while t is a suitable (positive) constant.

Expression (3) can of course be derived with $S_{i,j}^{sup}$ and thus $Var(\mathbf{f}_r)$ adapted to the supervised case.

3.2 Illustration

SLS is briefly compared with the correlation coefficient, a widely used criterion for feature selection. The objective is to show the ability of the method to achieve feature selection and its greater capability detect non linear relationships between each feature and the output. Two artificial problems are considered.

The first one consists of 8 features $X_1 \dots X_8$ uniformly distributed on $[0; 1]$. The output is defined as:

$$Y_1 = \cos(2 \pi X_1 X_2) \sin(2 \pi X_3 X_4). \tag{6}$$

The second consists of 4 features $X_1 \dots X_4$ uniformly distributed on $[0; 1]$. The output is defined as:

$$Y_2 = X_1^2 X_2^{-2}. \tag{7}$$

The sample size is 1000 in both cases and 1000 datasets are randomly generated for each problem. The comparison criterion is the percentage of cases for which the 4 (2) informative features are the 4 (2) best rated.

For the first problem this percentage is 93% with SLS and 25% with the correlation coefficient. For the second problem, the percentages are 100% and 32% respectively. The advantage of SLS in these two simple cases is thus obvious.

4 Semi-supervised Laplacian Score

First experiments showed promising results concerning the use of SLS for supervised regression problems with a large number of data points. When the number of supervised samples is small, however, unsupervised samples have also to be taken into account.

The semi-supervised feature selection algorithm proposed in this paper is based on the developments in the two previous sections. More precisely, LS and SLS are both based on the locality preserving power of the features. The difference comes from the fact that locality (measured by the distance between samples) is defined from the unsupervised part of data for LS and from the output for SLS. A quite intuitive idea is thus to compute the distance between two samples from their outputs if both are known, and from the unsupervised part of the data otherwise.

Consider a semi-supervised regression problem consisting in the training set X and an output vector $Y = [y_1 \dots y_s] \in \mathbb{R}^s$, $s \ll m$.

The first step is to define a matrix d of distances between each pair of data points:

$$d_{i,j} = \begin{cases} (y_i - y_j)^2 & \text{if } y_i \text{ and } y_j \text{ are known} \\ \frac{1}{n} \sum_{k=1}^n (f_{k,i} - f_{k,j})^2 & \text{otherwise.} \end{cases} \tag{8}$$

In the second case, the distance is normalized by the number of features n in order to keep it comparable to the distance computed from the output.

A matrix S^{semi} is then built as follows:

$$S_{i,j}^{semi} = \begin{cases} e^{-\frac{d_{i,j}}{t}} & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are close and } y_i \text{ or } y_j \text{ is unknown,} \\ C \times e^{-\frac{d_{i,j}}{t}} & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are close and } y_i \text{ and } y_j \text{ are known,} \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

Two points are considered as close if one is among the k nearest neighbors of the other one.

The (positive) constant C allows us to give more weight to the information coming from the supervised part of the data, as it is believed to be more important than the unsupervised one for the feature selection problem.

One can then define $D^{semi} = \text{diag}(S^{semi}\mathbf{1})$, $L^{semi} = D^{semi} - S^{semi}$ and $\tilde{\mathbf{f}}_r = \mathbf{f}_r - \frac{\mathbf{f}_r^T D^{semi} \mathbf{1}}{\mathbf{1}^T D^{semi} \mathbf{1}} \mathbf{1}$.

The criterion for semi-supervised feature selection, called Semi-Supervised Laplacian Score (SSLS), is eventually:

$$SSLS_r = \frac{\tilde{\mathbf{f}}_r^T L^{semi} \tilde{\mathbf{f}}_r}{\tilde{\mathbf{f}}_r^T D^{semi} \tilde{\mathbf{f}}_r} \times SLS_r, \quad (10)$$

where SLS_r is the Supervised Laplacian Score (computed on the supervised samples only). The criterion thus combines the influence of both the unsupervised and the supervised part of the data, giving however this last part more importance.

5 Experimental Results

In this section the interest of the proposed semi-supervised approach is illustrated on three real-world data sets.

The first one is the Juice dataset. The goal is to estimate the level of saccharose of an orange juice from its observed near-infrared spectrum. 218 spectra samples with 700 points are available. The dataset can be downloaded from the website of the UCL’s Machine Learning Group ¹.

The second one is the Nitrogen dataset, containing originally 141 spectra discretized at 1050 different wavelengths. The objective is the prediction of the nitrogen content of a grass sample. The data can be obtained from the Analytical Spectroscopy Research Group of the University of Kentucky². In order to reduce the huge number of features, each spectrum is represented by its coordinates in a B-splines base as a preprocessing [10]. 105 features are built this way.

The last one is the Delve-Census data set, for which only the 2048 first samples are considered. The data is available from the University of Toronto³. Originally,

¹ <http://www.ucl.ac.be/mlg/>

² <http://kerouac.pharm.uky.edu/asrg/cnirs/>

³ <http://www.cs.toronto.edu/delve/data/census-house/desc.html>

each sample consists in 139 demographic features about a small region and the objective is to predict the median price of houses in each region. However, only 104 features are considered here, since those which are too correlated with the output have been removed for the experiments.

The performances of the feature selection algorithms are evaluated by the root mean squared error (RMSE) of a 5 nearest neighbors prediction model.

First features are selected on the training set with only a few randomly selected supervised samples. The model is then used to predict the output of the points of an independent test set. For the prediction step, as a too small number of labeled data would not allow the model to perform correctly, all the samples in the training set are supposed to be labelled.

This procedure ensures that the performances reflect the quality of the feature selection itself, and are not too much influenced by the prediction model. The algorithms are tested with 7 and 10 supervised samples for the two first (smaller) data sets, and with 70 and 100 supervised samples for the larger Delve Census dataset. The RMSE is estimated through a 5-fold cross validation procedure repeated 10 times.

Parameter t is set to 1, 5 neighbors are considered for computing the unsupervised (II) and supervised (A) score, while 30 neighbors are considered for the semi-supervised score (9). Indeed, the number of supervised samples being small, increasing the number of neighbors considered in the analysis allows to take such samples into account. The parameter C in (9) is set to 5. This moderate value gives a large importance to the supervised samples, but still gives a significant weight to the information coming from the unsupervised data points. The maximum number of selected features is 100. Before any distance computation, features are normalized by removing their mean and dividing them by their standard deviation.

Figure II first shows how the use of a few labeled data can improve the feature selection procedure for regression purposes when compared with the unsupervised approach. Indeed, as expected, the unsupervised approach (LS) is obviously the one performing the worse on all three datasets and the prediction performances are greatly improved by the use of only a small number of supervised samples.

Moreover, the proposed SSLS also performs better than the correlation coefficient for the three examples. This is particularly obvious for the Juice and the Delve Census data sets where the RMSE obtained with the SSLS is never larger than the one obtained with the correlation coefficient.

Eventually, results on the Juice data set underline the interest of the semi-supervised SSLS approach over the supervised SLS method. This indicates that the knowledge coming from the unsupervised samples is efficiently taken into account in the feature selection procedure. Results on the Nitrogen data set are also in favour of the SSLS, which leads in most of the cases to a smaller RMSE. It also reaches the lowest global RMSE with both 7 and 10 supervised samples. Results on the Delve Census dataset are slightly better for the supervised approach.

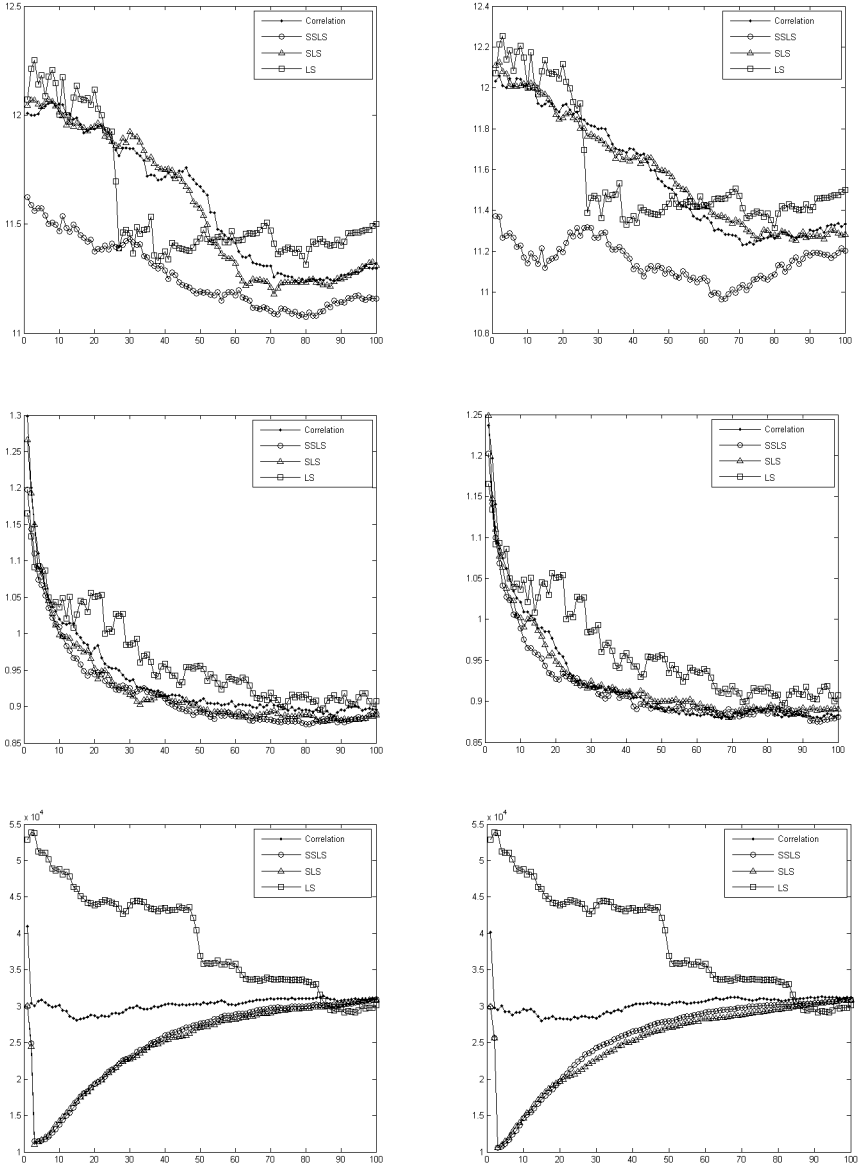


Fig. 1. RMSE as a function of the number of selected features with 7 (left) and 10 (right) supervised samples. From top to bottom: Juice, Nitrogen and Delve Census data set.

6 Conclusions and Future Work

In this paper, two feature selection algorithms are introduced for regression problems. Both are inspired by the Laplacian Score (LS), a recently introduced unsupervised feature selection criterion and are based on the locality preserving power of the features. In other words, the algorithms first select feature which are coherent with a distance measure between samples.

In supervised learning, the distances are evaluated with the output only, leading to the Supervised Laplacian Score (SLS). In the case of semi-supervised learning, distances are computed with the output if they are known or with the data points otherwise. The semi-supervised score obtained this way is then combined with the SLS to produce the semi-supervised Laplacian score (SSLS).

Experiments demonstrate the interest of the proposed approach, especially for the semi-supervised feature selection problem. More precisely, for the problems considered here, SSLS is shown to be superior to the correlation coefficient and to the unsupervised approach. Moreover, it also outperforms its supervised version on two datasets, showing the interest of considering unsupervised samples for the feature selection when the number of supervised points is too low.

Further work could be focused on new ways to combine the information coming from the supervised and unsupervised part of the data, as only the product has been considered here.

References

1. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. *J. Mach. Learn. Res.* 3, 1157–1182 (2003)
2. Battiti, R.: Using Mutual Information for Selecting Features in Supervised Neural Net Learning. *IEEE T. Neural. Networ.* 5, 537–550 (1994)
3. Mitra, P., Murthy, C.A., Pal, S.K.: Unsupervised Feature Selection Using Feature Similarity. *IEEE T. Pattern. Anal.* 24 (2002)
4. He, X., Cai, D., Niyogi, P.: Laplacian Score for Feature Selection. In: *Advances in Neural Information Processing Systems (NIPS)*, vol. 17 (2005)
5. Chapelle, O., Schölkopf, B., Zien, A.: *Semi-Supervised Learning*. MIT Press, Cambridge (2007)
6. Zhu, X., Goldberg, A.B.: *Introduction to Semi-Supervised Learning*. Morgan & Claypool Publishers, San Francisco (2009)
7. Zhao, Z., Liu, H.: Semi-supervised Feature Selection via Spectral Analysis. In: *7th SIAM International Conference on Data Mining* (2007)
8. Quinzán, I., Sotoca, J.M., Pla, F.: Clustering-Based Feature Selection in Semi-supervised Problems. In: *9th International Conference on Intelligent Systems Design and Applications*, pp. 535–540 (2009)
9. Chung, F.R.K.: *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics 92. American Mathematical Society, Providence (1997)
10. Rossi, F., Delannay, N., Conan-Guez, B., Verleysen, M.: Representation of Functional Data in Neural Networks. *Neurocomputing* 64, 183–210 (2005)

Detection of Transients in Steel Casting through Standard and AI-Based Techniques

Valentina Colla¹, Marco Vannucci¹, Nicola Matarese¹, Gerard Stephens²,
Marco Pianezzola³, Izaskun Alonso⁴, Torsten Lamp⁵, Juan Palacios⁶,
and Siegfried Schiewe⁷

¹ Scuola Superiore Sant'Anna, Viale Rinaldo Piaggio, 56025 Pontedera, Italy

² Tata Steel UK Ltd

³ Riva Acciaio Spa

⁴ GERDAU Sidenor I+D

⁵ VDEh-Betriebsforschungsinstitut GmbH

⁶ TECNALIA Research & innovation

⁷ ArcelorMittal Ruhrort GmbH

colla@sssup.it, {mvannucci,n.matarese}@sssup.it

Abstract. The detection of transients in the practice of continuous casting within a steel-making industry is a key task for the prediction of final product properties but currently a direct observation of this phenomenon is not available. For this reason in this paper several standard and soft-computing based methods for the detection of transients from plant data will be tested and compared. From the obtained results it emerges that the use of a fuzzy inference system based on experts knowledge achieves very satisfactory results correctly identifying most of the transient events present in the databases provided by different companies.

Keywords: transient detection, neuro-fuzzy systems, industrial problem.

1 Introduction

A *transient* in any signal can be roughly described as a sudden change of the signal itself. In a more mathematical way a *transient* can be undirectly defined as the phenomenon which, when present in a signal, makes the Fourier expansion of the signal composed by an infinite number of sinusoids [1]. The problem of transient detection is common to many fields. Transient detection is faced for instance in industrial electronics where the detection of electrical transients is performed in order to avoid circuits malfunctionings or damages [2]; a similar problem is encountered and faced in the field of electrical engineering [3] and in the design of audio filters [4] where the identification of transient is used for signal preprocessing. The problem of transient detection has been faced by means of different approaches: standard techniques such as Fast Fourier Transform [5] or artificial intelligence. For instance in [6] neural nets are used for acoustic transients and in [7] a combination of Wavelet Transform and fuzzy logic is adopted. This work proposes a comparison among traditional and AI based

methods for the detection of transient events within an industrial framework. In the described application, related to steel casting, the detection of transients in a weight signal, is fundamental for improving the steel products quality.

2 The Industrial Problem

Continuous casting is a sub-process where the liquid steel produced in the basic oxygen furnace or in the electric arc furnace is cast into a container called *ladle* which subsequently fills the tundish. Finally the liquid steel in the tundish passes through a nozzle to the mould for the manufacturing of the final product. During the casting the ladle progressively transfers the liquid steel to the tundish which in turn passes the material to the mould. When the level of the material in the tundish lowers, further liquid steel is transferred to the tundish through the ladle. Currently this latter operation takes place when the weight of the tundish - which is constantly monitored - decreases under a fixed threshold. The tundish can be refilled with a slightly different kind of steel with respect to the previous material load and this fact can affect some properties (e.g. mechanical properties) of the final product. To estimate the final product properties, the percentages of the different steels in the mix must be known: they can be estimated from the different loads of liquid material transferred by the ladle to the tundish. Unluckily on some plants, due to practical reasons related to the industrial framework, the exact moment when the refill of the tundish takes place (the so-called *transient*) is not tracked: the only way to establish the beginning and the end of the transient is an *a-posteriori* analysis of the profile of the tundish weight. The automatic detection of the transients is not trivial: the signal measuring the tundish weight is extremely irregular and noisy, the characteristics and shape of the transients themselves are very variable (see the exemplar profiles depicted in Fig. III).

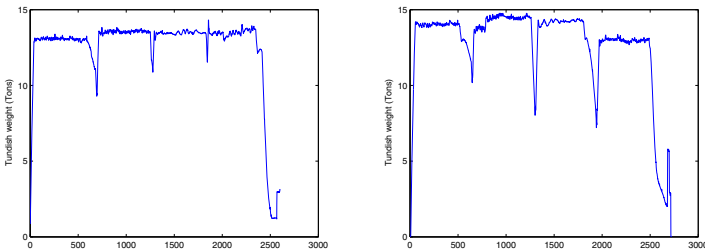


Fig. 1. Typical profiles of the tundish weight through time

3 Automatic Transient Detection

In this section the various methods attempted for the automatic detection of transient events are described; all these methods are based on the analysis of

the tundish weight profiles since that is the only available information. These methods have been based on the information provided by expert technicians working on the continuous casting machine of several steel-making industries in Europe. The support of experts for the identification of transient events in the casting machine is needed due to the extremely manifold nature of transients: the features of such events (e.g. profiles and shapes) are extremely variable and can be easily confused with other operating conditions of the machine.

3.1 Crisp Rules Based Method

On the basis of the suggestions of expert personnel a set of common features characterizing transients have been pointed out. These features mainly involve the absolute value of the weight of the tundish and the value of the its first and second derivative in each point of the signal as the combination of these factors mainly determines the shape of the profile. These considerations have been put up in order to create a set of rules for the identification of the points potentially belonging to the transients within an examined profile. These identification is singly applied to all the points of the profile so as to assign to each of them a binary value expressing their belonging to a possible transient. The main drawback of this method is the difficulty of the implementation of the rules due to the complex interactions of the various considered features which are hard to express in crisp mathematical terms. Once the critical points have been identified, they are grouped into sets of contiguous points to form a set of *potential* transients; each of these groups are further examined in order to evaluate if they can correspond to *real* transients. This discrimination is made through simple rules involving the length of the potential transients and the variability ranges of the tundish weight in such intervals. According to the tests, this latter check is extremely reliable as it correctly identifies the real transients among the potential ones. As the filter used for picking out real transients shows extremely good performance, it is used for the same purpose coupled to the other presented methods.

3.2 Transient Detection through the Use of a Fuzzy Inference System

The main criticality faced developing the crisp rules based method is the implementation of the rules themselves as they involve many interacting variables and quantities whose definition is not clear and univocal as they are derived from the subjective suggestions and the experience of personnel working on the steel plant. For overcoming these criticalities an alternative method exploiting as the first one the knowledge provided by experts was developed on the basis of a fuzzy inference system (FIS) [8]. The main advantage of this approach lies in the possibility of easily formalizing all the knowledge provided by experts, implementing in a natural and flexible way the desired relations between input and output variables which were impossible or extremely hard to express through

the crisp rules and which can lead to unsatisfactory results due to inaccuracies and rigidity of the inference system caused by the complexity of the rules. The designed FIS takes as inputs three features extracted for each point of the tundish weight profile and processes them through a set of rules expressed by means of natural language which describe the human reasoning for the detection of transients. The FIS returns as output for each point of the profile a numeric value in the range $[0;1]$ quantifying the *probability* for that point to belong to a transient. The features used as input by the FIS are the following:

derivative quantifies the variation of the tundish weight for a specific point. In this work the absolute value of the first derivative of the tundish weight is taken into account as for the design of this specific FIS it is not necessary to specify whether the weight variation is positive or negative. According to the experts' knowledge high variations can correspond to transient events.

derivative variability is used to measure the variability of the first derivative. The higher the value of this variable the more probably the examined point belongs to a transient.

relative tundish weight is the ratio between the tundish weight in a point and the average tundish weight. This variable is taken into account because transients are characterized by a relative low tundish weight with respect to the other normal casting situations

Each fuzzy variable is associated to its fuzzy sets which are used to define the inference system by means of natural language:

- derivative (**D**) : low; high
- derivative variability (**DV**) : low; medium; high
- relative tundish weight (**RTW**): low; normal
- transient probability (**TP**): low; medium; high

Each fuzzy set is associated to a trapezoidal membership function whose shape and position in the numeric domain of the corresponding variable has been determined on the basis of the experts' knowledge. The FIS rules mimic the reasoning described by the experts, such as in the following examples:

```
IF DV is high & RTW is low THEN TP is high
IF DV is NOT high & RTW is low & D is low THEN TP is low
IF RTW is normal THEN TP is low
```

The rules are managed by a Mamdani-type fuzzy inference system [9] which uses the MIN function for the implementation of the AND operator and of the implication method, the MAX function for the OR operator and the aggregation method while the defuzzification is implemented by means of the centroid method.

The system returns for each point a value in the range $[0;1]$ which represents the degree of membership of the examined point to a transient event; this value is subsequently compared to a threshold value equal to 0.5 to determine whether the point belongs or not to a transient event. Once each point of the profile has been classified through the FIS, the same points-grouping method and potential transient filter described in section 3.1 is performed so as to point out the final detected transients among the potential ones.

3.3 Feed-Forward Neural Networks for the Identification of Transients

A further approach based on the use of feed forward neural networks (FFNN) has been attempted for the detection of transient events. This approach belongs to the supervised learning methods family, thus it requires an a-priori classification of the variable to predict (here the membership of a point to a transient event). For this reason all the available tundish weight profiles have been processed by human experts in order to assign to each point a binary class 0/1 denoting such membership (1-valued points). The NN inputs are the same three variables used by the FIS based method while the output of the network is a binary value representing the membership of the examined point to a transient. The FFNN was trained by exploiting the 75% of all the available observations while the remaining 25% was used for testing the method and assessing its performance.

The adopted NN is a multi-layer perceptron feed forward neural network (MLP-FFNN) [10] with one hidden layer layer, three inputs and one output. The network was trained by means of a variation of the backpropagation algorithm including Bayesian regularization [11] in order to improve its generalization capabilities. Within this work several configuration of this kind of network have been tested varying the number of neurons included in the hidden layer.

When a tundish weight profile is processed, input variables are calculated and passed to the NN, which provides as output the membership (0/1) of each point to a transient. This information is used to form the set of potential transients which are finally filtered as for the other methods in order to select the real ones.

3.4 Transient Detection through Wavelet Decomposition

The solution proposed in this section is based on Wavelet Transform (WT) [12], a signal analysis method that consist in projecting a signal to a family of basis functions generated by translating and dilating a single function which is called *mother wavelet*. In literature there are many works exploiting WT for the detection of transient disturbances [13]. One of the biggest advantages of this technique lies in the ability of decomposing complex information into elementary forms involving coarse approximation and different levels of details.

Here the WT based on Haar mother wavelet decomposes the original weight profile and the reconstructed coefficients are extracted. The basic idea of this heuristic approach is to take into account two (and eventually more) different levels of decomposition of the original signal; in this case a level-6 (D6) and level-5 (D5) are used. A lower level of decomposition follows better the original data with respect to a higher level, as depicted in Fig. 2. Once the decomposition has been done, the search of an eventual transient is pursued on the decomposed signals (D5 and D6) which are noise and fluctuations free with respect to the original one. An empirically determined parameter α called *degree of sensitivity* has been introduced with the aim of indicating the *depth* of the transient to be identified: a big value states the identification of a clear-cut transient. In order to check the starting and the ending point of the transient a comparisons among the coefficients of the two different levels is performed for each point i :

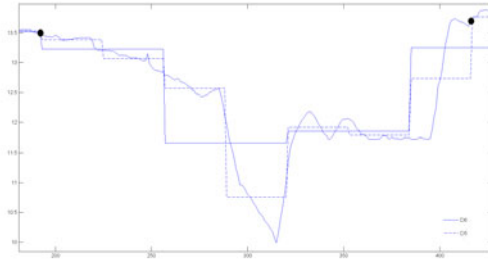


Fig. 2. A sample of the results provided by the wavelet decomposition method: in the picture the tundish weight and levels 5 (D5) and 6 (D6) of decomposition are shown

```

If ( D6(i) >= D6(i+1) + ALPHA ) then
    while ( D5(i) + ALPHA <= D5(i-1) ) i = i - 1;
startTransient = i;
    
```

The first condition finds a possible start of the transient, while the internal loop condition adjusts the point in more accurate way following back the coefficient of the considered level of decomposition which fits more faithfully the initial data. A specular approach was used to find the end of the transient.

4 Numerical Results

For the testing (and the training in the case of MLP-FFNN) of the methods described in section 3 a wide database containing about 60000 punctual observations recording the trend of tundish weight has been exploited. These records refer to more than 30 different heats and include 50 transient events. The database is composed by two datasets provided by two distinct steel-making companies: the different origin of the datasets enhances reliability and efficiency of the developed detection system as it allows its tuning on a wider range of cases.

The systems based on crisp rules, FIS and WT exploit the whole available dataset for the testing of the method. In particular each profile is processed in order to find possible transient events. In the case of the MLP-FFNN method, due to the necessity of training the system, the 75% of the casts (uniformly taken from both the steel-making companies) have been used for the training of the neural network while the remaining 25% for the test. The results obtained by the tested methods are shown in Tab. 1 for both the datasets in terms of detected transients. Noticeably none of the proposed methods risen any false alarm. Some qualitative results of the FIS-based method are shown in Fig. 4

The results achieved by the crisp rules-, FIS- and WT-based methods are very satisfactory as these approaches can detect correctly most of the transient events present in the dataset; in particular the best performing method among those tested is the FIS based one whose performance is significantly better than

Table 1. Results obtained by the tested methods on the two industrial databases

Method	Dataset	Detected transient
Rule base	Company 1	77%
FIS		95%
MLP-FFNN		55%
Wavelet		85%
Rule base	Company 2	85%
FIS		100%
MLP-FFNN		35%
Wavelet		95%

the one achieved by the crisp rules proving the efficiency of the fuzzy framework with respect to the crisp one for this task. The performance of the MLP-FFNN classifier are not satisfactory: in that case less than 50% of the transient are spotted. This can be due to the absence of apriori knowledge of the problem embedded into this approach. On the other hand the WT-based approach obtains very satisfactory results detecting about 90% of the transient events.

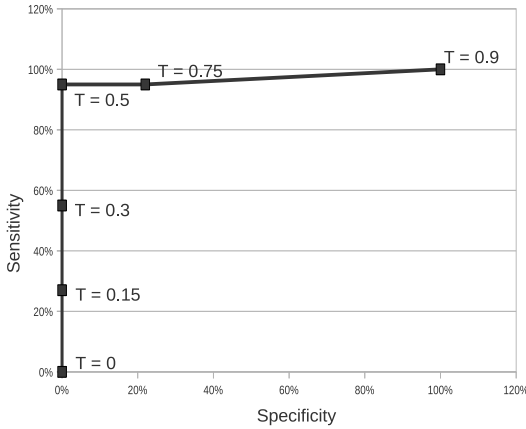


Fig. 3. The ROC curve related to the choice of the FIS activation threshold within the FIS-based method

The specificity of the FIS-based methods has been investigated for varying values of the threshold applied to the output of the inference system which determines whether a point belongs or not to a transient. This operation was done in order to select the best performing threshold. For this purpose the calculation of false alarms rate has been adjusted to meet the peculiarity of the proposed application in facts in such framework there are not discrete observations to be classified but, within a tundish weight profile, any couple of points could potentially be classified as an alarm (false or not). In order to overcome this limit the false alarms rate has been calculated on the basis of a set of *critical* parts

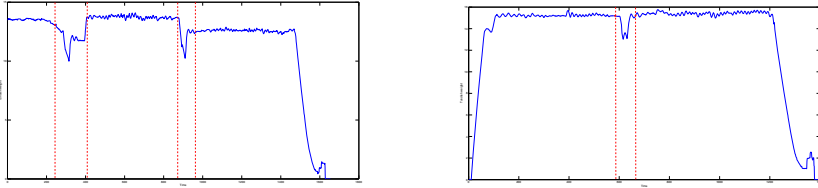


Fig. 4. Two sample results obtained by the use of the FIS-based method

of the profiles which could be classified as belonging to a transient. The results obtained by the FIS classifier have been calculated for different values of the FIS activation threshold are shown in table 2 and by means of a ROC curve in figure 3. The results of this investigation justify the choice that has been made for the threshold value.

Table 2. Results obtained by the FIS based method for different values of the FIS activation threshold

Threshold	Sensitivity	Specificity
0.15	95%	78%
0.3	95%	78%
0.5	95%	100%
0.75	55%	100%
0.9	27%	100%

5 Conclusions and Future Work

The automatic detection of transient events in the framework of the continuous casting process has been faced by means of different techniques. The experience of plant expert workers has been exploited for the design of a crisp rule based system and of a FIS reflecting the human reasoning for performing such task. FFNN- and a WT-based approaches have been also tested on this problem. The results obtained by the tested methods are generally satisfactory since most of the transient events are correctly spotted with the exception of the FFNN-based method. The goodness of the results achieved by the FIS proves the efficiency of the integration of human knowledge into the model via a fuzzy reasoning. When more plant data are available, the FIS-based system will be further tested and the possibility of exploiting new data to tune its parameters will be investigated.

Acknowledgments

The work described in the present paper was developed within the project entitled “Effect of Transients on Quality of Continuously Cast Product” (Contract No. RFSR-CT-2009-00005) that has received funding from the Research Fund

for Coal and Steel of the European Community. The sole responsibility of the issues treated in the present paper lies with the authors; the Commission is not responsible for any use that may be made of the information contained therein.

References

1. Smith, J.: Introduction to digital filters with audio applications. W3K Publishing (2008)
2. Ker, M.D., Yen, C.C.: New transient detection circuit for on-chip protection design against system-level electrical transient disturbance. *IEEE Trans. on Industrial Electronics* 57(10), 3533–3543 (2010)
3. Petersen, R., Pant, P., Lopez, P., Barton, A., Ignowski, J., Josephson, P.: Voltage transient detection and induction for debug and test. *Test Conference* (2009)
4. Kwong, M.D., Lefebvre, R.: Transient detection of audio signals based on an adaptive comb filter in the frequency domain. In: 37th Asilomar Conf. on Signals, Systems and Computers, vol. 1, pp. 542–545 (2003)
5. Plett, M.I.: Transient Detection With Cross Wavelet Transforms and wavelet coherence. *IEEE Trans. Signal Proc.* 55(5), 1605–1611 (2007)
6. Pao, Y.H., Hemminger, T.L., Adams, D.J., Clary, S.: An episodal neural-net computing approach to the detection and interpretation of underwater acoustic transients. In: *IEEE Conf. Neural Networks for Ocean Engineering*, pp. 21–28 (1991)
7. El Safty, S., Ghariieb, S., El Latif Badr, A., Mansour, M.: A wavelet fuzzy expert technique for classification of power transformer transients. In: *Int. Conf. Power Syst. Tech. PowerCon* (2006)
8. Baldwin, J.F.: Fuzzy Logic and Fuzzy Reasoning. *International Journal of Man-Machine Studies* 11, 465–480 (1978)
9. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies* 7(1), 1–13 (1975)
10. Zhang: Neural networks for classification: a survey. *IEEE Trans. on Systems, Man and Cybernetics. Part C* 30(4) (November 2000)
11. Foresee, H.: *Proceedings of the International Joint Conference on Neural Networks* (June 1997)
12. Daubechies: Ten lectures on wavelets, CBMS-NSF conference series in applied mathematics. SIAM, Philadelphia (1992)
13. Datta, V., Anand, G.V.: Transient Detection in Non-Gaussian Noise by Wavelet Packet Transform. In: *Int. Conf. Sig. Proc. & Comm., Bangalore, July 18-21*, pp. 1–5 (2010)
14. Walker: *A Primer on Wavelets and Scientific Applications* (1999)

Oesophageal Voice Harmonic to Noise Ratio Enhancement over UMTS Networks Using Kalman-EM

Marouen Azzouz, Begoña García Zapirain, Ibon Ruiz, and Amaia Méndez

Deustotech-Life Unit, Deusto Institute of Technology, University of Deusto
Avda. de las Universidades 24, Bilbao, Spain, 48007
{mbgarciazapi, ibruiz, amaia.mendez}@deusto.es

Abstract. Oesophageal voice is characterized by its extremely low intelligibility. An algorithm based on Kalman Expectation Maximization (EM) has been developed. The noise presented in the model, state and measurement noise has been optimized in order to improve the algorithm results. The database consists of “a” phonemes of several patients having undergone a total laryngectomy. Additionally, the effect of the algorithm on the UMTS mobile communication context has been tested. The tests show that the algorithm gives the best results when it is used as state noise an oesophageal noise and brown noise as measurement noise. The global percentage enhancement is 75.78%.

Keywords: Mobile communication, GPRS/UMTS networks, Speech enhancement, Kalman filters, Oesophageal Speech.

1 Introduction

Laryngectomy is the removal of the larynx and separation of the airway from the mouth, nose and oesophagus. The laryngectomee breathes through an opening in the neck: a stoma. The operation is carried out in cases of laryngeal cancer. However, many laryngeal cancer cases are only treated with radiation and chemotherapy or other laser procedures, and laryngectomy is performed when those treatments fail to conserve the larynx [1]. Consequently, complete recovery is not yet universal or common. Nevertheless, we can try to improve the way in which oesophageal speech is perceived and thus help oesophageal speaking people in the vital human need that communication is. Nowadays, technological development has led to a vast range of possibilities that would never have been possible two decades ago. This project may be included knowledge in three different fields: GPRS/UMTS networks, speech enhancement and oesophageal speech. The terms speech enhancement and speech cleaning properly refer to improvement in the quality or intelligibility of a speech signal and the reversal of degradations that have corrupted it respectively. In practice, however, the two terms are used interchangeably. In fact, oesophageal speakers even encounter speaking difficulties with direct speech, so we can imagine that it would not be easy for them to transmit easily understood information over the phone.

The main concern of this project has been to try and meet this challenge by taking advantage of technological progress and merging knowledge in telecommunications networks, speech enhancement and oesophageal speech itself. We intend to participate

in enhancing oesophageal speech generally, and particularly using telecommunications networks. We tried to obtain sensitive improvement results regarding the HNR criteria for oesophageal sequences transmitted over GPRS/UMTS and filtered with a Kalman filter.

The core objectives identified at the beginning of this project can be expressed as follows:

1. Testing AMR-NB (Adaptive Multi-Rate Not Balanced) coding HNR improvement for oesophageal sequence. Considering the particular nature of oesophageal speech, the answer is not immediate and thus requires experimenting.
2. Finding the noise colour producing the best HNR improvement. We know that oesophageal speech is incompatible with the white noise AR-model.
3. Deciding on whether to filter oesophageal speech before or after AMR-NB coding. Since oesophageal speech will be transported over a GPRS/UMTS network, it is important to know whether the enhancement process will occur before or after AMR-NB coding.

In order to achieve these aims, we had to find out which algorithm goes best with oesophageal speech: whether we should apply simple Kalman filtering or use Kalman-EM (Expectation Maximization) instead.

2 Methodological Fundamentals

Oesophageal speakers bring about speech by insufflating air into the oesophagus through the mouth or noise. The insufflated air produces a controlled belch that is shaped by the articulators and radiated from the lips. However, insufflations limit duration between air injection gestures and are associated with an undesired audible injection noise, referred to as “injection gulp”. The main idea is to integrate this noise into the modelling of oesophageal speech in Kalman Filtering and correlate it to the speech and not the measurements, which means that we need to modify the Kalman algorithm and add a coloured state noise. Furthermore, we would like to maintain coloured measurement noise. The solution proposed is that of pre-filtering the measurement of noise in order to “whiten” it; then apply the KEM (Kalman Expectation Maximization) algorithm and finally render the measurement noise.

2.1 Kalman Filtering with Coloured Measurement Noise

The state-space speech model can be written as follows [1]:

$$\text{Measurement: } y(n) = s(n) + v(n) \tag{1}$$

$$\text{State equation: } s(n) = -\sum_{i=1}^p a_{i,n} \cdot s(n - i) + w(n) \tag{2}$$

where $\{y_i\}_{1 \leq i \leq N}$ and $\{s_i\}_{1 \leq i \leq N}$ represent the measurements respectively and the state $\{a_{i,k}\}_{1 \leq i \leq p}$ are LPC coefficients for the speech over the analysis frame indexed by k .

$$\text{The state-space model is: } x(n + 1) = A_n \cdot x(n) + g \cdot w(n) \tag{3}$$

$$\text{with: } x(n) = [s(n - p + 1) \dots s(n - 1) s(n)]^T, \quad g = [1 \ 0 \ \dots \ 0]_{p \times 1}^T \quad \text{and}$$

$$A_n = \begin{bmatrix} -a_{1,n} & \dots & -a_{p,n} \\ 1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{p \times p}$$

$$y(n) = h^T \cdot x(n) + v(n) \tag{4}$$

where $h = [1 \ 0 \ \dots \ 0]_{p \times 1}^T$

$v(n)$ and $w(n)$ are samples at instant n of the measurement and state noise respectively. At this moment, we suppose that they are uncorrelated, centred, white and Gaussian.

Introducing coloured noise, the system's equations become [2], [3] and [4]:

$$y(n) = \bar{h}^T \cdot \bar{x}(n) \tag{5}$$

$$\bar{x}(n) = \bar{A}_n \cdot x(n - 1) + \bar{g} \cdot w(n) \tag{6}$$

where, $\bar{g} = \begin{pmatrix} g & 0 \\ 0 & g_v \end{pmatrix}$, $\bar{h}^T = (h^T \ h_v^T)$, $\bar{A}_n = \begin{pmatrix} A_n & 0 \\ 0 & A_v^n \end{pmatrix}$, $\bar{x}(n) = \begin{pmatrix} x(n) \\ v(n) \end{pmatrix}$,

$$v(n) = [v(n) \ v(n - 1) \ \dots \ v(n - q + 1)]^T, \quad g_v = h_v = [1 \ 0 \ \dots \ 0]_q^T \tag{7}$$

where $v(n)$ is a coloured noise. LPC modelling for the noise is:

$$v(n) = -\sum_{i=1}^q b_{i,n} \cdot v(n - i) + e(n) \tag{8}$$

where $\{b_{i,k}\}_{1 \leq i \leq p}$ are LPC coefficients for speech over the analysis frame indexed by k and $e(n)$ is a White Gaussian centred noise.

As described above, the Kalman filter addresses the general problem of trying to estimate the state of a discrete time-controlled process governed by a linear set of equations. However, this algorithm needs to be adapted to the specifics of our project, which is oesophageal speech processing.

2.2 EM Concept

Supposing we know \bar{A}_n (7), we would not need the EM method because the optimal solution is given in the Minimum Mean Square Error sense with the Kalman filter. But oesophageal sequences are seldom clear and thus require optimal estimation of the state space parameters before applying the filter. The proposed solution is to estimate these parameters dynamically within the algorithm in an iterative way. This means estimating the same parameters as many times as necessary until we have a satisfactory estimation.

Let $\theta = [a_{i,1 \leq i \leq p} \ g_w \ b_{i,1 \leq i \leq q} \ g_v]$ be the parameter we try to estimate within every iteration of the EM method. This parameter should maximize the probability of obtaining the data we are trying to estimate.

$x = [s \ v]$ is denoted as the “complete data” since each and every observation $z(n)$ depends on these variables that constitute the system's “unobserved state”. However, the choice of the complete data has to be exact, which means that it must include all the unobserved state, but no more, since it reflects directly on both the performance of the algorithm and the calculation.

The main idea behind the EM method is that, even though we do not know $x(n) = [s_p(n) v_q(n)]$, maximization of $p(x|\theta)$ can lead to having an estimate of the parameter in the Maximum Likelihood sense: $\hat{\theta} = \operatorname{argmax}_{\theta} \{p_x(x|\theta)\} = \operatorname{argmax}_{\theta} \{\ln(p_x(x|\theta))\}$.

Given the fact that we do not know the complete data, we will work on the expectation of $p_x(x|\theta)$ given the knowledge of the observed data and the current value of $\hat{\theta}$ called $\hat{\theta}_i$ with respect to the EM iterations, hence the name Expectation in the EM method. It consists of finding $Q(\theta|\hat{\theta}_i) = E_{\hat{\theta}_i}[\ln(p_x(x|\theta)) | z]$.

The maximization step (M-step) consists of obtaining the estimate at the $i + 1^{\text{th}}$ iteration given by $\hat{\theta}_{i+1} = \operatorname{argmax}_{\theta} \{Q(\theta|\hat{\theta}_i)\}$, as described in [5].

2.3 Database

No commercial databases available including oesophageal voices. Thus, the whole used private database was recorded with the help of a local otolaryngologist and the local laryngectomee association, who kindly helped in this task. The final database was composed by 316 utterances: from these, 119 were of patients with slight pathologies (mostly nodules and polyps), 108 were oesophageal voices of laryngectomized persons and 89 healthy voices. With all this information, different tests were performed along the entire research work, checking the accuracy of the algorithms and analyzing the results. From now onwards, a subset of 12 samples of oesophageal voice will be reflected in the rest of the results section in order to keep a balance between readability and richness of the details.

3 Experiment Design

Figure 1 is a block diagram containing the essential “units” or components showing the design flow adopted in the algorithm. The codec and filtering process is like a kind of black box that takes the original oesophageal speech as input and renders the final enhanced sequence.

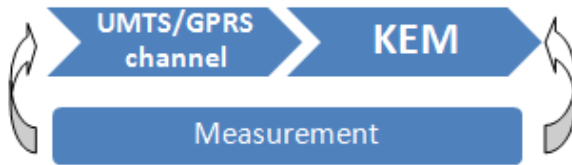


Fig. 1. General block diagram

HNR measurement is taken before coding/decoding the oesophageal sequence. We then filter the decoded sequence and measure the HNR value again. This design flow has been questioned and proved to give better results as regards the HNR criteria than using Kalman filtering followed by UMTS/GPRS coding.

3.1 UMTS/GPRS Channel

This block represents the AMR-NB codec used over the GPRS/UMTS network and is described in Figure 2. This codec operates with eight bit rate modes ranging from 4.75 kbps to 12.2 kbps. The channel used for this work is the 12.2 kbps channel, also called Enhanced Full rate (EFR). This block consists of two steps: encoding and decoding. It is mandatory to specify the channel. Channels are indexed from 0 to 7. We used the 12.5 kbps/s channel, which corresponds to channel 7. We chose this channel for several reasons:

- Although EFR helps to improve sound quality over the phone, it consumes just 5% more energy than the GSM Full Rate coding standard.
- In 1995, EFR was selected by ETSI as the standard industry codec for GSM/DCS networks.
- Finally, technology is constantly progressing, so improving oesophageal sound quality over the actual best bite rate codec sets a solid ground and insurance for upcoming technologies that oesophageal speech quality can be even better enhanced over networks.

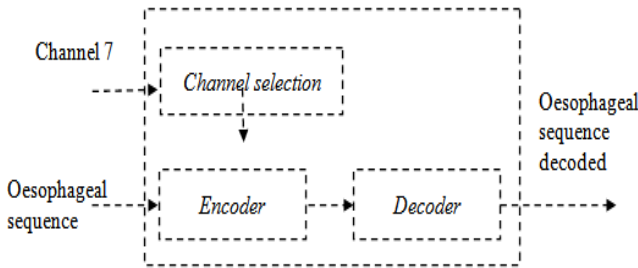


Fig. 2. AMR-NB codec block diagram

3.2 Kalman Filtering Block

The Kalman filtering block has, in fact, been subjected to many changes, for many Kalman variations have been tested with the HNR criteria in four noise colours: white, pink, brown and oesophageal non-speech noise. Kalman-EM was chosen as regards the HNR criteria. The next step was to inject a specific oesophageal noise for the state equation instead of the white noise along with the brown noise in measurement equations [6] and [7].

3.3 Harmonic to Noise Ratio (HNR)

HNR is a general evaluation of noise present in the analyzed signal. It is defined as (4), with $rp(0)$ and $rap(0)$ being the respective energies of the periodic and aperiodic components [8]:

$$HNR = \frac{r_p(0)}{r_{ap}(0)} \quad (9)$$

The measurements were made with the help of MDVP from Kay Electronics [9], important software that provides good estimations of a signal's parameters.

4 Results

This section presents a comparison between the HNR parameters of the processed and original signal from a private database consisting of "a" phonemes of several patients having undergone a total laryngectomy (no public databases available). Several performances were carried out, although only some of the obtained results will be described.

Table 1. KEM algorithm with white, pink and brown noise

Utterance	Original HNR (dB)	White HNR (dB)	Pink HNR (dB)	Brown HNR (dB)
a1	-0,40	0,76	-0,07	2,56
a2	-4,21	-2,07	-2,66	-0,95
a3	-7,79	-5,85	-6,03	-4,41
a4	-7,97	-5,69	-5,65	-5,43
a5	0,13	0,86	1,29	2,01
a6	-6,4	-5,32	-5,33	-3,30
a7	-7,01	-6,37	-6,70	-5,06
a8	-6,65	-4,92	-4,39	-4,08
a9	-9,86	-8,28	-8,48	-5,90
a10	-3,92	-2,92	-2,92	-2,32
a11	-4,05	-2,25	-1,81	-1,35
a12	-3,22	-2,24	-1,75	0,06

The different algorithms tested are presented as follows: the effect of applying the UMTS coder, the effect of applying the Kalman-based enhancement algorithm (Kalman + EM concept), the effect of combining UMTS & KEM and the effect of combining UMTS & KEM with oesophageal state noise and one of the coloured noise (white, pink or brown) as measurement noise.

- As mentioned above, we want to know whether HNR increases or decreases after applying the AMR-NB coder for "raw" oesophageal speech. In the results of this performance it can be notice that the AMR-NB coder brings about significant improvement compared to the original oesophageal signal. The percentage of enhancement according to the performance criteria is 35.98%.
- Three different noises were used in order to research the proposed algorithms (the Kalman-EM and Kalman_block algorithm): white, pink and brown, which were injected as measurement noises; the state noise is white. Table 1 shows the differences between the KEM algorithms for oesophageal voice with white, pink

and brown noises respectively. A global performance enhancement can be appreciated in all the tests. The HNR percentage enhancement is 42.72% for the Kalman_block algorithm and 54.11% for the Kalman-EM for the best choice of measurement brown noise.

- In this section we intend to discover whether it is better to apply KF before or after GPRS/UMTS coding. It was decided to use the KEM algorithm with a brown measurement noise. We notice that when the AMR-NB coder is applied after KEM, the results do not follow a fixed behaviour pattern. Although the enhancement is positive compared to the original sequences, it is not constantly positive compared to KEM-filtered sequences. When applying KEM filtering after AMR-NB codec, we detect a global enhancement. The improvement is slight but “stable”.
- At this point, we know that the best design flow is to apply the AMR-NB coder and then filter the coded sequence with Kalman-EM with brown measurement noise. HNR variations after applying oesophageal noise will now be discussed. We applied this experience under the same circumstances, which means that we applied the KEM algorithm with colored state noise for AMR-NB coded sequences. First, we applied esophageal noise as a measurement noise and brown noise as state noise and then we changed the reference noises. The best choice, it seems, is to apply an oesophageal state noise and a brown measurement noise. The process chosen offers significant HNR improvement: the global percentage enhancement is 75.78% compared to the original ones.

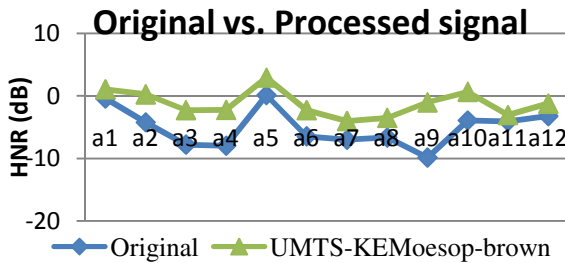


Fig. 3. HNR before (blue) and after (green) processing with oesophageal noise as state noise and brown noise as measurement noise

5 Conclusions and Future Work

As regards the first step, it can be concluded that HNR increases after we apply the GSM codec for oesophageal speech. The sensation of intelligibility is greater in oesophageal voices after having passed them through the coder. This is due to the fact that the GSM coder filters the oesophageal noise present in the voices, thus producing a greater final sensation. The second step is to deal with the coloured measurement noise, as well as trying to find which noise we should use. For this purpose we used 4 noises, namely: pink, brown, white and oesophageal non-speech noise. The best results were obtained when applying oesophageal noise as state noise and brown

noise as measurement noise. The next step was to determine whether it is better to apply Kalman filtering before or after the GSM codec. We noticed that applying the coder after Kalman filtering not only decreases HNR but also “deteriorates” the oesophageal sequence. This means that oesophageal sequences may lose the smooth form they gain after applying the filter. The process chosen offers significant HNR improvement: global percentage enhancement is 75.78% compared to initial oesophageal sequences among the sequences tested. This has led us, however, to say that brown measurement noise and oesophageal state noise are the best choice. It is possible that the results of this project could have been further enhanced since we tried just three coloured noises among a vast range of possibilities. We could also have tried some different Kalman implementations. Another possibility is to use a voice activity detector so that we can estimate noise magnitude, for example, and then perhaps obtain a better enhanced sequence. Moreover, we could extend this research work by working on different GPRS channels. Although working on the other AMR-NB channels would simply be a repetition of the same task 8 times, it would be a good chance to test different bite rate channels and thus learn more about oesophageal speech behaviour.

Acknowledgments. The authors wish to acknowledge the help of the “Asociación Vizcaína de Laringectomizados”, whose members voluntarily lent their voices for this investigation. Special mention should also go to the Education, University and Research Department from the Basque Government for their support of the project.

References

1. Paliwal, K.K., Basu, A.: A Speech Enhancement Method Based on Kalman Filtering. In: Proc. ICASSP 1987, pp. 177–180 (1987)
2. Gabrea, M.: Robust Adaptive Kalman Filtering-based Speech Enhancement Algorithm. In: Proc. ICASSP 2004, pp. 301–304 (2004)
3. García, B., Vicente, J., Ruiz, I., Alonso, A., Loyo, E.: Oesophageal Voices: Glottal Flow Restoration. In: ICASSP 2005, pp. IV-141–144 (2005)
4. Gibson, J., Koo, B., Gray, S.: Filtering of coloured noise for speech enhancement and coding. *IEEE Trans. Signal Process.* 39(8), 1732–1742 (1991)
5. Gannot, S., Burshtein, D., Weinstein, E.: Iterative and sequential Kalman filter-based speech enhancement algorithms. *IEEE Trans. Speech, Audio Process.* 6(4), 373–385 (1998)
6. Goh, Z., Tan, K.-C., Tan, B.: Kalman-filtering speech enhancement method based on a voiced-unvoiced speech model. *IEEE Trans. Speech, Audio Process.* 7(5), 510–524 (1999)
7. Sorqvist, P., Handel, P., Ottersten, B.: Kalman filtering for low distortion speech enhancement in mobile communication. In: Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, Munich, Germany, vol. 2, pp. 1219–1222 (April 1997)
8. Severin, F., Bozkurt, B., Dutoit, T.: HNR extraction in voiced speech, oriented towards voice quality analysis. In: Proc. EUSIPCO 2005, Antalya, Turkey (2005)
9. Zelcer, S., Henri, C., Tewfik, T.L., Mazer, B.: Multidimensional voice program analysis (MDVP) and the diagnosis of pediatric vocal cord dysfunction. *Ann. Allergy Asthma Immunol.* 88, 601–608 (2002)

Study of Various Neural Networks to Improve the Defuzzification of Fuzzy Clustering Algorithms for ROIs Detection in Lung CTs

Alberto Rey, Alfonso Castro, and Bernardino Arcay

Faculty of Computer Science, Campus Elviña, University of A Coruña, Spain
{alberto.rey,alfonso.castro,bernardino.arcay}@udc.es

Abstract. The detection of pulmonary nodules in CT images has been extensively researched because it is a highly complicated and socially interesting matter. The classical approach consists in the development of a computer-aided diagnosis (CAD) system that indicates, in phases, the presence or absence of nodules. A common phase of these systems is the detection of regions of interest (ROIs), that may correspond to nodules, in order to reduce the searching space. This paper evaluates the use of various neural networks for the defuzzification of the output of fuzzy clustering algorithms, in order to improve the detection of true positives and the reduction of false positives. Also, they are compared to the results from a support vector machine (SVM).

1 Introduction

The high complexity of the lung structure, with its large amount of branches, and the difficulty in detecting among a great variety of lung diseases, have turned the analysis of pulmonary images into one of the most studied areas of medical image analysis.

Within this area, the lung carcinoma is particularly interesting, due to its high incidence in modern society and its remarkably high mortality rate; early detection remains essential and may in fact increase the survival rate in certain cases up to 50% [1].

Even though the most commonly used image modality is low-cost thoracic radiography, it has become more customary to use detailed definition images obtained from high resolution CTs: these can be acquired in time intervals below one respiration and reach resolutions below 1 mm.

The analysis and interpretation of these scans by radiologists is a difficult and time-consuming task, due to the large amount of information provided and the fact that nodules are hardly different from other non-pathological structures. To overcome these difficulties, a wide variety of CAD systems has been developed [2] [3]; our research group is developing a system of this kind.

Currently, our system focuses on the detection of ROIs. In previous works [4] [5], we analysed several fuzzy clustering algorithms that provide good results for the detection of ROIs in lung CTs, even though the number of false positives

provided by those algorithms is not optimal. Traditionally, the output of the algorithms is defuzzified by assigning the pixel to the cluster with the highest membership value. Our objective is to analyse the use of different neural networks to improve defuzzification and to reduce false positives and increase true positives, the inputs being the memberships calculated by the algorithms. Also, we compare their results with those of an SVM.

2 Materials

For the purposes of our analysis, we used a training and testing set that consists in several high-resolution CT lung images provided by the LIDC (Lung Image Database Consortium) database [6]. The LIDC project was an initiative of the NCI (National Cancer Institute) to create a reference repository of CT lung images for the evaluation and implementation of CAD systems for the detection of pulmonary nodules.

The repository provides a varied set of annotated lung images, especially low-dose helical CT scans, in which each image has associated relevant data regarding the presence or absence and the characteristics of any observed nodule(s). This data is stored in an XML file that specifies the type and contour of the nodule observed in the image.

The images are stored according to the DICOM standard in size 512x512, with a pixel size that varies from 0.5 to 0.8 mm and a grayscale of 16 bits in Hounsfield Units (HU).

3 Fuzzy Algorithms Analysed

The detection task of the initial ROIs was carried out by three different fuzzy clustering algorithms. We selected these algorithm types because they provide the best results [4] and [5] with a similar dataset of CT images from LIDC.

3.1 SFCM (Spatial Fuzzy C-Means)

This algorithm [7] is a modified version of the standard FCM (Fuzzy C-Means): it has increased robustness against noise and achieves more homogeneous regions by using a spatial function.

During the first step, this algorithm obtains the initial memberships for each pixel applying the traditional FCM algorithm, with omission of its iterative process. The subsequent step calculates the spatial function value for each pixel in the image with function $h_{ij} = \sum_{k \in NB(x_j)} u_{ik}$, where $NB(x_j)$ represents a resizable square window, centered on each pixel x_j under consideration.

The next step consists in calculating the spatial membership function with the help of the previous information:

$$u'_{ij} = \frac{u_{ij}^p h_{ij}^q}{\sum_{k=1}^C u_{kj}^p h_{kj}^q}, \quad (1)$$

where p and q are control parameters for the importance of functions u_{ij} and h_{ij} . Finally, the new centroids are obtained by the same FCM procedure:

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m x_i}{\sum_{i=1}^N u_{ij}^m} . \tag{2}$$

The iteration process will stop when the maximal difference between two centroids lies below a predetermined threshold. Otherwise, it will recalculate the FCM in order to commence further iteration.

3.2 SKFCM (Spatial Kernelized Fuzzy C-Means)

This algorithm, proposed in [8], introduces a penalty factor with spatial neighborhood information to the KFCM (Kernelized Fuzzy C-Means). The kernel function allows us to transform the original low dimension space into a higher space, where complex nonlinear problems can be treated more efficiently. The present work only uses the Gaussian radial basis function kernel.

During the initialization of the algorithm, the number of clusters c , the initial class centroids v , the initial memberships u and the threshold ϵ must be determined.

In the first step, the memberships factors are calculated with the function:

$$u_{ik} = \frac{\left((1 - K(x_k, v_i)) + \frac{\alpha}{N_R} \sum_{r \in N_k} (1 - u_{ir})^m \right)^{\frac{-1}{(m-1)}}}{\sum_{j=1}^c \left((1 - K(x_k, v_j)) + \frac{\alpha}{N_R} \sum_{r \in N_k} (1 - u_{jr})^m \right)^{\frac{-1}{(m-1)}}} . \tag{3}$$

Finally, the centroids are updated as follows:

$$v_i = \frac{\sum_{k=1}^n u_{ik}^m K(x_k, v_i) x_k}{\sum_{k=1}^n u_{ik}^m K(x_k, v_i)} . \tag{4}$$

As in the other algorithms, repeat these steps until condition $\max_{i,k} \|u_{ik}^t - u_{ik}^{t-1}\| \leq \epsilon$ is satisfied, where epsilon is a determined threshold.

3.3 MKSFCM (Modified Kernelized Spatial Fuzzy C-Means)

This algorithm, proposed by Castro et Al. in [5], combines the algorithms SFCM as introduced in Section 3.1 and the KFCM proposed in [9]. Its primary objective is to combine the main advantages of both methods: homogeneity and robustness against noise and outliers. This algorithm consists of the following steps:

1. Calculation of the membership function:

$$u_{jk} = \frac{(1/d^2(x_j, V_k))^{1/(q-1)}}{\sum_{j=1}^C (1/d^2(x_j, V_k))^{1/(q-1)}} , \tag{5}$$

where $d^2(x_j, V_k) = K(x_j, x_j) - 2K(x_j, V_k) + K(V_k, V_k)$, q determines the fuzziness of the clusters, C is the number of clusters, V are the centroids, and K is the kernelized function.

2. Introduce the spatial information as in the SFCM procedure, applying Eq. [11](#) to obtain the membership factors.
3. The new kernel matrix $K(x_j, \hat{V}_k)$ and $K(\hat{V}_k, \hat{V}_k)$ is calculated as follows:

$$K(x_j, \hat{V}_k) = \phi(x_j) \cdot \phi(\hat{V}_k) = \frac{\sum_{i=1}^N (u_{ik})^q K(x_i, x_j)}{\sum_{i=1}^N (u_{ik})^q} . \quad (6)$$

4 Neural Networks and SVM Analysed

Neural networks are widely applied in the field of pattern recognition and classification. Several network architectures can be applied to these fields, but we have selected the approaches that are commonly used in medical imaging, such as Backpropagation network and Radial basis network. [10](#)

The standard Support Vector Machines were designed specifically for binary classification and regression estimation. However, they have also been extended to multi-class problems. We selected the multi-class approach provided by Weston and Watkins [11](#) that tries to solve the problems in one single optimization.

4.1 Backpropagation Neural Network (BPNN)

This approach follows the architecture of the Feed-Forward networks, which consists of a group of neurons arranged in multiple layers using unidirectional connections, with weights, between nodes in the adjacent layers. It uses the backpropagation supervised learning algorithm to dynamically alter the values of weights and bias for each neuron in the network, so as to minimize the output error.

In this analysis, we have selected a common three-layer structure with one hidden layer, using the sigmoid and linear transfer functions in the hidden and output nodes, respectively. The modification of the weights is carried out using different kinds of training algorithms in the categories of gradient descent, conjugate gradient, and quasi-newton, which are derivations of the basic backpropagation algorithm.

4.2 Radial Basis Function Neural Network (RBFNN)

The architecture of this network is composed by a three-layer feed-forward network. The neurons of the hidden layer compute their output through a radial basis function, i.e. the symmetric Gaussian function.

The RBFNN can be trained by different approaches, but this paper uses a hybrid learning approach that can be divided into two stages, combining unsupervised and supervised strategies. In the first step, the parameters of the Gaussian function are obtained by using an unsupervised algorithm, i.e. a variation of the k-Means proposed in [12](#). This allows us to reduce the computational cost, which is very high for the original algorithm due to the use of a special, tree-shaped data structure with additional data.

Finally, the output weight is calculated by a supervised method, i.e. the least mean square.

4.3 Multi-class SVM

This type of SVM was introduced by Weston and Watkins and proposes a method to solve multi-class problems in a single optimization approach.

This approach constructs k two-class decision rules, where the m th function $w_m^T \phi(x) + b_m$ separates training vectors of the m class from the other generated vectors. These k decisions functions are obtained by solving a single optimization problem.

The formulation of the optimization is as follows:

$$\begin{aligned}
 w^{ij}, b^{ij}, \xi^{ij} \quad & \frac{1}{2} \sum_{m=1}^k w_m^T w_m + C \sum_{i=1}^l \sum_{m \neq y_i} \xi_i^m w_{y_i}^T \phi(x_i) \\
 & + b_{y_i} \geq w_m^T \phi(x_i) + b_m + 2 - \xi_i^m, \\
 \xi_t^{ij} \geq 0, \quad & i = 1, \dots, l, \quad m \in \{1, \dots, k\} \setminus y_i, \quad (7)
 \end{aligned}$$

where the decision function is $argmax_{m=1, \dots, k} (w_m^T \phi(x) + b_m)$.

5 Evaluation Metrics

For the purposes of this paper, we selected 23 preprocessed CT images that contain a representation of the different types of lung nodules that might appear, i.e initial phase, juxtapleural, isolated, etc., and that are acquired from different thoracic zones.

We began by applying the fuzzy algorithms to the training set and using the obtained memberships as input to the neural networks and SVM. This training set is constructed according to the following steps:

1. Select a set of pixels from each cluster, proportional to the percentage of each cluster in the segmented image. For example, if one cluster represents 20% of a segmented image, select only 20% of pixels from this cluster. Thereby, not only the training set is reduced in size, but the original characteristics of the input space are maintained as well. We observed that better results were obtained by separating the ROI marked as nodule into two sets: one set with the true nodule pixels, another set with the remaining marked pixels, representing the false positives. Note that the results of this analysis follow this approach.
2. Obtain the membership factors for the previously selected pixels, using the non-defuzzified output of the fuzzy algorithms. These factors for each pixel are disposed as a row in the training set and represent one input for the networks and SVM.
3. Repeat steps 1 and 2 for each image that serves as an input for the training set.

The quality of the obtained results is quantified by means of true positives (TP) and false positives (FP), because the mask images only have information

about the nodule region, omitting the information referring to the other elements of the CT image.

The results are represented in the form of charts and tables that summarize the best results obtained with different selections of parameters. The main aim is to observe the enhancement of ROI detection of the fuzzy algorithms and its stability, when they are combined with the neural networks and SVM.

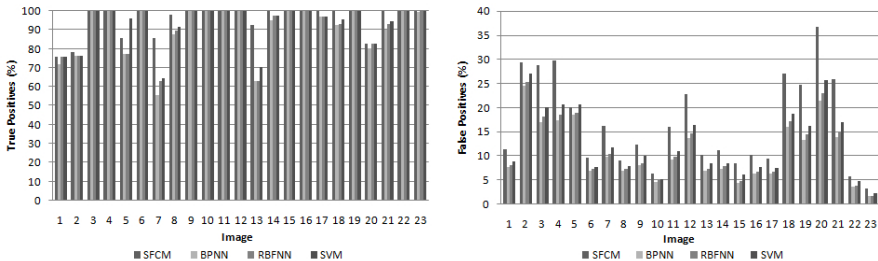
6 Results

Space limitations oblige us to present only the results obtained for the different combinations of fuzzy algorithms with networks and SVM using their optimal parameters, and to omit the results obtained with other selections of parameters. In the case of network architecture, we varied the number of neurons of the hidden layer in the [5, 60] range, and used 3 different training functions and various weights initializations. For the case of SVM, we tested three kernel types, varying their gamma parameter and regularization cost C in the [0.05, 2] and [10, 600] ranges, respectively.

The results obtained by combining the SFCM algorithm with both networks maintain the true positives rates of the fuzzy algorithms (Fig. 1a), providing values above 80% for most images, even though this percentage decreases by 25%-30% for Images 7 and 13. RBFNN presents slightly higher true positives rates than BPNN, the greater oscillation being of 8% for Image 7. In terms of false positives (Fig. 1b), the network approaches yield good results, achieving a reduction above 10% for various images and showing an overall 3% decrease for the rest, the BPNN obtaining the lowest rates.

In comparison with the SVM technique, these results are similar; the main difference lies in the improvement of the true positives rate of Image 5 by 18% and the slight general worsening of around 1% with regard to false positives.

In essence, the networks provide good results in maintaining the true positives rates of the fuzzy segmentation and significantly reducing the false positive rates.



(a) True positives rates.

(b) False positives rates.

Fig. 1. Rates for the SFCM and its combinations

Tests for the SKFCM algorithm with its combinations (Fig.2a, Fig.2b) obtained very different results for each network. We therefore analysed each network combination separately. The BPNN approach reveals a misclassification problem in several cases, e.g. Images 7 and 21, with success rates below 26%, and Images 4 and 20, with false positives reaching 37% and 41%, respectively. RBFNN obtains a high true positives ratio of up to 75% for most images, but this result is nevertheless negative due to the increase in false positives rates in some images and mainly by serious classification errors. Moreover, the SVM combination shows a strong decrease in false positives rates, reaching 23% in the case of Image 22, and 5%-15% for the remaining images. Nevertheless, as in the BPNN case, the rate of true positives drops to 45% for several images.

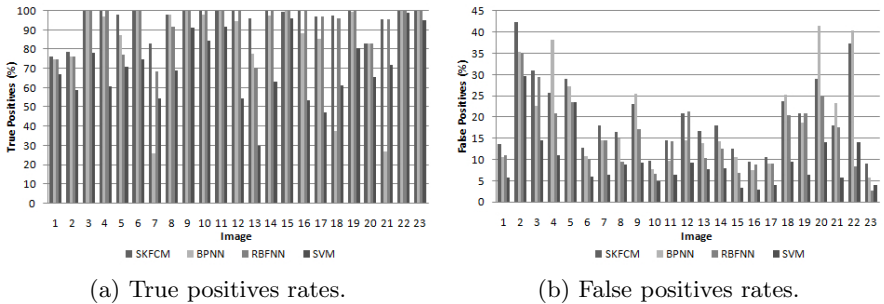


Fig. 2. Rates for the SKFCM and its combinations

Finally, Fig.3a and Fig.3b, indicate the results obtained by the MKSFCM algorithm and its combinations. The BPNN combination shows a stable behaviour, maintaining the success rate for most images in concordance with the fuzzy segmentation, although there exist some noteworthy variations, such as a worsening of 15% for Images 7 and 13 and an increase above 8% for Images 1,2, and 8. In Fig.3b, the false positives rate decreases for most cases an average of 2%, except for Images 2 and 8, where the rate of increase is the same. This fact is offset by the improvement of the true positives rate for these images.

The RBFNN combination obtains a general decrease of the false positives rate for all dataset, Fig.3b, but this result is not good because the success rate drops by 30% for several cases (i.e 2,5,7,18), which reveals a classification problem, since the pixels are assigned to another cluster.

The SVM technique, in combination with this algorithm, provides a similar result than the BPNN with regard to true positives, since it maintains its rates except for slight variations in some images. The false positives rate decreases for the major part of the dataset in terms of a stable 2% without any remarkable increase.

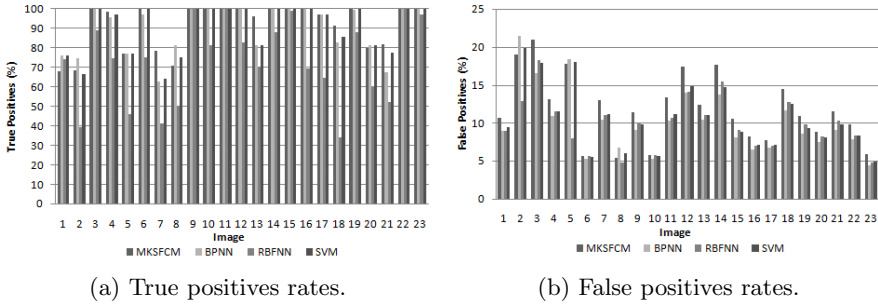


Fig. 3. Rates for the MKSFCM and its combinations

The BPNN and SVM combinations present a stable behaviour, obtaining good rates in both true positives and false positives. Even so, the decrease of the false positives rates is not too high, due mainly to the very good results obtained by the MKSFCM algorithm in the initial segmentation of ROIs.

7 Conclusions

This paper has analysed the combination of various fuzzy clustering algorithms and two types of network architectures, as well as an approach of the SVM technique. Our main purpose was to achieve a good method to improve the defuzzification phase of the fuzzy analysed algorithms in order to obtain better segmentation of the ROIs, reducing the false positives rates without losing true nodule pixels.

The combination of SFCM algorithms with networks obtains good results, reducing the false positives rate for all images in a significant percentage, reaching up to 10% and maintaining the true positives rate of the initial segmentation for most images at over 80%. In comparison with the SVM approach, the results are quite similar, but with a slight but irrelevant percentage of false positives by the SVM.

The SKFCM combinations present variable results, all with classification problems. It should be noted that even though the RBFNN combination obtains good values for both rates, it has serious problems with the classification of some images that may affect the subsequent stages of CAD implementation, nullifying our goals. These combinations do not provide good results in data sets with high variability, principally due to the difficulty in differentiating the nodules from the other lung structures.

In the case of combinations with the MKSFCM algorithm, the BPNN and SVM approaches obtain the better results, preserving the true positives rate of the initial segmentation in most images and even increasing it for some. Moreover, both methods provide a general reduction of the false positives rate of 2%, remaining below 15% for the major part of the dataset.

Therefore, and according to this analysis, the SFCM combinations with networks and SVM and those with MKSFCEM, with the exception of RBFNN, obtain valuable improvements in the defuzzification phase, that in our opinion can be incorporated into the nodule detection CAD system.

Acknowledgments. This work was supported by project 09SIN006PR of the I+D Suma Call of the Plan Galego de Investigación, Desenvolvemento e Innovación Tecnolóxica (INCITE), Xunta de Galicia.

References

1. Weir, H.K., et al.: Annual report to the nation on the status of cancer, 1975-2000. *Journal of the National Cancer Institute* 95(17), 1276–1299 (2003)
2. Lee, Y., Hara, T., Fujita, H., Itoh, S., Ishigaki, T.: Automated detection of pulmonary nodules in helical CT images based on an improved template-matching technique. *IEEE Transactions on Medical Imaging* 27(7), 595–604 (2001)
3. Lei, C., Li, X., Jie, Z., Wufan, C.: Automated lung segmentation algorithm for CAD system of thoracic CT. *Journal of Medical Colleges of PLA* 23(4), 215–222 (2008)
4. Castro, A., Bóveda, C., Rey, A., Arcay, B.: An analysis of different clustering algorithms for ROI detection in high resolutions CT lung images. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) *ICCVG 2010. LNCS*, vol. 6374, pp. 241–248. Springer, Heidelberg (2010)
5. Castro, A., Arcay, B.: Comparison of various fuzzy clustering algorithms in the detection of ROI in lung CT and a modified kernelized-spatial fuzzy c-means algorithm. In: *Proc. of 10th IEEE Int. Conf. on Inf. Tech. and Appl. in Biom., Corfu., Greece* (2010)
6. Armato III, S.G., McLennan, G., McNitt-Gray, M.F., Lung Image Database Consortium Research Group: Lung Image Database Consortium: Developing a Resource for the Medical Imaging Research Community. *Radiology* 232, 739–748 (2004)
7. Chuang, K., Tzeng, H., Chen, S., Wu, J., Chen, T.: Fuzzy C-means clustering with Spatial information for image segmentation. *Computerized Medical Imaging and Graphics* 30, 9–15 (2006)
8. Zhang, D.-Q., Chen, S.-C.: A novel kernelized fuzzy C-means algorithm with application in medical image segmentation. *Artificial Intelligence in Medicine* 32(1), 37–50 (2004)
9. Zhong, W.D., Wei, X.X., Jian, Y.P.: Fuzzy C-Means clustering algorithm based on kernel method. In: *Proceedings of the Fifth International Conference on Computational Intelligence and Multimedia Applications, ICCIMA 2003* (2003)
10. Haykin, S.: *Neural Networks: A Comprehensive Foundation*, 2nd edn. Prentice-Hall, Englewood Cliffs (1998)
11. Weston, J., Watkins, C.: Multi-class support vector machines. In: Verleysen, M. (ed.) *Proc. ESANN 1999*, Brussels, Belgium (1999)
12. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* 24(7), 881–892 (2002)

Differential Evolution Optimization of 3D Topological Active Volumes

J. Novo, J. Santos, and M.G. Penedo

Computer Science Department, University of A Coruña (Spain)
{jnovo,santos,mgpenedo}@udc.es

Abstract. The Topological Active Volumes is an active model focused on 3D segmentation tasks. It provides information about the surfaces and the inside of the detected objects in the scene. The segmentation process turns into a minimization task of the energy functions which control the model deformation. We used Differential Evolution as an alternative evolutionary method that minimizes the decisions of the designer with respect to other evolutionary methods such as genetic algorithms. Moreover, we hybridized Differential Evolution with a greedy search to integrate the advantages of global and local searches at the same time that the segmentation speed is improved. Moreover, we included in the local search the possibility of topological changes to perform a better adjustment in complex surfaces.

Keywords: Deformable contours, Genetic algorithms, Differential evolution, Image segmentation.

1 Introduction and Previous Work

The active nets model [1] was proposed as a variant of the deformable models [2] that integrates features of region-based and boundary-based segmentation techniques. To this end, active nets distinguish two kinds of nodes: internal nodes, related to the region-based information, and external nodes, related to the boundary-based information. The former model the inner topology of the objects whereas the latter fit the edges of the objects.

The Topological Active Net model and its extension to 3D, that is, the Topological Active Volume (TAV) model [3], were developed as an extension of the original active net model. The model deformation is controlled by energy functions in such a way that the mesh energy has a minimum when the model is over the objects of the scene. The TAV model is an active model focused on segmentation tasks that makes use of a volumetric distribution of the nodes. It integrates information of edges and regions in the adjustment process and allows to obtain topological information inside the objects found. This way, the model, not only detects surfaces as any other active contour model, but also segments the inside of the objects. The model has a dynamic behavior by means of topological changes in its structure, that enables accurate adjustments and the detection of several objects in the scene.

There is very little work in the optimization of active models with genetic algorithms (GA), mainly in edge or surface extraction [4,5] in 2D tasks. For instance, in [4] the author developed the “genetic snakes”, this is, snakes that minimize their energy by means of genetic algorithms. In [6] the authors proved the superiority of a global search method by means of a GA in the optimization of the Topological Active Nets model in 2D images. The results showed that the GA is less sensitive to noise than the usual greedy optimizations and does not depend on the parameter set or the mesh size.

Regarding 3D images, the author in [7] used 3D “active cubes” to segment medical images, where the automatic net division was a key issue. Since the greedy energy-minimization algorithm proposed was sensitive to noise, an improved greedy algorithm inspired by a simulated annealing procedure was also incorporated. To our knowledge, there are not works using evolutionary algorithms for the optimization of 3D deformable models, except our previous work [8], where we extended the GA with new defined operators for the segmentation process using TAV structures. The genetic approach overcame some drawbacks, basically in images with different types of noise, with regard to the work proposed in [3].

In this paper, we used Differential Evolution (DE) [9][10] as an alternative evolutionary method. Moreover, we hybridized DE with a greedy method, so we can join the advantages of the global and local search methods. This paper is organized as follows: Section 2 introduces the basis of the TAV model. Section 3 briefly explains the DE used in the model optimization. In Section 4 representative examples are included to show the capabilities of the different approaches. Finally, Section 5 expounds the conclusions.

2 Brief Description of Topological Active Nets

A Topological Active Volume (TAV) is a discrete implementation of an elastic n -dimensional mesh with interrelated nodes [3]. The model has two kinds of nodes, internal and external, which represents different object features: the external nodes fit the edges whereas the internal nodes model the internal topology.

As in other deformable models, the state of the model is governed by an energy function, composed of internal and external energy terms. The internal energy controls the shape and the structure of the net whereas the external one represents the external forces which govern the adjustment process. The energies are composed of several terms and in all the cases the aim is their minimization.

Internal energy terms. The internal energy depends on first and second order derivatives which control the contraction and bending of the mesh, respectively:

$$E_{int}(v(r, s, t)) = \alpha(|v_r(r, s, t)|^2 + |v_s(r, s, t)|^2 + |v_t(r, s, t)|^2) + \beta(|v_{rr}(r, s, t)|^2 + |v_{ss}(r, s, t)|^2 + |v_{tt}(r, s, t)|^2) + 2\gamma(|v_{rs}(r, s, t)|^2 + |v_{rt}(r, s, t)|^2 + |v_{st}(r, s, t)|^2) \quad (1)$$

where the subscripts represents partial derivatives and α , β and γ are coefficients controlling the first and second order smoothness of the net.

External energy terms. The external energy represents the features of the scene that guide the adjustment process:

$$E_{ext}(v(r, s, t)) = \omega f[I(v(r, s, t))] + \frac{\rho}{\aleph(r, s, t)} \sum_{n \in \aleph(r, s, t)} \frac{1}{\|v(r, s, t) - v(n)\|} f[I(v(n))] \quad (2)$$

where ω and ρ are weights, $I(v(r, s, t))$ is the intensity value of the original image in the position $v(r, s, t)$, $\aleph(r, s, t)$ is the neighborhood of the node (r, s, t) and f is a function of the image intensity, which is different for both types of nodes. If the objects to detect are bright and the background is dark, the energy of an internal node will be minimum when it is on a position with a high grey level. Also, the energy of an external node will be minimum when it is on a discontinuity and on a dark point outside the object. So, the function f is defined as:

$$f[I(v(r, s, t))] = \begin{cases} IO_i(v(r, s, t)) + \tau IOD_i(v(r, s, t)) & \text{internal nodes} \\ IO_e(v(r, s, t)) + \tau IOD_e(v(r, s, t)) \\ + \xi(G_{max} - G(v(r, s, t))) & \\ + \delta GD(v(r, s, t)) & \text{external nodes} \end{cases} \quad (3)$$

where τ , ξ and δ are weights, G_{max} and $G(v(r, s, t))$ are the maximum gradient of the image and the gradient of the input image in node position $v(r, s, t)$, IO is a term we called “In-Out” and IOD a term called “distance In-Out”, and $GD(v(r, s, t))$ is a gradient distance term. The IO terms minimize the energy of those individuals with the external nodes in background intensity values and the internal nodes in object intensity values meanwhile the terms IOD act as a gradient: for the internal nodes (IOD_i) its value minimizes towards brighter values of the image, whereas for the external nodes its value (IOD_e) is minimized towards low values (the background).

The optimizations with a greedy algorithm [3] and with a genetic algorithm [8] consider a global energy as the sum of the different terms, weighted with the exposed parameters. The adjustment process consists in minimizing these energy functions. In the case of the greedy algorithm, the mesh is placed over the whole image and, in each step, the energy of each node is computed in its current position and in its nearest neighborhood. The position with the lowest energy value is selected as the new position of the node. The algorithm stops when there is no node in the mesh that can move to a position with lower energy.

3 Differential Evolution

Differential Evolution (DE) [9][10] is a population-based search method. DE creates new candidate solutions by combining existing ones according to a simple formulae of vector crossover and mutation, and then keeping whichever candidate solution has the best score or fitness on the optimization problem at hand. The central idea of the algorithm is the use of difference vectors for generating perturbations in a population of vectors. This algorithm is specially suited for optimization problems where possible solutions are defined by a real-valued vector. The basic DE algorithm is summarized in the pseudo-code of Figure 1.

One of the reasons why Differential Evolution is an interesting method in many optimization or search problems is the reduced number of parameters that

1. Initialize all individuals x with random positions in the space.
2. Until a termination criterion is met, repeat the following:
 - For each individual x in the population do:
 - 2.1 Pick three random individuals x_1, x_2, x_3 from the population they must be distinct from each other and from individual x .
 - 2.2 Pick a random index $R \in \{1, \dots, n\}$, where the highest possible value n is the dimensionality of the problem to be optimized.
 - 2.3 Compute the individual's potentially new position $y = [y_1, \dots, y_n]$ by iterating over each $i \in \{1, \dots, n\}$ as follows:
 - 2.3.1 Pick $r_i \in U(0, 1)$ uniformly from the open range $(0, 1)$.
 - 2.3.2 If $(i = R)$ or $(r_i < CR)$ let $y_i = x_1 + F(x_2 - x_3)$, otherwise let $y_i = x_i$.
 - 2.4 If $(f(y) < f(x))$ then replace the individual x in the population with the improved candidate solution, that is, set $x = y$ in the population.
3. Pick the agent from the population that has the lowest fitness and return it as the best found candidate solution.

Fig. 1. Differential Evolution Algorithm

are needed to define its implementation. The parameters are F or differential weight and CR or crossover probability. The weight factor F (usually in $[0, 2]$) is applied over the vector resulting from the difference between pairs of vectors (x_2 and x_3). CR is the probability of crossing over a given vector (individual) of the population (x) and the candidate vector y . Finally, the index R guarantees that at least one of the parameters (genes) will be changed in such generation of the candidate solution.

As Feoktistov [11] indicates, the fundamental idea of the algorithm is to adapt the step length ($F(x_2 - x_3)$) intrinsically along the evolutionary process. At the beginning of generations the step length is large, because individuals are far away from each other. As the evolution goes on, the population converges and the step length becomes smaller and smaller.

In our application each individual encodes a TAV. The genotypes code the Cartesian coordinates of the TAV nodes. If a component of a mutant vector (candidate solution) goes off its limits, then the component is set to the bound limit. In this application it means that, in order to avoid crossings in the net structure, each node coordinate cannot overcome the limits established by its neighbors.

Moreover, the usual implementation of DE chooses the base vector x_1 randomly or as the individual with the best fitness found up to the moment (x_{best}). To avoid the high selective pressure of the latter, the usual strategy is to interchange the two possibilities across generations. Instead of this, we used a tournament to pick the vector x_1 , which allows us to easily establish the selective pressure by means of the tournament size.

4 Results

We selected representative artificial and real CT images to show the capabilities and advantages of the DE approach and its hybridization with a greedy local search. All the processes used a population of 1000 individuals. The tournament size to select the base individual x_1 in the DE runs was 3% of the population.

We used a fixed value for the CR parameter (1.0) whereas we used a maximum value of 0.6 for the F parameter. In the different applications of the equation which determines a candidate solution (step 2.3.2 in Figure 1), we used a random value for F between 0.2 and such maximum value (for each node), parameters that were experimentally tuned to provide the best results in most of the images. This allows that each node can move its position in a different intensity, although in the direction imposed by the difference vector ($x_2 - x_3$), which facilitates that each node can independently fall in its best location, as the object boundaries in the case of the external nodes. This strategy provided us with the best results in all the images.

First, we performed a comparison with a GA. Figure 2 compares the evolution of the energy (fitness) of the best individual and the average energy of the population over the generations, using DE, a GA with only one evolutionary phase and a GA with two phases. The two phases are: a first one, whose aim is to produce a population of individuals that cover the object in the image, and a second one, with a different set of energy parameters, to refine the adjustment, as detailed in 3. So, the first phase provides a rough boundary detection meanwhile the second phase provides a better boundary segmentation and distribution of nodes. In this last case, the graph shows only the evolution of the second phase, from generation 200, as the energy of the first phase is not comparable because of the different energy parameters. Moreover, these fitness evolutions were the result of an average of 20 different evolutionary processes with different initial populations. This comparison was made testing the different approaches in the object of Figure 3, using the same energy parameter set in all cases.

As it can be seen, the convergence of the GA process with one phase (dashed lines) is the worst. This is because this strategy requires large initial individuals that cover the object to segment, and this large nets have to be progressively approxi-

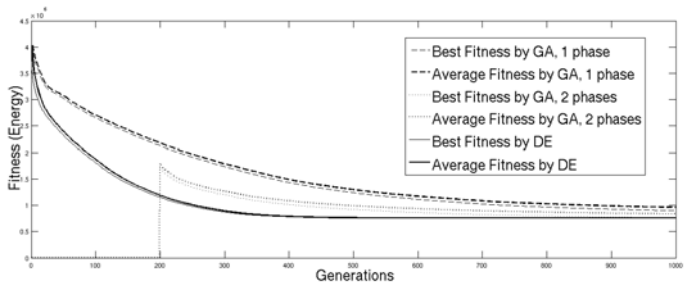


Fig. 2. Best individual fitness (energy) and average fitness of the population with the different evolutionary processes. The curves are an average of 20 different runs with different initial populations.

imated to the boundary of the object. This is solved with the definition of the two different evolutionary phases in the GA process with different tasks (dotted

lines) but these phases require different tuned energy parameters. However, with the DE process, with only one phase and initial large individuals we obtain a significant faster convergence (solid lines).

In Figure 3 the best individual at different generations is shown with all the mentioned processes. We can see the best individual in intermediate generations (50 and 200, 1st and 2nd columns) and the final result (3rd column) in generations 500 (DE) and 1000 (GA versions). The small inset shows a 2D slice of the 3D object. The GA with only one phase and large initial individuals (2nd row) moves slowly the external nodes to the object boundary (basically through mutations in individual nodes), situation that is overcome with the GA with 2 phases (1st row). In this case, the first phase is focused in surrounding the contour of the object so we obtain the external nodes well placed in less generations. However, the DE approach, with only one phase and large initial individuals, thanks to the way that it produces new individuals (a simultaneous perturbation in the location of all the nodes) can quickly obtain a population surrounding the object, producing a faster correct segmentation with a correct distribution of nodes.

4.1 Hybridization of Differential Evolution and Greedy Search

We combined DE with the greedy local search with two aims: to integrate the advantages of global and local searches, and to obtain faster segmentations. A number of greedy steps was applied in all the genotypes of the population used by the DE process. A greedy step implies the application of the greedy movements in all the nodes of the codified TAV. The number of steps was a small number (randomly between 0 and 4) to minimize the falling in local minima. Moreover, the greedy algorithm was applied only in particular generations of the DE process.

Figure 4, upper graph, shows different evolutions of the best individual over the generations using different configurations of the hybrid approach and compared with the DE and the greedy approach. In this case, the greedy steps were applied to the individuals each 10 generations. The graphs of fitness evolution are an average of 20 evolutionary runs of the corresponding algorithm with

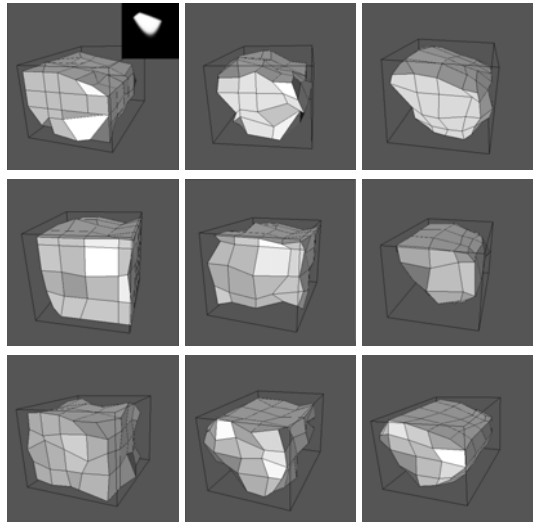


Fig. 3. Best individual across generations in different evolutionary processes. Inset: slice of the original image. 1st row, classic GA process, two evolutionary phases. 2nd row, classic GA process, one evolutionary phase. 3rd row, differential evolution process.

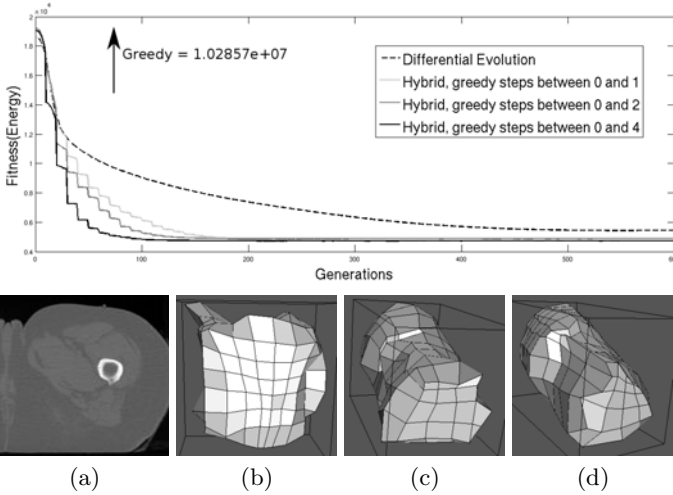


Fig. 4. Upper graph, best individual evolution comparing the DE approach with different hybrid approaches and the greedy method. Bottom part, (a) CT slice of the object, (b) final result with the greedy method, (c) final result with DE, (d) final result with hybridized DE, greedy steps between 0 and 4.

different initial populations. As the graphic shows, the more greedy steps are used, the faster the energy minimization is, but the higher is the predominance of the greedy minimization with respect to the DE minimization. So, we can use a hybrid combination that uses a relative small number of greedy steps to speed up the process without penalizing the robustness of the DE methodology.

The example corresponds to a 3D image of a humerus composed by CT slices, where the nodes have to overcome the flesh surrounding the bone. The greedy approach provided a poor segmentation with all nodes stuck in the flesh contour (Figure 4(b)). Meanwhile, the hybrid approach (with greedy steps between 0 and 4 in this case) was able to overcome these difficulties (Figure 4(d)) obtaining even a better segmentation than a single DE (Figure 4(c)). The greedy method also helps to obtain a more homogeneous internodal distribution and even to find a better adjustment with the external nodes with respect to a single DE method. This can be seen in Figure 4, upper part, where the hybrid combinations reached a lower energy value.

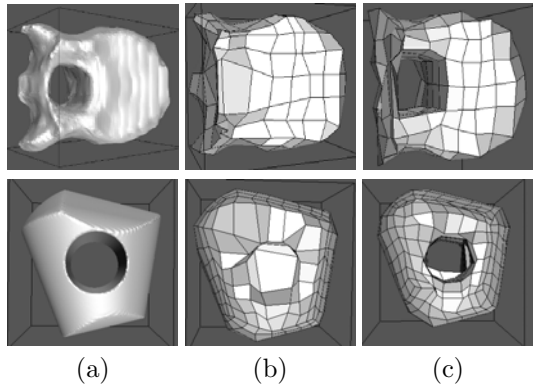


Fig. 5. Best final results obtained using the hybridized DE. (a) Original image. (b) Final results without topological changes. (c) Final results with topological changes.

Regarding computing times, in an Intel Core 2 at 2.83 GHz, the evolution of the DE approach across the 600 generations (1000 individuals used in this example) required an average of 95 minutes, whereas the hybrid combination (steps between 0 and 4) required an average time of 8 minutes to obtain the same fitness as the best value of the DE alternative (in generation 50).

Finally, we included the possibility of topological changes provided by the greedy method, as proposed and detailed in [3]. Thus, the 3D mesh presents the possibility of a better adjustment to complex objects with holes or concavities. Figure 5 shows different representative segmentation examples with objects that require topological changes. In this case we used the depicted hybrid method, applying again a random greedy number of steps between 0 and 4 to all the individuals each 10 generations. In both cases, we present the final result with the hybrid method, including topological changes (Figure 5(c)) or not (Figure 5(b)). As it can be seen, the region with a hole can be detected, but with topological changes it can be also segmented.

5 Conclusions

In this paper, it was proposed a new evolutionary methodology for the optimization of the TAV model. The implemented DE approach introduced some important advantages with respect to the classic GA algorithm. In particular, the previous set of genetic operators was substituted by a single one, which implied more simplicity in order to minimize the decisions that have to be made by the designer. The alternative GA method required different phases that were integrated by a single one using DE. The proposed method also provided a faster convergence and better results, as it was shown in the graphics. Moreover, it was developed a hybrid combination of the DE with the greedy local search, integrating the advantages of both strategies: the global search overcame the possible presence of noise in the image whereas the greedy search helped to speed up the segmentation, which also introduced the possibility of topological changes to perform better adjustments and segmentations in complex surfaces.

Acknowledgments. This paper was funded by the Ministry of Science and Innovation of Spain (project TIN2007-64330) and by the Consellería de Industria, Xunta de Galicia (Grant contracts 10/CSA918054PR and 10TIC009CT).

References

1. Tsumiyama, K.S.Y., Yamamoto, K.: Active net: Active net model for region extraction. *IPSI SIG notes* 89(96), 1–8 (1989)
2. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision* 1(2), 321–323 (1988)
3. Barreira, N., Penedo, M.G.: Topological Active Volumes. *EURASIP Journal on Applied Signal Processing* 13(1), 1937–1947 (2005)

4. Ballerini, L.: Medical image segmentation using genetic snakes. In: Proceedings of SPIE: Application and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation II, vol. 3812, pp. 13–23 (1999)
5. Séguier, R., Cladel, N.: Genetic snakes: Application on lipreading. In: International Conference on Artificial Neural Networks and Genetic Algorithms (2003)
6. Ibáñez, O., Barreira, N., Santos, J., Penedo, M.G.: Genetic approaches for topological active nets optimization. *Pattern Recognition* 42, 907–917 (2009)
7. Bro-Nielsen, M.: Active nets and cubes. Technical Report 13, IMM, Technical University of Denmark (1994)
8. Novo, J., Barreira, N., Santos, J., Penedo, M.G.: Topological active volumes optimization with genetic approaches. In: XII Conference of the Spanish Association for the Artificial Intelligence, vol. 2, pp. 41–50 (2007)
9. Price, K.V., Storn, R.M.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11(4), 341–359 (1997)
10. Price, K.V., Storn, R.M., Lampinen, J.A.: *Differential Evolution. A Practical Approach to Global Optimization*. Natural Computing Series. Springer, Heidelberg (2005)
11. Feoktistov, V.: *Differential Evolution: In Search of Solutions*. Springer, NY (2006)

Genetic Algorithms Applied to the Design of 3D Photonic Crystals

Agustín Morgado-León¹, Alejandro Escuin¹, Elisa Guerrero¹, Andrés Yáñez¹,
Pedro L. Galindo¹, and Lorenzo Sanchis²

¹ Universidad de Cádiz, Departamento de Lenguajes y Sistemas Informáticos,
Grupo de “Sistemas Inteligentes de Computación”,
C.A.S.E.M. 11510 – Puerto Real (Cádiz), Spain

² Universidad de Valencia, UMDO – Instituto de Ciencias de los Materiales,
P.O. Box 22085, 46071 – Valencia, Spain

{agustin.morgado, alejandro.escuin, elisa.guerrero, andres.yanez,
pedro.galindo}@uca.es, lorenzo.sanchis@uv.es

Abstract. We aim at determining the optimal configuration of photonic crystal structures capable of carrying out a certain optical task. An exhaustive search would require a high computational cost, in this work we show how genetic algorithms can be applied to reliably find an optimal topology of three-dimensional photonic crystals. The fitness, representing the performance of each potential configuration, is calculated by means of finite element analysis. Different experiments are presented in order to illustrate the potential of this 3D design approach.

1 Introduction

Photonic Crystals (PCs) [8], [14] are periodic dielectric structures that have a band gap that forbids the propagation of a certain frequency range of light. In much the same way that the atomic lattice of a semiconductor establishes an electronic bandgap between conduction and valence bands, a periodic distribution of materials of disparate optical properties can create a photonic bandgap [12].

The distribution of those scattering elements allows for the design of PCs with different photonic band gaps. Dielectric structures act as scattering elements which prevent light propagation in a specific frequencies band, what marks its importance. To prevent the propagation of light, the periodicity of dielectric structures has to be, at least, a half of the wavelength of light. This property enables to control light more effortlessly, producing effects that are impossible to achieve with conventional optics, as for instance waveguides that permits 90 degree bends with a 100% rate of transmission.

By varying certain parameters of the photonic crystals, such as its geometry or the relative sizes of scattering elements, the reflective properties of PCs can be altered. The idea of controlling light by means of photonic crystals has led to many proposals for the creation of novel devices [3], [7], [9], including different types of focusing elements [1], [5].

There exist three types of photonic crystals depending on whether the dielectric spatial distribution is in one (one-dimensional photonic crystals), two (two-dimensional photonic crystals) or three (three-dimensional photonic crystals) directions.

A one-dimensional photonic crystal is made of slabs with alternating dielectric constants, as shown in Fig. 1 (1D). It is the simplest possible structure and its distribution has to be perpendicular to the direction of propagation of light. This type of photonic crystal can act, for example, as a Bragg mirror [7].



Fig. 1. Examples of one- (1D), two- (2D), and three-dimensional (3D) photonic crystals. The main feature of a photonic crystal is the periodicity of dielectric materials along different axes.

When the dielectric function varies along two of its axes and homogeneously along the third direction, the result is a two-dimensional photonic crystal. The most common distribution of this type of material consists of a square lattice of infinite dielectric columns on a XY plane, as shown in Fig. 1 (2D). Unlike the first, two-dimensional photonic crystals can prevent light from propagating in any direction within the plane. Among its usual applications excels their use in photonic-crystal fibers.

The last type of photonic structure is the three-dimensional photonic crystal [13]. It is usually presented as spheres (Fig. 1 (3D)) separated periodically along three directions, which can also be distributed by a woodpile. Practical realizations of these devices should be fast, simple and accurate as well as meet certain requirements such as low price and good performance. Thus previous computer-based simulations become crucial in order to measure the suitability of 3D PCs' design before fabrication.

We consider how to determine the optimal configuration of 3D PC structures capable of carrying out a specific optical task. From an initial configuration formed by a certain number of scattering elements, the goal is to find the optimal number and position of these scattering elements which provides maximum performance.

Since direct solvers involve a very significant time-consuming direct search in a large parameter space, several inverse design techniques have been devised, among which Genetic Algorithms (GAs) have been successfully applied to 2D PCs [4].

In this work we extend this 2D approach to the design of 3D PC structures, in particular the proposed methodology is illustrated for 3D PC lenses whose specific task consists in focusing light at a prefixed spot, named focal point.

For 3D PC structures the determination of the performance involves more complex and time-consuming calculations of the electric and magnetic fields than 2D PCs. There exist different numerical methods to calculate the performance of PC structures such as MST (Multiple Scattering Theory) [6], [11], FDTD (Finite Difference Time Domain) [5], [10], [12], [15] or FEM (Finite Elements Method) [2], which has been chosen in this work to calculate electromagnetic fields.

The remainder of the paper is organized as follows. Section 2 describes the particular GAs implementation, showing the encoding of the 3D PC lenses and the different genetic operators applied in this work. Section 3 shows two different simulations in order to illustrate the application of the whole methodology. Finally, conclusions and future work are presented in section 4.

2 Genetic Algorithms

GAs are meta-heuristic methods based on biological principles to find the optimal solution. This method starts with an initial set of random solutions called population and applies genetic operators such as mutation and crossover to evolve these potential solutions to find the best one. To evaluate each individual a fitness function is used.

The fitness function of a PC lens is defined as the capacity to guide the light bundle to a set point (focus). In this work is calculated by means of FEM (Fig. 2). This simulation method divides the problem into smaller parts, known as *finite elements*. From the finite elements a system of linear equations is solved and, finally, smoothing conditions and partial derivatives are applied to obtain the final solution. This solution is the value of the electric and the magnetic fields of a complete photonic crystal structure.

Consequently, we define the problem as the intensity of the magnetic field at a specific focal point must be maximized, but the optimal arrangement of scattering elements which can do this, is not known. The goal is to find the optimal topology that will produce that specific magnetic field objective.

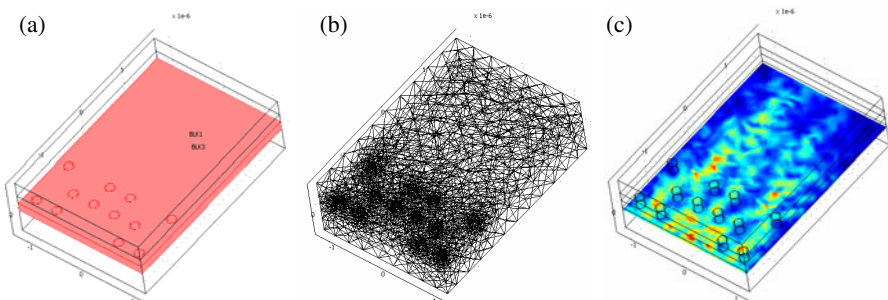


Fig. 2. (a) 3D PC Model formed by 12 scattering elements. (b) Finite Element Mesh. (c) Magnetic field norm determined by FEM, representing the fitness function of the problem.

2.1 Encoding

The photonic crystal model considered is a slab with square lattice of cylindrical holes which act as scattering elements. The tunable parameter in the problem is defined as the presence or absence of each cylindrical hole.

Fig. 3a shows the XY plane of a PC model. The scattering elements are distributed symmetrically (Fig. 3b) and perpendicularly to the direction of wave propagation. For this reason, the search space is restricted to symmetrical configurations, this effectively reduces the problem size by 50%, i.e. the distribution of half of the cylinders is considered and the search space is composed of all the lenses formed by combinations of the $N/2$ scattering elements. Then the search space size is reduced from 2^N to $2^{N/2}$ (Fig. 3c).

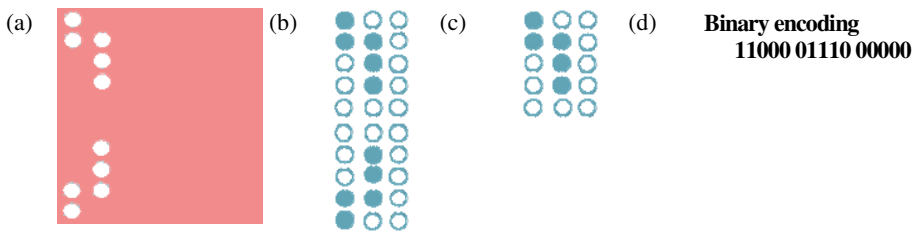


Fig. 3. Encoding example of a photonic crystal evolved from an initial configuration of 10x3 scattering elements. (a) PC configuration showing the position of the 10 scattering elements which form the evolved model. (b) Distribution of these scattering elements, showing the original position of each scattering element, in blue the presence of a cylinder and in white the absence of a scattering element. (c) Arrangement of half of the cylinders considered in the optimization problem due to the symmetrical property of these structures. (d) Binary string that encodes this PC model.

A binary string is used to encode the photonic crystal according to the symmetrical distribution of its scattering elements (Fig. 3d). Digits 1s of this string represent the existence of cylindrical air holes and digits 0s represent the absence of these scattering elements.

2.2 Selection, Recombination and Mutation

The lenses will be selected by applying the *roulette method*, which is based on assigning a probability value to every candidate solution. In our case, this selection will depend of the accumulated relative intensity of the magnetic field norm. Two individuals are chosen randomly based on these probabilities to produce an offspring. The PCs with higher fitness will have more probability to be selected than PCs with lower fitness.

The offspring, named *crossing solution*, is generated by a recombination operator applying a random binary string, where digits 1s indicate that genes from first parent must be taken and digits 0s that genes from second parent must be used (Fig. 4a).

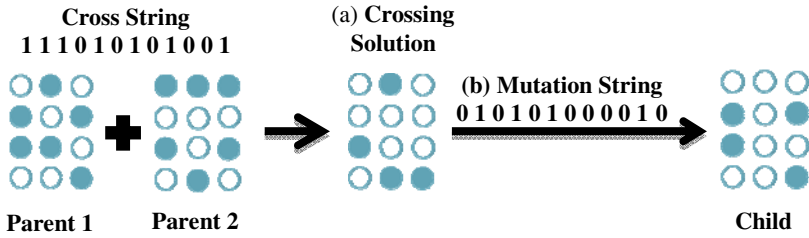


Fig. 4. Application of recombination and mutation operators. (a) The first step is a genetic recombination of two selected parents to obtain a crossing solution, then (b) this solution will mutate to a final child.

In order to guarantee diversity and to avoid reaching a local maximum, a mutation operator is used. Mutation prevents inbreeding and it is based on a probability, which depends on the nature of the problem. If the problem is discrete, a high mutation probability will help us to find the optimal solution, but if the problem is continuous, as in our case, we will need a low mutation probability because small changes in the topology of cylinders might yield better performance.

Based on the mutation probability a random binary string is generated (Fig. 4b). Digits 1s represent the genes that will mutate and digits 0s represent the genes that will not change. All individuals in the subset may mutate or not, and the output of this stage is an individual child.

As many children as the size of the subset must be generated, their performance is calculated and then the best individuals from the parents and the children will be selected to form the new generation of candidate solutions.

The iterative execution of these genetic operators leads the PC's topologies increasingly towards the optimum. This process is continued until there is no improvement of the best fitness value after a certain number of iterations.

3 Experimental Results

The first experiment is the design of a 3D photonic crystal consisting of three layers, which dimensions are $x = 3.4E-6m$, $y = 2.3808E-6m$ and $z = 1.0E-6m$. The dielectric component is a Gallium Arsenide (GaAs) slab surrounded by air.

The thickness of GaAs slab is $1.5E-7m$ with a 6×3 square lattice of cylindrical air holes with a lattice constant of $3.968E-7m$. The radius of these cavities is $8.2E-8m$. The source is a sinusoidal plane wave of 327.64 THz, which is propagated along the x -axis. The goal of the optimization process involves maximizing intensity at prefixed focal point, $(x_f, y_f, z_f) = (0, 0, 0)$.

Due to the symmetry of the structure, the search space size is formed by $2^{N/2} = 2^9$ possible PCs configurations. The population of GAs is composed of 25 individuals and each individual takes around ten minutes to be calculated in an Intel Xenon with 8 cores and 16GB RAM. The crossing probability is 50% and the mutation probability is 10%. This process continues until there is no improvement of the best fitness value after 4 iterations or when a maximum number of generations, set to 100, is reached.

Fig. 5a shows magnetic field norm intensity profiles along the x-axis from the best individuals of certain generations and Fig. 5b shows the position in the model of this profile. In each generation intensity at the focal point (x=0) is increasing, and the process converges after fourth generation. The best individual of the last generation is the optimal solution.

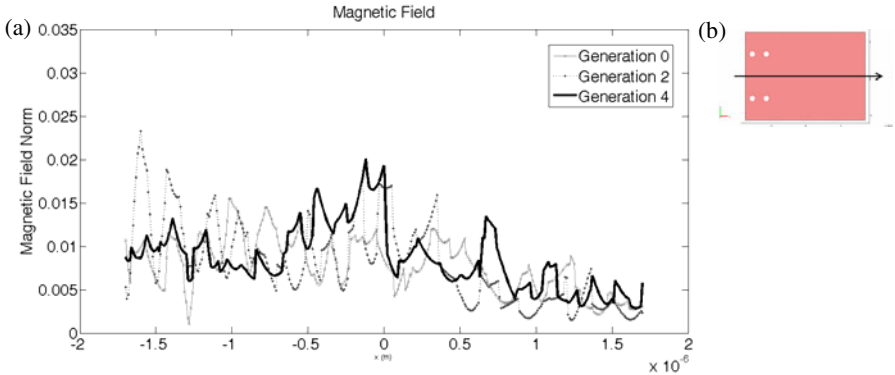


Fig. 5. (a) Magnetic field norm profiles along x-axis of the best individuals in generations 0, 2 and 4 when considering an initial lens configuration of 6x3 cylindrical holes. (b) Black line over the PC model showing the line profile of the magnetic field norm along the x-axis that has been taken to compare the performance of the best PC lens after three different generations.

Fig 6 represents the optimal solution, which is formed by 4 cylindrical holes, and the corresponding magnetic field norm. The intensity value achieved at the focal point is 0.0193 A/m.

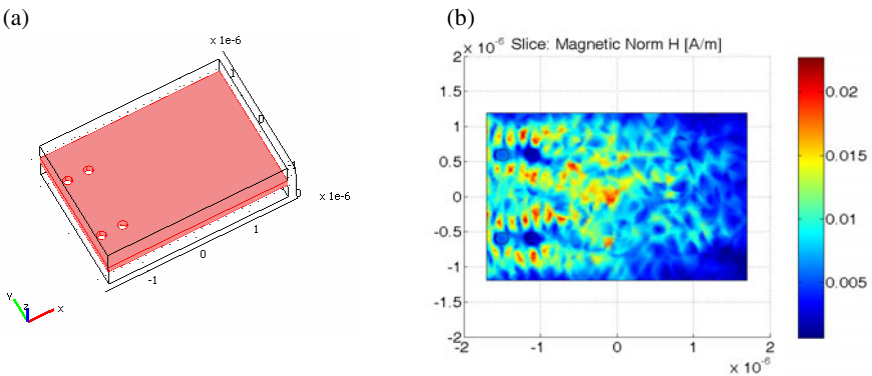


Fig. 6. (a) Optimal solution formed by 2 pairs of cylinders (the non-optimized arrangement is formed by 6x3 cylindrical holes). (b) 2D Magnetic field norm

Second experiment involves the design of a lens formed by 10x3 cylindrical holes in the slab of GaAs. The dimensions of this model are $x=3.4E-6$ m, $y= 3.9680E-6$ m and $z=1.0E-6$ m. Slab thickness, cylinders radius, lattice spacing, source, focal distance and probabilities of crossing and mutation are the same as previous experiment.

Fig. 7a shows the evolution on performance for the best individuals of generations 0, 4 and 9 and Fig. 7b shows a schema of one profile. After 9 generations the process converges towards an intensity value of 0.0306 A/m. Fig. 8 shows the optimal arrangement of scattering elements, formed by 10 cylindrical holes over a map of the intensity of the magnetic field norm.

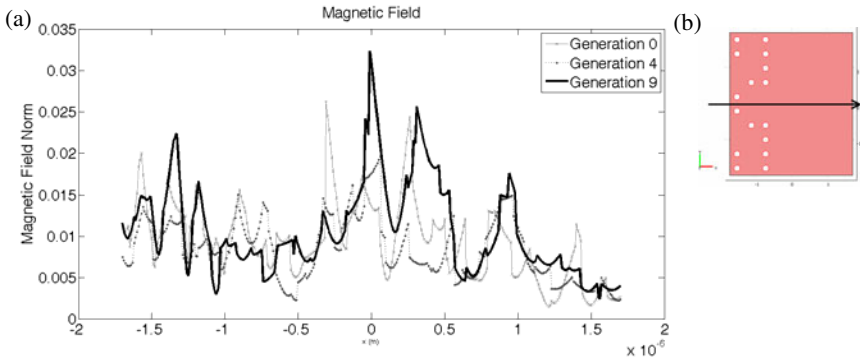


Fig. 7. (a) Magnetic field norm profiles along x-axis of the best individuals of generations 0, 4 and 9 when considering an initial lens configuration of 10x3 cylindrical holes. (b) Black line over the PC model showing the line profile along the x-axis that has been taken to compare the performance of the best PC lens after three different generations.

Second model takes more time in the convergence process since a more complex model has been considered in this experiment, therefore the search space size and the fitness computation time have been increased, but better results have been obtained in terms of performance.

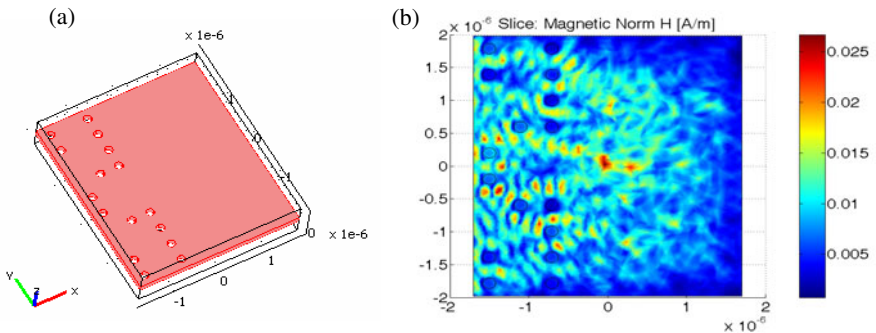


Fig. 8. (a) Optimal PC lens formed by 7 pairs of scattering elements. (b) Magnetic field intensity map. Maximum intensity is achieved at focus with a value of 0.0306 A/m.

4 Conclusions and Future Works

We have presented a methodology based on inverse design to model three-dimensional photonic crystal structures. GAs have been applied in order to find the

optimal configuration and performance is evaluated by FEM. This methodology has been implemented for the design of two three-dimensional PC lenses with different complexity. More complex lenses are more efficient but the computation time to converge is higher.

Now we are addressing the goal of adding other tunable parameters to the optimization process, namely, taking into account different values of radius and relative position of the cylindrical holes.

Future work will be addressed to parallelize this method and execute it in the cluster of the University of Cádiz to reduce computing time. Other numerical methods to calculate electric and magnetic fields are also being studied.

References

1. Boutayeb, H., Tarot, A., Mahdjoubi, K.: Focusing Characteristics of a Metallic Cylindrical Electromagnetic Band Gap Structure with Defects. *Progress in Electromagnetic Research* 66, 89–103 (2006)
2. Courant, R.: Variational Methods for the Solution of Problems of Equilibrium and Vibrations. *Bull. Amer. Math. Soc.* 49, 1–23 (1943)
3. Flück, E., Hulst van, N.F., Vos, W.L., et al.: Near-Field Optical Investigation of Three-Dimensional Photonic Crystals. *Physical Review E: Statistical, nonlinear, and soft matter physics* 68, 15601 (2003)
4. Hakansson, A., Cervera, F., Sanchez-Dehesa, J.: Sound Focusing by Flat Acoustic Lenses without Negative Refraction. *Applied Physics Letters* 86, 054102-054102-3 (2005)
5. Haxha, S., AbdelMalek, F.: A Novel Design of Photonic Crystal Lens Based on Negative Refractive Index. *PIERS online* 4, 296–300 (2008)
6. Ishimaru, A. (ed.): *Electromagnetic wave propagation, radiation, and scattering*. Prentice-Hall, Englewood Cliffs (1991)
7. Joannopoulos, J., Meade, R., Winn, J., et al.: *Photonic crystals: Molding the flow of light*. Princeton University Press, Princeton (2008)
8. John, S.: Strong Localization of Photons in Certain Disordered Dielectric Superlattices. *Phys. Rev. Lett.* 58, 2486 (1987)
9. Levi, B.G.: Progress made in Near-Field Imaging with Light from a Sharp Tip. *Phys. Today* 52, 18–20 (1999)
10. Minin, I., Minin, O., Kotlyar, V., et al.: Subwavelength Diffractive Photonic Crystal Lens. *Microwave Conference, China-Japan Joint*, 756-757 (2008)
11. Sanchis, L., Hakansson, A., Lopez-Zanon, D., et al.: Integrated Optical Devices Design by Genetic Algorithm. *Applied Physics Letters* 84, 4460–4462 (2004)
12. Taflove, A., Hagness, S.: *Computational electrodynamics: The finite-difference time-domain method*, 3rd edn. Artech House Publishers, Boston (2005)
13. Yablonovitch, E., Gmitter, T.J., Leung, K.M.: Photonic Band Structure: The Face-Centered-Cubic Case Employing Nonspherical Atoms. *Phys. Rev. Lett.* 67, 2295 (1991)
14. Yablonovitch, E.: Inhibited Spontaneous Emission in Solid-State Physics and Electronics. *Phys. Rev. Lett.* 58, 2059–2062 (1987)
15. Cui, Y., Liu, K., Foland, S., et al.: Electro-Thermally Tunable Silicon Photonic Crystal Lens. In: *IEEE 23rd International Conference on Micro Electro Mechanical Systems (MEMS 2010)*, pp. 188–191 (2010)

Sliding Empirical Mode Decomposition for On-line Analysis of Biomedical Time Series

A. Zeiler¹, R. Faltermeier², A.M. Tomé³, C. Puntonet⁴, A. Brawanski²,
and E.W. Lang¹

- ¹ CIML Group, Biophysics, University of Regensburg, D-93040 Regensburg, Germany
elmar.lang@biologie.uni-regensburg.de
- ² Clinic of Neurosurgery, University Hospital Regensburg, D-93040 Regensburg, Germany
rupert.faltermeier@clinic.uni-regensburg.de
- ³ DETI/IEETA, Universidade de Aveiro, 3810-193 Aveiro, Portugal
ana@ieeta.pt
- ⁴ DATC, EEIS, Universidad de Granada, 38170 Granada, Spain
carlos@atc.ugr.es

Abstract. Biomedical signals are in general non-linear and non-stationary. *Empirical Mode Decomposition* in conjunction with *Hilbert-Huang Transform* provides a fully adaptive and data-driven technique to extract *Intrinsic Mode Functions* (IMFs). The latter represent a complete set of orthogonal basis functions to represent non-linear and non-stationary time series. Large scale biomedical time series necessitate an on-line analysis which is presented in this contribution. It shortly reviews the technique of EMD and related algorithms, discusses the newly proposed *slidingEMD* algorithm and presents some applications to biomedical time series from neuromonitoring.

1 Introduction

Recently an empirical nonlinear analysis tool for complex, non-stationary time series has been pioneered by N. E. Huang et al. [2]. It is commonly referred to as *Empirical Mode Decomposition* (EMD) and if combined with Hilbert spectral analysis it is called *Hilbert - Huang Transform* (HHT). It adaptively and locally decomposes any non-stationary time series in a sum of *Intrinsic Mode Functions* (IMF) which represent zero-mean amplitude and frequency modulated oscillatory components. The EMD represents a fully data-driven, unsupervised signal decomposition technique and does not need any *a priori* defined basis system. The empirical nature of EMD offers the advantage over other empirical signal decomposition techniques like *empirical matrix factorization* (EMF) of not being constrained by conditions which often only apply approximately. Especially with biomedical signals one often has only a rough idea about the underlying modes and mostly their number is unknown. Furthermore, large scale biomedical time series recorded over days necessitate an on-line analysis while EMD can analyze data only globally so far. This contribution will review the technique of empirical mode decomposition and its recent extensions, propose an on-line EMD variant called *slidingEMD* and discuss some biomedical applications related to neuromonitoring.

2 Empirical Mode Decomposition

The EMD method was developed from the assumption that any non-stationary and non-linear time series consists of different simple intrinsic modes of oscillation. The essence of the method is to empirically identify these intrinsic oscillatory modes by their characteristic time scales in the data, and then decompose the data accordingly. Through a process called *sifting*, most of the *riding waves*, i.e. oscillations with no zero crossing between extrema, can be eliminated. The EMD algorithm thus considers signal oscillations at a very local level and separates the data into locally non-overlapping time scale components. It breaks down a signal $x(t)$ into its component IMFs obeying two properties:

1. An IMF has only one extremum between zero crossings, i.e. the number of local minima and maxima differs at most by one.
2. An IMF has a mean value of zero.

Note that the second condition implies that an IMF is stationary which simplifies its analysis. But an IMF may have amplitude modulation and also changing frequency.

The Standard EMD Algorithm. The sifting process can be summarized in the following algorithm. Decompose a data set $x(t)$ into IMFs $x_n(t)$ and a residuum $r(t)$ such that the signal can be represented as

$$x(t) = \sum_n x_n(t) + r(t) \quad (1)$$

Note that a residuum is a non-oscillatory signal which does not fulfill the conditions for an IMF. Sifting then means the following steps:

- Step 0: Initialize: $n := 1, r_0(t) = x(t)$
- Step 1: Extract the n -th IMF as follows:
 - a) Set $h_0(t) := r_{n-1}(t)$ and $k := 1$
 - b) Identify all local maxima and minima of $h_{k-1}(t)$
 - c) Construct, by cubic splines interpolation, for $h_{k-1}(t)$ the envelope $U_{k-1}(t)$ defined by the maxima and the envelope $L_{k-1}(t)$ defined by the minima
 - d) Determine the mean $m_{k-1}(t) = \frac{1}{2}(U_{k-1}(t) - L_{k-1}(t))$ of both envelopes of $h_{k-1}(t)$. This running mean is called the low frequency local trend. The corresponding high-frequency local detail is determined via a process called *sifting*.
 - e) Form the $(k) - th$ component $h_k(t) := h_{k-1}(t) - m_{k-1}(t)$
 - 1) if $h_k(t)$ is not in accord with all IMF criteria as given above, increase $k \rightarrow k + 1$ and repeat the Sifting process starting at step [b]
 - 2) if $h_k(t)$ satisfies the IMF criteria then set $x_n(t) := h_k(t)$ and $r_n(t) := r_{n-1}(t) - x_n(t)$
- Step 3: If $r_n(t)$ represents a residuum, stop the sifting process; if not, increase $n \rightarrow n + 1$ and start at step 1 again.

The sifting process separates the non-stationary time series data into locally non-overlapping intrinsic mode functions (IMFs) which are *locally orthogonal*. Global orthogonality is not guaranteed as neighboring IMFs might have identical frequencies at different time points (typically in < 1% of the cases). After having extracted all IMFs, they can be analyzed further by applying the Hilbert transform or processing them in any other suitable way [3], [4].

3 Sliding EMD

The application of EMD to biomedical time series is limited by the size of the working memory of the computer. Hence in practical applications only relatively short time series can be studied. However, many practical situations like continuous patient neuromonitoring ask for an *on-line* processing of the recorded data. Recently, a blockwise processing, called *on-line EMD*, has been proposed [5]. The method is still in its infancy and needs yet to be developed to a robust and efficient *on-line* technique. Following we will propose an efficient and robust *online* EMD algorithm which we call *slidingEMD*.

3.1 The Principle

In a first step, the recorded time series, encompassing N samples, is split into segments, each encompassing m samples, which can be analyzed with EMD. Simply adding the IMFs extracted from the different segments together would induce boundary artifacts, however. This is illustrated in a simple example in Fig. 1.

Such strong boundary artifacts can be avoided when a sliding window, encompassing m samples, is shifted over the time series with a step size of k samples. Choosing this step size such that $n = \frac{m}{k} \in \mathbb{N}$, neighboring windows can be joined without having discontinuities or gaps at the boundary. With this choice, every sample is represented n - times in overlapping windows corresponding to consecutive shifts of the time series and a mean sample value can easily be estimated. If the conditions $n \in \mathbb{N}$ and $\frac{N}{m} \in \mathbb{N}$ hold with N the number of samples, the total number M of windows, generated through shifting the time series n times, is given by $M = (N - m)/k + 1$. The time series in every window is decomposed by EMD into j IMFs $c_{m_{ij}}(t)$ and a local residuum $r_{m_i}(t)$ according to

$$x_{m_i}(t) = \sum_j c_{m_{ij}}(t) + r_{m_i}(t) \tag{2}$$

whereby the number of sifting steps is kept equal in all segments. Resulting IMFs are collected in a matrix with corresponding sample points forming a column of the matrix with $n = \frac{m}{k}$ entries. Columns corresponding to the beginning or end of the time series are deficient, hence are omitted from further processing. This assures that all columns contain the same amount of information to estimate average IMF amplitudes at every time point in each segment. This finally yields average IMFs according to (n - size of the ensemble)

$$c_j(t_\tau) = \frac{1}{n} \sum_i^{i+n} c_{m_{ij}}(t_\tau) \tag{3}$$

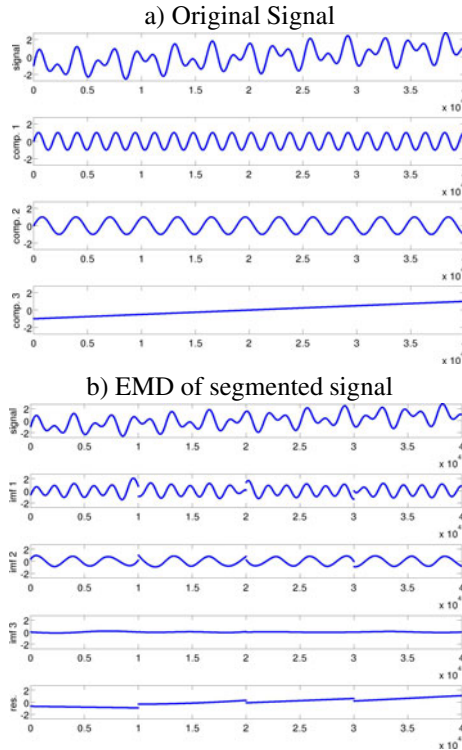


Fig. 1. a) Toy signal (top trace) and underlying signal components: $\sin(7t)$ (second trace), $\sin(4t)$ (third trace) and the trend $0, 1 \cdot t - 1$ (bottom trace). b) EMD decomposition of the toy signal. The time series has been segmented into 4 segments and decomposed with EMD. After joining the resulting IMFs together, boundary effects become clearly visible at $t = 1000$, 2000 and $t = 3000$.

3.2 Properties of *slidingEMD*

Contrary to global EMD, the local residues estimated with *slidingEMD* for every segment may turn into low frequency oscillations when joined together into an average global residuum. By choosing the segment size properly it may be determined which oscillations should appear as distinct IMFs and which should be absorbed as apparent local trends into the respective residues. These apparent local trends which combine to low frequency oscillations in the final average global residue may be down-sampled and subsequently analyzed with *slidingEMD* as well. This process can be repeated until finally a truly monotonous trend remains. Hence, this cascaded application of *slidingEMD* acts as a low frequency filter for long-term oscillations and trends in biomedical time series.

Similar to *Ensemble Empirical Mode Decomposition* (EEMD), also with *slidingEMD* an averaging over differently decomposed data sets is achieved. While due to added

noise, with EEMD a given sample is associated with different amplitudes, with *slidingEMD* the same amplitude is associated with different samples in different shifted segments. This latter behavior alleviates effects related with a non-unique data decomposition via EMD. Furthermore, artifacts resulting from end effects loose their impact via averaging. Finally, segmentation with proper window size and step size does not result in boundary artifacts after combination of the local IMFs. *SlidingEMD* is further similar to *local EMD* in that the stopping criterion needs to be valid only locally, i.e. within the window considered. This substantially reduces *oversifting* and also reduces the number of necessary sifting steps until the stopping criterion is met locally. Finally note that while local IMFs fulfill all defining conditions, this is not necessarily true for the resulting average IMFs though the related deviations should be small always.

3.3 Application of *slidingEMD* to Brain Status Data

In this section the potential of *slidingEMD* to analyze and decompose real brain status data will be explored. The time series concern arterial blood pressure (ABP), intracranial brain pressure (ICP) and partial oxygen pressure (POP) recordings extending over many days and have been sampled with a sampling rate of $\tau^{-1} = 0.2Hz$ or $\tau^{-1} = 1Hz$, respectively. First the dependence of the decomposition onto segment size and step size will be investigated. Next it will be studied how low frequency oscillations can be separated into the residuum via *slidingEMD*. The latter often correspond to non-stationary signal components which are hard to deal with using classical signal processing paradigms. This *detrending* has been discussed in the literature already by [6] and [1] applying EMD. However, if not only a monotonous residuum but also certain low frequency oscillations need to be separated out, a subsequent sophisticated analysis of the estimated IMFs is necessary. To the contrary, *slidingEMD* provides a very simple and efficient way to achieve this goal by simply varying the segment size accordingly.

Optimizing step size and segment size. Analyzing biomedical signals, their component signals are usually not known with sufficient precision. Yet the quality of any decomposition needs to be assessed to identify an optimal step size and window size, respectively. The goal of any decomposition is to extract all stationary IMFs as well as the non-stationary residuum or trend. Following, residua will be estimated with *slidingEMD* varying the step size but keeping the segment size fixed. Next residues corresponding to subsequent step sizes will be subtracted point-wise and the squared differences summed and divided by the number of samples, i.e.

$$Q = \frac{1}{N} \sum_{t=1}^N \left(r_i(t) - r_j(t) \right)^2 \quad (4)$$

Here N designates the number of samples, $r_i(t)$ represents the residue for step size i and $r_j(t)$ the corresponding residue for the subsequent step size j . The segment sizes $m = 2^q$, $q = 8 - 11$ always formed integer multiples of the step sizes. Fig. 2 illustrates the extracted residua obtained with a segment size $m = 2048$ and a step size $\Delta t = 8$.

The estimated mean square difference Q for two corresponding residues i, j of the ABP, ICP and POP time series quickly declines to very small values for $i \leq 128, j \leq 64$

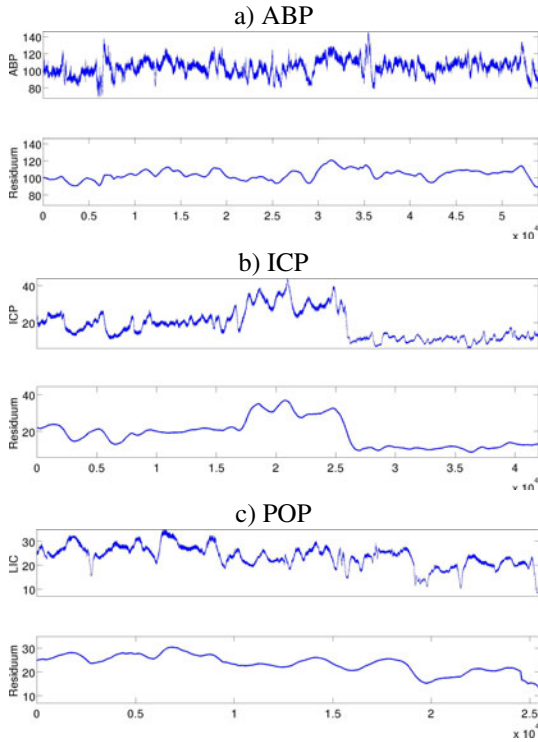


Fig. 2. a) Arterial blood pressure (ABP) time series, b) intracranial brain pressure (ICP) time series, c) Partial oxygen pressure (POP) time series and corresponding residua estimated with *slidingEMD* using a window size $m = 2048$ and a step size $\Delta t = 8$

and all segment sizes investigated. This also means that under all conditions investigated the reconstruction quality quickly saturates as function of the step size. Hence, the estimated residues are very similar under such conditions. The size n of the ensemble thus is the determining quantity concerning the reconstruction quality. An ensemble size $2^4 \leq n \leq 2^6$ provides a good trade-off between reconstruction quality and computational load in good correspondence with results obtained with artificial toy signal time series.

Comparative analysis of brain status data with EMD and *slidingEMD*. The following study compares the residues estimated with *slidingEMD* with the residues plus the sum of low frequency IMFs estimated with EMD. First the mean square difference between the residues estimated with either *slidingEMD* ($r_s(t)$) or EMD ($r_e(t)$) is calculated. Next, the residuum $r_e(t)$ and the lowest frequency IMF is added and the sum is subtracted from the residuum $r_s(t)$. Next, the residuum $r_e(t)$ and the two lowest frequency IMFs are added and the sum is subtracted from the residuum $r_s(t)$. This process is iterated until the difference is negligible.

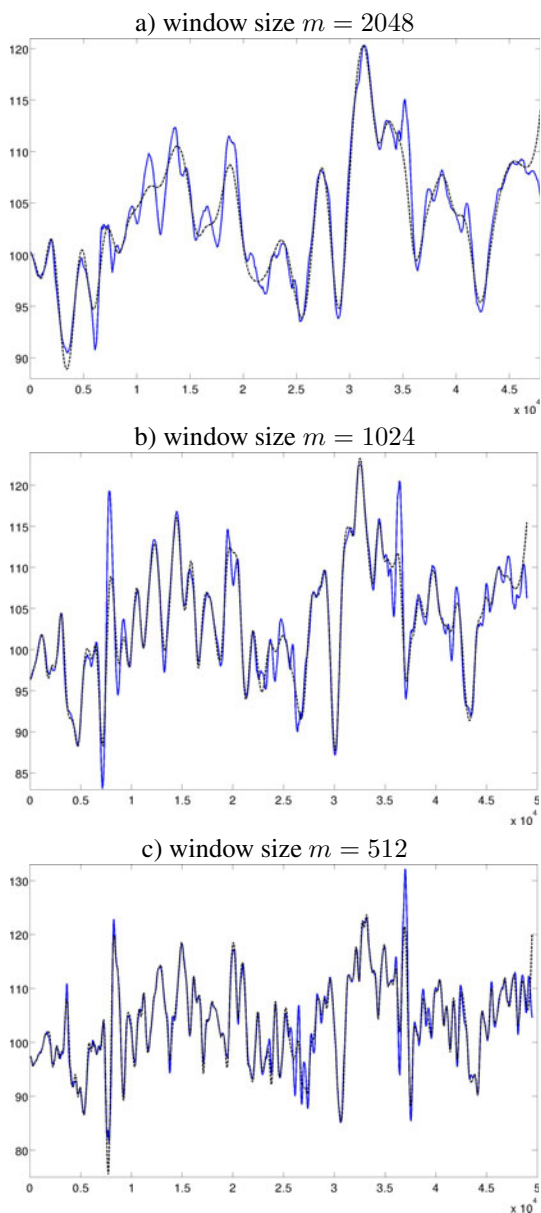


Fig. 3. Residuum $r_s(t)$ (full line) estimated with *slidingEMD* using a window size of a) $m = 2048$, b) $m = 1024$ and c) $m = 512$ samples and the sum of the respective residuum plus low frequency IMFs, i.e. $r_e(t) + IMF_i + \dots + IMF_{14}$ where a) $i = 10$, b) $i = 9$, c) $i = 8$, estimated with EMD (broken line)

$$Q_{n+2} = \frac{1}{N} \sum_{t=1}^N \left(r_s(t) - \left(r_e(t) + \sum_{i=0}^n c_{14-n}(t) \right) \right)^2 \quad (5)$$

The result of this iteration ($n = \{0, 1, \dots, 13\}$) yields minima of $Q_{n+2}(n)$ for $n = 4, 5, 6$ using segment sizes $m = 2^q$, $q = 9 - 11$ and a step size $k = 2$. It becomes obvious that the residuum estimated with *slidingEMD* using a segment size $m = 2048$ and the residuum plus IMFs 10 to 14, all estimated from applying EMD, are very similar (see Fig. 3). Much the same holds true in the following constellations: segment size $m = 1024$ and $r_e(t) + IMF_9 + \dots + IMF_{14}$ and segment size $m = 512$ and $r_e(t) + IMF_8 + \dots + IMF_{14}$.

The results prove that *slidingEMD* yields a decomposition well in accord with standard EMD. However, *slidingEMD* offers the additional advantage of being a true *on-line* algorithm which, furthermore, is based on an ensemble of estimates. It thus provides a robust estimate of underlying intrinsic mode functions. It is also as flexible as standard EMD in detrending applications and allows to extract the stationary part of originally non-stationary biomedical time series data. For practical applications, hence, *slidingEMD* should be preferred.

Acknowledgment

Support by the DAAD and the BFHZ - CCUFB is gratefully acknowledged.

References

1. Flandrin, P., Gonçalves, P., Rilling, G.: Detrending and denoising with empirical mode decomposition. In: EUSIPCO 2004, pp. 1581–1584 (2004)
2. Huang, N.E., Shen, Z., Long, S.R., Wu, M.L., Shih, H.H., Zheng, Q., Yen, N.C., Tung, C.C., Liu, H.H.: The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis. *Proc. Roy. Soc. London A* 454, 903–995 (1998)
3. Quiroga, R.Q., Arnhold, J., Grassberger, P.: Learning driver-response relationships from synchronization patterns. *Phys. Rev. E* 61, 5142 (2000)
4. Quiroga, R.Q., Kraskov, A., Kreuz, T., Grassberger, P.: Performance of different synchronization measures in real data: A case study on electroencephalographic signals. *Phys. Rev. E* 65, 041903 (2002)
5. Rilling, G., Flandrin, P., Goncalès, P.: On empirical mode decomposition and its algorithms. In: *Proc. 6th IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing* (2003)
6. Wu, M.-C., Struzik, Z.R., Watanabe, E., Yamamoto, Y., Hu, C.-K.: Temporal evolution for the phase histogram of ECG during human ventricular fibrillation. In: *AIP Conf. Proc.*, vol. 922, pp. 573–576 (2007)

Suitability of Artificial Neural Networks for Designing LoC Circuits

David Moreno, Sandra Gómez, and Juan Castellanos

Natural Computing Group, Technical University of Madrid,
Campus de Montegancedo, 28660 Boadilla del Monte (Madrid) Spain
d.mnavas@alumnos.upm.es, sgomez@eui.upm.es, jcastellanos@fi.upm.es

Abstract. The simulation of complex LoC (Lab-on-a-Chip) devices is a process that requires solving computationally expensive partial differential equations. An interesting alternative uses artificial neural networks for creating computationally feasible models based on MOR techniques. This paper proposes an approach that uses artificial neural networks for designing LoC components considering the artificial neural network topology as an isomorphism of the LoC device topology. The parameters of the trained neural networks are based on equations for modeling microfluidic circuits, analogous to electronic circuits. The neural networks have been trained to behave like AND, OR, Inverter gates. The parameters of the trained neural networks represent the features of LoC devices that behave as the aforementioned gates. This would mean that LoC devices universally compute.

Keywords: LoC, Lab-on-a-Chip, MOR, Microfluidic devices, Nanofluidic devices, Artificial neural networks.

1 Introduction

Lab-on-a-Chip (LoC) is a field of research and technological development with the goal of constructing highly integrated compact devices for integrating multiple laboratory functions on a single chip with a minimized size. The integration is carried out on monolithic platforms which permit the integration of micro(nano)fluidic functionalities and components necessary to accomplish one or more biochemical or chemical processes. Microfluidic lab-on-a-chip (LoC) systems have been studied for more than a decade and have many applications in biology, medicine, and chemistry [7,8]. LoC devices perform chemical analysis involving sample preparation, mixing, reaction, injection, separation analysis and detection [1]. The most highly integrated Lab-on-a-Chip devices include all processes and devices on a single chip or card so that the introduction of an unprocessed sample leads to the output of an analytical result – an “answer” – from that same chip [4].

Simulating these types of devices and their components requires a great number of parameters and complex partial differential equations. An efficient way of simulating LoC devices involves functional decomposition into a series of interconnected blocks which work to create models for use in the decomposition when

first principle models are not possible. These blocks are the mixer, the injector, and the separator. For the mixer and separator, band shape assumptions are used with analytical techniques for simplifying the partial differential equations into several ordinary differential equations. For the injector, the resulting modeled outputs are described by a finite number of performance functions; hence numerical techniques are used to describe these functions in a finite domain (see [4]).

The structure of this paper is as follows. Section 2 is the background where we present the theoretical basis that supports the approach proposed by other researchers [4] who use artificial neural networks for simulating the behavior of components of a set of microfluidic devices. Section 3 presents our proposal for designing LoC devices through artificial neural networks which map the topology of the LoC. Section 4 presents conclusions and future work.

2 Artificial Neural Networks for Simulating LoC Components

Simulation of complex LoC devices is a very expensive process from the point of view of computational resources. In general, numerical solutions for partial differential equations are necessary to achieve an appropriate simulation. Processing these solutions is costly. As the design of LoC systems requires many repeated simulations, iterative design using numerical simulation is computationally infeasible [1]. The proposal [2] simplifies this process applying MOR (Model Order Reduction) for splitting the spatial dependency of device behavior, extracting the most typical characteristics of the governing equations and, hence, reduces the complexity of the problem. In particular, applying MOR is undertaken to reduce the number of parameters in the simulation. The methodology proposed in [1] uses the results of MOR applied to specific LoC components as input for training an artificial neural network. This trained neural network simulates the behavior and performance of the specified LoC components. Finally, the result obtained is a model for designing LoC components generated through the neural architecture. This approach is tested in [1] modeling an injector component because it defines the shape and quantity of analytes that will be used for separation and analysis. The injector components modeled were cross, double-tee and gated-cross. Modeling an injector is a very difficult task. The approach [1] tries to simplify the mathematical model maintaining accuracy with respect to physical features. The steps for modeling the injector described in [1] are

1. The π - Buckingham theorem is used for reducing the physical parameter space. The reduced parameters are dimensionless.
2. The process is carried out in the dimensionless parameter space to obtain the minimal number of numerical simulations and so reduce cost.
3. An artificial neural network is constructed to analytically describe the parameter space.
 - (a) The selected network topology is feed-forward back-propagation.

- (b) The network learns the functional mapping without knowing the underlying physical basis.
4. The trained network is converted into an explicit algebraic function appropriate for use in any software environment for executing a simulation or synthesis process.

The results obtained in this research indicate that the injector model obtained by neural networks is very close to the result obtained by numerical simulations. This result is fascinating because, as described above, numerical simulations are not only expensive to carry out but time-consuming. This approach demonstrates that artificial neural networks can obtain models very close to the results obtained in numerical simulations at lower resources cost.

3 Artificial Neural Networks for Designing LoC Devices

The main idea involves considering the topology of an artificial neural network as an isomorphism between this network and a microfluidic device (LoC). This isomorphism permits the network to be trained in a specific function and to transform elements such as nodes, edges, weights and activation functions into elements and features of a microfluidic device. This approach is based on the analogy between electronic circuits and microfluidic circuits shown in the next equation (for more details, see [3])

$$\Delta P = L_f \frac{dQ}{dt} + R_f Q + \frac{1}{C_f} \int Q dt \quad (1)$$

Where P is pressure, t is time, Q is the rate of volumetric flow, L is inductance (due to the inertia of the fluid), R is resistance (due to the transversal forces of the channel walls), C is capacity (due to the compressibility of the fluid). In most of the microfluidic circuits, the third member of equation 1 can be omitted.

For a channel with transversal fixed section and circular shape, fluidic resistance is expressed by

$$R_f = \frac{8\mu l}{\pi r_c^4} \quad (2)$$

Where μ is the viscosity dynamic of the fluid, l is the length of the channel and r_c is the radius of the channel.

Fluidic inductance is defined by

$$L_f = \frac{\rho l}{A_c} \quad (3)$$

Where ρ is the density of the fluid, l is the length of the channel and A_c is the transversal area of the channel.

Now, Q can be expressed in the following way

$$Q = G_f \Delta P \quad (4)$$

Where G_f is the fluidic conductance, Q can be expressed as the sum of input rates of volumetric flow

$$Q = \sum_i Q_i \quad (5)$$

Based on the equations above, we can model each component of the neural network with the following features:

- The whole of the nodes are LoC chambers where reactions and processes take place.
- The input nodes receive rates of volumetric flow Q_i as inputs.
- The weights of the edges are R_{fij} . This paper considers the rate of volumetric flow Q as constant, so L_f is not modeled.
- The output nodes return $\Delta P_j = \sum_i Q_i R_{fij}$ as output.

Artificial neural networks have been trained to behave like AND, OR and inverter gates. The possible values and the equivalences between digital inputs and flows are

- 0 is equal to any value lower than a flow unit
- 1 is equal to any value equal or greater than a flow unit.

3.1 AND Gate

The truth table of the AND gate is shown in Table 1

Table 1. Truth Table of the AND Gate

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

A simple artificial neural network that implements this gate has two input nodes and one output node, with two edges, each from input node to output node. The weights of the edges are equal to 0.5 and the threshold for the activation function is 1. With these values, the equation for fluidic resistance is

$$R_f = \frac{8\mu l}{\pi r_c^4} = 0,5 \quad (6)$$

Variables can take different values. This represents the degrees of freedom provided in terms of the possibilities that the lab-on-chip design permits. The dynamic viscosity of fluids can be determined by fixing the radius channel and its length with known resistance

$$\mu = \frac{0,5\pi r_c^4}{8l} \tag{7}$$

Similarly, it is possible to determine the radius channel by fixing the dynamic viscosity, channel length and resistance

$$r_c = \sqrt[4]{\frac{\mu 8l}{0,5\pi}} \tag{8}$$

Finally, it is possible to determine the length of the channel by fixing its radius, the dynamic viscosity and resistance

$$l = \frac{0,5\pi r_c^4}{8\mu} \tag{9}$$

The value for the output node is

$$\Delta P = 0,5(Q_1 + Q_2) \tag{10}$$

If the pressure applied is greater than or equal to the unit of pressure, the flow through the node is the desired one.

3.2 OR Gate

The truth table of the OR gate is shown in Table 2

Table 2. Truth table of the OR gate

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

A simple artificial neural network that implements this gate has two input nodes and one output node, with two edges, each from input node to output node. The weights of the edges are equal to the value 1 and the threshold for the activation function is 1.

With these values, the next equation for the fluidic resistance is

$$R_f = \frac{8\mu l}{\pi r_c^4} = 1 \tag{11}$$

Variables can take different values. This represents the degrees of freedom provided in terms of the possibilities that the lab-on-chip design permits. The dynamic viscosity of fluids can be determined by fixing the radius channel and its length with known resistance

$$\mu = \frac{\pi r_c^4}{8l} \quad (12)$$

Similarly, it is possible to determine the radius channel by fixing the dynamic viscosity, channel length and resistance

$$r_c = \sqrt[4]{\frac{\mu 8l}{\pi}} \quad (13)$$

Finally, it is possible to determine the length of the channel by fixing its radius, the dynamic viscosity and resistance

$$l = \frac{\pi r_c^4}{8\mu} \quad (14)$$

The value for the output node is

$$\Delta P = Q_1 + Q_2 \quad (15)$$

If the pressure applied is greater than or equal to the unit of pressure, flow through the node is the desired flow.

3.3 Inverter

Inspired by “NOT A AND B” gate in [6], this is an inverter gate adapted to the features of LoC devices. We consider it impossible to have negative resistance, but the flows can interfere with one another. The base of the inverter is an artificial neural network that implements an XOR gate with an input always equal to 1. Figure 1 (obtained with JavaNNS [5]) shows the network used to create our inverter, where the thresholds of the activation functions are below those of the hidden nodes and y .

Let x_1 be the input which always has the value 1, and let x_2 be the input to be inverted. Table 3 shows the truth table of the XOR gate where the rows of interest are in italics, i.e. those whose x_1 input is equal to 1. Fluidic resistance does not have negative values, so we can consider the negative weights as channels from input flow to channels whose weights in the neural network are positive; thus the channels, with negative weight-resistance, will oppose the positive flow. In the Figure 1, a “negative” channel goes to the channel represented in the neural network through the edge from the input x_1 to the node in the hidden layer.

The values for R_f to be considered for the each channel are (with absolute values)

$$R_{f_{x_1 \text{ hidden}}} = \frac{8\mu l}{\pi r_c^4} = 6.508 \quad (16)$$

$$R_{f_{x_1 y}} = \frac{8\mu l}{\pi r_c^4} = 4.661 \quad (17)$$

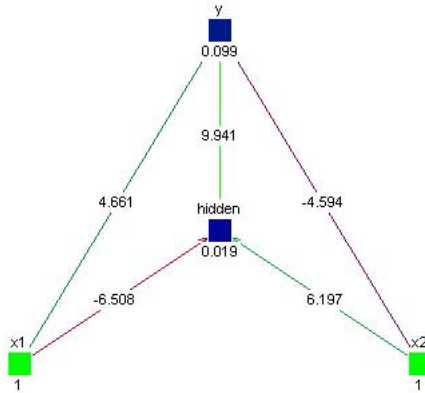


Fig. 1. XOR Network

Table 3. Truth Table of the XOR gate

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

$$R_{fx_2hidden} = \frac{8\mu l}{\pi r_c^4} = 6.197 \quad (18)$$

$$R_{fx_2y} = \frac{8\mu l}{\pi r_c^4} = 4.594 \quad (19)$$

$$R_{hidden y} = \frac{8\mu l}{\pi r_c^4} = 9.941 \quad (20)$$

The values for the ΔP , both hidden node and output node are

$$\Delta P_{hidden} = R_{fx_1hidden}Q_1 + R_{fx_2hidden}Q_2 \quad (21)$$

$$\Delta P_y = R_{fx_1y}Q_1 + \Delta P_{hidden} + R_{fx_2y}Q_2 \quad (22)$$

4 Conclusions and Future Work

This paper has presented, in section 2, other studies demonstrating the viability of neural networks for simulating lab-on-a-chip (LoC) components. The contribution of this paper is to demonstrate the suitability of mapping artificial neural network topology into a lab-on-a-chip (LoC) using the neural network as a design

tool. AND, OR and inverter gates have been used to prove the potential of this approach. It is important to emphasize that these three gates are necessary for a device, abstract or real, to compute universally. In this sense, it is possible to affirm this approach makes universal computation in a lab-on-a-chip (LoC) conceivable. In general, this idea opens possibilities for designing any type of circuit to be used to represent the functionalities performed in a lab-on-a-chip (LoC). Our approach permits the creation of more complex operations over samples e.g. “if element x_1 is present and element x_2 is not but element x_3 is, the microfluidic device must do a specific task” in a straightforward way. The circuits designed using this methodology should be tested by means of lab-on-a-chip (LoC) simulation tools. Comparisons between theoretical and simulated results could support this new methodology. Future research can be focused on taking into account the first element of equation 1, i.e. the influence of inductance with respect to the variation of volumetric flow over time in pressure. Neural networks able to solve differential equations could be used for this purpose.

References

1. Magargle, R., Hoburg, J.F., Mukherjee, T.: Microfluidic injector models based on artificial neural networks. In: Chakrabarty, K., Zeng, J. (eds.) *Design Automation Methods and Tools for Microfluidics-Based Biochips*, pp. 215–233. Springer, Heidelberg (2006)
2. Wang, Y., Bedekar, A.S., Krishnamoorthy, S., Sundaram, S., Singhal, A.K.: Model Order Reduction (MOR). In: Li, D. (ed.) *Encyclopedia of Microfluidics and Nanofluidics*, pp. 1382–1391. Springer, Heidelberg (2008)
3. Dutta, P., Horiuchi, K., Jubery, T.Z.: Microfluidic Circuits. In: Li, D. (ed.) *Encyclopedia of Microfluidics and Nanofluidics*, pp. 1151–1158. Springer, Heidelberg (2008)
4. Garcia-Cordero, J.L., Ricco, A.J.: Lab-on-a-Chip (General Philosophy). In: Li, D. (ed.) *Encyclopedia of Microfluidics and Nanofluidics*, pp. 962–969. Springer, Heidelberg (2008)
5. JavaNNS (Java Neural Network Simulator), <http://www.ra.cs.uni-tuebingen.de/software/JavaNNS>
6. Prakash, M., Gershenfeld, N.: Microfluidic Bubble Logic. *Science* 315(5813), 832–835 (2007)
7. Reyes, D.R., Iossifidis, D., Auroux, P.A., Manz, A.: Micro Total Analysis Systems. 1. Introduction, Theory and Technology. *Anal. Chem.* 74(12), 2623–2636 (2002)
8. Auroux, P.-A., Iossifidis, D., Reyes, D.R., Manz, A.: Micro Total Analysis Systems. 2. Analytical Standard Operations and Applications. *Anal. Chem.* 74(12), 2637–2652 (2002)

Aeration Control and Parameter Soft Estimation for a Wastewater Treatment Plant Using a Neurogenetic Design

Javier Fernandez de Canete, Pablo del Saz-Orozco, and Inmaculada Garcia-Moral

Dpt. System Engineering and Automation, Campus de Teatinos s/n
29071 Malaga, Spain
{canete,pablo.saz,gmoral}@isa.uma.es

Abstract. Biochemical oxygen demand and chemical oxygen demand are the most important parameters for wastewater management and planning, which represents the oxygen consumption from degradation of organic material. Insufficient levels of dissolved oxygen prevent the successful degradation of organic matter present, whereas too high levels cause a waste of energy and hence decreased efficiency. Therefore, the need for controlling dissolved oxygen through adequate aeration and sludge pumping operations is of great importance. This paper proposes the use of artificial neural networks applied both to the prediction of both oxygen demand parameters starting from secondary variable measurements and to the control of dissolved oxygen in aeration tanks for a nonlinear wastewater treatment model benchmark. Genetic algorithms are used for the automatically choice of the optimum control law based on the neural network model of the plant. The results show how this combined scheme can be effectively employed in aeration control.

Keywords: Wastewater process, artificial neural networks, genetic algorithms, aeration optimal control.

1 Introduction

Continuous challenges are nowadays required from wastewater treatment plants in order to satisfy new constraints in terms of quantity and quality of the discharged effluent for the compliance with stringent environmental regulations at minimum costs. In addition to plant improvements attained through the adoption of new equipment technologies, it is necessary to apply optimal control schemes in order to both meet the restrictions and minimizing the operation costs. The optimizing of wastewater process has been studied by several authors, mainly focusing on the activated sludge process. Different approaches can be found in the literature, for example in [1-3].

Besides this, wastewater treatment plants processes such as aeration, chemical feeds and sludge pumping are usually controlled by online sensor measurements. Nevertheless, biochemical oxygen demand is one of the major parameters for wastewater management and planning and is normally measured off-line as biochemical oxygen demand (*BOD*) and chemical oxygen demand (*COD*), both

parameters being in laboratory measured using a five days period for its computation [4]. Therefore, it is necessary to estimate the value of these off-line primary variables throughout the use of secondary variables, such as dissolved oxygen, mixed suspended solids, etc, which are measured on-line.

Artificial neural networks (ANN) can be considered from an engineering viewpoint, as a nonlinear heuristic model useful to make predictions and data classifications, and have been also used as a soft sensor for process control. In recent years some software sensors have been proposed in the wastewater environment, some examples are found at [5-6]. On the other hand, genetic algorithms (GA) have been applied successfully in the field of biological wastewater treatment for optimization and control [7-8].

In this paper, adaptive neural networks are applied both to the prediction of the biochemical oxygen demand starting from on-line secondary variable measurements, and to the modeling of a nonlinear wastewater process. Both approaches of neural network and GA have been implemented using specific tools coded in MATLAB and has been applied to a wastewater process model formed by five aeration tanks with a secondary settler. The results show that neural networks model could be employed successfully in estimating the daily *BOD* and *COD* in the inlet of wastewater treatment plant and the ANN combined with GA scheme can be effectively used for aeration control with better results as compared to PID controllers.

2 Wastewater Process Control

The activated sludge process (ASP), is globally the most common method of wastewater treatment. In this process, biomass (which is called activated sludge) suspended in the wastewater to be treated is cultivated and maintained in an aerated bioreactor tank. The wastewater is purified, i.e. organic carbon, nitrogen and phosphorus are removed, during its retention in the bioreactor. The bioreactor is followed by a clarifier basin, in which the biomass is separated by gravitational settling and pumped backward to the bioreactor, and the treated wastewater is directed as overflow to further treatment or to discharge. Excess activated sludge is removed from the process and treated separately. The schematic flow sheet of the process is presented in fig. 1.

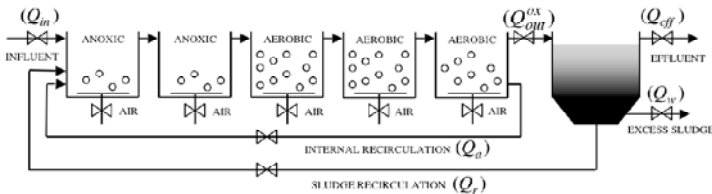


Fig. 1. Schematic of the wastewater process

In the present paper we are considering the wastewater process model presented as a result of the European research projects COST action 624 and 628 [9]. This model is used as a recognized benchmark for testing different control schemes approaches. The system is formed by 5 biological tanks and a secondary settler. Those first two tanks are anoxic while the three last ones are aerated. There are two sludge recirculation lines both from the settler and from the last aeration tank to the input line.

3 Neurogenetic Control

Probably the first step for controlling a wastewater process is to choose the controlled variables. This could be an obvious task for more general systems, but this is a complex task for this concrete process. In some references, Skogestad presents different hints for selecting the controlled variables and specially for those systems which are controlled under a cost function [10-11].

Nevertheless, some of the fundamental controlled variables as *COD* or *BOD* should be measured in laboratory, so they are not available for on-line control. Besides, there are constraints to be considered for some of these variables (table 1)

Table 1. Effluent constraints

	Adopted effluent constraints	Units
Total nitrogen N_t	18	gNm^{-2}
BOD_5	10	$gBODm^{-3}$
Total COD	100	$gCODm^{-3}$
Suspended solids TSS	30	$gSSm^{-3}$

On the other hand, the proposed control scheme should assure not violating the constraints (or violating them as less as possible) with an optimum operational cost. The cost function should penal every violation and should increase its value with the energy consumption.

3.1 Cost Function

The main output of a wastewater process is not usually marketable as is thrown to a river or to the sea, so the main objective of this kind of processes is to meet the constraints marked by the legislations at the lowest price. So a cost study function should be defined in order to actuate in the direction of minimizing it, by considering pumping costs, aeration costs, mixing energy and sludge disposal costs.

Firstly it should be considered the pumping cost E_p . That cost is directly on the recycled flow rate Q_a , the internal sludge recirculation rate Q_r and in the excess sludge flow rate Q_w . Following the expression by [12], the consumed energy follows

$$E_p = \frac{1}{T} \int_{T_0}^{T_0+T} \left(0.004Q_a(t) + 0.008Q_r(t) + 0.05Q_w(t) \right) dt \tag{1}$$

The aeration energy E_a can be expressed as follows in the expression

$$E_a = \frac{S_0^{sat}}{1.8 \cdot T \cdot 1000} \int_{T_0}^{T_0+T} \sum_{i=1}^5 (V_i \cdot K_{La,i}(t)) dt \tag{2}$$

where S_0^{sat} is the oxygen saturation concentration and $K_{La,i}$ $i=1\dots5$, is the oxygen mass transfer coefficients.

The anoxic zones should be mixed in order to avoid settling conditions, so energy should be supplied. This mixing energy E_M follows the next expression

$$E_M = \frac{1}{T} \int_{T_0}^{T_0+T} \sum_{i=1}^5 \begin{cases} 0.005 \cdot V_i dt & K_{La,i} < 20 \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

Assuming a constant cost of the energy $K_e = 0.09\text{€}/Kwh$, the total energy cost can be expressed as $J_e = K_e(E_p + E_a + E_M)$. To this energy cost, a sludge disposal cost should be added. If a constant cost $K_D = 80\text{€}/Tm$ is considered, the disposal cost can be expressed as

$$C_D = \frac{1}{T} \int_{T_0}^{T_0+T} \sum_{i=1}^5 (K_D \cdot TSS_w(t) \cdot Q_w(t)) dt \tag{4}$$

so, the total cost will be $J = J_e + C_d$.

In this particular case, the violation of the restrictions should be taken in account in the cost function. As a model based predictive control scheme is going to be used, a good cost function should be definite positive and increasing with the value of both the consumption and the number of violations. So the following function is proposed as a cost function in this kind of problems

$$D = K_1 \frac{\sum_i^n J_i}{n \cdot J_{max}} \cdot K_2 \frac{NV}{n^5} \tag{5}$$

where J_i is the cost in each step, J_{max} is the maximum cost associated to the physical limits of the different actuators (the pumps and the aerators), n is the horizon in the predictive strategy, NV is the predicted number of violations for each set of inputs in the n steps and K_1 and K_2 are two constants, which values are related to the relative values between the operation costs and the restrictions violations costs.

3.2 Model Based Predictive GA-Control

The model based predictive control uses a model of the process to predict the future output of the system according to the possible control actions. This prediction is used to determine the optimum input that minimizes a certain behavior criteria. Obviously an accurate model of the plant is a first necessary step for applying this strategy.

For complex optimization problems derivative-free schemes as genetic algorithms have been receiving increasing amounts of attention due to derivative freeness and its flexibility. So, in the present paper a neural-network-based identifier is proposed while the cost function is optimized through the use of genetic algorithms. A diagram of the control strategy is shown in fig. 2.

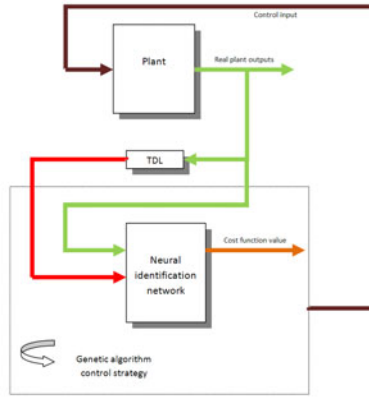


Fig. 2. Neuro-genetic control strategy

4 Neural Network Identification

One of the main problems in controlling wastewater plants is the fact that two of the main effluent quality variables as are the *COD* and the *BOD* have to be measured in a laboratory. So their values aren't available for an on-line control strategy. Therefore the neural network responsible for acquiring the dynamic behavior of the plant also can be considered as an estimator for its capability of determining the value of those variables from other ones which can be measured in a reliable and cheaper way.

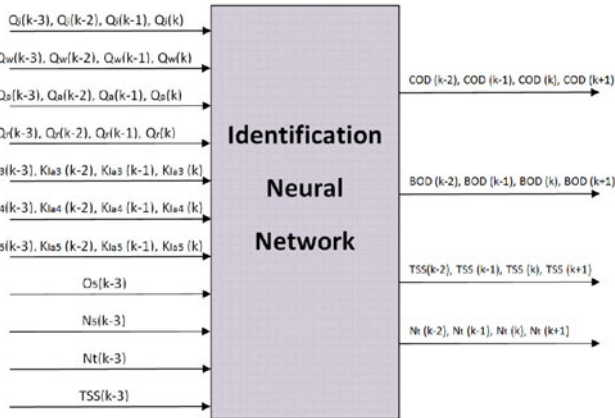


Fig. 3. Neural identification model structure

In this case the input to the identification neural network are flows Q_{in} , Q_w , Q_a , Q_r , oxygen mass transfer coefficients for tanks 1-3 $K_{La,1}$, $K_{La,2}$, $K_{La,3}$, dissolved

nitrogen in tank 5 N_5 , dissolved oxygen in tank 5 O_5 , and the control inputs suspended solids TSS , total nitrogen N_t and the outputs are given by the COD , the BOD , the TSS and the N_t in the effluent. In the fig 3 a diagram of the identification network can be seen.

The identification network has a two hidden layers structure with, respectively, 25 and 10 neurons. It was trained using a Levenberg-Marquardt algorithm [13]. The training set consisted in 1000 training data from different experiments and it was validated against an independent experiment. In fig 4 the validation results can be seen by software sensing COD and BOD , thus the other two variables TSS and N_t can be measured. As it can be seen, both mean errors are below 2%.

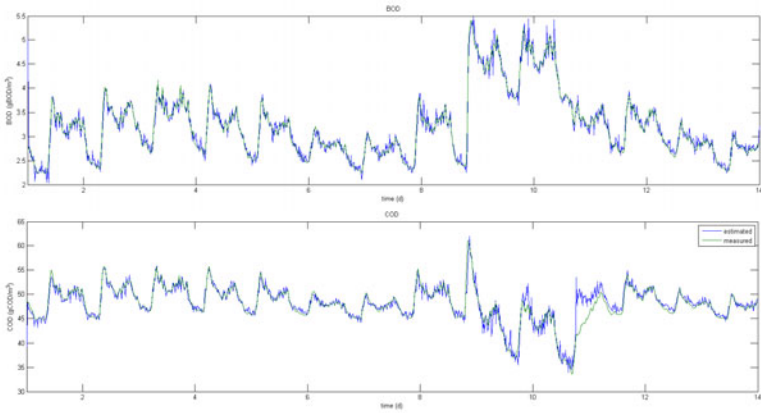


Fig. 4. Neural identification validation

5 Results

In order to validate the control strategy a simulation has been performed with the conditions provided by the COST simulation benchmark. Using a model predictive control with a time horizon of 4 steps (according to the identification model), the genetic algorithm was selected with a population of 70 habitants, 65 generations and a codifying depth of 8 bits per variable. The results of the control scheme for a value of $K_1 = 1.5$ and $K_2 = 5.5$ can be seen in fig. 5.

A review of both the effluent quality and the energy consumption can be seen in table 2, while the mean consumption was of 1652 €/day. Both results improve the classical fine tuned multi-PID control strategy (further information of these results can be found in [11] and [14]. At it can be seen both a reduction in power consumption and in violations can be obtained.

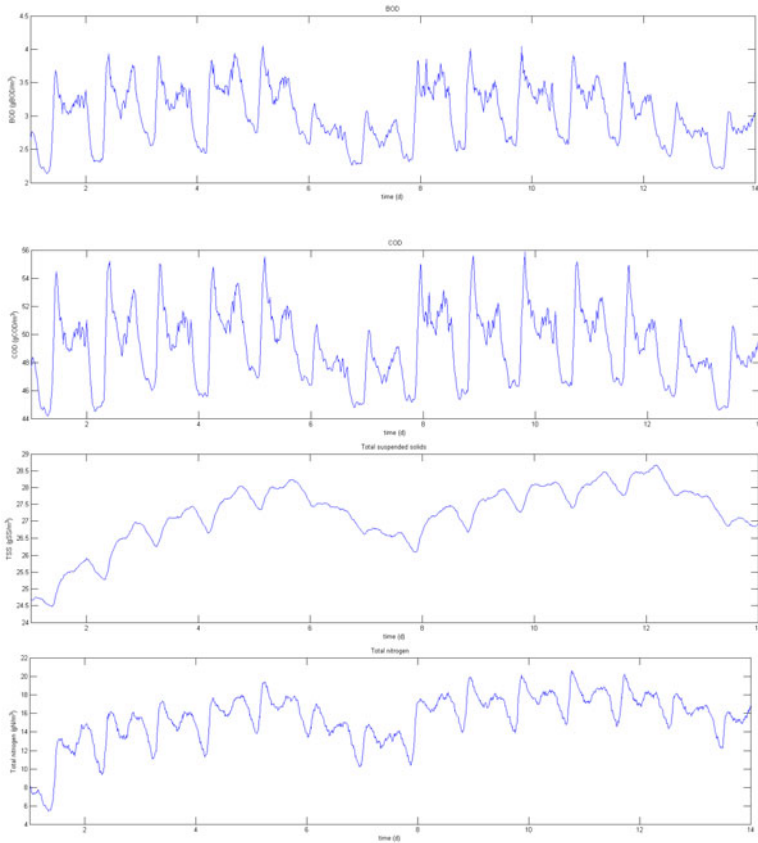


Fig. 5. Concentrations for BOD , COD, TSS and N_t at the effluent for the GA-controller

Table 2. Effluent results

	Average	Violations (in % of the operating time)
Total nitrogen	15.42	10.01
BOD ₅	3.01	0
Total COD	49.05	0
Suspended solids	27.20	0

6 Conclusions

In this paper a control strategy based in an identification neural network and in a genetic algorithm for optimizing both the consumption and the number of regulations violations is proposed for a biological wastewater system. The complete scheme was tested against the COST benchmark simulation model. The presented procedure has

also the advantage of being able to predict the concentration of some variables which measurement is expensive or/and time consuming, so it can be also used to predict the effluent's quality and use that prediction in an on-line control scheme.

References

1. Fikar, M., Chachuat, B., Latifi, M.A.: Optimal Operation of Alternating Activated Sludge Processes. *Control Eng. Prac.* 13(7), 853–861 (2005)
2. Azwar, S., Hussain, M.A., Ramachadran, K.B.: The Study of Neural Network Based Controller for Controlling Dissolved Oxygen Concentration in a Sequencing Batch Reactor. *Bioprocess. Eng.* 28, 251–265 (2006)
3. Biswas, P., Bose, P., Tare, V.: Optimal Choice of Wastewater Treatment Train by Multi-Objective Optimization. *Eng. Optimiz.* 39(2), 125–145 (2007)
4. Chapman, D.: *Water Quality Assessments*. Chapman and Hall, London (1992)
5. Auburn, C., Theilliol, D., Harmard, J., Steyer, J.P.: Software Sensor Design for COD Estimation in an Anaerobic Fluidized Bed Reactor. *Water Sci. Technol.* 43(7), 115–120 (2001)
6. Civelekoglu, G., Perendec, A., Yigit, N., Kitis, M.: Modelling Carbon and Nitrogen Removal in a Industrial Wastewater Treatment Plant Using an Adaptive Network Based Fuzzy Inference System. *Clean* 37(6), 617–625 (2007)
7. Doby, T.A., Loughlin, D.H., De los Reyes, F.L., Ducoste, J.J.: Optimization of Activated Sludge Designs Using Genetic Algorithms. *Water Sci. Technol.* 45(6), 187–198 (2002)
8. Holenda, B., Domokis, S., Rédey, A., Fazakas, J.: Dissolved Oxygen Control of the Activated Sludge Wastewater Treatment Process Using Model Predictive Control. *Comput.Chem. Eng.* 32, 1270–1278 (2008)
9. Copp, J.B.: *The COST Simulation Benchmark: Description and Simulation Manual*
10. Skogestad, S.: Control Structure Design for Complete Chemical Plants. *Comput. Chem. Eng.* 28, 219–234 (2004)
11. Mulas, M., Baratti, R., Skogestad, S.: Controlled Variables Selection for a Biological Wastewater Treatment Process. In: 8th IFAC Symposium on Dynamics and Control of Process Systems, Leuven (2010)
12. Alex, J., Benedetti, L., Copp, J., Gernaey, K., Jeppson, U., Nopens, I., Pons, M., Rieger, L., Rosen, C., Steyer, J., Vanrolleghen, P., Winkler, S.: Benchmark Simulation Model n°1. Technical Report, IWA Taskgroup on Benchmark Control Strategies for WWTPs (2008)
13. Baruch, I., Mariaca-Gaspar, C.R.: A Levenberg-Marquardt Learning Applied for Recurrent Neural Identification and Control of a Wastewater Treatment Bioprocess. *Int. J. Intell. Syst.* 24, 1094–1114 (2009)
14. Wang, M.: Impact of Dissolved Oxygen Performance on the Operational Cost of Activated Sludge Process. MSc. Thesis, Cranfield university (2007)

Pulse Component Modification Detection in Spino Cerebellar Ataxia 2 Using ICA

Rodolfo V. García¹, Fernando Rojas², Jesús González², Luis Velázquez³, Roberto Rodríguez³, Roberto Becerra¹, and Olga Valenzuela⁴

¹ Network Department, University of Holguín. Cuba
rodgarberm@gmail.com

² Dept. of Computer Architecture and Technology, University of Granada. Spain
frojas@atc.ugr.es

³ Centre for the Research and Rehabilitation of Hereditary Ataxias
"Carlos J. Finlay", Holguín. Cuba

⁴ Dept. of Applied Mathematics, University of Granada

Abstract. Anomalies in the oculomotor system are well known symptoms in patients with severe Spino Cerebellar Ataxia 2 form of autosomal dominant cerebellar ataxias (ADCA), including the main parameters used to describe saccadic movements. Also a combination of a pulse and step components constitutes an accepted model of the saccadic generation system. In the present work, Independent Component Analysis (ICA) is used to separate both components, revealing a significant difference in the time of the pulse component to reach its maximum value. This result suggests its use in order to obtain a classification tool to diagnose this disease. Five electro-oculographic records of patients of SCA2 ataxia and five control subjects were processed with the ICA based algorithm with this objective.

Keywords: biomedical engineering, computer aided diagnosis, independent component analysis, ataxia, SCA2, electro-oculography.

1 Introduction

There are almost 800 patients and 8000 presymptomatic relatives in Cuba at risk of developing some autosomal dominant cerebellar ataxia (ADCAs), in the next few years. This is a heterogeneous group of dominantly inherited neurological disorders characterized by progressive ataxia that results from degeneration of the cerebellum and its afferent connections [1].

This form of ataxia occurs commonly in persons of Spanish ancestry in north-eastern Cuba, a figure much higher than that found in western Cuba or in other parts of the world. Reported high prevalence is probably the result of a founder effect, but might be due to an interaction between a mutant gene and an unidentified environmental neurotoxin [2,3].

The ocular movement records have been widely used in processing and classification of biological signals and pathological conditions, specifically saccadic movements in response to a visual stimulation, are considered amongst the most useful tools in the

study of neurological pathologies . Several studies have reported oculomotor abnormalities in ADCA [4,5], slowness of saccades has been suggested as a relatively characteristic finding in SCA2 [6-9].

A commonly accepted model of the saccadic system states that the programming of the saccades consists of a pulse and a step, the pulse command is used to move the eye to a new position and the step command keeps the eyes at the new position [10,11].

2 ICA Based Pulse and Step Decomposition

Independent Component Analysis (ICA) [12], is widely used to process biomedical signals [13,14], including clinical electroencephalography records and electrocardiography, amongst others.

Independent component analysis is aimed to find a linear transformation given by a matrix \mathbf{W} , which mixes the underlying independent components \mathbf{x} (in this case saccadic pulse and step), to produce the observed signals \mathbf{y} :

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{x} \quad (1)$$

Based in preliminary works [15-18], ICA was applied in order to decompose pulse and step components of the horizontal saccadic movements, consequently computing the time to the maximum of the pulse component and finally establishing differences between patients and control subjects.

Beginning from the time of the occurrence of stimulus position change, each response to the stimulus change during a fixed time of 1250 ms was used as an observed variable, conforming a rectangular matrix with dimension m (number of responses) by n (numbers of points of each response):

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1n} \\ S_{21} & S_{22} & \dots & S_{2n} \\ \vdots & \vdots & \dots & \vdots \\ S_{m1} & S_{m2} & \dots & S_{mn} \end{bmatrix} \quad (2)$$

A preliminary Scree test checked the presence of only two components with a significant content, corresponding to the step and pulse components respectively, then the Infomax ICA [19] algorithm was applied to get the independent components, taking the first two components with the largest contribution to the variance of the data.

3 Experiment Setup

In this work, we apply ICA to electro-oculographic records of five patients of ataxia SCA-2 and five healthy subjects, diagnosed and classified in the “*Centro de*

Investigación y Rehabilitación de las Ataxias Hereditarias de Holguín (Centre for the Research and Rehabilitation of Hereditary Ataxias, CIRAH)”, to obtain both, pulse and step components, and computing the time of the maximum value of pulse component in order to make a classification of the subjects.

All the records were carried out by the medical staff of CIRAH, each subject was placed in a chair, with a head fixation device to avoid head movements, the variables were collected by a two channels electronystagmograph (Otoscreen, Jaeger-Toennies, D-97204 H6chberg, Germany), recording conditions are as follows: electrodes of silver chloride placed in the external borders of right eye (active electrode) and left eye (reference electrode), high pass filtering 0.002 Hz, low pass filtering 20 Hz, sensitivity 200 $\mu\text{V}/\text{division}$, and sampling frequency 200 Hz. For stimulus generation was used a black screen CRT display showing a white circular target with an angular size of 0.7° , using an stimulation angle of 30° . The stimulus and patient response data are automatically stored in ASCII files by Otoscreen electronystagmograph.

4 Experimental Results and Discussion

Pulse and step components were obtained for healthy and ataxia sick subjects, as it was previously expected. The visual inspection shows well defined morphology in the expected waveform of both components for every record, Fig. 1 shows the results for a healthy subject (left) and a patient (right).

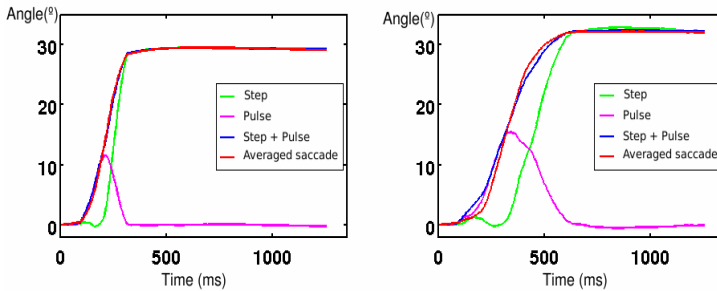


Fig. 1. Obtained components for healthy (left) and patient (right)

As can be noticed in Figure 1, the time to the maximum of the pulse component has an increased latency in a patient when compared to a healthy subject, suggesting this parameter as a good candidate for classification. Although other parameters such as the latency at the start of the saccade, averaged saccade or pulse and step component durations could also be used, the chosen latency to the maximum of the pulse component shows better performance in classification, as it does not need a threshold value. Nevertheless, the use of several parameters in conjunction is not discarded for further research, although a higher number of samples would be necessary.

Table 1 shows the values of this latency for every subject, and the mean and standard deviation of the two categories of subjects, while Fig. 2 is a graphical representation of the data, and the averaged consolidation.

Table 1. Times to the maximum of the pulse component for control subjects and patients

						Mean	Std. Dev.
Subject	S1	S2	S3	S4	S5	Control subjects	
Time (ms)	43	44	43	37	43	42,00	2,83
Subject	E1	E2	E3	E4	E5	Patient	
Time (ms)	68	98	119	84	99	93,60	18,98

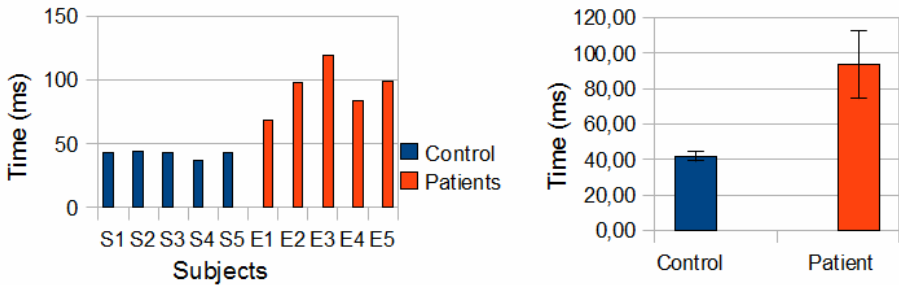


Fig. 2. Times to the maximum of pulse component for every subject (left) and averaged results (right)

5 Conclusions

There exists a remarkable difference in the time to reach the maximum of the pulse component for every subject, depending of his condition, well reflected in the consolidated means and standard deviation. No overlapping was found between patients and control subjects.

Therefore, this parameter is an adequate classificatory criterion to determine if a subject is a healthy subject or a patient of ataxia. An important task to do is the use of more subjects to confirm this result, and try to improve the method in order to evaluate the effects produced by ataxia on other parameters of the pulse and step components.

Acknowledgments. This work has been supported by the Genil Start-up Project for Young Researchers (<http://genil.ugr.es>) “Processing and Classification of Electro-oculography (EOG) Data for Ataxia SCA-2 Diagnosis” (PYR-2010-23) from the CEI BioTIC GENIL (CEB09-0010) of the CEI Program from the MICINN. The authors would like to thank the rest of the personnel in the Centre for the Research and Rehabilitation of Hereditary Ataxias “Carlos J. Finlay”, Holguín, (Cuba) for their support and collaboration.

References

1. Velázquez, L.: Ataxia espino cerebelosa tipo 2. Principales aspectos neurofisiológicos en el diagnóstico, pronóstico y evaluación de la enfermedad. Ediciones Holguín (2006)
2. Auburger, G., Orozco Diaz, G., Ferreira Capote, R., Gispert Sanchez, S., Paradoa Perez, M., Estrada del Cueto, M., et al.: Autosomal dominant ataxia: Genetic evidence for locus heterogeneity from a Cuban founder-effect population. *American Journal of Human Genetics* 46(6), 1163–1177 (1990)
3. Orozco, G., Estrada, R., Perry, T.L., Araña, J., Fernandez, R., Gonzalez-Quevedo, A., et al.: Dominantly inherited olivopontocerebellar atrophy from eastern Cuba: Clinical, neuropathological, and biochemical findings. *Journal of the Neurological Sciences* 93(1), 37–50 (1989)
4. Engel, K.C., Anderson, J.H., Gomez, C.M., Soechting, J.: Deficits in ocular and manual tracking due to episodic ataxia type 2. *Movement Disorders* 19(7), 778–787 (2004)
5. Schöls, L., Linnemann, C., Globas, C.: Electrophysiology in spinocerebellar ataxias: Spread of disease and characteristic findings. *The Cerebellum* 7(2), 198–203 (2008)
6. Wadia, N., Pang, J., Desai, J., Mankodi, A., Desai, M., Chamberlain, S.: A clinicogenetic analysis of six Indian spinocerebellar ataxia (SCA2) pedigrees. The significance of slow saccades in diagnosis 121(12), 2341–2355 (1998)
7. Seifried C., Velázquez-Pérez L., Santos-Falcón N., Abele M., Ziemann U., Almaguer L.E., et al.: Saccade velocity as a surrogate disease marker in spinocerebellar ataxia type 2 [Internet] (2005), <http://www.scopus.com/scopus/inward/record.url?eid=2-s2.0-22144471158&partnerID=40&rel=R8.2.0> (cited November13, 2008)
8. Velázquez-Pérez, L., Seifried, C., Santos-Falcón, N., Abele, M., Ziemann, U., Almaguer, L.E., et al.: Saccade velocity is controlled by polyglutamine size in spinocerebellar ataxia 2. *Annals of Neurology* 56(3), 444–447 (2004)
9. Velázquez-Pérez, L., Almaguer-Mederos, L., Santos-Falcón, N., Hechavarría-Pupo, R., Sánchez-Cruz, G., Paneque-Herrera, M.: Spinocerebellar ataxia type 2 in Cuba. A study of the electrophysiological phenotype and its correlation with clinical and molecular variables. *Revista de Neurología* 33(12), 1129–1136 (2001)
10. Robinson, D.: The mechanics of human saccadic eye movement. *J. Physiol.* 174(2), 245–264 (1964)
11. Optican, L.M., Robinson, D.: Cerebellar-dependent adaptive control of primate saccadic system. *J. Neurophysiol.* 44(6), 1058–1076 (1980)
12. Hyvärinen, A., Karhunen, J., Oja, E.: *Independent component analysis*. John Wiley and Sons, Chichester (2001)
13. James, C.J., Hesse, C.W.: *Independent component analysis for biomedical signals*. *Physiol. Meas.*, <http://iopscience.iop.org/0967-3334/26/1/R02> (cited 10:19:31)
14. Jung, T.-P., Makeig, S., Lee, T.-W., McKeown, M.J., Brown, G., Sejnowski, T.J.: *Independent Component Analysis of Biomedical Signals*
15. Alvarez, T.L., Daftari, A., Semmlow, J.: *Independent Component Analysis of Divergence Eye Movements*. In: *Proceedings of 2nd International IEEE EMBS Conference on Neural Engineering*, pp. 608–611 (2005)

16. Gayed, B.A., Alvarez, T.L., Daftari, A., Semmlow, J.L., Pedrono, C.: Independent component analysis reveals transient component of divergence varies with initial position [Internet]. In: Proceedings of the Northeast Conference Bioengineering (2005), <http://www.scopus.com/inward/record.url?eid=2-s2.0-28444454426&partnerID=40&md5=c8fca8b6a42d1f90ac5d7f9b20ed42ab> (cited April 21, 2010)
17. Chen, Y.-F., Chen, T., Semmlow, J.: Decomposition of Control Signals for Saccade and Vergence Eye Movements Using Independent Component Analysis. *Journal of Medical and Biological Engineering* 24(1), 23–28 (2004)
18. Lin, H.-H., Chen, Y.-F., Chen, T., Lee, Y.-Y.: Decomposition of saccadic pulse and step components by independent component analysis [Internet]. In: Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences, METMBS 2004, pp. 124–128 (2004), <http://www.scopus.com/scopus/inward/record.url?eid=2-s2.0-11144264106&partnerID=40> (cited December 3, 2008)
19. Bell, A.J., Sejnowski, T.: An information-maximization approach to blind separation and blind deconvolution. *Neural Computation* 7, 1129–1159 (1995)

Early Pigmentary Retinosis Diagnostic Based on Classification Trees

Vivian Sistachs Vega¹, Gonzalo Joya Caparrós², and Miguel A. Díaz Martínez³

¹ Applied Mathematics Department. Havana University, Cuba

² Electronic Technology Department. Malaga University, Spain

³ General Math. Department. Polytechnic Superior Institute “J. A. Echeverría”, Havana, Cuba
vivian@matcom.uh.cu, gjoya@uma.es, migue@ind.cujae.edu.cu

Abstract. In this work we analyze different classification tree based techniques (CART, Bagging and Boosting), evaluating their performance with respect to their capability to reduce error rate and correct pattern classification. As a case of study we propose the classification of Pigmentary Retinosis patients through electroretinograms. Pigmentary Retinosis is the most frequent retina dystrophy (1/5000). The electroretinogram (ERG) constitutes a fundamental test in the study of this type of dystrophy since the wide clinical heterogeneity of visual diseases. Besides, retina electrophysiological study can provided information that may be used to predict the disease before the apparition of symptoms and allows us to corroborate the affectation degree on the dystrophic process of cones and canes. As experimental database we use a set of 148 electroretinograms , which is part of a retrospective study carried out by the Cuban National Reference Center of Pigmentary Retinosis.

Keywords: Pigmentary Retinosis, Classification and Regression Trees (CART), Bagging, Boosting.

1 Introduction

Classification Trees based methods have become one of the most flexible, intuitive and powerful tools to analyze and explore complex data structures. They constitute a good option when facing problem consisting on classification or prediction and one needs to generate rules that can be easily explained.

Decision or classification trees allow to find key variables that identify the members of each current group and to formulate rules to predict the group assigned to a new individual or pattern. Also, they allow us to deploy graphical and statistical analysis results, providing a reliance measure of the correctness of the classification.

Classification techniques are widely used on human activity, and can serve for two proposes: on the one hand, they can be used when one doesn't know the classes beforehand and one wants to find clusters or groups with differentiated features into the observed set (non-supervised learning); on the other hand, they can be used when the different classes are known and the objective is to establish a rule set that allows us to classify a new observation in one of the existing classes (supervised learning) [11].

This classification process has two fundamental steps. The first one is known as learning step (training phase) on which the algorithm builds the classifier, learning

from known observation set. On the second step (operation phase), the model is used to classify new patterns, and for it, is necessary to estimate the predictive precision of the classifier. These techniques have been extensively used on the last decades in many branches of the science and engineering, and especially in medical sciences where decision making is based on the analysis of high number of data and none totally specified rules.

In this paper we carry out an approach to classification problems, particularizing in classification tree and classification to decision making techniques: Classification and Regression Trees (CART), Bagging, and Boosting (section 2). Then, we present the variables and features of our case of study: classification of an electroretinogram set from phenotypic characteristics (section 3), and we present and discuss our results on early diagnosis of Pigmentary Retinosis using extracted decision rules (section 4). Finally, conclusions are exposed (section 5).

2 The Classification Problem

Classification problem frequently presents situations on which we need to find a mathematical model that allows us to predict a categorical variable. When one finds the model by means of a supervised learning algorithm, the process has two fundamental steps as it is known: training phase on which the classification algorithm builds the final classifier learning from a set of known observations classes; test phase where the model have to classify news patterns with the required predictive precision.

If training and test are accomplished on the same pattern set, the resultant estimation would probably be optimistic. Therefore the most accurate thing to do will be to use a test set (formed by the observations and their corresponding classes), randomly selected from the data set and different from the training patterns [4].

On the case where dependent variable is continuum, a regression model has to be used, obtaining a regression tree. On the other hand, when dependent variable is categorical, a classification model is used, obtaining a classification tree [6].

The tree is structured as follows:

It begins with a root node containing all the sample observations. Descending along the tree, the data are configured in a branch way, in data subsets mutually exclusive that become more homogeneous and less impure in each level. The tree is composed by different node layers: the root node; interior nodes which descend from a father node and produce two children nodes; and terminal or leaf nodes which haven't got children. Every node is partitioned considering all the possible binary ramifications of the way $X_j \leq s$ when the predictive variable is continuum, or in the way $X_j \in S$ when our variable is categorical, where s is a threshold value for variable X_j and S is a non-empty subset of possible categories of X_j . For each node, the chosen threshold values (or sets) will be these producing most homogeneous children. The ramification kindness will be given by the impurity decrease. After this Division process, one can apply a Pruning Process and a Final Selection process.

Other aspect to take account in classification problems is that to increase the precision we have to recourse to aggregate several classifiers. There exist several alternatives,

among them are applying the same classification over different modified versions of the training set. Some of these techniques are relatively new and have been studied in the past few years. Some of those techniques are Bagging and Boosting.

2.1 Classification Trees Problem

Classification and Regression Trees (CART). Suppose $Y=\{1,2,\dots,k, \dots,K\}$. Then, a CART algorithm classify the observations of the node m in the class $k(m) = \arg \max_K \hat{p}_{mk}$ where \hat{p}_{mk} is the proportion of observations of the node m belonging to class k . The purity of an answer is measured then as the proportion of cases with the same response variable value. A completely pure set is that in which all cases belong to the same class. Later the best ramification will be the one who produces less impure children. There exists three impurity measures Q_m :

- 1. Gini Index.

$$\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^K \hat{p}_{mk} (1 - \hat{p}_{mk}) \tag{1}$$

- 2. Entropy

$$\sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk} \tag{2}$$

- 3. Classification error.

$$1 - \hat{p}_{mk} \tag{3}$$

The two last ones are the most used as ramification criteria because its sensitivity to the change and the growth of the tree, while the first one is frequently used as an impurity measure in the branch pruning ([6], [7]). Once found the best ramification we partition the data on the two following regions:

$$R1(j, s) = \{X|X_j \leq s\} \text{ and } R2(j, s) = \{X|X_j > s\}$$

Then, we proceed to repeat this process of ramification in each one of these branches until a certain stop criteria is accomplished (for example, until the error decrease is greater than a threshold, or some node has achieved the minimum pre-established size to be a parental node, or when the division of this produces children of minimum size).

Bagging. Bagging proposed in [2], is based on the bootstrapping and aggregation methods. Starting from a training set $X=\{X_1, \dots, X_n\}$ by means of random extraction with replacement of the same number of elements than the original set of n elements, we obtain B bootstrap samples $X_b=\{X_{1b}, \dots, X_{nb}\}$ where $b = 1,2, \dots, B$. In some of these samples the presence of noise (bad observations) will be reduced, causing that the constructed classifier presents a better behavior than the classifier constructed with the original set.

Therefore Bagging can be used to construct a better classifier when the training set has noisy observations. Its algorithm is shown in [6].

Boosting. Boosting is a method that enhances a classifiers precision using the classification method as a subroutine to produce a high precision classifier on the training set. Its most used variant is AdaBoost.M1 proposed in [5].

Every time a classifier is generated, the weights are actualized assigning them higher weight to the badly classified observations and lower weight to the correctly classified, taking these new samples as the training set of the next classifier. The general idea is to force the new classifier to minimize the error committed [1], [3].

2.2 Evaluation of the Classifier Accuracy

An interesting aspect on classification problems is to know how is going to be the behavior of the classifier with new observations. It is very important than test observations don't be used in the process of learning of the classifier. On this sense exists two strategies, frequently used: training set and test set (holdout), cross validation with k partitions [4].

Training and test sets. It consists on partitioning the data set randomly into two independent sets: a training and a test set. Generally takes the 75% of the data as training set and the remaining 25% as test set. The training set is used to obtain the classifier and the test to estimate the precision.

Cross validation with k-partitions. At the beginning the data are partitioned randomly in k subsets mutually exclusives D_1, D_2, \dots, D_k of approximately equal size. Then, the process of training and testing is preformed k times of the following way: on the i th iteration the partition D_i is used as a test set and the remaining subsets as training sets. Finally the precision is estimated as the total number of correct classifications on the k iterations, divided by the total of individual of the initial set.

Some examples of headings are: "Criteria to Disprove Context-Freeness of Collage Languages", "On Correcting the Intrusion of Tracing Non-deterministic Programs by Software", "A User-Friendly and Extendable Data Distribution System", "Multi-flip Networks: Parallelizing GenSAT", "Self-determinations of Man".

3 Case Study: Lost Patterns Classification on an Electroretinogram (ERG) from Phenotypical Characteristics

Our case of study is a part of a retrospective study initiated on 2008 and carried out by the National Reference Center of Pigmentary Retinosis in Cuba [8],[9]. Its goal is to find decision rules that allow us to classify patients (individuals with vision problems) to whom were analyzed certain phenotypical characteristics as visual field (cv), visual acuity (av), age , evolution time ($timevol$), and inherence mode ($heren$), in a type of ERG (so this is the variable to be classified, that we call erg), since this test gives us a more exact degree of affectation of every cellular system to corroborate the affectation of the dystrophic cell process.

A database of 74 patients (148 eyes) was analyzed because they were the patients that fulfill the inclusion criteria and canes and cones differentiated ERG. In spite of being Retinosis a bilateral disease is usual on ophthalmological studies to analyze the behavior of both eyes, since they don't present necessarily the same visual field and visual acuity behavior.

It was used the R software 2.12.0 (2010) [10], where the dependent variable was the *erg* and the used independent ones were the continuous *av*, *age*, and *timevol*; and the categorical *cv*, *heren*, *sex* (See table 1).[12]

Table 1. Study Variables

Variable	Code	Categories
erg: ERG pattern (Nominal scale)	RBC	Registrable ERG with canes-cones
	RCB	Registrable ERG with cones-cane
	RND	Registrable ERG undefined pattern
	NR	Unregistrable ERG
herengen: Inheritance (Nominal scale)	AD	Autosomal Dominant
	AR	Autosomal Recessive
	LX	Recessive linked to X
av: Visual Acuity (Continuum)	av	
cvcod : Visual Field (Ordinal Scale)	N1	Escotomaanular
	N2	Concentric reduction over 30 °
	N3	Concentric reduction to 30 °
	N4	Concentric reduction to 20 °
	N5	Concentric reduction ≤ 10 °
	N6	Normal
	N7	Central Escotoma
Timevol: Time evolution (Continuum)	timevol	
age (Continuum)	age	
sex (Nominal Scale)	sex	male (m), female (f)

4 Results and Discussion

In our problem, methods CART, Boosting and Bagging have been applied to classification of a qualitative variable (*erg*). The implementation of algorithms is carried out on the software R.

We define a training sample of 98 ERG patterns randomly selected of the 148 patients observed. The rest of the patients (50) constitute the test set to validate the resultant classifiers as predictor. The validation criterion is the classification error obtained. Simulations have been carried out for each one of the three explained methods. We have repeat the learning and validation process 10 times for each method.

Figure 1 shows the classification tree obtained by CART for the 8th simulation. From this tree, we can obtain decision rules that could help a specialist to diagnose the degree of affectation of the disease and to predict the retinosis type (see Table 2). The classification error rate was 0.22 which is the smallest of the ten trees obtained from the CART (see table 3).

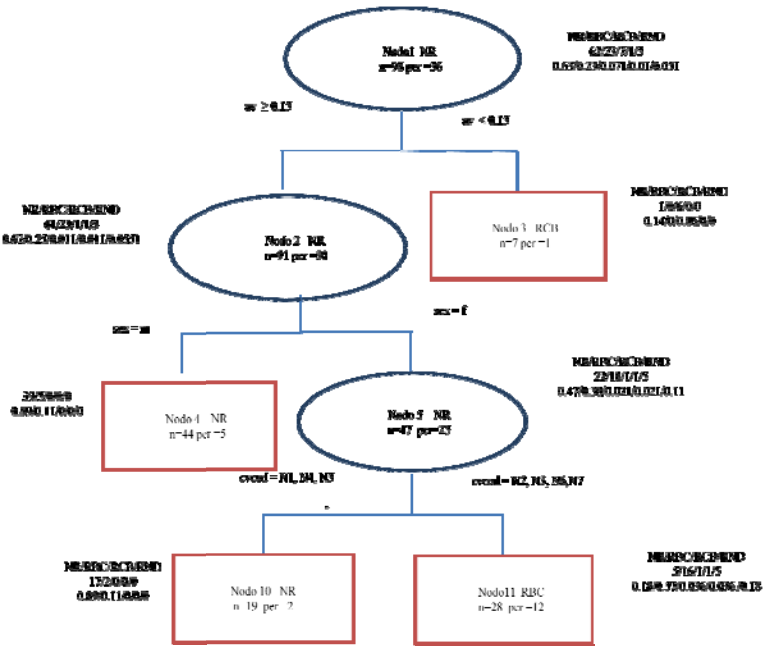


Fig. 1. CART of the training sample 8

Table 2. Decisión Rules

Decision Rules
If $AV < 0.15$ then the patient is $erg = RCB$
If $AV \geq 0.15$ and $sex = male$ then the patient is $erg = NR$
If $AV \geq 0.15$, $sex = female$ and $SV = (N2 \text{ or } N3 \text{ or } N6 \text{ or } N7)$ then $erg = RCB$
If $AV \geq 0.15$, $sex = female$ and $SV = (N1 \text{ or } N2 \text{ or } N5)$ then the patient is $erg = NR$

Applying the algorithms Boosting and Bagging to the selected data in the training sample 25 classification trees were used, i.e. the number of iterations of the algorithms. With these we obtain more than one classification tree but the method allows to obtain a case prediction, which consists in a weighted vote of the decision from the 25 decision trees obtained. The obtained classification prediction errors are shown in the table 2 for the ten simulations. One can appreciate that Boosting produce a prediction error 0.14 (2nd simulation) and Bagging produce a prediction error 0.20 (9th simulation). This comparison obtained from table 2 allows us to affirm than in this problem of Pigmentary Retinosis evaluation, Boosting and Bagging methods have the best performance.

We have determined which is the relative significance of each variable (the number of times the variable is selected in the data division) on the classification trees construction of Boosting and Bagging (see results in figures 2 and 3). Observe that the most relative significant variables resulted to be visual acuity (AV) shown on figure 2.b and visual field (cvcod) shown in figure 2.a. In both cases the inheritance was the least significant.

Table 2. Classification Errors

Samples	CART	Boosting	Bagging
1	0.36	0.26	0.24
2	0.32	0.14	0.24
3	0.36	0.26	0.30
4	0.24	0.24	0.24
5	0.34	0.22	0.26
6	0.28	0.22	0.24
7	0.30	0.15	0.22
8	0.22	0.20	0.24
9	0.36	0.20	0.20
10	0.28	0.16	0.26

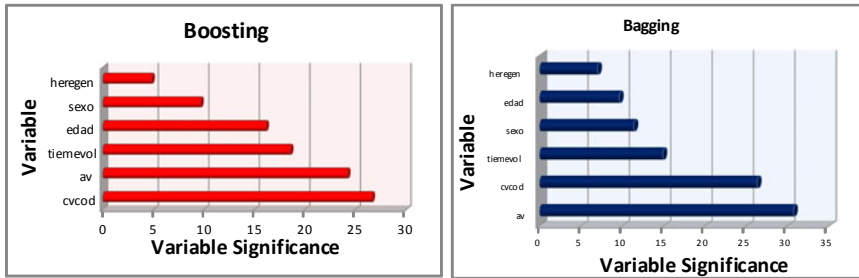


Fig. 2. a) Boosting variable significance; b) Bagging variable significance

5 Conclusions

In this work we have study CART, Boosting and Bagging methods applied to the classification of ERG for a early Pigmentary Retinosis diagnosis . Our results show that the methods Boosting and Bagging require a computational effort greater than CART method, but they show to be more robust than the last one. Although CART produces the worst performance taking account classification errors, from a practical point of view, it can be interesting by offering a tree that allows visualizing the decision rules to establish early diagnosis.

The most significant variables to Retinosis diagnostics using ERG are visual acuity and visual field, which allow to forecast the kind of dystrophy on the patient with a classification error percentage of the 22% that can be improved by the specialists applying the boosting with a classification error of the 14%, case in which would need to implement the procedure to realize the predictions of this method. It is especially relevant the low significance of the inheritance.

Currently, we are improving the classification process by applying a cross validation based learning and testing other classification methods.

Acknowledgements. This work has been partially supported by AECID (Agencia Española de Cooperación Internacional para el Desarrollo), Project D/023835/09 and D/030223/10. Thanks to Dr. Alberto Barrientos of the Pigmentary Retinosis Center in Cuba to provide data and suggestions.

References

1. Alfaro, E., Gámez, M., García, N.: Multiclass corporate failure prediction by Adaboost. *M1. International Advances in Economic Research* 13(3), 301–312 (2007)
2. Breiman, L.: Bagging predictors. *Machine Learning* 24(2), 123–140 (1996)
3. Breiman, L.: Arcing classifiers. *The Annals of Statistics* 26(3), 801–849 (1998)
4. Breiman, L., Friedman, J., Olshen, R., Stone, C.: *Classification and regression trees*. Wadsworth, Belmont (1984)
5. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: *Proceedings of the 13th Int. Conf. on Machine Learning*, pp. 148–156. Morgan Kaufmann, San Francisco (1996)
6. Hastie, T., Tibshirani, R., Friedman, J.: *The elements of statistical learning: Data mining, inference and Prediction*. Springer, Heidelberg (2005)
7. Kuncheva, L.I.: *Combining Pattern Classifiers Methods and Algorithms*. John Wiley & Sons, Chichester (2004)
8. Pagon R.A., Daiger S.P.: *Retinitis Pigmentosa. A Overview* (2005), <http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=gene&part=rp-overview>
9. Peláez, O.: *Retinosis Pigmentaria: Experiencia Cubana*, pp. 25-30, Ed. Científico-Técnica, La Habana, Cuba (1997)
10. R Development Core Team R: *A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria (2010), <http://www.R-project.org/> ISBN 3-900051-07-0
11. Yohannes, Y., Hoddinott, J.: *Classification and Regression Trees: an introduction*. Technical guide #3!International Food Policy Research Institute 2033 K Street, N.W.Washington, D.C. 20006 U.S.A (1999)
12. Venables, W.N., Smith, D.M.: *the R Development Core Team Notes on R: A Programming Environment for Data Analysis and Graphics Version 2.10.1* (2009)

Audio-Cued SMR Brain-Computer Interface to Drive a Virtual Wheelchair

Francisco Velasco-Álvarez¹, Ricardo Ron-Angevin¹, Leandro da Silva-Sauer¹, Salvador Sancha-Ros¹, and María José Blanca-Mena²

¹ Dpto. Tecnología Electrónica, E.T.S.I. Telecomunicación, Universidad de Málaga
{fvelasco, rra, sauer, ssancha}@dte.uma.es

² Dpto. Psicobiología y Metodología de las Ciencias del Comportamiento, Facultad de Psicología, Universidad de Málaga
blamen@uma.es

Abstract. In this work, an electroencephalographic analysis-based, self-paced (asynchronous) brain-computer interface (BCI) is proposed to control a virtual wheelchair using three different navigation commands: turn right, turn left and move forward. In order to reduce the probability of misclassification, the BCI is to be controlled with only two mental tasks (relaxed state versus imagination of right hand movements) using an audio-cued interface. Six healthy subjects participated in the experiment. After two training sessions controlling a wheelchair in a virtual environment using both a visual and auditory interface, all subjects successfully controlled the wheelchair in the last session, where the interface was only auditory. The obtained results support the use of the proposed interface to control a real wheelchair without the need of a screen to provide visual stimuli or feedback.

Keywords: Brain-computer interface (BCI), virtual environment (VE), navigation, asynchronous, motor imagery (MI), mental tasks, auditory.

1 Introduction

A Brain-computer interface (BCI) is a system that enables a communication that is not based on muscular movements but on brain activity. This activity can be measured through electroencephalographic (EEG) signals. Several EEG signals can be detected, resulting in different types of BCI. Sensorimotor rhythm-based BCIs (SMR-BCI) are based on the changes of μ and β rhythms [1]. These rhythms correspond to specific features of the EEG signals characterized by their frequencies that can be modified by voluntary thoughts. When a person performs a movement, it causes a synchronization/desynchronization in this activity (event-related synchronization/desynchronization, ERS/ERD) which involves a rhythm amplitude change. However, the outstanding property of these signals is their behaviour when someone merely imagines movements (motor imagery, MI), because this causes a similar amplitude change [2]. This relevant characteristic is what makes SMR suitable to be used as input for a BCI.

Our work is focused on the control of a wheelchair. Before people can use it in a real situation, it is necessary to guarantee enough control to avoid dangerous scenarios. Virtual reality (VR) is a suitable tool to provide subjects with the opportunity to train and test the application. Among those BCI systems aimed at navigation, several of them can be found in which the subject moves in only one direction ([3, 4]), so the user needs only one control command. Some systems offer the user more ease of use, by means of more commands. In [5], a simulated robot performs two actions ('turn left then move forward' or 'turn right then move forward') in response to left or right hand MI. A more versatile application can be found in [6] with three possible commands (turn left, turn right, and move forward) selected with three MI tasks. Having a higher number of commands makes it easier to control the virtual wheelchair, since the subject has more choices to move freely (by means of an information transfer rate increase). Some studies proved that the best classification accuracy is achieved when only two classes are discriminated ([7, 8]). As the systems mentioned above match the number of commands to the number of mental tasks, none of them fulfills at the same time two basic characteristics in this kind of systems: free mobility and safety (a classification error corresponds to a wrong command, which can cause dangerous situations in the real world). In order to provide different commands without making the BCI performance worse, a new paradigm, based on discrimination of only two classes (one active mental task versus any other mental activity), which enabled the selection of four navigation commands, was proposed in [9]. The mapping of two states into four commands was achieved with a bar that turned in a circle and pointed to the various commands placed around it. When subjects wanted to select the command the bar was pointing to, they carried out one MI task to extend the bar. The other mental state was relaxed state, which made the bar keep turning. An improvement of this interface was proposed in [10], providing an asynchronous BCI (where the subject controls the timing of the interaction) allowing subjects to freely navigate in a VE. In these previous works, a graphical interface with a feedback bar was shown to the subject; however, this could be a problem if the system is conceived to finally control a real wheelchair. In this way, a laptop coupled to the wheelchair has been occasionally used ([11, 12]) to provide visual feedback or stimuli (e.g. in a system based on Steady-State Visual Evoked Potentials, or in a visual oddball paradigm). Nevertheless, this solution may not be adequate to control a wheelchair, as it could limit the subject's field of view and, at the same time, distract him from the task of controlling the wheelchair for having to look at the computer screen. If a BCI system is to be proposed that allows a subject to control a wheelchair, it should let the user watch the environment at all times. This study is a continuation of the work in [10], in which the visual paradigm was intended to be replaced by an auditory one, thus avoiding the need for the subject to look at the screen in order to execute the navigation commands. In fact, in the graphical interface proposed in [10], the visual feedback is not necessary, as the only essential information that subjects need to receive is the cue that indicates which command is being pointed by the bar. The experiment herein presented establishes a comparison between the use of a graphical interface, and an interface only guided by auditory cues. It is worth mentioning different meanings of the word "auditory" with

reference to a BCI: most of the works use audio stimuli to elicit event-related potentials [13-18]; others use the auditory stimuli for feedback or as cues for task presentation when subjects have to self-regulate the mental task (by means of slow cortical potentials, [19] or SMR, [20]). The use in the paradigm proposed herein suits the latter case: subjects hear an audio cue which signals them which navigation command can be selected, so they decide whether to carry out the MI task to select it, or to wait for the next command. Regarding the feedback, the actual movement of the virtual wheelchair represents how subjects are performing in the control of their SMR.

2 Methods

2.1 Subjects and Data Acquisition

Six naïve and healthy subjects participated in the study, with ages (mean \pm SD) 21.8 ± 1.7 years. They were selected from a higher number of volunteers according to their error rate obtained in a previous session without feedback (see next section). The EEG was recorded from two bipolar channels. The active electrodes were placed 2.5cm anterior and posterior to electrode positions C3 and C4 (right and left hand sensorimotor areas, respectively) according to the 10/20 international system. The ground electrode was placed at the FPz position. Signals were amplified by a 16 channel biosignal g.BSamp (Guger Technologies) amplifier and then digitized at 128 Hz by a 12-bit resolution data acquisition NI USB-6210 (National Instruments) card.

2.2 Initial Training and Signal Processing

Before using the system to test the two paradigms, subjects had to follow an initial training that consisted of two sessions: a first one without feedback and a second one providing continuous feedback. Those subjects who obtained a low error rate in the first session continued with the experiment. These two training sessions were used for calibration purposes. The training was the same as the one used in [21] and was based on the paradigm proposed by the Graz group [3]. It consisted of a virtual car that dodged a water puddle in the road, by moving left or right according to the mental task carried out (relaxed state or right hand MI), see Fig. 1. The training protocol consisted of two sessions: a first one without feedback and the other one providing continuous feedback. An offline processing of the first session [21] made it possible to determine the parameters for the feedback session. The same parameters were used to calibrate the system for the virtual environment (VE) exploration sessions. This processing is based on the procedure detailed in [23], and consisted of estimating the average band power of each channel in predefined, subject-specific reactive frequency (manually selected) bands at intervals of 500 ms. The obtained reactive frequency bands (Mean \pm SD) were centred at 11 ± 0.77 Hz; the average band width was 3.6 ± 0.81 Hz. In the feedback session, the movement of the car was computed on-line every 31.25 ms as a result of a Linear Discriminant Analysis (LDA) classification. The trial paradigm and all the algorithms used in the signal processing were implemented in MATLAB.

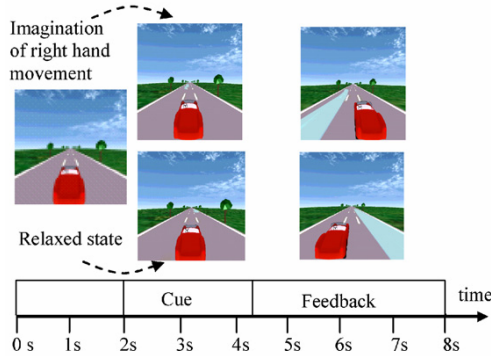


Fig. 1. Timing of one trial of the training with feedback

2.3 Navigation Paradigm

The main objective of our research is to provide an asynchronous BCI system which, by the discrimination of only two mental states, offers the user several navigation commands to be used in a VE. An asynchronous (or self-paced) system must produce outputs in response to intentional control as well as support periods of no control [24]; those are the so-called intentional control (IC) and non-control (NC) states, respectively. Both states are supported in this study: the system waits in a NC state in which an NC interface is shown (Fig. 2a). The NC interface enables subjects to remain in the NC state (not generating any command) until they decide to change to the IC state, where the control is achieved through the IC interface (Fig. 2b).

The NC interface consists of a semi-transparent vertical blue bar placed in the centre of the screen. The bar length is computed every 62.5 ms as a result of the LDA classification: if the classifier determines that the mental task is right hand MI, the bar extends; otherwise, the bar length remains at its minimum size. In order to switch from the NC to the IC state, the subject must accumulate more than a “selection time” with the bar over the “selection threshold”. If the length is temporarily (less than a “reset time”) lower than the selection threshold, the accumulated selection time is not reset, but otherwise it is set to zero.

The IC interface is similar to the one presented in [9]: a circle divided into three parts, which correspond to the possible navigation commands (move forward, turn right and turn left), with a bar placed in the centre of the circle that is continuously rotating clockwise. The subject can extend the bar carrying out the MI task to select a command when the bar is pointing at it. The way the selection works in this interface is the same as in the NC interface, with the same selection and reset time and the same selection threshold. In the IC interface, another threshold is defined (stop threshold, which is lower than the selection threshold, and not visible to the subject). When it is exceeded, the bar stops its rotation in order to help the subject in the command selection. Subjects receive audio cues while they interact with the system.

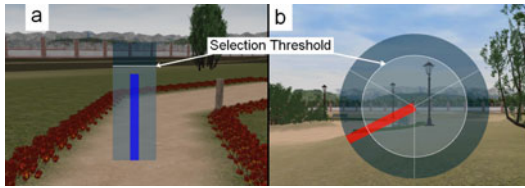


Fig. 2. NC interface (a) and IC interface (b)

When the state changes from IC to NC they hear the Spanish word for ‘wait’; the reverse change is indicated with ‘forward’, since it is the first available command in the IC state. Finally, every time the bar points to a different command, they can hear the corresponding word (‘forward’, ‘right’ or ‘left’).

Once a command is selected, the virtual wheelchair starts moving forward or turning left or right at a fixed speed and the bar (that has already stopped, since it exceeds the stop threshold) changes its color to red. The movement is maintained as long as the bar length is above the selection threshold (this means that the subject is still carrying out the MI mental task). If the bar is temporarily below this threshold (less time than the reset time), the movement stops, but the system allows the subject to continue the same movement if the bar again exceeds the selection threshold. While it happens, the bar maintains its red color to indicate this possibility to the subject. In the case that the bar remains under the selection threshold longer than the reset time, the bar changes its color to blue and continues rotating (if it is under the stop threshold) so that the subject can select a command again. The position of the rotating bar does not change; it takes its rotation up again from the same point at which it last stopped to select a command. This way, the subject can select the same command several times in a row, in case the reset time passes without the subject wanting to stop the movement. From the results obtained with this mode of operation ([10]), it can be deduced that the paradigm could be used for training and, eventually, to control a real wheelchair.

Auditory Interface. In the last session of the experiment, after subjects had got used to the visual interface and the simultaneous audio cues, the graphics were removed so subjects could only see the VE. This final test is the main objective of this paper.

2.4 Experimental Procedure

The experiments carried out by the subjects consisted of three runs in which they had to follow a prefixed path to reach, as fast as they could, an avatar placed at the end of it. This path was located in a 3D virtual park. The first two sessions were carried out with the visual interface, and the last one, without it. The first one was considered an adaptation session; so in the Results section we will include a comparison between the second and the third one. If the movement led the subjects out of this path, the wheelchair collided with an invisible wall, so the movement finished. After a collision happened, the bar continued its rotation from the point where it stopped, as in the case when a subject finished the selection of a command. The path consisted of five sections

with a total length of 38 m. These sections were placed so that subjects had to achieve at least two turns of approximately 45° in each direction. The dimensions of the virtual wheelchair were 1.17 x 0.75 m. Subjects were looking at a large stereoscopic screen (2 x 1.5 m) placed at a distance of 3 m, wearing polarized glasses and earphones.

3 Results

Even though visual mode has already been tested in a previous work ([10]), in this study it was necessary to include one session in order to establish a case study (where each subject would pass through different experimental situations). Furthermore, in this experiment there are important differences: that experiment was carried out in a different VE (the new one is more immersive due to the high realism and the use of stereoscopic vision) and with a different path to follow.

In order to establish a comparison between the performances with both paradigms, a Wilcoxon test was applied to the recorded data.

Table 1. Results for each subject and interface: Time needed to complete the path, number of collisions and number of commands of each type used (Forward, Right, Left and Total)

Subject	Interface	Time(s)	Coll	F	R	L	Total
1	Visual+Auditory	551	8	19	17	18	54
	Auditory	647	6	20	32	18	70
3	Visual+Auditory	381	9	15	12	14	41
	Auditory	521	9	20	19	14	53
4	Visual+Auditory	249	2	7	7	2	16
	Auditory	658	16	22	19	16	57
5	Visual+Auditory	749	19	26	25	22	73
	Auditory	486	13	18	13	15	46
7	Visual+Auditory	686	21	29	21	19	69
	Auditory	441	10	18	14	11	43
8	Visual+Auditory	544	8	17	25	17	59
	Auditory	775	14	24	35	33	92

Table 2. For each parameter and interface: Mean (M), standard deviation (SD), sum of negative rank (SNR), sum of positive rank (SPR), Wilcoxon T-statistic (T) and statistical significance (p)

Parameter	Interface	M(s)	SD(s)	SNR	SPR	T	p
Time	Visual+Auditory	526.67	186.56	9	12	9	.42
	Auditory	588	126.37				
Collisions	Visual+Auditory	11.17	7.31	7.5	7.75	7.5	.56
	Auditory	11.33	3.67				
Forward	Visual+Auditory	18.83	7.91	9	12	9	.42
	Auditory	20.33	2.34				
Right	Visual+Auditory	17.83	7.28	6	15	6	.23
	Auditory	22.00	9.30				
Left	Visual+Auditory	15.33	7.03	3	7	3	.31
	Auditory	17.83	7.78				
Total	Visual+Auditory	52.00	20.98	7	14	7	.28
	Auditory	60.17	18.26				

The obtained results show that there are not significant differences between the two interfaces for any of the parameters ($p > 0.05$); in other words, the use of the auditory interface does not make the results worse, so both are equally effective.

4 Discussion and Conclusion

The herein proposed training process, associating the visual and auditory interfaces, has proven valid for a future auditory-only interaction. As quoted by [20], a subject in an initial experimental situation with the visual interface associated to the auditory one shows fast acquisition of the interaction commands. Here, learning consists of internalizing the image and its timing in the previous sessions in order to subsequently respond only to the auditory stimulus. This statement is also supported by a self-report questionnaire applied after the execution of the experiment, in which all the subjects declared they had managed to associate the auditory stimulus to the mental task they intended to execute. They also stated that such stimulus did not become a source of distraction that could lead to a mistake.

Another noticed factor is that typical deviations are relatively high in all the variables, which indicates that each subject has adapted himself in a particular manner to each interface. On the other hand, no direct relationship is noticed between each type of interface and the execution of the path (number of commands, collisions and total time), which leads to deducing that none of the interfaces is intrinsically better than the other to interact with the system.

The objective of the research was to confirm the utility of an exclusively auditory interface, as it would make interaction of a handicapped person in a wheelchair easier. Success in the execution of the path with no significant differences in the measures of the evaluated variables in both modes of interaction shows the effectiveness of the proposed method. We thus conclude that both types of interface are equally effective for communication with a BCI system.

Acknowledgments. This work was partially supported by the Innovation, Science and Enterprise Council of the Junta de Andalucía (Spain), project P07-TIC-03310.

References

1. Kübler, A., Müller, K.R.: An introduction to brain-computer interfacing. In: Dornhege, G., Millán, J., de, R., Hinterberger, T., et al. (eds.) *Toward Brain-Computer Interfacing*, pp. 1–25. MIT Press, Cambridge (2007)
2. Neuper, C., Pfurtscheller, G.: Motor imagery and ERD. In: Pfurtscheller, G., Lopes da Silva, F.H. (eds.) *Event-related desynchronization. Handbook of Electroencephalography and Clinical Neurophysiology. Revised Series*, vol. 6, pp. 303–325. Elsevier, Amsterdam (1999)
3. Leeb, R., Settgast, V., Fellner, D., et al.: Self-Paced Exploration of the Austrian National Library through Thought. *International Journal of Bioelectromagnetism* 9, 237–244 (2007)
4. Leeb, R., Friedman, D., Müller-Putz, G.R., et al.: Self-Paced (Asynchronous) BCI Control of a Wheelchair in Virtual Environments: A Case Study with a Tetraplegic. *Computational Intelligence and Neuroscience* (2007)
5. Tsui, C.S.L., Gan, J.Q.: Asynchronous BCI Control of a Robot Simulator with Supervised Online Training. In: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. (eds.) *IDEAL 2007. LNCS*, vol. 4881, pp. 125–134. Springer, Heidelberg (2007)

6. Scherer, R., Lee, F., Schlögl, A., et al.: Toward Self-Paced Brain-Computer Communication: Navigation through Virtual Worlds. *IEEE Transactions on Biomedical Engineering* 55, 675–682 (2008)
7. Kronegg, J., Chanel, G., Voloshynovskiy, S., et al.: EEG-Based Synchronized Brain-Computer Interfaces: A Model for Optimizing the Number of Mental Tasks. *IEEE transactions on neural systems and rehabilitation engineering* 15, 50–58 (2007)
8. Obermaier, B., Neuper, C., Guger, C., et al.: Information Transfer Rate in a Five-Classes Brain-Computer Interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 9, 283–288 (2001)
9. Ron-Angevin, R., Díaz-Estrella, A., Velasco-Álvarez, F.: A Two-Class Brain Computer Interface to Freely Navigate through Virtual Worlds. *Biomed. Tech.* 54, 126–133 (2009)
10. Velasco-Álvarez, F., Ron-Angevin, R., Blanca-Mena, M.J.: Free Virtual Navigation using Motor Imagery through an Asynchronous Brain-Computer Interface. *Presence: Teleoperators and Virtual Environments* 19, 71–81 (2010)
11. Iturrate, I., Antelis, J.M., Kübler, A., et al.: A Noninvasive Brain-Actuated Wheelchair Based on a P300 Neurophysiological Protocol and Automated Navigation. *IEEE Transactions on Robotics* 25, 614–627 (2009)
12. Rebsamen, B., Guan, C., Zhang, H., et al.: A Brain Controlled Wheelchair to Navigate in Familiar Environments. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 18, 590–598 (2010)
13. Furdea, A., Halder, S., Krusienski, D.J., et al.: An Auditory Oddball (P300) Spelling System for Brain-Computer Interfaces. *Psychophysiology* 46, 617–625 (2009)
14. Halder, S., Rea, M., Andreoni, R., et al.: An Auditory Oddball Brain-Computer Interface for Binary Choices. *Clinical Neurophysiology* 121, 516–523 (2010)
15. Hill, N.J., Lal, T.N., Bierig, K., et al.: An Auditory Paradigm for Brain-Computer Interfaces. *Advances in Neural Information Processing Systems* 17, 569–576 (2005)
16. Kanoh, S., Miyamoto, K.: A Brain-Computer Interface (BCI) System Based on Auditory Stream Segregation. In: *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., EMBC - Pers. Healthc. through Technol.*, pp. 642–645 (2008)
17. Klobassa, D.S., Vaughan, T.M., Brunner, P., et al.: Toward a High-Throughput Auditory P300-Based Brain-Computer Interface. *Clinical Neurophysiology* 120, 1252–1261 (2009)
18. Sellers, E.W., Donchin, E.: A P300-Based Brain-Computer Interface: Initial Tests by ALS Patients. *Clinical Neurophysiology* 117, 538–548 (2006)
19. Pham, M., Hinterberger, T., Neumann, N., et al.: An Auditory Brain-Computer Interface Based on the Self-Regulation of Slow Cortical Potentials. *Neurorehabil. Neural Repair* 19, 206–218 (2005)
20. Nijboer, F., Furdea, A., Gunst, I., et al.: An Auditory Brain-Computer Interface (BCI). *J. Neurosci. Methods* 167, 43–50 (2008)
21. Ron-Angevin, R., Díaz-Estrella, A.: Brain-Computer Interface: Changes in Performance using Virtual Reality Techniques. *Neurosci. Lett.* 449, 123–127 (2009)
22. Guger, C., Schlögl, A., Neuper, C., et al.: Rapid Prototyping of an EEG-Based Brain-Computer Interface (BCI). *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 9, 49–58 (2001)
23. Guger, C., Edlinger, G., Harkam, W., et al.: How Many People are Able to Operate an EEG-Based Brain-Computer Interface (BCI)? *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11, 145–147 (2003)
24. Schlögl, A., Kronegg, J., Huggins, J.E., et al.: Evaluation Criteria for BCI Research. In: Dornhege, G., Millán, J., de Hinterberger, J.R., et al. (eds.) *Toward Brain-Computer Interfacing*, pp. 327–342. The MIT Press, Cambridge (2007)

A Domotic Control System Using Brain-Computer Interface (BCI)

Rebeca Corralejo, Roberto Hornero, and Daniel Álvarez

Grupo de Ingeniería Biomédica, E.T.S.I. de Telecomunicación, Universidad de Valladolid,
Paseo Belén 15, 47011 Valladolid, Spain
rebeca.corralejo@uva.es, robhor@tel.uva.es,
dalvgon@ribera.tel.uva.es

Abstract. The aim of this study was to develop a Brain Computer Interface (BCI) application to control domotic devices usually present at home. Electroencephalographic (EEG) activity was recorded from users' scalp and sensorimotor rhythms were used to control the BCI. Our application uses the BCI2000 general purpose system. We studied four feature extraction algorithms: AR model, Fourier transform, wavelet transform and matched filter. An AR model achieved the best accuracy. Our application allows to control several devices, such as a TV set, a Hi-Fi system, a DVD player and the lights of a room. Five users from a disability reference center in León (Spain) with severe physical and cognitive disabilities took part in the study, achieving accuracies up to 69.9%. Domotic BCI applications could be really useful for disabled people. However, further research is needed.

Keywords: Brain Computer Interface (BCI), electroencephalogram (EEG), feature extraction, feature translation, real-time applications, disabilities.

1 Introduction

A Brain-Computer Interface (BCI) is a communication system that monitors the brain activity and translates specific signal features that reflect the user's intent into commands that operate a device [1]. The method most commonly used for monitoring brain activity in BCIs is the electroencephalography (EEG). The EEG is a non-invasive method that requires relatively simple and inexpensive equipment and it is easier to use than other methods [2].

BCIs can be classified into two groups according to the nature of the input signals. Endogenous BCIs depend on the user's control of endogenic electrophysiological activity, such as amplitude in a specific frequency band of EEG recorded over a specific cortical area [2]. BCIs based on sensorimotor rhythms or slow cortical potentials (SCP) are endogenous systems and often require extensive training. Other systems depend on exogenous electrophysiological activity evoked by specific stimuli and they don't require extensive training [2]. BCIs based on P300 potentials or visual evoked potentials (VEP) are exogenous systems.

The aim of the present pilot study consisted in the development of a BCI application for disabled people to interact with the devices usually present at home.

The designed BCI application uses the μ (8-12 Hz) and β (16-24 Hz) sensorimotor rhythms. The mentioned rhythms present variations in EEG over motor cortex when a self-generated movement is carried out and also when a subject observes the movement or imagines making the same [3]. Usually, to control these BCIs, users have to imagine different motor tasks. In this study, we used a paradigm consisted of two classes: motor imagery of left and right hand. For right hand motor imagery, the μ rhythm amplitude is attenuated over the left hemisphere. For left hand motor imagery, it is attenuated over the right hemisphere. This attenuation is known as Event Related Desynchronization (ERD) and it is prominent contralateral to the intended movement [4]. After signal acquisition, several signal processing methods are applied to determine the user's intent. Firstly, feature extraction methods obtain specific signal features. Then, selection methods choose the most significant ones that encode the user's purpose. Finally, classifiers translate them into device commands.

This study analysed four feature extraction algorithms: autoregressive (AR) model, short-term Fourier transform (STFT), continuous wavelet transform (CWT) using complex Morlet wavelets and μ rhythm-matched filter. Previous studies showed these algorithms could be useful in BCI applications [4], [5], [6]. As classification method we used a linear classifier, described in [7]. This algorithm is included in the BCI2000 general purpose system [7] that we used to develop the domotic application. Our application exploits the user's ability for moving a cursor in 1D by means of two classes of motor imagery to navigate through a menu that access to several devices. Specifically, the application controls a television set, a DVD player, a mini Hi-Fi system and the lights of a room. Five users from a dependence and disability reference center in León (Spain) tested and evaluated our BCI application. These users are people with severe disabilities, so they are potential users of BCI systems.

This communication is organized as follows: Section 2 introduces subjects and EEG recording details. In Section 3, the proposed feature extraction algorithms are described. Section 4 details the designed BCI application and in Section 5, results are presented. Finally, in Section 6, our results are discussed and conclusions are drawn.

2 Data Recording and Subjects

We used a g.USBamp biosignal amplifier (g.tec, Austria) with 16 channels to record EEG signals. Eight channels were recorded monopolarly with the left ear serving as reference and the right ear as ground. Signals were sampled at 128 Hz and bandpass-filtered between 2 and 60 Hz. Three main channels were located over the motor cortex (C3, Cz, C4) and the other five ones over the front, temporal and parietal areas (F3, F4, T7, T8, Pz) to make a Laplacian spatial filtering of the main channels.

The performance of four feature extraction algorithms was evaluated by a trained user. Signals were recorded from a healthy subject (female, 26 years) during feedback sessions. Every single trial started with a bar at the left or right of the screen, based on the CursorTask block from BCI2000. The task consisted in moving a cursor through motor imagery (left or right hand movements) to get the target showed in the screen. If subject got the target, that trial counted as a hit, otherwise as a miss. The experiment consisted of 10 runs with 20 trials each (100 trials per class) for each feature extraction algorithm. Accuracy was calculated as the percentage of hits to the

sum of hits and misses. The algorithm that achieved the best accuracy was included in the domotic application.

The performance of the application was evaluated by users from a disability reference center. Five subjects (2 males and 3 females from 40 to 57 years) with severe disabilities, both physical and cognitive ones, were included in the study. They had different pathologies as Arnold-Chiari malformation, ataxia, cerebral palsy or brain injury with tetraparesis. The experiment had three stages: learning, training and application. During the learning stage, a word in the screen indicated the proposed task (motor imagery of left or right hand). Subsequently, an offline analysis was developed to find out the specific frequency bands of sensorimotor rhythms for each subject. In the training stage, subjects had to move a cursor horizontally to get a bar through motor imagery. Finally, in the application stage, three simple control sequences were proposed to assess our application. They are summarized in Table 1. Trials' length was variable from 8 to 12 s (feedback period variable from 4 to 8 s).

3 Methodology

3.1 Autoregressive Model (AR)

In an autoregressive model of order p , each time point of a time series is represented as a fixed linear combination (AR coefficients) of the last p data points [8]:

$$c(t) = \sum_{i=1}^p k_i c(t-i) + e(t) \cdot \tag{1}$$

We estimated AR coefficients using the Burg algorithm [9]. Model order was set to 16 and window length to 0.5 s [5]. Two configurations were used:

AR1. AR spectrum estimated in bands with a width of 3 Hz, centered between 0 and 30 Hz. The used feature is the corresponding to the specific μ rhythm of each subject.

AR2. AR spectrum estimated in bands with a width of 1 Hz, centered between 0 and 30 Hz. The used features are the specific to μ and β rhythms of each subject.

Table 1. Proposed sequences to evaluate the BCI application by potential users: A) Hi-Fi system, B) TV and C) Lights

<i>Seq. A:</i> Switch on Hi-Fi system and select radio		<i>Seq. B:</i> Switch on TV, modify channel and volume		<i>Seq. C:</i> Switch on lights and change its color	
A.1	Select Music Menu	B.1	Access TV Menu	C.1	Select Music Menu
A.2	Perform selected action	B.2	Switch on the TV	C.2	Select DVD Menu
A.3	Switch on Hi-Fi system	B.3	Select Channel Up	C.3	Select Lights Menu
A.4	Select Radio Function	B.4	Select Channel Down	C.4	Perform selected action
A.5	Perform selected action	B.5	Perform selected action	C.5	Switch on the lights
		B.6	Select Volume Up	C.6	Select Change Color
		B.7	Perform selected action	C.7	Perform selected action

3.2 Short-Term Fourier Transform (STFT)

The Short-Term Fourier Transform (STFT) is used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time:

$$STFT\{\mathbf{x}[n]\} \equiv X(m, \omega) = \sum_{n=-\infty}^{\infty} \mathbf{x}[n] \mathbf{w}[n-m] e^{-j\omega n}, \quad (2)$$

where $\mathbf{x}[n]$ is the signal of interest (EEG) and $\mathbf{w}[n]$ a window function (Hamming, 1 s) [5], [10]. Thus, each sample of the STFT covers a spectral width of 1 Hz. Frequency bands power related to the specific μ and β rhythms were our features.

3.3 Continuous Wavelet Transform (CWT) Using Complex Morlet Wavelets

This algorithm is based on the winner algorithm from BCI Competition 2003 for Data Set III. It consists in filtering EEG data with complex Morlet wavelets [4]. These specific wavelets take the form of modulated Gauss impulses with a characteristic eigenfrequency ω_0 . To localize them in the time and frequency domain, they have to be scaled (scaling factor s) and temporally shifted (according to τ):

$$\Psi_{\tau, s(f)}(t) = \frac{1}{\sqrt{s(f)}} \pi^{-\frac{1}{4}} e^{(i\omega_0 \frac{t-\tau}{s(f)})} e^{-\frac{1}{2}(\frac{t-\tau}{s(f)})^2}. \quad (3)$$

We calculated CWT instantaneous amplitude for EEG segments (0.5 s) using two complex Morlet wavelets centered to the specific μ and β rhythms of each subject.

3.4 μ Rhythm-Matched Filter (MF)

This algorithm creates a parametric model for the μ rhythm that is evident in the scalp recorded EEG of most of healthy adults [6], [11]. Firstly, the fundamental frequency f_F of the characteristic μ rhythm was determined. Then, it was decomposed in terms of a discrete number of phased-coupled sinusoidal components [11]. We calculated the amplitudes and phases of the fundamental peak and the two main harmonics (A_m , ϕ_m). The MF was modeled as the sum of the three first harmonics present in the real rhythm. To avoid jitter effect during real-time processing, we used a complex MF: the real part using cosine functions and the complex one using sine functions:

$$MF_{real}(n) = \sum_{m=1}^3 A_m \cos\left(\frac{2\pi n m f_F}{f_S} + \phi_m\right), \quad (4)$$

$$MF_{imag}(n) = \sum_{m=1}^3 A_m \sin\left(\frac{2\pi n m f_F}{f_S} + \phi_m\right). \quad (5)$$

Thus, a template of 0.25 s length were obtained [5]. Incoming EEG data segments were circularly convolved with the template and the maximum was used as feature.

4 Designed Domotic BCI Application

The proposed BCI application exploits the user's ability of moving a cursor in 1D by means of two classes of motor imagery, in order to navigate through a menu which allows to control several devices. Fig. 1 shows the designed application. Its interface is divided in two parts. On the left frame there are two bars, a red one on the left and a blue one on the right, and a green circle cursor. The user, by means of motor imagery of right and left hand, moves the cursor horizontally to get the desired bar. The blue bar allows to navigate through the application menu, showed in the right frame, whereas the red one is used to perform the action that is selected in the menu at that moment. As mentioned before, the application menu appears on the right frame. It consists of four buttons that allow to access the submenus to control the TV set, the DVD player, the Hi-Fi system or the lights of the room. When a user accesses to a submenu, several buttons appear showing different actions over a specific device. Fig. 2 shows two examples. Navigation through different menus is done in a circular way turning clockwise. To operate the devices, i.e. to perform the commands, our application uses an infrared (IR) emitter device connected to the PC. The IR emitter has the ability of learning signals from remote controls, so it is possible to replicate and use them to control any IR device.

5 Results

5.1 Feature Extraction Algorithms

A simple 1D movement control application was used to evaluate the accuracy obtained with each feature extraction algorithm. The results are summarized in Table 2. The best accuracy was achieved by an AR model with the configuration AR1. This algorithm was included as feature extraction algorithm in the domotic application.



Fig. 1. Interface of the designed BCI application for controlling domotic and electronic devices

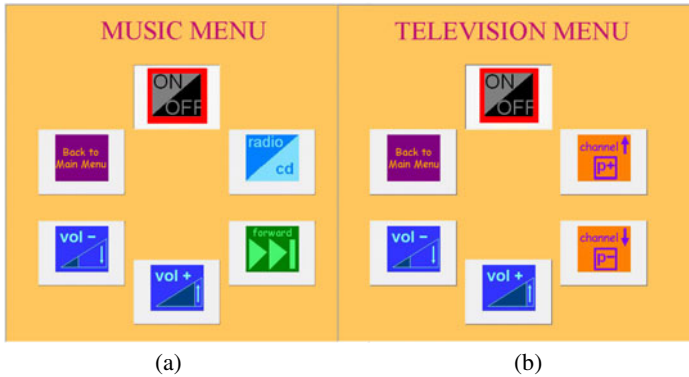


Fig. 2. Two submenus of our application to control: (a) the Hi-Fi system, (b) the TV set.

5.2 Evaluation of the BCI Application by Potential Users

Although five users took part in the experiment, only two of them (users 1 and 3) were able to control suitably a cursor in 1D. Only these users reached the third stage and used the BCI application. Regarding the remaining users (users 2, 4 and 5), one of them left the experiment during the training stage. The other two users reached the second stage but they could not control the cursor movement with enough accuracy after two weeks of training.

Our results are summarized in Table 3. It includes accuracy, percentage of sequences totally completed and the average of necessary steps to complete a sequence regarding to the necessary minimum steps, i.e. without misses. We considered a miss when the user selected the running bar (red) instead of the navigation bar (blue) so the application performed a non-required command. However, when the user selected the navigation bar instead of the running one, i.e. he skipped the desired command and selected the next one, we did not consider that as a miss. The user had to select the next buttons until he reached the desire one again. However, that caused the user needed more steps to complete the proposed sequence of commands. Because of that reason, sometimes users needed more steps to complete a sequence than the minimum (5 or 7 steps).

Accuracy results were worse than expected for both users. They were able to complete totally the sequence in many cases but usually they needed more steps than the strictly necessities. User 1 had good accuracy results for sequence A (69.85%), completing it 47 times out of 100 and achieving an Information Transfer Rate (ITR) [12] of 7.38 bit/min. For sequences B and C, accuracies were lower respect to sequence A and the user completed them 26 and 33 times out of 100, respectively. Probably, it occurred because sequence A required fewer steps than the other two, so it was easier for the users. As expected, the average of needed steps increased with the sequence's length. User 3 achieved high accuracy for sequence B (66.78%), completing it 31 times out of 100 and obtaining an ITR of 5.66 bit/min. However, user 3 was not able to complete correctly sequence C, finishing it always at intermediate points.

Table 2. Accuracy of proposed feature extraction algorithms achieved for a healthy subject

	Accuracy (%)		
	Average	Maximum	Minimum
AR1	92.08	100.00	83.33
AR2	90.18	100.00	72.22
STFT	83.10	88.89	73.68
CWT	82.50	94.44	63.64
MF	81.07	93.75	58.82

Table 3. Results achieved for users 1 and 3 using the domotic application: accuracy, percentage of sequences totally completed and the average of necessary steps to complete a sequence regarding to the necessary minimum steps (5 for seq. A, 7 for seq. B and C)

		Accuracy (%)	Sequences completed (%)	Average steps per completed sequence
<i>User 1</i>	Sequence A	69.85	47.00	6.00 (5.00)
	Sequence B	47.96	26.00	8.29 (7.00)
	Sequence C	47.93	33.00	9.14 (7.00)
<i>User 3</i>	Sequence A	47.13	27.00	10.67 (5.00)
	Sequence B	66.78	31.00	9.20 (7.00)
	Sequence C	51.42	0.00	0.00 (7.00)

6 Discussion and Conclusions

The performance of four extraction features algorithms was analyzed [4], [5], [6]. The AR models achieved the highest accuracies. The AR1 configuration with a frequency band width of 3 Hz and centered on the μ rhythm achieved the highest accuracy: 92.08%. Our results suggest these models could be the most adequate to extract features in real-time BCI systems. Thus, an AR model with the AR1 configuration was used as feature extraction method in the designed BCI application.

On the other hand, we developed a real-time BCI application. The main purpose of this application was to control different devices present at home through the EEG signal. Therefore, this application could be really useful for people with severe disabilities that cannot interact easily with their usual environment. However, this research showed that not all the subjects of the study were able to control a cursor in 1D. Additionally, users who were able, they did not achieved a high accuracy with the real application. Probably, it could be due to two reasons: the limited training time and the cognitive problems they present. The study was limited to five weeks and approximately three sessions per week. Probably with more time and an intensive training, users could have increased their accuracy. Our results showed that accuracy from a healthy subject was higher compared to these users' performance. While a healthy subject was able to control a cursor in 1D with more than 90% of accuracy, these users achieved lower values. It could be due to their cognitive disabilities. They have problems to concentrate in a mental task for several seconds, so they probably did not carry out the motor imagery tasks correctly. Their results improved along with training but they had more difficulties than healthy people or those with only physical

disabilities. Further studies should take these limitations into account extending the training period and even designing other kind of BCI systems. Maybe, a P300-based BCI, which does not require extensive training [2], could achieve higher accuracy.

In summary, our results suggest AR models are adequate for EEG feature extraction in BCI systems. Furthermore, BCI systems could be very useful to control devices present at home, making interaction with these devices easier for disabled people. However, we found significant differences between people with cognitive problems and without them. It seems that cognitive disabilities are an important limitation to use μ rhythm-based BCI systems, so longer training periods or other type of BCI systems will be required.

References

1. Wolpaw, J.R., Birbaumer, N., Heetderks, W.J., McFarland, D.J., Peckham, P.H., Schalk, G., Donchin, E., Quatrano, L.A., Robinson, C.J., Vaughan, T.M.: Brain-Computer Interface Technology: A Review of the First International Meeting. *IEEE Trans. Rehab. Eng.* 8, 164–173 (2000)
2. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-Computer Interfaces for Communication and Control. *Clin. Neurophysiol.* 113, 767–791 (2002)
3. Pineda, J.A., Allison, B.Z., Vankov, A.: The Effects of Self-Movement, Observation, and Imagination on μ Rhythms and Readiness Potentials (RP's): Toward a Brain-Computer Interface (BCI). *IEEE Trans. Rehab. Eng.* 8(2), 219–222 (2000)
4. Lemm, S., Schafer, C., Curio, G.: BCI Competition 2003-Data Set III: Probabilistic Modeling of Sensorimotor μ -Rhythms for Classification of Imaginary Hand Movements. *IEEE Trans. Biomed. Eng.* 51(6), 1077–1080 (2004)
5. Corralejo, R., Hornero, R., Abásolo, D., Temprano, J.: Comparación de Métodos de Extracción de Características para su Aplicación a Brain Computer Interface. *Actas del XXVI Congreso Anual de la Sociedad Española de Ingeniería Biomédica*, 174–177 (2008)
6. Krusienski, D.J., Schalk, G., McFarland, D.J., Wolpaw, J.R.: A μ -Rhythm Matched Filter for Continuous Control of a Brain-Computer Interface. *IEEE Trans. Biomed. Eng.* 54(2), 273–280 (2007)
7. Schalk, G., Mellinger, J.: *A Practical Guide to Brain-Computer Interfacing with BCI2000*. Springer, London (2010)
8. Dornhege, G., Blankertz, B., Curio, G., Müller, K.R.: Boosting Bit Rates in Noninvasive EEG Single-Trial Classifications by Feature Combination and Multiclass Paradigms. *IEEE Trans. Biomed. Eng.* 51(6), 993–1002 (2004)
9. Marple, S.L.: *Digital Spectral Analysis with Applications*. Prentice Hall, New Jersey (1987)
10. Portnoff, M.R.: Time-Frequency Representations of Digital Signals and Systems Based on Short-Time Fourier Analysis. *IEEE Trans. Acoust. Speech Signal Process.* ASSP 28(1), 55–69 (1980)
11. Pfurtscheller, G., da Silva, F.H.L.: Event-Related EEG/MEG Synchronization and Desynchronization: Basic Principles. *Clin. Neurophysiol.* 110, 1842–1857 (1999)
12. Blankertz, B., Dornhege, G., Kunzmann, V., Losch, F., Curio, G.: The Berlin Brain-Computer Interface: Machine Learning-Based Detection of User Specific Brain States. In: Dornhege, G., Millán, J., Hinterberger, T., McFarland, D.J., Müller, K.R. (eds.) *Towards Brain-Computer Interfacing*, The MIT Press, Cambridge (2007)

A Dictionary-Driven SSVEP Speller with a Modified Graphical User Interface

Ivan Volosyak, Anton Moor, and Axel Gräser

Institute of Automation (IAT) - University of Bremen
Otto-Hahn-Allee, NW1, 28359 Bremen, Germany

volosyak@iat.uni-bremen.de

<http://www.iat.uni-bremen.de/>

Abstract. BCI spellers are mainly composed of an interface, by which alphanumerical characters are presented to users, and a classification system, which identifies the target character online by using the acquired EEG data. In this study, we proposed modifications both to the graphical user interface and to the classification system of the SSVEP-based speller, in order to improve the usability of the Bremen-BCI and to consequently increase the information transfer rate (ITR). The performances of writing three phrases (including every letter of the alphabet at least once) using the standard Bremen-BCI speller and the proposed dictionary-driven SSVEP speller with a modified interface were compared with each other. Seven subjects, in copy spelling mode, used a conventional speller and an improved BCI speller incorporating a custom-built dictionary with a mean ITR of 29.98 ± 5.79 bit/min and of 32.71 ± 9.18 bit/min, respectively. For the vast majority of users, this new dictionary-driven BCI realization resulted in a significant increase in spelling performance.

Keywords: BCI (Brain-Computer Interface), SSVEP (Steady-State Visual Evoked Potential), LCD (liquid crystal display), speller.

1 Introduction

A brain-computer interface (BCI) is a technical system that acquires and analyzes brain neural activity patterns in real time to translate them into control commands for computers or external devices [1]. Steady-state visual evoked potentials (SSVEP) are the continuous brain responses elicited at the visual and parietal cortical area under visual stimulation with a specific constant frequency; they can be used for BCI and are described as reliable in the relevant literature [2]. Presently, the SSVEP approach provides the fastest and the most reliable communication paradigm for the implementation of a non-invasive BCI system. High information transfer rates are essential for a BCI in order to become a practical device for communication and control, such as a speller application, and are important in order to control an external device.

Several methods for improving the information transfer rate (ITR) are available. In this study we address this issue by incorporating a custom-built dictionary into a conventional classification system. Although the idea of supporting

BCI spellers with dictionaries, in general, is not new [3,4,5], there are - to the best of our knowledge - no published studies that have attempted to realize this idea with a BCI speller based on the SSVEP paradigm. Therefore, we decided to improve our conventional SSVEP-based Bremen-BCI speller by incorporating a custom-built dictionary. These changes involved several necessary modifications in the graphical user interface (GUI), which are specific to each particular BCI paradigm, and hence they constitute the main novelty presented in this paper.

The paper is organized as follows: the second section describes the experimental setup, introduces the dictionary employed, and presents details about the modified interface for SSVEP spellers. The results are presented in the third section, followed by a discussion and conclusion in the final section.

2 Methods and Materials

2.1 Subjects

The SSVEP copy-spelling experiments were conducted with seven healthy volunteer subjects. All subjects use a computer screen at work on a daily basis and had little or no previous experience with BCI systems. Three of the subjects were male, and four were female, with a mean age of 34.3 years (std=14.04, range 18-55). All subjects had normal or corrected-to-normal vision. The subjects did not receive any financial reward for participating in this study.

2.2 Signal Acquisition

Subjects were seated approximately 60 cm in front of the LCD screen (22" Samsung SyncMaster 2233RZ with the vertical refresh rate of 120 Hz and resolution of 1680 x 1050 pixels) of the desktop computer running the Bremen-BCI software. The graphical user interface of the SSVEP based Bremen-BCI speller that was used is shown in the Fig. 1. The desktop PC has an Intel Core i7 CPU 975 (3.33GHz) processor running Microsoft Windows XP Professional. The EEG data were recorded from the surface of the scalp via 8 standard Ag/AgCl EEG electrodes. They were placed on sites P_Z , PO_3 , PO_4 , O_1 , O_Z , O_2 , O_9 , O_{10} ; AF_Z was used for ground and A_2 (right earlobe) was used for the reference electrode of the international system of electrode placement. Standard abrasive electrolytic electrode gel was applied between the electrodes and the skin in order to bring impedances below 5 k Ω . An EEG amplifier g.USBamp (Guger Technologies, Graz, Austria) was used and the sampling frequency was 128 Hz. During the EEG acquisition, an analog bandpass filter between 2 and 30 Hz, and a notch filter around 50 Hz (mains frequency in Europe) were applied directly in the amplifier.

2.3 Experimental Setup

After completing the consent form, subjects were prepared for the EEG recording described above. A short familiarization run was carried out in order to introduce the speller application to the subjects. No parameter adaptations were

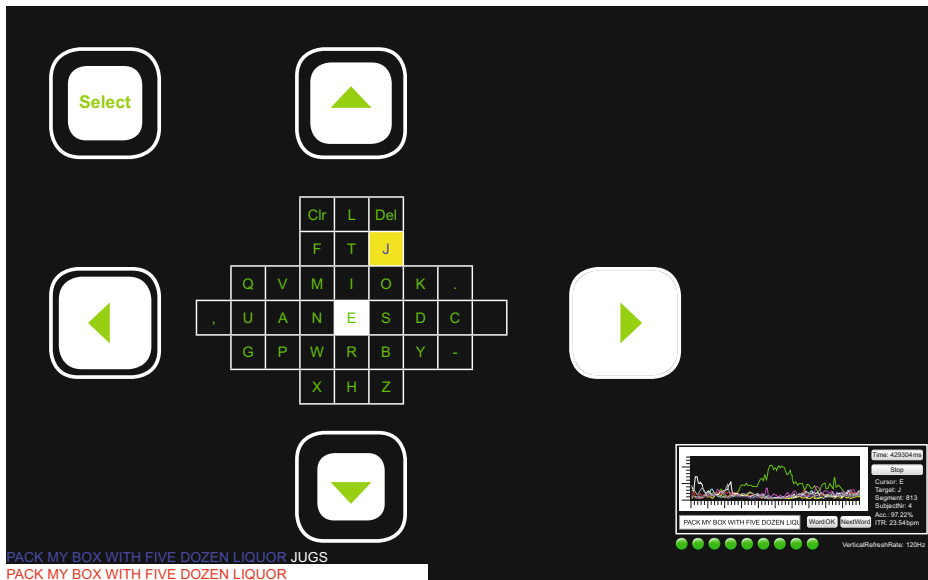


Fig. 1. GUI of the SSVEP based BREMEN-BCI during an online experiment. A subject was spelling the text “PACK MY BOX WITH FIVE DOZEN LIQUOR JUGS”, the command “right” was about to be executed. A cursor can be navigated left, right, up and down until the desired letter is reached. With the “select” command a letter is selected and displayed at the bottom of the screen. At the beginning of the experiment and after every selection, the cursor automatically moves back to the initial letter ‘E’.

performed at this stage. The assessment task was to spell three texts – including the words “BCI”, “BRAIN”, and a pangram (a sentence using every letter of the alphabet at least once) with the two SSVEP spellers: the conventional Bremen-BCI system and a modified speller incorporating a custom-built dictionary. The two copy spelling words “BCI” and “BRAIN” were the same for all subjects, but the pangrams were different for each subject. The order in which these three terms were presented to the user was determined randomly in order to avoid adaptations. Each trial ended automatically when the subject correctly spelled the text in question (or when the subject chose to stop spelling due to any reason such as visual fatigue - which has not happened in this study). Misspellings were to be corrected with the ‘Del’ option located at the top-right of the matrix. The entire session took an average of about 60 minutes per subject. All data collected during the experiment were recorded anonymously.

Conventional SSVEP-based Bremen-BCI speller. The graphical user interface (GUI) of the conventional Bremen-BCI speller is presented in Fig. 1. It consists of a virtual keyboard with 32 characters (letters and special symbols) and five white stimulation boxes. These boxes are located at the outer edges and upper left corner of the screen and flicker with different frequencies. As the quality of the SSVEP response depends on the stability of the frequencies, the

five stimulation frequencies that are used in this experiment were selected on the basis of the refresh rate of the LCD screen (120 Hz) that produces the stimuli: 6.67 Hz (“select”), 7.50 Hz (“left”), 8.57 Hz (“right”), 10.00 Hz (“up”), and 12.00 Hz (“down”). In the previous studies [6,7] the selection of these frequencies was thoroughly discussed. This setup, as opposed to having a LCD for the GUI and a separate LED board for the visual stimuli, is much more convenient for the users as they do not have to shift their gaze too much. In contrast to the previous software implementation of this GUI application, as described, e.g., in [7], where the EEG classifier was implemented as a separate software module (Microsoft MFC application) connected via a Transmission Control Protocol/Internet Protocol (TCP/IP) link to the GUI (Nokia QT application), in the most recent software realization the complete software module was re-developed as a single executable (developed as a Microsoft DirectX 9 application). The second improvement consists in the automatic measuring of impedances between EEG electrodes (represented as nine circles, eight signal electrodes and the reference electrode, presented in the EEG block located at the lower right edge of the screen) during the subject preparation phase. By applying of an appropriate electrode gel the impedances between the electrodes and the skin should be bring below $5\text{ k}\Omega$ during the standard preparation procedure. The impedances are measured sequentially and are represented online as follows: below $5\text{ k}\Omega$, in green; above $5\text{ k}\Omega$ and below $10\text{ k}\Omega$, in yellow; and the impedances of electrodes above $10\text{ k}\Omega$ were shown in red (cf. Fig. 2(a)). All these modifications improve the comfort and easy use of the SSVEP-based Bremen-BCI.

At the beginning of each trial, the cursor is located in the middle of the virtual keyboard, over the letter ‘E’, and all flickering boxes are presented in their default size of 125 x 125 pixels. During the spelling task, by focusing the user’s attention on one of the four flickering boxes, the cursor is navigated by the commands “left”, “right”, “up” and “down” until the desired letter is reached. With the “select” command a character is selected and displayed at the bottom of the screen, as shown in Fig. 1. Audio feedback, i.e. the name of the classified command, follows after every recognized command. After each selection, the cursor automatically moves back to the initial letter ‘E’. The number of commands required for the letter selection varies from letter to letter - from the minimum of one (selection of the letter ‘E’ in the middle of the speller layout with just one “select” command) to a maximum of five commands (e.g., the selection of the letter ‘G’, which requires four movement commands and the following selection). The letters are arranged according to their frequency of occurrence in the English language, and the additional special characters are located at the edge.

During the experiment, the sizes of the flickering boxes vary in relation to the SSVEP amplitude. The white frames around each stimulus box (their fixed size is 205 x 205 pixels each) represent the maximum size that a stimulus can reach without classification. This assists the user in knowing whether a command has been executed. This novel continuous real-time visual feedback about the power of SSVEP signals additionally improves the time behaviour of the BCI system [8].



Fig. 2. GUI of the dictionary-driven SSVEP-based Speller with a modified GUI. After spelling two letters, up to six words to choose from are suggested on the right side of the button “Go”. If the desired word is amongst those, as it is shown in Fig. 2 (a), the BCI user can switch to layout 2 by means of the “Go” command. In layout 2 (shown in the Fig. 2 (b)) the whole word can be written by selecting the corresponding flickering box. In case of erroneously selecting the “Go” command, the user can switch back to layout 1 with the “back” command.

Navigation can move beyond the layout boundaries. For example, it is possible to go from the letter ‘L’ to the letter ‘H’ by choosing the “up” command. The box at the bottom of the screen contains - for the copy spelling mode - the word that is to be spelled, and the already spelled text. Fig. 1 shows a screenshot taken during the online spelling task. The SSVEP classification was performed on-line every 13 samples (ca. 100ms) on the basis of the adaptive time segment length of the acquired EEG data. More details about the SSVEP detection method used can be found in [8].

Dictionary-driven SSVEP-based Speller with a modified GUI. The graphical user interface of our proposed speller system with a custom-built dictionary unit (DU) consists of two different layouts, as shown in Fig. 2. Layout 1 is almost equivalent to the conventional Bremen-BCI speller shown in Fig. 1. The only difference is the additional button “Go”. After the subject has spelled two consecutive letters, up to six word suggestions starting with these two letters are presented to the BCI user. In the initial stage, our custom-built dictionary was filled with 49’142 commonly used words of the English language. Each word was initialized with an occurrence frequency of zero (therefore initial word suggestions will be delivered in alphabetical order). If the desired word is found between these suggestions, the user can switch to layout 2 (shown in the Fig. 2(b)) by means of the “Go” command. In layout 2, each flickering box corresponds to the complete word (including a space character after the word), which allows for a significant increase in spelling performance, especially for words with many letters. Even for short words this feature is advantageous, because of automatically adding the word separator (space) after each word. This character is located at the right edge of the letter layout and, with the conventional Bremen-BCI speller, it requires at least 5 commands to be selected (4 times “right” with the following selection).

Similar to the T9 technology, we implemented a feature which stores the words and phrases the user often uses. It speeds up the process by offering the most frequently used words first. It then lets the user access other choices, as well. To each word in the dictionary, we assign an integer variable (frequency of occurrence), which will be incremented after this word was spelled. This is exactly the reason why we selected different pangrams for each user (otherwise the order of experiments would matter – the first user of the system would be discriminated against users who will start with the already updated dictionary). The dictionary can be expanded by adding missing words, making it possible for them to be recognized in the future. After introducing a new word in the conventional way by spelling all letters one by one, the next time the user tries to produce that word it will be found in the predictive dictionary.

This custom-built dictionary is realized on the basis of the software library SQLite version 3.6.23.1. The stimulation frequencies that are used for additional buttons in the modified GUI (six flickering boxes in layout 1 and up to seven frequencies in layout 2, cf. Fig. 2) were also selected on the basis of the refresh rate of the LCD screen (120 Hz) that produces the stimuli: 6.32 Hz (“Go”, “back” in layout 2), 10.91 Hz (sixth word suggestion in layout 2).

Table 1. Copy spelling results. The crossed out letters (e.g. ~~Æ~~) represent misspelled text, which was corrected with “Del” option during the experiment. The space character is represented as ~~_~~. The letters automatically added by the dictionary unit after selection in layout 2 are presented in superscript.

#		Copy spelling tasks (in bold) and effectively spelled words
1	BCI	BRAIN MY EX PUB QUIZ CROWD GAVE JOYFUL THANKS
	BCI	BRAIN MY_EX_PUB QUIZ_CROWD_GAVE_JOYFUL_THANKS
	BCI	BRAIN MY_EX_PUB^QUIZ^CR^OWD^GAVE^JOYFUL^THANKS^
2	BCI	BRAIN WAXY AND QUIVERING JOCKS FUMBLE PIZZA
	BCI	BRAIN WAXY_ AND_ QUIVERING_ JOCKS_ FUMBLE_ PIZZA
	BCI	BRAIN WAXY^AND^QUIVIE^RING^JOCKS^FUMBLE^PIZ^A^
3	BCI	BRAIN COZY SPHINX WAVES QUART JUG OF BAD MILK
	BCI	BRAIN COZY_SPHINX_WAVES_QUART_JUG_OF_BAD_MILK
	BCI	BRAIN COZY^SPHINX^WAVES^QUART^JUG^OF^BAD^MILK
4	BCI	BRAIN A VERY BAD QUACK MIGHT JINX ZIPPY FOWLS
	BCI	BRAIN A_VERY_BAD_QUACK_MIGHT_JINX_ZIPPY_FOWLS
	BCI	BRAIN A^VERY^BAD^QUACK^MIGHT^JINX^ZIPPY^FOWLS
5	BCI	BRAIN PACK MY BOX WITH FIVE DOZEN LIQUOR JUGS
	BCI	BRAIN PACK_MY_BOX_WITH_FIVE_DOZEN_LIQUOR_JUGS
	BCI	BRAIN PACK_MY^BOX^WITH^FIVE^DOZEN^LIQUOR^JUGS
6	BCI	BRAIN FEW QUIPS GALVANIZED THE MOCK JURY BOX
	BCI	BRAIN FEW_QUIPS_GALVANIZED_THE MOCK_JURY_BOX
	BCI	BRAIN FEW^QUIPS^GALVANIZED^THE^MOCK^JURY^BOX
7	BCI	BRAIN KVETCHING FLUMMOXED BY JOB W ZAPS IRAQ
	BCI	BRAIN KVETCHING_FLUMMOXED_BY_JOB_W_ZAPS_IRAQ
	BCI	BRAIN KVETCHING^FLUMMOXED^BY^JOB^W^ZAPS^IRAQ^

3 Results

Table 1 presents the copy spelling results (categories for each individual subject: copy spelling tasks, words spelled with the conventional Bremen-BCI speller, results with the dictionary unit). The resulting spelling times for each SSVEP speller application are reported in Table 2. The number of commands with corresponding accuracies, and the information transfer rates (calculated as described in 1) are presented for the conventional Bremen-BCI speller. The number of targets $N = 32$ (all letters and special characters of the layout) was used for the calculation of the ITR. It is important to mention that for this system realization, two ways of ITR calculations are available: on the command basis ($N = 5$ in Fig. 1), or on the basis of the letters used ($N = 32$ in Fig. 1). The ITR values vary depending on the number of targets used. This phenomenon was already discussed in our previous work [9]. On average, over all spelled texts, assuming the same weights of each of three spelling tasks, subjects performed in the copy spelling mode using the conventional speller, and using the improved BCI speller incorporating a custom-built dictionary, with a mean information transfer rate of 29.98 ± 5.79 bit/min and of 32.71 ± 9.18 bit/min, respectively. For the second case, the ITR was calculated on the basis of the text to be spelled (it was assumed that the user has spelled the complete text letter by letter).

Table 2. BCI spelling performance over all copy spelling words

Subject	Without dictionary unit (DU)									With DU		
	BCI			BRAIN			TEXT			BCI	BRAIN	TEXT
	Time	Acc.	ITR	Time	Acc.	ITR	Time	Acc.	ITR	Time	Time	Time
#	[s]	[%]	[bpm]	[s]	[%]	[bpm]	[s]	[%]	[bpm]	[s]	[s]	[s]
1	32.91	100.00	27.35	52.51	100.00	28.57	707.99	92.70	16.98	27.02	68.45	478.97
2	19.09	100.00	47.14	47.53	100.00	31.56	723.73	87.76	15.67	27.63	26.00	569.77
3	15.95	100.00	56.44	26.34	100.00	56.95	292.09	91.49	40.15	14.12	12.39	272.29
4	35.55	80.00	27.74	46.41	85.71	33.49	395.99	86.79	33.41	13.71	29.45	572.10
5	23.97	100.00	37.55	40.12	100.00	37.39	516.14	93.33	22.58	36.46	48.55	562.45
6	35.24	100.00	25.54	96.38	85.71	16.13	438.04	97.50	25.79	32.91	22.65	606.84
7	28.84	100.00	31.20	79.73	77.78	21.23	676.41	82.76	17.92	34.63	21.23	543.36
Mean	27.36	97.14	36.14	55.00	92.74	29.16	535.77	90.33	24.64	26.64	32.67	515.11
SD	7.88	7.56	11.69	24.99	9.43	13.37	170.11	4.90	9.20	9.35	19.28	113.99

4 Discussion and Conclusion

Although there were 49'142 words in the initial version of the customized dictionary, the selection of pangrams shown in Table 1 was inaccurate. On the one hand, we limited ourselves by designing the dictionary merely on the basis of common English words. On the other hand, many words used in pangrams did not exist in this dictionary. This was the main reason for some subjects, who reported that they would subjectively prefer the previous speller realization without a dictionary - due to simplicity reasons. The other problem consists in the fact that the words in the dictionary are mainly in singular form, but the pangrams contained many words in the plural. These problems exist only in this initial software implementation and could easily be solved in future implementations. The ITR values for the proposed speller with DU present just an rough estimation, the ITR was not designed for an additional input from a predictive speller, and should be calculated e.g. in "output characters per minute" as introduced in [5].

This study shows that the performance of dictionary-driven spellers is highly dependent on the target word. If there is only a small number of similar words in the dictionary, BCI users can attain a significant increase in spelling performance, especially for words with many letters. A modified graphical user interface (with one additional frequency to detect the command "Go" in layout 1, cf. Fig. 2) does not affect the classifier performance.

Our future work will address further improvements both in the proposed dictionary unit and in the GUI of the modified SSVEP speller.

Acknowledgment. The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant BRAIN, n° 224156, and from a Marie Curie European Re-Integration Grant RehaBCI, n° 224753.

References

1. Wolpaw, J., Birbaumer, N., McFarland, D., Pfurtscheller, G., Vaughan, T.: Brain-computer interfaces for communication and control. *Clin. Neurophysiol.* 113, 767–791 (2002)
2. Vialatte, F.B., Maurice, M., Dauwels, J., Cichocki, A.: Steady-state visually evoked potentials: Focus on essential paradigms and future perspectives. *Prog. Neurobiol.* 90, 418–438 (2010)
3. Felton, E., Lewis, N., Wills, S., Radwin, R., Williams, J.: Neural signal based control of the dasher writing system. In: 3rd International IEEE/EMBS Conference on CNE 2007, pp. 366–370 (May 2007)
4. Ahi, S.T., Kambara, H., Koike, Y.: A Dictionary-Driven P300 Speller With a Modified Interface. *IEEE Trans. Neural Syst. Rehab. Eng.* 19(99), 6–14 (2010)
5. Ryan, D.B., Frye, G.E., Townsend, G., Berry, D.R., Mesa, G.S., Gates, N.A., Sellers, E.W.: Predictive spelling with a P300-based brain-computer interface: Increasing the rate of communication. *Hum. Comput. Interact.* 27(1), 69–84 (2011)

6. Volosyak, I., Cecotti, H., Gräser, A.: Optimal visual stimuli on LCD screens for SSVEP based Brain-Computer Interfaces. In: Proc. 4th Int. IEEE/EMBS Conference on Neural Engineering NER 2009, pp. 447–450 (May 2009)
7. Volosyak, I., Cecotti, H., Gräser, A.: Impact of frequency selection on LCD screens for SSVEP based brain-computer interfaces. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) IWANN 2009. LNCS, vol. 5517, pp. 706–713. Springer, Heidelberg (2009)
8. Volosyak, I.: SSVEP-based Bremen-BCI– boosting information transfer rates. *J. Neural Eng.* (in press, 2011)
9. Cecotti, H., Volosyak, I., Gräser, A.: Evaluation of an SSVEP based Brain-Computer Interface on the command and application levels. In: Proc. 4th Int. IEEE/EMBS Conference on Neural Engineering NER 2009, pp. 474–477 (May 2009)

Non-invasive Brain-Computer Interfaces: Enhanced Gaming and Robotic Control

Reinhold Scherer^{1,2}, Elisabeth C.V. Friedrich³, Brendan Allison¹,
Markus Pröll¹, Mike Chung², Willy Cheung²,
Rajesh P.N. Rao², and Christa Neuper^{1,3}

¹ Institute for Knowledge Discovery, Graz University of Technology,
Krenngasse 37, 8010 Graz, Austria

² Computer Science and Engineering, University of Washington,
Box 352350, Seattle, WA 98195-2350, USA

³ Department of Psychology, University of Graz,
Universitätsplatz 2/III 8010 Graz, Austria
`reinhold.scherer@tugraz.at`

Abstract. The performance of non-invasive electroencephalogram-based (EEG) brain-computer interfacing (BCI) has improved significantly in recent years. However, remaining challenges include the non-stationarity and the low signal-to-noise ratio (SNR) of the EEG, which limit the bandwidth and hence the available applications. In this paper, we review ongoing research in our labs and introduce novel concepts and applications. First, we present an enhancement of the 3-class self-paced Graz-BCI that allows interaction with the massive multiplayer online role playing game World of Warcraft. Second, we report on the long-term stability and robustness of detection of oscillatory components modulated by distinct mental tasks. Third, we describe a scalable, adaptive learning framework, which allows users to teach the BCI new skills on-the-fly. Using this hierarchical BCI, we successfully train and control a humanoid robot in a virtual home environment.

1 Introduction

Brain-computer interface (BCI) technology allows direct interaction with the environment by recording and translating the user's brain activity in real-time. Electroencephalographic (EEG) signals, i.e., bioelectrical potentials recorded from the scalp, are the most common non-invasive source of brain signals in BCIs [1]. Major problems confronting BCI developers are the non-stationarity and inherent variability of EEG signals and the low signal-to-noise ratio (SNR). These characteristics, among others, hamper reliable detection and translation of on-going EEG patterns into messages and hence limit the bandwidth. This is aggravated by the fact that the brain itself is a highly adaptive system. No a priori fixed mappings between the brain and the application exist. User feedback training and machine learning are required to optimize the interplay.

The two major types of EEG features used in BCIs are evoked potentials (EP) and oscillatory components (transient changes of spectral components are

also known as event-related desynchronization/synchronization, ERD/S). EPs are stereotypical brain responses to external/perceptual events that are stable over time and require little adaptation on part of the user. EP-based BCIs do not require long training and achieve high information transfer rates (ITR, usually <30 bit/min), but rely on external stimuli to elicit EPs. ERD/S activity can be induced internally, i.e., users intentionally perform distinct mental tasks (MTs) to send specific messages to the BCI. ERD/S-based systems require longer training and achieve lower ITRs, however, users are independent from external events and can initiate communication whenever required.

In this paper, we review research currently being performed in our laboratories that aims to increase effective bandwidth through intelligent processing, reduce errors, and shorten ERD/S BCI training. By identifying the best control signals for each subject, and incorporating context awareness into more intelligent software, we can extend the control possible with BCIs. We are expanding the self-paced 3-class ERD/S Graz-BCI [2] and have added control of the massive multiplayer online role-playing game (MMORPG) World of Warcraft (WoW) (Blizzard Entertainment, Inc.). We are aiming at identifying the MTs that yield the best performance (stable ERD/S) within subjects. We are also working on a new adaptive hierarchical architecture that allows the user to customize the functionality of applications and teach the BCI new skills on-the-fly.

2 ERD/S-Based Interaction with World of Warcraft

Recently, we introduced the self-paced 3-class ERD/S BCI and a customized graphical user interface (GUI) that allowed BCI user operating Virtual Google Earth (Google, Inc., Mountain View, CA) [2]. Based on this 3-class approach, we developed a novel intelligent GUI that lets the user play the MMORPG WoW.

WoW is a very popular video game that offers individual players or parties of players a virtual universe to explore and interact with. The main goal of the game is to team up with other players to fight monsters and complete various quests. Computer keyboard and mouse are used to control the avatar. In order to achieve hands-free control we developed a network controlled application, which simulates specific time based mouse and keyboard inputs. For visual feedback, the WoW native application-programming interface LUA was used to extend the standard game interface with an add-on that allows displaying BCI feedback. When the BCI detected left hand, right hand, or foot motor imagery (MI), the user received real-time feedback in form of arrows pointing to the left, right and forwards, respectively, which mapped to the navigation commands “rotate left”, “rotate right” and “move forward” (Fig. 1a-c). The length of the arrow (4 increments) corresponds to the quality of detection, i.e., the BCI has to detect MI for a configurable minimum time before the avatar starts moving. The avatar continues executing the selected action as long as the user performs MI. Users can start, stop and switch between MI tasks as required (self-paced mode).

In WoW, many actions involve specific points of interest (POIs). For example, the player must typically go to a certain POI to accept a quest from a non-player

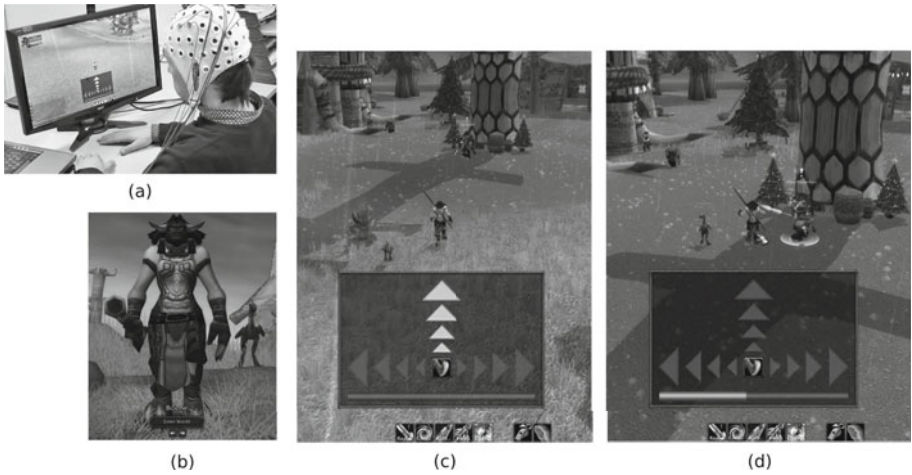


Fig. 1. BCI-based interaction with WoW. (a) Picture of the BCI player in front of the computer screen. (b) Game avatar (level 10 Tauren hunter). (c) The avatar in the starting area Mulgore. At the bottom of the screen, the in-game add-on shows the BCI-operated navigation arrows. In the screenshot, the arrow is pointing upwards and consequently the avatar is walking forward. (d) The avatar is close to a non-playable character that offers him a quest. The timeout bar for selecting critical events, located below the navigation arrows, starts growing from left to right. If the user wants to accept the quest, he has to wait a predefined dwell time. Otherwise, he can move away and cancel the action.

character (NPC), engage specific enemies, gather collectible resources, or loot objects. With three directional control, a fourth, binary degree of freedom can be generated by using a timeout, which we use for confirming critical actions, like attacking an opponent within range or accepting new quests. Whenever the player's avatar comes within the range of a POI, noncritical actions (e.g. getting a gatherable resource) are executed immediately. Critical actions can be accepted by staying within the range of the POI without moving for a predefined dwell time. During this time period, the timeout bar located below the navigation arrows starts extending. Once fully extended, the action is performed (Fig. 1d).

This control strategy lets players explore and interact with the virtual game environment in many ways. Users can play hundreds of quests, engage a wide range of enemies in numerous environments, win loot from defeated enemies and awards from NPCs, and improve their character in many ways. All of these actions could be implemented socially. Indeed, most players in WoW do not play alone, but conduct quests and other actions in cooperation with a party consisting of other human players. Hence, this new WoW BCI system had a much more complex and engaging task structure than our prior work, as well as more intelligent BCI processing software that incorporated context to enable more complicated, high-level control with a limited bandwidth.

3 Evaluation of Mental Tasks for Robust Control

In the previous study we used kinesthetic MI, which is the classical MT used for controlling ERD/S BCIs. According to our experience, however, there is a high variability in the on-line performance (e.g. [2]). This is particularly evident during early training and when optimizing the system complexity (e.g. limiting the number of EEG sensors) is required to increase practicality. We therefore started the systematic investigation of MTs of different modalities with the aim to identify MTs that are statistically stable over long periods of time (days) and thus may increase performance and reduce training time [3].

In order to identify such MTs, we recorded thirty channel EEG data from 9 naïve able-bodied female subjects during 7-s cue-guided imagery trials on four different days. Electrodes were evenly distributed over the head. Thirty trials per class were recorded on each day in randomized order. MTs included:

1. Mental rotation (ROT): Visualization of 3-dimensional L-shaped figure rotating in the 3-d space.
2. Word association (WORD): Generation of as many words possible beginning with the presented letter (e.g. B = bank, bold, buy, etc).
3. Auditory imagery (AUD): Imagination of listening to a familiar tune (melody) without articulating the words.
4. Mental subtraction (SUB): Calculation of successive elementary subtractions from the presented problem (e.g. $105-6 = 99$, $99-6 = 93$, etc).
5. Spatial navigation (NAV): Imagination of navigating through a familiar house (flat) thereby focusing on orientation.
6. Imagery of faces (FACE): Imagination of the face of the best female friend.
7. Motor imagery (MI) of the right hand: Imagination of repetitively squeezing a hand-sized ball with the own right hand.

The EEG was band pass filtered 0.5-100 Hz, amplified and digitized at 256 Hz. Additionally, electrooculographic (EOG) activity was recorded from two electrodes placed on the outer canthus of the left eye and above the nasion. Recorded EEG signals were visually scored by experts and trials contaminated with muscle or eye movement activity within the imagery period were excluded from further analysis. Two distinct classes are required to transfer bits of information and consequently as first step we examined binary classification performances. To get a first glimpse on the stability of MTs, the classical method of common spatial patterns (CSP) was used to design class specific spatial filters in the 8-30 Hz frequency band, and Fisher linear discriminant analysis (LDA) to classify the log-transformed normalized variance from 4 projections [4]. Each day was analyzed independently to rank the discriminability of the imagery pairs and to evaluate the variability between days. To get an overview of timing and dynamics of the induced EEG patterns, trials were subdivided into thirteen 1-s data segments with 0.5 s overlap. For each time segment and imagery pair, CSPs and LDA were computed and evaluated using a 10-times 10-fold cross-validation procedure.

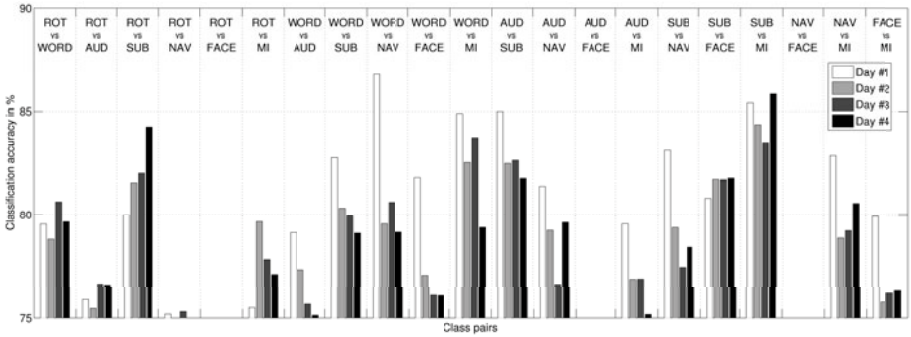


Fig. 2. Mean classification performance (10-times 10-fold cross-validation), averaged over nine subjects, based on classifying pairs of mental tasks on four different days

Fig. 2 summarizes initial results, including mean cross-validated accuracies, averaged over all participants, for each class pair and day. The mean accuracy for each day was computed by averaging subject-specific peak accuracies within the 7-s imagery period. Class pairs SUB vs. MI, AUD vs. SUB and SUB vs. FACE achieved binary classification accuracies $>80\%$ on every single day. The computed accuracies are comparable to accuracies computed between different motor imagery tasks reported in the literature. Our off-line results suggest that the identified MTs induce EEG patterns that are statistically distinct and stable over time, and thus appropriate for BCIs. We are now conducting feedback experiments to confirm these findings.

4 Toward Hierarchical Adaptive BCIs for Robotic Control

There is growing interest in intelligent and context-aware software for robotic devices. Many disabled people cannot use high bandwidth communication systems, and seemingly simple tasks like getting a glass of water may take far too long if the user must control low-level details of this task. Hence, intelligent robotic devices often allow users to accomplish high-level goals with a single command. In [5], we used a P300-based BCI to send high-level commands to a semi-autonomous humanoid robot that physically interacted with real objects in an augmented reality environment. To autonomously perform these tasks, the robot requires prior knowledge of the environment and substantial artificial intelligence. For example, to pour a glass of water from a bottle in a refrigerator, robots must navigate in the given environment, know the locations of cups and the bottle, manipulate objects, and heed safety protocols.

We are now developing a new generation of scalable, user-adaptive BCIs that combine the advantages of process-directed and goal-directed control. Users could utilize imagery to teach the BCI new commands, which are then made available for selection using evoked potentials (e.g., the P300). This leads to a

hierarchical hybrid BCI wherein lower-level actions are first learned and later semi-autonomously executed using a higher-level command, thereby improving accuracy and freeing the user from tedious ongoing process-oriented control. Continuing the above example, the user first navigates the robot to the kitchen, and hence teaches the robot the task “Go to kitchen” from a specific starting point. To make this flexible and universally applicable, the robot has to generate an internal representation of the environment that allows for finding the path to the kitchen from any location in the user’s environment. In a first experiment we used radial basis function models to learn navigation policies [6]. Usually, however, not enough training data is available and so learned models may not be reliable enough to predict the path. To overcome this problem, we investigate the use of uncertainty for guiding the robot’s behavior. We use Gaussian processes (GPs, [7]) for learning high-level commands and exploit the fact that they provide a measure of uncertainty in their output. When the uncertainty in a given region of the task space is too high (e.g., due to lack of training in that area), the BCI switches to user control for further guidance rather than continuing to execute the unreliable and potentially dangerous high-level command. Such uncertainty-guided decision-making is critical for real-world BCI applications, such as BCI-control of a robotic wheelchair or helper robot, where user safety and the safety of those around the robot are of paramount importance.

The current prototype of the hierarchical BCI for robotic control is composed of the three components: A steady-state visual EP-based (SSVEP) BCI; a hierarchical menu and learning system; and the humanoid robot using a simulation of the robot that mimics the physics of the real world. The three components interact closely: The hierarchical adaptive menu system displays available commands as flashing stimuli for the user to choose using the SSVEP BCI. The user makes the desired selection by focusing on the desired command in the menu. The BCI detects the option the user is focusing on and sends its classification output to the hierarchical menu system, which in turn sends a command to the robot and switches to the next appropriate menu. The robot executes the command it receives, which can be either a lower-level command such as turn left/right or a higher-level learned command. Finally, the user closes the control loop by observing the simulated robots action and making the next desired selection based on the updated menu.

Up to three flickering stimuli (12 Hz, 15 Hz and 20 Hz) were presented on a TFT computer screen with a refresh rate of 60 Hz. The view of the robot in its environment was shown in a larger immersive setting above (Fig. 3.a). One channel EEG was recorded (60 Hz notch; sampling rate 256 Hz) bipolarly from electrodes placed at Cz and Oz. SSVEPs were calculated by applying the Fast Fourier Transform to 1-s segments of EEG every 0.5 s. The frequency with the highest average power among the three target frequencies within the past 4-s was classified as the users choice.

The structure of the hierarchical menu system is depicted in Fig. 3.b. Users could choose to train either new or existing skills by navigating the robot through the virtual environment. During navigation, the robot moved forward

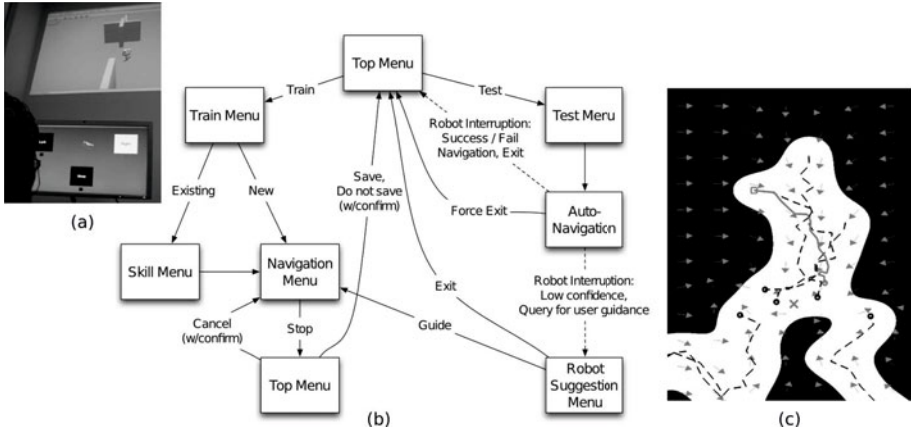


Fig. 3. Hierarchical learning framework. (a) Picture of the experimental setup showing the subject in front of the SSVEP stimulation screen and the humanoid robot simulation placed above. The user is in the low-level control mode (Navigation Menu) and navigating the robot by selecting the commands left, right or stop. (b) Hierarchical menu structure. The boxes represent the available (sub)menus. Each arrow and the associated label indicate the option that has to be selected to arrive at the sub-menu. (c) Navigation traces for GP model learning. Dashed black lines represent training routes (low-level control). The symbol "o" indicates the end point of a route. The target destination is marked with the symbol "x". The continuous gray line shows the route of the robot during the test mode (high-level command). The robot follows the directions learned by the GPs (vector field). Dark areas mark regions of high uncertainty.

automatically. Available navigation commands included left, right and stop. After the user demonstrated and saved samples of the task, the BCI learns the task via GPs. Learned skills can be called and tested (Test Menu). The learned model guides the humanoid robot through the environment. Whenever the robot enters regions of uncertainty (high variance in the GP model), the system interrupts the operation and asks the user for assistance, i.e., to either guide the robot and collect more data, or to exit and return to the top-level menu. During execution of a high-level command, the robot queries the BCI for navigation direction based on the current position (in this feasibility study the global positioning system, GPS, was used). If the GP model is used, one obtains both a predicted mean value as well as the variance of the prediction. This variance can be related to the confidence of the BCI in the learned model: high variance implies low confidence in the predicted navigational command and vice versa. For the current implementation, we used a simple threshold to decide when the robot should ask the user for guidance based on this confidence metric.

First preliminary results from four subjects using the hierarchical BCI indicate that the proposed system reduces the number of required selections and decreases the navigational time compared to low-level control. Fig. 3c shows navigation traces of GP model learning and guidance.

5 Conclusion and Future Direction

Non-invasive BCIs have evolved in recent years, and can now interact with a variety of applications and artificial devices. However, there is still room for improvement. We need better models of brain function and methods that predict brain activity, more practical recording equipment, and novel concepts on how to optimize brain-computer adaptation and extend progress to new applications and goals. In this paper, we reviewed research in our labs that enhances the usefulness of BCIs and paves the way for new applications. We introduced a novel GUI for the 3-class self-paced Graz-BCI that allows playing the MMORPG WoW. Players can not only navigate through the virtual world, but also interact with the virtual environment, acquire and complete game quests, and improve their character. We demonstrated that a proper combination of MTs contributes towards a more robust classification of ERD/S patterns. We proposed a new adaptive hierarchical architecture that allows a user to teach the BCI new skills on-the-fly. These learned skills are later invoked directly as high-level commands. We plan to continue this research direction, and explore new tasks and task combinations (such as in hybrid BCIs) and alternate applications and devices.

Acknowledgements

This work was supported in part by the ICT Collaborative Project BrainAble (247447), the NeuroCenter Styria, the National Science Foundation (0622252 & 0930908), the Packard Foundation, and the Office of Naval Research.

References

1. Mason, S.G., Bashashati, A., Fatourechi, M., Navarro, K.F., Birch, G.E.: A comprehensive survey of brain interface technology designs. *Ann. Biomed. Eng.* 35(2), 137–169 (2007)
2. Scherer, R., Schlögl, A., Lee, F., Bischof, H., Janša, J., Pfurtscheller, G.: The Self-Paced Graz Brain-Computer Interface: Methods and Applications. *Computational Intelligence and Neuroscience*, Article ID 79826 (2007)
3. Friedrich, E., Scherer, R., Neuper, C.: Consistency over time and across tasks of brain-computer interface-relevant mental strategies. In: *Proc. TOBI Workshop, Graz, Austria, February 3-4*, p. 68 (2010)
4. Ramoser, H., Müller-Gerking, J., Pfurtscheller, G.: Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Trans. Biomed. Eng.* 8(4), 441–446 (2000)
5. Bell, C.J., Shenoy, P., Chalodhorn, R., Rao, R.P.N.: Control of a humanoid robot by a noninvasive brain-computer interface in humans. *J. Neural. Eng.* 5, 214–220 (2008)
6. Chung, M., Cheung, W., Scherer, R., Rao, R.P.N.: Towards Hierarchical BCIs for Robotic Control. In: *Proc. IEEE EMBS NE 2011, Cancun, Mexico (2011)* (to appear)
7. Rasmussen, C.E.: Gaussian Processes in Machine Learning. In: Bousquet, O., von Luxburg, U., Rätsch, G. (eds.) *Machine Learning 2003. LNCS (LNAI)*, vol. 3176, pp. 63–71. Springer, Heidelberg (2004)

An EEG-Based Design for the Online Detection of Movement Intention

Jaime Ibáñez, J. Ignacio Serrano, M. Dolores del Castillo, Luis Barrios,
Juan Álvaro Gallego, and Eduardo Rocon

Grupo de Bioingeniería, Spanish National Research Council (CSIC)
Ctra. Campo Real km 0.200 - La Poveda, 28500, Arganda del Rey (Madrid), Spain

Abstract. The development of EEG-based wearable technologies for real-life environments has experienced an increasing interest over the last years. During activities of daily living, these systems need to be able to distinguish predefined mental states from the ongoing EEG signal, and these states of interest can be given after long periods of inactivity. A detector of the intention to move that is conceived to be used in real-time is proposed and offline validated with an experimental protocol with long intervals of inactivity that are also used for the detector's validation.

1 Introduction

The EEG-based BCI technology has experienced a great development in the last decade in fields like rehabilitation [4], entertainment, or impaired subjects support during activities of daily living (ADL) [9]. Nevertheless, there is still a major challenge to be addressed: to develop a technology that can be easily set-up and used in real life conditions. Any BCI system must fulfill some essential requirements in order to be used for ADL: 1) a reduced number of EEG positions; 2) the system must be able to run online, taking into account the variability existing among subjects; 3) the system must be able to work asynchronously; and 4) the protocols used must be realistic, with long intervals of user's inactivity between tasks.

Many BCIs have successfully faced the first three requirements [10]. However, the experimental protocols that most of the BCI studies use have high restrictions on the timing of the tasks performance, and this leaves less freedom to the measured subjects to execute the tasks at their own pace. Letting the measured subjects to self control the timing of the tasks executions (asynchronous paradigms) helps developing more real scenarios where the performance of the BCIs can be properly validated. During real conditions, where a person normally performs tasks after long intervals of inactivity, it becomes useful to discriminate between the mental processes of interest and the ongoing basal (non task-related) EEG signal. A BCI that is intended to recognize such predefined mental processes operates as a switching system that is in charge of controlling posterior systems [6].

The BCI system presented in this paper aims at detecting the intervals preceding the execution of voluntary motor actions in patients suffering from neurological tremor. It is optimized for working in real scenarios, successfully fulfilling the aforementioned requirements of a BCI system for ADL. The detection of the patient's intention to move can help differentiating the voluntary actions from the undesired tremor. This can be of great interest for subsequent strategies for tremor compensation.

The detector presented here must be able to anticipate the execution of voluntary movements, for that reason only EEG-detectable neurophysiological processes that anticipate this kind of tasks can be used. There are two well documented neurophysiological phenomena that start before a voluntary action: 1) the Bereitschaftspotential (BP), which corresponds to the early stage of the Movement Related Cortical Potentials (MRCP), which are the variations of the DC amplitude of the EEG signal related to the execution of motor tasks [11] [3]; and 2) the Event-Related Desynchronization (ERD), which refers to the decrease of the EEG signal power in the alpha and beta bands related to the performance of motor tasks, [8].

The online single trial analysis of the BP for the detection of movement intention presents an important drawback: the part of the BP to be used is mainly the early-BP, that has low amplitude, and it is only visible after averaging over a set of trials large enough. The ERD overcomes this problem since it does not have a transitory period between the "synchronized state" and the "desynchronized state". Typically, ERD-based systems analyze the desynchronization phenomenon over time windows of 1-2 seconds length to assert that the desynchronization corresponds to a motor task [12]. Nevertheless, only a few studies with ERD take advantage of its anticipatory characteristic, and they present synchronous-BCIs discriminating between pre-defined tasks, [7]. To our knowledge, the only EEG-based movement intention detector that has been reported up-to-date is a recent system developed by Bai et al., [2]. They present an online BCI detecting the movement intentionality. The system evaluates the best features of the signal spectrum and decides whether a voluntary movement is to be performed with a low false positive rate and an average anticipation time longer than 600ms. However, this work has only been tested with healthy subjects (22.6 +/- 2.4 years old). Besides, it includes a visual feedback to avoid performing movements when EEG artifacts are measured and uses 27 electrodes for the calibration session.

ERD-based intention detection in healthy subjects depends on the ability to detect the subject-specific EEG patterns associated to the desynchronization process. This detection becomes more difficult when measuring patients suffering from neurological disorders, like the tremor patients analyzed in the present study. The effects of the tremor related neurological disorders on the desynchronization patterns have been analyzed [5], especially in the case of Parkinson's disease, where a decrease in the amplitude of the ERD and a delay in its observation have been documented.

The BCI system proposed in this paper is specifically optimized for being used in real time conditions while subjects perform ADL. A Naïve Bayes classifier is used, because it is optimal in terms of computational load while presenting similar classifying performance than other classifying methods [1]. Additionally, a reduced number of electrode positions over the motor cortex has been used. The experimental protocol proposed for validating this movement intention detector consists of large intervals of motor inactivity and well-separated self-paced movement executions. The results obtained show that the EEG ERD features preceding a voluntary movement can be successfully detected in real-time conditions in controls and tremor patients. Our BCI system presents a high rate of movement anticipations and the number of false positives generated during long periods of non-activity is very low for most of the subjects.

2 Methods

2.1 Subjects

Two groups of right-handed subjects were selected in the present study: 2 control subjects (1 female and 1 male) and 2 age-unmatched patients (all males) with essential tremor and thus presenting kinetic tremor. All of them participated in this study with the approval of the ethical committee of the Hospital Central de Valencia. The patients were 70 and 72 years old each and the two controls were 27 and 28 years old. The selection of two different groups of population was done in order to demonstrate the system's adaptability, and not to perform a study on the special ERD features of the tremor patients

2.2 Data Acquisition

EEG signals were recorded from FC3, FCz, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CPz and CP4 of the international 10-20 system, with a gTec amplifier and Au scalp electrodes. The reference was set to the common potential of the two earlobes and Fz was used as ground. The amplifier was set to filter the signal between 0.5 and 60 Hz, and an additional 50-Hz notch filter was used. Synchronized with the EEG acquisition, the movements performed with the right arm and hand were detected by means of inertial sensors (manufactured by Tech MCS, Madrid, Spain). The EEG, IMUs and acoustic reference locations were all stored at 256Hz.

In order to get reference-free estimations of the EEG signals, all the 13 channels were spatially filtered. A Laplacian filter was used for the electrode positions C3, C1, Cz, C2, and C4. The surrounding channels were spatially filtered by means of a Common Average Reference filter. The resulting 13 reference-free channels were the data sources used for the analysis of the ERD.

2.3 Paradigm

The measurement sessions with each subject were divided into 3 minutes length runs. In each run, the measured subjects were asked to stay steady and to

perform periodically a motor task consisting of a single wrist extension followed by a return into a resting position. An acoustic reference sounded 10s after each movement onset to indicate the measured subjects that a trial had finished and it assured a separation long enough between two consecutive movements for the subject's mental activity to get back into a resting or basal state. The movement executions with less than 3 seconds between the acoustic reference and the next movement were removed in the posterior analysis to ensure the voluntariness of the movements taken into account. This was accomplished because typically, the desynchronization in voluntary movements starts 2 seconds before the onset of the movement [8], and the length of the evoked potential caused by a simple acoustic reference is lower than 1 second.

Each valid trial consisted of an initial acoustic reference time followed by a self-chosen period of no motor activity higher than 3 seconds, an execution of the motor task and an additional 10s period of time of no motor activity. An average of 5 runs of 3 minutes each were recorded with all the subjects. This resulted in an average set of 50 trials performed per subject. The average time interval between movement executions was 21.31s.

2.4 Online Classifier Implementation

The main issues to take into account when characterizing the ERD in the intervals of EEG data preceding a voluntary movement are: 1) The desynchronization is best seen over the motor area, and it presents asymmetric spatial patterns [9]; 2) The desynchronization may be detected in both the alpha and beta bands and the specific frequencies at which it is most prominent are subject-dependent [8]; 3) The ERD patterns present high inter- and intra-subject variability [3].

The variability among subjects, and also among different days of measurements with a same subject, makes essential to adapt the classifier to each subject measured. The design of an ERD-based movement intention detector that is optimized for each measured subject requires taking into account his/her best desynchronization channels and frequencies.

Feature Extraction: For each one of the 13 channels, the optimal desynchronization frequency is obtained by comparing the average basal and movement power spectral densities (PSDs) of the training trials. The movement intervals are taken in each trial from 2 seconds before the movement onset until the movement ending. The basal intervals are taken from each movement ending until 3 seconds before the next movement onset. The PSDs are obtained with a frequency resolution of 0.5 Hz. The best frequency for each channel is obtained by maximizing the normalized distance between the mean power values of all the training trials at each frequency.

In each channel, the average ERD is obtained at its optimal frequency, by filtering the training trials with Butterworth band-pass filters of 2Hz width (4th order). The resulting curves are used for estimating the anticipation of the desynchronization for each channel. The four most predictive channels are selected for the subsequent modeling and classification of the pre-movement state.

Movement prediction, classifier update and threshold selection: The training examples used to generate the model of the premovement state correspond to the logarithmic power values from 1 second before each movement onset to 1 second after the same movement onset for all the training trials. As the logarithm of the power values measured in each channel follow a normal distribution, Gaussian functions are used for modeling the premovement state.

A Naïve Bayes classifier is used for deciding whether each new observed example obtained from the validation EEG signal corresponds to a premovement state. The classifier uses as input data the logarithmic power values obtained in each one of the four selected channels at their corresponding optimal frequencies. The Welch's method with Hamming windowing is used in order to get the power estimations for time intervals of 2 seconds length.

The probability of the premovement condition is evaluated separately in each one of the four selected channels and the final probability that the Bayesian classifier outputs is the combination of the four estimations.

The trade-off between the number of movements anticipated and the number of false activations generated, as well as the length of the estimations of movement intention are used to select the optimal threshold of the classifier.

3 Results

The system's validation was performed offline. The classification of each run used the rest of the runs measured with the same subject as the training dataset. The test runs were classified continuously (each complete 3 minutes run is analyzed, as in an online scenario). No artifacts were removed in order to simulate an online scenario.

In order to evaluate the classifier performance we decided to follow an analysis mode based on events instead of a sample-by-sample test. This approach means that the parameters selected for the validation and optimization of the detector are based on activation units (AU), which are the consecutive set of 2 or more classifier estimations that are over the decision threshold, so these consecutive activations may be considered as a singular classification representing that an ERD process must be stable on time [13]. According to this approach, we define the conditions for an AU to be classified as a true-positive classified event (TP) or as a false positive classified event (FP) and two indicators for testing the classification results based in these two values:

$$Precision = TP / (TP + FP) \quad Recall = TP / (TP + FN)$$

where

TP (true positive): AU intersecting or containing the interval $[-0.5 : 0]$.

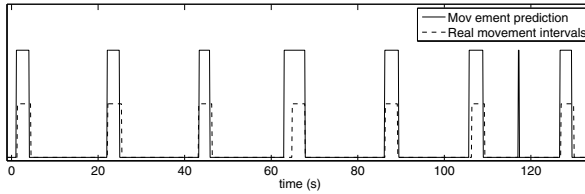
FP (false positive): AU intersecting the interval (*MovementEnding* : -0.5).

FN (false negative): Non detected movement.

Based on these criteria, a summary of the results achieved with the 4 subjects of study is shown in Table 1. The classifier is able to anticipate an average of 64% of voluntary movements and in the three best cases the recall is over

Table 1. Classification results of the four subjects

Code	Number of movements	Recall (%)	Precision (%)	Recall (%) (without anticipation)	Precision (%) (without anticipation)	Avg. L. true AUs (s)	Avg. L. false AUs (s)	Total L. False AUs / Total L. Rest State (%)	Mean Anticipation (s)
C01	40	70	42%	95	49	3.50	0.70	2.7	-0.82
C02	43	72	36%	91	42	4.13	1.20	11.6	-1.22
P01	65	34	23%	91	44	3.43	0.95	7.2	-0.99
P02	60	80	69%	98	73	3.29	0.51	1.1	-0.70

**Fig. 1.** Seven movements detected and one wrong activation before the last movement

70%. The average precision result (42%) represents a reduced percentage of false activations considering that, in average, the 81% of the time corresponds to the basal state of the subjects while the premovement condition is found just in the 2.5% of the time measured (the remaining 16.5% of the time corresponds to the movement intervals). Moreover, as can be observed in table 1, the average ratio between the total length of wrong activations and the total length of the basal state is 5.6%, and for the two best cases (C01 and P02), this ratio is just 2.7% and 1.1% respectively. This means that in the best case, around 5 false predictions in a minute (480 prediction per move are generated) will be obtained. These results represent a low percentage of false activations during long intervals of inactivity, which are critical for the correct performance of the BCI switch.

An interval of the patient P02 with 7 correctly classified movements and 1 wrong activation during 120 seconds of the protocol is shown in Figure 1.

On the other hand, the classification results obtained with P01 show that the anticipation in this case is not achieved (only a recall of 34% and a precision of 23%), although, as can be inferred from the movement detections without anticipation results shown in columns 3 and 4 of table 1, the detection of the movement related desynchronization is still good for this patient (the recall and precision results of the detection of movement executions, are similar to the results obtained with the rest of the measured subjects). This may be because the patient's ERD characteristics do not present average anticipations large enough.

Finally, the amount of anticipation obtained with this detector can be observed in the last column of table 1. These anticipation results were obtained from the onsets of the true positive AUs and they show that an average anticipation of 0.93 seconds is achieved. These results reflect the high accuracy of the detector for time locating and anticipating movement onsets.

4 Discussion and Conclusion

This study presents a BCI-switch capable of detecting the intention to move in control subjects and in subjects suffering from tremor-related pathologies. The detector is based on the characterization of the ERD on those channels and frequencies where it anticipates the most with respect to the movement onset. The ERD features vary among subjects, and in some cases it has been observed a lack of anticipation of the voluntary movement affecting the detection of movement intention, while the movement detection without anticipation was still possible. The capability of anticipating movements in such cases should be accomplished by reducing the power estimation windows, and this could increase seriously the amount of false detections during intervals of non-activity. The selection of the classifier parameters as the optimal threshold, the power estimation window length, or the number of channels taken into account for the ERD characterization need to consider all the factors that describe the detector's performance, such as the recall, the precision, the number and length of the false activations per minute or the distances between the AUs' onsets and the movement onsets. It will be further investigated how to select the optimal parameters in an online situation, so that the best classification results can be achieved with each user.

The main purpose of the system here presented was to be able to adapt to each subject characteristics, and because of this reason, the results obtained depended on the particular features of the ERD for each subject.

Finally, the Bayesian classifier proposed allows updating the detector's parameters every time a new movement is detected. With the accumulated knowledge, the actual online design of the BCI detector presented in this study is possible and will be accomplished in future works.

The ability to anticipate voluntary movements in control subjects and tremor patients has been evaluated during an experimental protocol presenting long intervals of subjects' inactivity. The anticipatory property of the ERD phenomenon has been used in order to develop the movement intention detector, and a methodology has been proposed for the detector's automatic configuration. The results achieved with three out of four subjects are encouraging, with recall results better in some cases than in the paper of Bai et al. [2]. The detector's ability to detect the mental states of interest has proven to be highly accurate despite the fact of detecting the previous intervals of the voluntary movements, where a high variability among single trials is observed.

Acknowledgements

This work was supported by the Spanish MINISTERIO DE CIENCIA E INNOVACIÓN within the IMAGENeuroMAPS Project (TEC2006-13966-C03-03, Integration of Magnetic Resonance and Electroencephalography. Application to the Fundamentals and Use of Brain Computer Interfaces by Disabled People), and by the European Communities within the TREMOR Project (FP7-ICT-2007-224051, An ambulatory BCI-driven tremor suppression system based on functional electrical stimulation). e-mail: jaime.ibanez@iai.csic.es.

References

1. Bai, O., Lin, P., Vorbach, S., Li, J., Furlani, S., Hallett, M.: Exploration of computational methods for classification of movement intention during human voluntary movement from single trial EEG. *Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology* 118, 2637–2655 (2007)
2. Bai, O., Rath, V., Lin, P., Huang, D., Battapady, H., Fei, D.Y., Schneider, L., Houdayer, E., Chen, X., Hallett, M.: Prediction of human voluntary movement before it occurs. *Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology* (July 2010)
3. Blankertz, B., Dornhege, G., Lemm, S., Krauledat, M., Curio, G., Müller, K.: The Berlin Brain-Computer Interface: Machine learning based detection of user specific brain states. *Journal of Universal Computer Science* 12(6), 581–607 (2006)
4. Daly, J.J., Wolpaw, J.R.: Brain-computer interfaces in neurological rehabilitation. *The Lancet Neurology* 7(11), 1032–1043 (2008)
5. Lim, V.K., Hamm, J.P., Byblow, W.D., Kirk, I.J.: Decreased desynchronization during self-paced movements in frequency bands involving sensorimotor integration and motor functioning in Parkinson's disease. *Brain Research Bulletin* 71(1-3), 245–251 (2006)
6. Mason, S.G., Birch, G.E.: A Brain-Controlled Switch for Asynchronous Control Applications. *IEEE Transactions on Biomedical Engineering* 47(10), 1297–1307 (2000)
7. Morash, V., Bai, O., Furlani, S., Lin, P., Hallett, M.: Classifying EEG signals preceding right hand, left hand, tongue, and right foot movements and motor imageries. *Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology* 119(11), 2570–2578 (2008)
8. Pfurtscheller, G., Lopes da Silva, F.H.: Event-related EEG/EMG Synchronization and Desynchronization: Basic Principles. *Clinical Neurophysiology* 110, 1842–1857 (1999)
9. Pfurtscheller, G., Brunner, C., Schlögl, A., da Silva, F.H.L.: Mu Rhythm (de)synchronization and EEG Single-Trial Classification of Different Motor Imagery Tasks. *Neuroimage* 31, 153–159 (2006)
10. Popescu, F., Fazli, S., Badoer, Y., Blankertz, B., Müller, K.R.: Single trial classification of motor imagination using 6 dry EEG electrodes. *PloS one* 2(7), e637 (January 2007)
11. Shibasaki, H., Hallett, M.: What is the Bereitschaftspotential? *Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology* 117(11), 2341–2356 (2006)
12. Solis-Escalante, T., Müller-Putz, G., Brunner, C., Kaiser, V., Pfurtscheller, G.: Analysis of sensorimotor rhythms for the implementation of a brain switch for healthy subjects. *Biomedical Signal Processing and Control* 5(1), 15–20 (2010)
13. Townsend, G., Grainmann, B., Pfurtscheller, G.: Continuous EEG Classification During Motor Imagery-Simulation of an Asynchronous BCI. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 12(2), 258–265 (2004)

Auditory Brain-Computer Interfaces for Complete Locked-In Patients

M.A. Lopez-Gordo¹, Ricardo Ron-Angevin¹, and Francisco Pelayo Valle²

¹Department of Electronic Technology, University of Malaga
malopezg@uma.es, rra@uma.es

²Department of Computer Architecture and Technology, University of Granada
fpelayo@ugr.es

Abstract. Brain-computer interfaces (BCIs) are intended for people unable to do any muscular movement such as complete locked-in patients. Most of the BCIs make use of visual interaction with the user, either in form of stimulation or biofeedback. However, visual BCIs challenge the ultimate use of BCIs because they require the subjects to gaze, explore and coordinate the eyes using their muscles, thus ruling out complete locked-in patients. Despite auditory BCIs overcome the problem of the visuals, there are not many examples of them in the BCI literature. In this paper we review the research and main contributions to auditory BCIs, and compare them with visual BCIs, especially to communicate with complete locked-in patients.

Keywords: Brain-computer Interface, auditory, Event-related potential, EEG, vegetative, complete locked-in.

1 Introduction

The classical definition of BCI established in [1] says “A brain–computer interface is a communication system that does not depend on the brain’s normal output pathways of peripheral nerves and muscles”. Since then, the vast majority of these interfaces use visual interaction with the subjects either as stimulation or as biofeedback. Examples are the BCIs based on steady-state visual evoked potentials (SSVEP-BCI) or P300 (P300-BCI). In the SSVEP-BCI a set of flickers with different temporal frequencies [2][3] [4] or phase shifts [5] [6] are presented to the subject in a display or by means of LEDs controlled by external circuitry. Each flicker elicits a vigorous sinusoidal-shaped response of the same temporal frequency and phase shift as its corresponding flicker. Thus, the subject normally must gaze directly at one of the stimulus to elicit its corresponding response that will be easily recognized, classified and codified as a symbol. In the P300-BCI the response is evoked by means of an array of flashing characters [7][8][9][10]. In the typical oddball paradigm the subject must gaze at the specific character that he wants to communicate and perform a cognitive task, normally counting the number of times that it flashes. This elicits a P300 associated to the character and hence classified with accuracy. The underlying fundament of both types of BCIs is somehow contradictory with the definition of BCI since excludes patients unable to control their eye motion properly (e.g. patients in an advanced stage of sclerosis lateral amyotrophic with visual impairment or the vegetative).

In the two types of BCIs mentioned before, the subject must be able to explore a visual scene, shift gaze to different positions and stare during the length of a trial. An example of this is given in [11], where augmented reality is used to control a robot; the control of a prosthesis by means of flashing lights [2]; or the use of virtual reality environments to improve the feedback control, specifically for untrained subjects [12]. Finally, the experiment performed in [13] shows that the performance of a P300-BCI speller in healthy subjects depends in considerable measure on gaze direction.

There are few exceptions in the literature of visual BCI that do not require gazing. An example of that is [14]. The study presents a binary SSVEP-BCI that allows an efficient communication without the need of gazing. The information transmission rate (ITR) obtained was 0.64 bits per seconds and an accuracy of 90%. To achieve this performance two main factors were involved: on the one hand, the use of both amplitude and phase of the SSVEP in the classification and, on the other, the use of the absence of gaze (the “thousand-yard stare”). In [15] different conditions used overlapping or non-overlapping images to reveal dependence on gaze function. The data demonstrated that SSVEP differences sufficient for BCI control may be elicited by selective attention to one of two overlapping stimuli. Thus, some SSVEP-BCI may not depend on gaze control.

In broad words, and with some exceptions, visual BCIs do not work properly without the visual ability to explore, to gaze and a suitable ocular motion. In this regard, an auditory BCI seems a more appropriate solution for users unable to control their eyes. In the rest of this paper we present some examples of auditory BCIs and how they can extend the group of users beyond the visuals, namely, the complete locked-in or the vegetative.

2 Auditory BCIs

Despite the singular definition of a BCI, there are just few efforts to use other more suitable modality of stimulus than the visual such as the auditory. A justification for this is that BCIs based on the latter modality of stimuli have given rise to systems of lower performance and usability than the visuals, able to perform above 1 bit/second with an accuracy around 90% [16].

2.1 Oddball Paradigm: P300

For instant, in [17] sixteen healthy volunteers participated in an experiment in which they learned to increase or decrease the amplitude of sensorimotor rhythms. Half of the participants were presented with visual and half with auditory feedback. Participants in the auditory feedback group learned slower, but four out of eight reached, at the end of the experiment, an accuracy of over 70% comparable to the visual feedback group. They concluded that with sufficient training time an auditory BCI may be as efficient as a visual BCI. In [9] a two-group design was used to ascertain whether participants benefited from visual cues early in training. Group A received only auditory stimuli, whereas Group AV received simultaneous auditory and visual stimuli in initial sessions after which the visual stimuli were systematically removed. The results showed that the two groups achieved equivalent accuracy with

mean bit rates averaged about 2 bits/min, and maximum bit rates reached 5.6 bits/min. Another study with P300 [18] was designed to test a spelling system based on auditory BCI. The spelling system was tested on a group of healthy volunteers. Compared to the visual spelling system, users’ performance was lower. It was concluded that auditory evoked ERPs from the majority of the users could be reliably classified with high accuracies among these users. From all these studies, it can be concluded that the implementation of an auditory P300 BCI is a feasible possibility that achieves a reasonable classification accuracy and communication rate.

2.2 Dichotic Listening

The authors of [19] presented a novel fully auditory EEG-BCI based on a dichotic listening to natural speech. A dichotic listening [20] was established by two streams of stimuli, one per ear, that were presented concurrently to the subject through earphones with a repetition rate of 5 stimuli per second (ISI=200 ms) and interleaving interval (ILI) of 100 ms between streams (see Fig. 1). Two conditions were established: First, a dichotic listening condition wherein two distinct streams of letters were read out simultaneously (Condition 1) and second, the same as Condition 1 except that complete speeches were used rather than letters (Condition 2). With this simple procedure, it was achieved a reasonable performance in terms of ITR and accuracy, even with very short trials. Actually, in the experiment the maximum averaged ITR (6.3 bits/minute and 2.2 bits/minute for Session 1 and 2 respectively with healthy volunteers) was achieved with trials of less than one second of duration.

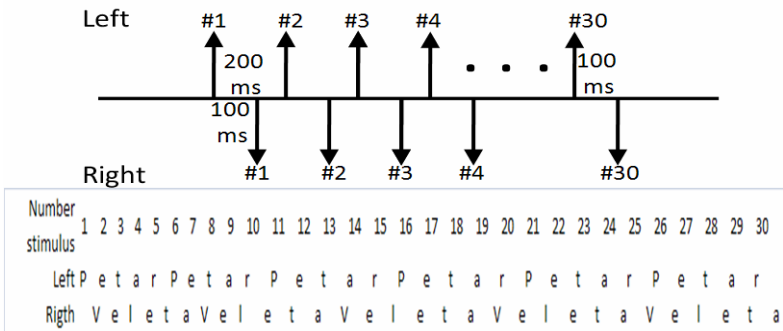


Fig. 1. Upper figure: Shows the typical structure of a trial of Condition 1. After the high beep the trial starts with two streams of 30 stimuli per ear (ISI=200 ms, ILI=100 ms). For the. Bottom figure: An example of the two streams of stimuli where the Spanish words “Petar” and “Veleta” are used. Adapted from [19].

This novel technique is of the interest of patients in an advanced state of cerebral deterioration, for whom a simple paradigm with a low cognitive effort is essential. The dichotic listening permits a more natural communication with complete locked-in or vegetative patients who still preserve a minimum level of consciousness, with promising results in terms of performance, usability, training and cognitive effort.

2.3 Other Paradigms

Some variants of the auditory P300-BCI based on the standard oddball paradigm have been also tried. The authors of [10] proposed an auditory BCI based on a paradigm similar to the standard oddball but including an additional target (i.e. two target stimuli, one frequent stimulus). Three versions were evaluated on twenty healthy participants in which the target stimuli differed in loudness, pitch or direction. It was achieved an average information transfer rate of up to 2.46 bits/min and accuracies of 78.5%. Other paradigms, such as the auditory Imagery of a familiar tune and Spatial Navigation Imagery through a familiar environment was used in [21]. In this experiment the maximum accuracy obtained 70.05% for the best combination of pair of electrodes used.

3 BCIs for the Vegetative State

Both auditory and proprioceptive BCIs are more convenient communication ways than the visual for complete locked-in patients [22]. The auditory BCIs have the benefit of extending their potential users beyond the visuals. Together with the locked-in, there is another category of patients who could benefit of a pure auditory BCI without any visual input, namely, patients with disorders of consciousness or vegetative who still preserve the necessary level of consciousness to process auditory stimuli and follow statements. In this regard, there are several studies that demonstrate that patients diagnosed as vegetative with the classical clinical procedures, still retain the ability to understand spoken commands and to voluntarily modify their brain activity [23][24][25][26], which are the necessary conditions to operate a BCI.

Different studies coincide in the convenience of a hierarchical approach to the assessment of cognitive function of patient in the vegetative state that have evaded detection using standard clinical methods. In [27] the assessment was established by means of an auditory scheme, from the speech recognition (lowest level) to the execution of volitional tasks ordered by means of verbal statements (highest level). In a first study the authors used the fMRI of a vegetative patient to measure neural responses during the presentation of spoken sentences, which were compared with responses to acoustically matched noise sequences (lowest level). Despite the appropriate neural response of the patient, it was not an unequivocal evidence that she was consciously aware. Therefore a second study was performed in which the vegetative patient was given verbal instructions to carry out two mental imagery tasks, namely spatial navigation and play tennis). This time, her neural responses were indistinguishable from those observed in healthy volunteers, thus being an unequivocal evidence of understanding external verbal instructions and the ability to execute volitional actions.

In [28] it was possible the assessment of cognitive function of patient in the vegetative state by means of an auditory-guided scheme in a fMRI study (Fig. 2). At the most basic level, sound perception was assessed by comparing auditory stimuli to silence. Once a response to sound was established, a speech-specific response was

assessed by comparing speech sounds to noise. The next level was the comparison between ambiguous and unambiguous sentences as a measure of comprehension. Finally, volition was assessed by evaluating imagery responses to verbal instructions. The patient who was being evaluated under this scheme completed all the levels. When the patient who was clinically diagnosed as vegetative was asked to imagine spatial navigation and play tennis, her response was statistically indistinguishable from that of healthy volunteers. Finally, in this study it is concluded that, despite the patient was diagnosed as vegetative, she retained the ability to understand spoken commands and to respond to them through her brain activity. In other words, she was a suitable candidate to control a BCI.

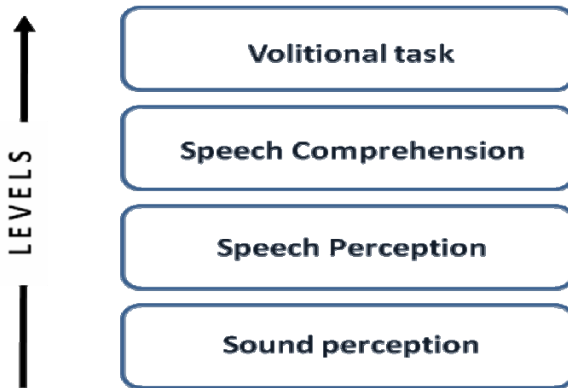


Fig. 2. A hierarchical approach to the evaluation of cognitive functions in the vegetative patients. Adapted from [28].

The fMRI analysis lacks of the necessary responsiveness for an efficient implementation of a BCI, thus bringing the EEG analysis a good chance to work around this disadvantage. In [29] it is proposed a different auditory hierarchical scheme that extends the scope of the fMRI studies reviewed before. It includes, at the top level, a choice of cognitive tasks, namely Oddball or imagery tasks, that are used as part of a BCI (see Fig 3). All the levels of the scheme are evaluated using EEG. As a requisite prior to the application of this auditory scheme is the presence of the N1 and P2, that ensures the participation of the auditory cortex in the processing of the stimulus. Then, the hierarchical scheme evaluates the presence of EEG activity beyond the 4Hz and follows with the detection of P300 and N400 after passive stimulation. In the following level a passive stimulation with instructions is presented. If the amplitudes of P300 and N400 are larger than those obtained in the previous level, then the volitional tasks will be presented. At this level either a 4-choice oddball-speller either the BOLD response during mental imagery or both will be performed. The scheme ends testing if the response of the last volitional level is good enough to control a BCI.

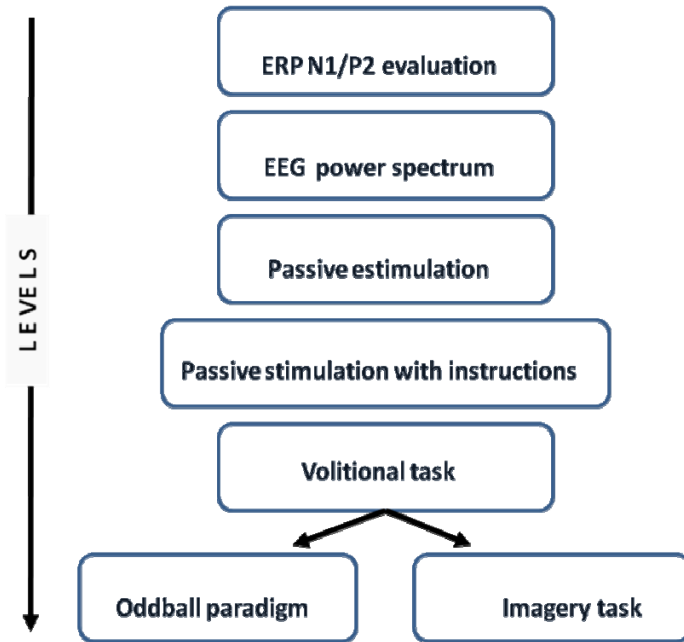


Fig. 3. Flow chart of an auditory hierarchical approach to use a BCI with vegetative patients. Adapted from [29].

4 Conclusions

The majority of the visual BCIs do not work properly without the visual ability to explore, to gaze and to control the eye motion, thus ruling out the possibility of use with complete locked-in patients. Although the auditory BCIs seem a more appropriate solution for this type of users, the number of implementations is very low in comparison with the visuals, due to important factors such as the high accuracy and ITR than the latter can achieve. Despite everything, there is a category of users that can not be conveniently covered with the visual BCIs, but with the auditory, namely, the complete locked-in or the vegetative.

There are few paradigms used in auditory BCI, mainly the oddball for the P300-BCI. Novel paradigms such as the auditory imagery or the dichotic listening to natural speech have given place to auditory BCI with promising results in terms of performance, usability and cognitive effort enough for a practical implementation with CLI or vegetative patients.

Acknowledgments. This work was financed by the Innovation, Science and Enterprise Council of the Junta de Andalucía (Spain), projects “BRAINS” P07-TIC-03310 and P06-TIC-02007, as well as the CASIP research group (TIC117 University of Granada, Spain).

References

- [1] Wolpaw, J.R., et al.: Brain-Computer Interface Technology: A Review of the First International Meeting. *IEEE Transactions On Rehabilitation Engineering* 8(2), 164–173 (2000)
- [2] Muller-Putzy, G.R., Pfurtscheller, G.: Control of an Electrical Prosthesis With an SSVEP-Based BCI. *IEEE Transactions on Biomedical Engineering* 55(1), 361–364 (2008)
- [3] Bakardjian, H., Tanaka, T., Cichocki, A.: Optimization of SSVEP brain responses with application to eight-command Brain-Computer Interface. *Neuroscience Letters* 469(1), 34–38 (2010)
- [4] Shyu, K., Lee, P., Liu, Y.: Dual-frequency steady-state visual evoked potential for brain computer interface. *Neuroscience Letters* 483(1), 28–31 (2010)
- [5] Lopez-Gordo, M.A., Prieto, A., Pelayo, F., Morillas, C.: Use of Phase in Brain-Computer Interfaces based on Steady-State Visual Evoked Potentials. *Neural Processing Letters* 32(1), 1–9 (2010)
- [6] Wang, Y., Gao, X., Hong, B., Jia, C., Gao, S.: Brain-Computer Interfaces Based on Visual Evoked Potentials. *IEEE Engineering in Medicine and Biology Magazine* 27(5), 64–71 (2008)
- [7] Ikegami, S., Takano, K., Saeki, N., Kansaku, K.: Operation of a P300-based brain-computer interface by individuals with cervical spinal cord injury. *Clinical Neurophysiology* (2010)
- [8] Kleih, S., Nijboer, F., Halder, S., Kübler, A.: Motivation modulates the P300 amplitude during brain-computer interface use. *Clinical Neurophysiology* 121(7), 1023–1031 (2010)
- [9] Klobassa, D., et al.: Toward a high-throughput auditory P300-based brain-computer interface. *Clinical Neurophysiology* 120(7), 1252–1261 (2009)
- [10] Halder, S., et al.: An auditory oddball brain-computer interface for binary choices. *Clinical Neurophysiology* 121(4), 516–523 (2010)
- [11] Kansaku, K., Hata, N., Takano, K.: My thoughts through a robot's eyes: An augmented reality-brain-machine interface. *Neuroscience Research* 66(2), 219–222 (2010)
- [12] Ron-Angevin, R., Diaz-Estrella, A.: Brain-computer interface: Changes in performance using virtual reality techniques. *Neuroscience Letters* (2008)
- [13] Brunner, P., Joshi, S., Briskin, S., Wolpaw, J.R., Bischof, H., Schalk, G.: Does the 'P300' speller depend on eye gaze? *Journal of Neural Engineering* 7(5), 056013 (2010)
- [14] Lopez-Gordo, M.A., Pelayo, F., Prieto, A.: A high performance SSVEP-BCI without gazing. In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–5 (2010)
- [15] Allison, B.Z., McFarland, D.J., Schalk, G., Zheng, S.D., Jackson, M.M., Wolpaw, J.R.: Towards an independent brain-computer interface using steady state visual evoked potentials. *Clinical Neurophysiology* 119(2), 399–408 (2008)
- [16] Gao, X., Xu, D., Cheng, M., Gao, S.: A BCI-Based Environmental Controller for the Motion-Disabled. *IEEE Transactions On Neural Systems And Rehabilitation Engineering* 11, 137–140 (2003)
- [17] Nijboer, F.: An auditory brain-computer interface (BCI). *Journal of Neuroscience Methods* 167(1), 43–50 (2008)
- [18] Furdea, A., et al.: An auditory oddball (P300) spelling system for brain-computer interfaces. *Psychophysiology* 46(3), 617–625 (2009)

- [19] Lopez-Gordo, M.A., Fernandez, E., Romero, S., Pelayo, F.: A Brain-computer interface Based on Natural Speech. under revision
- [20] Meyer, J.E., Robert, R.J., Bayles, J.D., Volkert, K., Evitts, P.E.: Dichotic listening: expanded norms and clinical application. *Archives of Clinical Neuropsychology* 17, 79–90 (2002)
- [21] Cabrera, A.F., Dremstrup, K.: Auditory and spatial navigation imagery in Brain–Computer Interface using optimized wavelets. *Journal of Neuroscience Methods* 174(1), 135–146 (2008)
- [22] Murguialday, A.R., et al.: Transition from the locked in to the completely locked-in state: A physiological analysis. *Clinical Neurophysiology* (2010)
- [23] Owen, A.M., Schiff, N.D., Laureys, S.: A new era of coma and consciousness science. *Progress in Brain Research* 177, 399–411 (2009)
- [24] Coleman, M.R., et al.: Do vegetative patients retain aspects of language comprehension? Evidence from fMRI. *Brain* 130(10), 2494–2507 (2007)
- [25] Laureys, S., Perrin, F., Bredart, S.: Self-consciousness in non-communicative patients. *Consciousness and Cognition* 16(3), 722–741 (2007)
- [26] Boly, M., et al.: When thoughts become action: An fMRI paradigm to study volitional brain activity in non-communicative brain injured patients. *NeuroImage* 36(3), 979–992 (2007)
- [27] Owen, A.M.: Detecting Awareness in the Vegetative State. *Science* 313(5792), 1402–1402 (2006)
- [28] Owen, A.M., Coleman, M.R.: Detecting Awareness in the Vegetative State. *Annals of the New York Academy of Sciences* 1129(1), 130–138 (2008)
- [29] Kübler, A.: Brain–Computer Interfaces for Communication in Paralysed Patients and Implications for Disorders of Consciousness. In: Laureys, S., Tononi, G. (eds.) *The Neurology of Consciousness*, p. 440. Elsevier, Amsterdam (2009)

Brain-Computer Interface: Generic Control Interface for Social Interaction Applications

C. Hintermüller, C. Guger, and G. Edlinger

g.tec medical engineering GmbH and Guger Technologies OG
Schiedlberg, Austria
hintermueller@gtec.at

Abstract. After suffering a more severe disease like spinal cord injury or stroke patients are often not able to interact or even communicate with their environment anymore, especially at the beginning of rehabilitation. Brain-computer interfaces (BCIs) can substitute this temporarily lost communication channels and might support rehabilitation by providing an alternative way for controlling a computer only by thoughts without any muscle activity. This enables the patient to communicate by writing letters on the screen, to stay socially in contact with friends or people outside the rehabilitation facility by participating in games like Second Life where they may appear as healthy persons. Another application is to control items in their room connected to the BCI system like the lights which can be turned off and on as it can be done in a virtual smart home without leaving the bed. In this paper a generic BCI interface is described which allows to control the aforementioned applications concurrently and transparently switch among them utilizing the P300 approach. The results of a recent study show that such a BCI can be used by patients suffering from cervical spinal cord injury almost as well as by healthy people which encourages us to think it may assist rehabilitation regarding the social aspect.

Keywords: brain-computer interface; BCI; P300; visual evoked potentials; speller; Second Life; Twitter.

1 Introduction

Many disorders, like spinal cord injury, stroke or amyotrophic lateral sclerosis (ALS), can affect or even completely damage the usual communication channels a person needs to communicate and interact with his or her environment. In such severe cases a brain-computer interface (BCI) might be the only remaining possibility to communicate [1]. An EEG-based BCI provides a new non-invasive communication channel between the human brain and a computer. The electrical brain activity during predefined mental tasks is analyzed and translated into corresponding actions intended by the user. But even for less severe levels of affection a BCI can improve quality of life allowing to control a computer, specially prepared electronic devices, or to stay in contact with friends through social networks and games, for example.

P300 evoked potential based BCIs can provide goal-oriented control and are mainly used for operating spelling devices [2], controlling computer games [3] or

performing navigation tasks e.g. moving a computer mouse [4]). The P300 evoked potential is elicited when an unlikely event occurs randomly between events with high probability. It manifests itself in a positive deflection in the amplitude of the EEG signal around 300 ms after a visual stimulus onset. Among the different classification techniques evaluated for P300 spellers, Fisher's linear discriminant analysis and the stepwise linear discriminant analysis yielded very robust performances [5]. The training of these classification methods to the individual brain activity can be accomplished within several minutes [6]. Guger et al. demonstrated that more than 70% of a sample population is able to use such a spelling setup with an accuracy of 100% [9].

This paper presents a generic UDP, XML based BCI command interface (according to the recently introduced international BCI interface standard, see www.gtec.at) and its application to implement control interfaces enabling social interactions like spelling, interacting virtually with other participants in Second Life, operating Twitter and controlling a virtual smart home. All of these applications are based on the P300 speller principle.

2 Material and Methods

For a P300 spelling device commonly a 6x6 matrix of different characters and symbols is presented on a computer screen [7]. It can be operated in a single-character mode where all characters are flashed individual in a random order or in row-column mode where a whole row or a whole column flashes. The flashing of exactly the character the subject wants to spell and thus is concentrating on is a relative unlikely event in either mode. Thereby a P300 component is induced in the EEG signal measured at 8 electrode positions mostly over occipital and parietal regions (see Figure 1) [8]. The P300 component reaches its maximum amplitude around 300 ms after the onset of the flash. For all other characters, rows or columns irrelevant to the subject no such P300 component is elicited.

During the training phase an arbitrary word like LUCAS or sequence of at least 5 control symbols is used to allow the BCI system to distinguish the characters or symbols the subject is supposed to concentrate on (targets), one after each other, from the remaining ones (non-targets). Whenever the currently focused letter flashes, the subject increments a mentally running counter. The EEG data of a specific time interval around each flash is used to train a linear discriminant analysis (LDA) classifier to distinguish between the typical EEG signal of the target and non target characters. The LDA classifier which has been trained with only a few target characters out of 36 possibilities is now able to identify the unknown target character reliably.

The EEG data were recorded with a g.MOBIIlab+ biosignal amplification unit (g.tec medical engineering GmbH, Austria) at 256 Hz sample rate and transferred to the computer wirelessly via Bluetooth®. A MATLAB/Simulink model controls the interface masks, processes the received data via a specific device driver and dispatches the targeted commands via the described UDP XML message passing interface. A notch filter (50 Hz or 60 Hz) and a band pass filter were applied to the signals in order to eliminate possible artifacts before they were down-sampled to 64

Hz. Data from 100 ms before each flash onset to 700ms afterwards were filtered and down-sampled again to get 12 feature values (i.e. samples) per channel and flash. These data chunks were sent to the LDA to determine if a target character flashed or not. The subjects were sitting in front of a computer screen and were instructed to relax as much as possible.

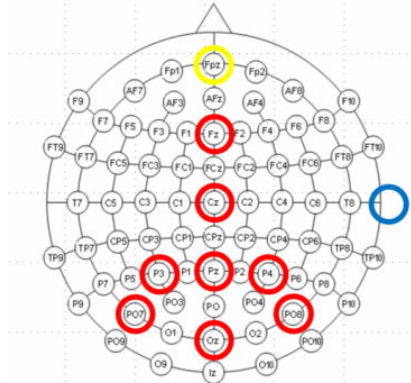


Fig. 1. Common electrode setup for P300 spellers according to [8]. Eight EEG electrodes are placed at Fz, Cz, P3, Pz, P4, PO7, Oz and PO8. The ground electrode is mounted on the forehead at Fpz (yellow circle) and the reference electrode is attached to the rig

2.1 Generic Control Interface

The interface was designed with the intention to control multiple applications and devices simultaneously and to provide a common way of sharing BCI control in a consistent manner among them. The various applications, like twitter (section 2.2) differ in the content of the P300 BCI matrix where letters may be replaced by symbols or phrases to control the associated applications.

In a first step the possible control states and the commands available in each state to move on to the next state are identified using unified modeling language (UML) diagrams, which yield a simple and clear overview of all required elements. Based on the diagrams, the detailed descriptions of the states and the actions to be taken along with the transition to the next state, an XML file describing the required masks, the position of the different symbols within the mask and the commands to call is generated for each application. Upon startup the interface description of the different applications are merged and symbols are added for switching among them. Figure 2 shows the UML diagram for the twitter application and the P300 mask resulting thereof. Whenever a control symbol or character is selected by the user, the BCI emits the command or character string associated with the symbol or executes the related action.

2.2 P300 Twitter Interface

Through the social network Twitter (Twitter Inc.) users can exchange messages. These messages can be sent over the Twitter website, by smart phones or by SMS

(Short Message Service). They're displayed on the author's profile page and are limited to 140 characters per message. Interfacing possibilities can be extended through the application programming interface provided by Twitter.

The basic BCI control mask was extended to add the necessary Twitter control commands in the first two lines (see Figure 2.a). The remaining characters are used for spelling resulting in a total of 54 possible symbols. Figure 2.b shows the Unified Modeling Language (UML) diagram of required actions for using the Twitter service.

The system was initially trained with 10 target symbols for the BCI user. Then another user was asking questions through Twitter and the BCI user had to answer each question every other day. A total of 9 questions were asked which means that the BCI user had to use the P300 interface on 9 different days whereas they selected between 6 and 36 characters each day.

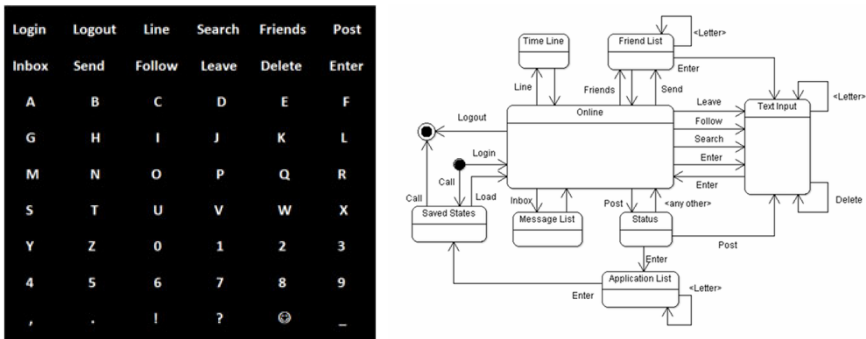


Fig. 2. The extended P300 interface control mask for Twitter (a) and the UML diagram of the actions required to use the Twitter service (b). Unlabeled actions indicate that the Twitter interface will execute them automatically. The switching to other Applications was implemented via the consecutive selection of “Post” and “Enter” command.

3 Results

To test the interface three applications, twitter, second life and a virtual smart home, were selected for the interface implementation. For the 3 systems the control states were identified using UML state diagrams (see Figure 2.b). Based on these, the control mask descriptions were generated merged and the symbols for switching between the applications were added. For the twitter application the switching was accomplished by a consecutive selection of “Post” and “Enter”, to display the list of applications to select from. With his approach the control mask (Figure 2.a), which was designed for the standalone version of the twitter applications, could be used without any change. The initial results achieved by a single subject for each of the applications are discussed in the following sections.

Table 1. Questions and text input with the P300 Twitter interface

Tweets	No. Characters	Duration [mm:ss]	Errors	No. Flashes	Time per character [s]
<i>Friend: Which kind of Brain-Computer Interface do you use?</i>					
BCI: P300 GTEC BCI	13	11:09	3	8	51
<i>Friend: Are you using the g.GAMMAsys?</i>					
BCI: EXACTLY!	7	06:18	1	8	54
<i>Friend: Active or passive electrodes? For explanation: the active system avoids or reduces artefacts and signal noise.</i>					
BCI: ACTIVE ELECTRODES	17	06:10	0	5	22
...					
<i>Friend: How long does it take to code the software for the BCI for TWITTER?</i>					
BCI: 3 WEEKS	7	03:13	1	4	28
<i>Friend: How many characters are you able to write within a minute?</i>					
BCI: 3 TO 4	6	03:15	0	5	33
<i>Friend: Did you get faster in writing during this period?</i>					
BCI: YES, FROM 2 TO 4 CHARACTERS	27	06:38	1	3	15

3.1 P300 Twitter Interface

Table 1 shows the results of the answers from the BCI user (bold sentences) to the 9 questions (italic sentences). The improvement of the user’s performance from the first session to the last one is impressive. For the first session it took the user 11:09 min to spell 13 characters with 3 mistakes. The user was instructed to correct any mistake before posting the answer on Twitter, yielding an average of 51 seconds selection time for each character. In contrast to this during the last session 27 characters were spelled within 6:38 min with only 1 mistake, with an average selection time of 15 seconds per character. At the same time the number of flashes per character decreased from 8 in the first session to only 3 flashes in the last.

3.2 Second Life and Smart Home Control

Compared to the twitter application three different hierarchically arranged control masks are required for the Second Life application. The P300 control interface for Second Life has been tested in first setups. Preliminary results indicate a very similar performance as the virtual smart home study conducted by Edlinger et al. [10]

In the virtual smart home study, some of the subjects achieved an accuracy of 95% in single character mode using between 3 and 10 flashes per character (mean 5.2 flashes per character) only. Table 2 lists the average accuracy achieved by the subjects

for each of the 7 control masks. Increased amplitudes of the P300 components in the EEG signals yielding better accuracies, can be observed on the larger masks compared to the smaller ones. This can be explained by the decreased probability of occurrence for each single character and therefore increased time which is required to flash all symbols.

Table 2. Accuracy, number of symbols, occurrence probability per symbol, number of flashes per symbol and mask (e.g. $25 \times 15 = 375$) and selection time per symbol for each mask

Mask	Accuracy	No. Symbols	Probability	No. Flashes	Time per character
Light	65.28%	25	4%	375	33.75 s
Music	76.11%	50	2%	750	67.50 s
Phone	63.89%	30	3.3%	450	40.50 s
Temperature	76.39%	38	2.6%	570	51.30 s
TV	65.74%	40	2.5%	600	54.00 s
Move	75.93%	13	7.7%	195	17.55 s
Go to	26.39%	22	4.5%	330	29.70 s

Interestingly, the subjects achieved only 26.39% accuracy for the “Go to” mask which is quite bad compared to the Light and Move control masks which have similar numbers of symbols. We think this is due to the different layout of the “Go to” mask. While all the other control masks were arranged in a matrix layout the symbols of “Go to” mask were placed at apparently random positions related to the appearance of the virtual apartment. More detailed results of the virtual smart home study can be found in [10].

4 Discussion

Extending the standard P300 speller by icons and symbols enables the user to control more complex scenarios. In [10] it is shown that a proper design of interface masks allows control of domotic devices in a virtual smart home study with comparable reliability as it has been reached for the spelling application. This paper discusses the usage of a BCI interface, to take part in social environments like controlling characters and actions in Second Life or operating other social interaction applications like sending or answering Twitter messages. The generic BCI interface which uses XML based description files simplifies the definition of the control masks. By merging the control masks of several applications like Twitter, second life and devices and also environments, like a virtual smart home, it is possible to use the different systems concurrently. To switch to another application it is sufficient to select the symbol or enter the symbol sequence used to load its control mask. If not running the application will be started and initialized to receive the control commands from the BCI.

In a recent study Guger et al. [9] evaluated a P300 speller (6 x 6 character matrix) in row-column mode with 100 healthy subjects (32 female and 68 male at the age of 27.9 ± 10.9) using a similar setup as described in this study. Using a subset of 81

subjects a total of 88.9% of them were able to use the speller with an accuracy level of 80-100% whereas 72.8% reached an accuracy of 100%. The average accuracy level was 91.1% with a spelling time of 28.8 seconds for a single character. Since all applications described in this study are based on the methods and the setup described in [9] we conclude from the presented case studies that comparable good results with respect to accuracy and usability as presented in [9] can be reached.

Acknowledgment

The research leading to these results has received funding from the European Community's, Seventh Framework Programme FP7/2007-2013 under BrainAble project, grant agreement n° 247447. The authors further gratefully acknowledge the funding by the European Commission under contract FP7-ICT-2009-247935 (Brain-Neural Computer Interaction for Evaluation and Testing of Physical Therapies in Stroke Rehabilitation of Gait Disorders) and under contract IST-2006-27731 (PRESENCIA).

References

1. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. *Clin. Neurophysiol.* 113(6), 767–791 (2002)
2. Farwell, L.A., Donchin, E.: Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalogr. Clin. Neurophysiol.* 70(6), 510–523 (1988)
3. Finke, A., Lenhardt, A., Ritter, H.: The MindGame: A P300-based brain-computer interface game. *Neural Networks* 22(9), 1329–1333 (2009)
4. Citi, L., Poli, R., Cinel, C., Sepulveda, F.: P300-Based BCI Mouse With Genetically-Optimized Analogue Control. *IEEE Trans. Neural Syst. Rehabil. Eng.* 16(1), 51–61 (2008)
5. Krusienski, D.J., Sellers, E.W., McFarland, D.J., Vaughan, T.M., Wolpaw, J.R.: Toward enhanced P300 speller performance. *J. Neurosci. Methods* 167(1), 15–21 (2008)
6. Ortner, R., Bruckner, M., Prückl, R., Grünbacher, E., Costa, U., Opisso, E., Medina, J., Guger, C.: Accuracy of a P300 Speller for People with Motor Impairments. In: *Proceedings of the IEEE Symposium Series on Computational Intelligence* (2011) (in press)
7. Donchin, E., Spencer, K.M., Wijesinghe, R.: The mental prosthesis: assessing the speed of a P300-based brain-computer interface. *IEEE Trans. Rehabil. Eng.* 8(2), 174–179 (2000)
8. Sellers, E.W., Krusienski, D.J., McFarland, D.J., Vaughan, T.M., Wolpaw, J.R.: A P300 event-related potential brain-computer interface (BCI): The effects of matrix size and inter stimulus interval on performance. *Biological Psychology* 73(3), 242–252 (2006)
9. Guger, C., Daban, S., Sellers, E.W., Holzner, C., Krausz, G., Carabalona, R., Gramatica, F., Edlinger, G.: How many people are able to control a P300-based brain-computer interface (BCI)? *Neuroscience Letters* 462(1), 94–98 (2009)
10. Edlinger, G., Holzner, C., Groenegress, G., Guger, C., Slater, M.: Goal-Oriented Control with Brain-Computer Interface. In: *Proceedings of the 5th International Conference on Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience*, pp. 732–740. Springer, Heidelberg (2009)

Variable Selection in a GPU Cluster Using Delta Test

A. Guillén¹, M. van Heeswijk², D. Sovilj², M.G. Arenas¹, L.J. Herrera,
H. Pomares¹, and I. Rojas¹

¹ Department of Computer Architecture and Computer Technology Universidad de Granada, Spain

² Department of Information and Computer Science,
Aalto University School of Science, Finland

Abstract. The work presented in this paper consists in an adaptation of a Genetic Algorithm (GA) to perform variable selection in an heterogeneous cluster where the nodes are themselves clusters of GPUs. Due to this heterogeneity, several mechanisms to perform a load balance will be discussed as well as the optimization of the fitness function to take advantage of the GPUs available. The algorithm will be compared with previous parallel implementations analysing the advantages and disadvantages of the approach, showing that for large data sets, the proposed approach is the only one that can provide a solution.

1 Introduction

The problem of variable selection is crucial by the time of design models that classify or perform regression so, the better the selection is, the more accurate models can be designed [3,13]. The approach to be taken in classification problems differs from the one taken in regression due to their differences in the output type (discrete and continuous respectively) so, this paper will consider the regression problem also known as function approximation. Formally, the function approximation problem is to determine, given a set of input/output pairs $(\mathbf{x}_i, y_i) \in R^d \times R$ $i = 1 \dots N$, design an unknown function F such as $F(\mathbf{x}_i) \approx y_i$. From this, the problem of variable selection can be defined as the search for the subset of variables that make possible to build a model that approximates the data as accurately as possible. Genetic Algorithms have been applied to many problems and variable selection is not an exception, however, the use of non-parametric noise estimators (independent from the models designed afterwards) has not been widely treated [11]. This paper presents a GA which has been implemented on a novel High Performance Computing (HPC) architecture using clusters of computers and Graphical Processing Units (GPUs) in order to compute the fitness of the individuals.

The rest of the paper is organised as follows: Section 2 introduces the Delta Test. Afterwards, Section 3 describes the design of the Parallel Genetic Algorithm that will perform the optimizations presented in the experiments included in Section 4. Finally, conclusions are discussed.

2 Delta Test in Variable Selection

In order to evaluate the goodness of an individual, the Delta Test (DT) [15] value obtained using the combination of variables will be used as it has been shown to be an adequate criterium [5]. The DT is a method to estimate the variance of the noise r_i , or the Mean Squared Error (MSE), that can be obtained without overfitting, this is:

$$y_i = F(\mathbf{x}_i) + r_i, \quad i = 1, \dots, N$$

where F is the unknown function. The DT can be formulated using the nearest neighbour formulation as

$$\text{Var}[r] \approx \delta = \frac{1}{2N} \sum_{i=1}^N (y_i - y_{NN(i)})^2,$$

with $\text{Var}[\delta] \rightarrow 0$ for $N \rightarrow \infty$

where the first nearest neighbour of a point \mathbf{x}_i in the R^d space is $\mathbf{x}_{NN(i)}$ and $y_{NN(i)}$ is the output of $\mathbf{x}_{NN(i)}$.

2.1 Computation of Delta Test Using Pre-calculated Distances

Computation of the nearest neighbour in the naive way involves calculating the distances between each pair of samples $\mathbf{d}_{i,j} = \sum_{m=1}^d (x_i^{(m)} - x_j^{(m)})^2$ and returning the smallest $\mathbf{d}_{i,j}$ and the corresponding index $NN(i)$ for each sample. Since the focus is on examining non-empty subsets of variables which can share individual elements, a lot of time is wasted recomputing the squared differences to obtain $\mathbf{d}_{i,j}$. A simple solution to decrease running time is to store that information into a $N(N-1)/2 \times d$ matrix, where each row contains precomputed squared differences for a pair of samples (x_i, x_j) . Given this matrix, computing all pairwise distances for a given variable subset $I \subseteq \{1, 2, \dots, d\}$ involves summing precomputed values for those I variables (i.e. the I -th columns of the matrix).

2.2 Computation of Delta Test on GPU

The computation of the k Nearest Neighbours (KNN) requires a big computational effort since it has to compute the pairwise distances between all the points. In [6] an implementation of the k -NN algorithm on a GPU [7] was presented, showing very large speed-ups compared to CPU times. We use this algorithm to determine the nearest neighbours ($\mathbf{x}_{NN(i)}$) to all input points (\mathbf{x}_i).

Differently from the approach in the previous subsection, the pairwise squared differences between all points are not pre-calculated, since it would not be feasible to keep this entire matrix in memory. However, even though we do not make

¹ The code is available at: <http://www.i3s.unice.fr/~creative/KNN/>

this optimization, computing the pairwise distances between the points can still be many times faster when using a fast GPU instead of the CPU.

Once all pairwise distances have been computed, a partial sort is performed in order to determine the nearest neighbour $\mathbf{x}_{NN(i)}$ (and their index $NN(i)$) to each of points \mathbf{x}_i . Finally, given these indices of the nearest neighbours, we can compute the Delta Test as explained in the beginning of this section.

3 Design of the Genetic Algorithm

The Genetic Algorithm (GA) is a well known optimization tool that has been applied to many problems.

3.1 GA Operators Description

When a GA is designed, there is a set of elements/operators that has to be defined depending on the problem to be solved. The first design decision is how an individual will represent a solution, this is, the solution encoding. The variable selection problem has a straight forward encoding using a binary chromosome whose length is equal to the number of variables. If a gene within the chromosome equals 1, the variable is selected, if it is 0, then the variable is discarded. This binary encoding has been widely treated in the literature so the classical, but effective, operators have been chosen:

1. Crossover: two-points binary
2. Mutation: Gene level, this is, if the individual is mutating select-unselect a random variable.
3. Selection: binary tournament selection
4. Elitism: Keep the best individuals from the previous generation.

3.2 Parallelizing the GA

Apart from the optimization of the computation of the fitness function using GPUs, the architecture available allows the algorithm to be distributed through several machines in a classical cluster manner. GAs are intrinsically parallel although modifications in the flowchart, distributing the populations, might improve results [20108].

Island Model. Among the several approaches proposed in the literature, the multi-deme distributed GA is one of the most popular due to its good behaviour [4812]. This implementation consists of evolving isolated populations on each island which is usually mapped into one processor. In our concrete case, since a processor might have several cores and several GPUs it might allocate several populations, one per each CPU/GPU pair.

In this type of algorithm, it is quite common to determine a migration operator or mechanism so the isolated islands share some individuals in order to collaborate and take advantage of the progress made by the others.

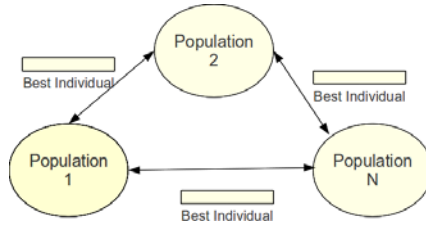


Fig. 1. Island migration scheme

The migration rate is a parameter that can be random [14], fixed [8], or autoregulated depending on the diversity of the population although this last approach can be looked from the perspective of the replacement policy, this is, migrate after a certain number of generations and, depending on the diversity of the population, accept the new individual.

The implementation selected in this algorithm was to migrate after 5 generations and always replaces the worst individual in the population by the incoming ones (in the same order they arrive from the other islands) as in [9]. The migration scheme is depicted in [1].

Population distribution in the cluster. Since we have a heterogeneous grid of computers, it is obvious that the time to complete a generation during the run will be different on each computer [1]. As the distributed populations perform a collective communication broadcasting individuals, the global performance of the algorithm would be injured if the fastest machines had to wait for the slower ones, wasting computing resources. In order to ameliorate this fact, the decision of setting different population sizes has been taken. Slower or overloaded machines will process less individuals, allowing these processes to require a smaller time to complete a generation. Therefore, during the collective communications, the waiting time for the synchronization will be reduced. Obviously, this approach is the same as increasing the predefined population size in the more powerful machines.

The question that arises from this policy is: how much the size of the population should be decreased/increased? The answer can be obtained empirically by measuring the time of one generation on each machine and obtaining the fraction between the fastest/slowest and the other time measurements. For example, if the time on *Machine*₁ is the double of *Machine*₂ the population size for *Machine*₁ should be the half (or the population size for *Machine*₂ should be doubled).

4 Experiments

4.1 Cluster Architecture

The cluster that was configured had the components described below that were interconnected as Figure 2 shows.

1. 1 Master node with 2 GPUs:

Processor:

- *model name* : (26) Intel(R) Core(TM) i7 CPU 930 @ 2.80GHz — *cache size* : 8192 KB
- *cpu cores* : 4 - *siblings* : 8

2 GPUs:

- *Graphics Processor*: GeForce GTS 450 — *CUDA Cores*: 192
- *Memory*: 1024 MB - *Memory Interface*: 128-bit
- *Bus Type*: PCIExpress x16 Gen1 - *PCI-E Max Link Speed*: 2500

2. 2 Local network node with 1 GPU:

Processor:

- *model name* : (23) Intel(R) Core(TM)2 Quad CPU Q9550 @ 2.83GHz — *cache size* : 6144 KB
- *cpu cores* : 4 - *siblings* : 4

GPU:

- *Graphics Processor*: GeForce 9800 GTX — *CUDA Cores*: 128
- *Memory*: 512 MB - *Memory Interface*: 256-bit
- *Bus Type*: PCIExpress x16 Gen2 - *PCI-E Max Link Speed*: 5000

Processor:

- *model name* : (15) Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz — *cache size* : 4096 KB
- *cpu cores* : 4 - *siblings* : 4

GPU:

- *Graphics Processor*: GeForce 8400 GS — *CUDA Cores*: 16
- *Memory*: 512 MB - *Memory Interface*: 64-bit
- *Bus Type*: PCIExpress x16 - *PCI-E Max Link Speed*: not available

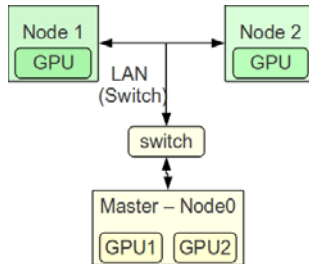


Fig. 2. Cluster of GPUs used in the experiments

4.2 Comparison with Previous Approaches

This section will analyse the performance and behaviour of the proposed algorithm. First, small data sets are compared with a previous work, afterwards, the algorithm is applied to a large real-world data set.

Small Datasets. The algorithm was compared with the approaches in [11] that was a previous parallel version of the algorithm implemented in a regular cluster. For the sake of a fair comparison, the stop criteria will be the same: execution time. In [11], the time limit is set to 600 seconds based on studies and recommendations from the industry arguing that is the maximum time that an operator is willing to wait to see a solution.

The same population sizes were used (50,100 and 150) and data sets processed were: 1) The Tecator data set²; the Tecator data set aims at performing the task

² http://lib.stat.cmu.edu/data_sets/tecator.

of predicting the fat content of a meat sample on the basis of its near infrared absorbance spectrum. The data set contains 215 useful instances for interpolation problems, with 100 input channels, and 3 outputs, although only one is going to be used (fat content). 2) The Anthrokids modified data set³: This data set consists on several measures to predict child’s weight, it has 1019 instances and 53 variables.

The results are shown in Table II. As the table reflects, the performance of the proposed algorithm is not impressive in comparison with the previous algorithm. In fact, it does not outperform the previous approach. However, it is remarkable that the difference between the solutions is not too big and, in some cases, the new algorithm outperforms the previous one demonstrating that the proposed approach is a good algorithm. The reason because the previous algorithm obtains, in general, better solutions for these data sets is because it is able to perform more generations due to the pre-calculation of the distance matrix as it was described in Section 2.

Table 1. Performance of the proposed algorithm (pGPU) against sequential and parallel algorithms for different data sets. Values of the DT obtained.

Data set	Population	Measurement	seq.	parallel(np=2)	parallel(np=4)	pGPU(np=4,4GPUs)
Anthrokids	50	Mean (DT)	0.01278 (11.5e-4)	0.01269 (14.2e-4)	0.01204 (12.6e-4)	0.01587 (8.1e-3)
	100	Mean (DT)	0.01351 (11.6e-4)	0.01266 (86.4e-4)	0.01202 (17.4e-4)	0.014553 (5.6e-4)
	150	Mean (DT)	0.01475 (12.1e-4)	0.01318 (11.2e-4)	0.01148 (9.9e-4)	0.01556(12e-4)
Tecator	50	Mean (DT)	0.13158(7.9e-4)	0.14297 (7.7e-3)	0.13976 (7.8e-3)	0.123803 (3.7e-3)
	100	Mean (DT)	0.13321 (3.1e-3)	0.13587 (2.4e-3)	0.13914 (8.6e-3)	0.132501 (3e-4)
	150	Mean (DT)	0.13146 (8.5e-4)	0.1345 (2.4e-3)	0.13522 (6.9e-3)	0.13197 (9.9e-4)

Large Datasets. The analysis of the previous results might discourage the use of GPUs instead of using a pre-calculation of the distances, however, when the data set starts becoming a little bit bigger, this second approach is not possible any more. The reason is because the application runs out of memory so, to use a cluster of GPUs it is not a matter of performance in time or quality results, it is a matter of being able to provide a solution.

As an example, it will be used part of the data set provided by the Spanish Institute of Statistics (Instituto Nacional de Estadística, INE) that contains data about marital dissolutions in Spain. Several problems arise from this data, and one of them is predicting the dissolution process length which is translated into the regression problem.

The data to be used consists of 19967 input samples of 20 variables which was divided into training (of 15385) a test (4568) sets. The size of the training set is too big to pre-calculate the distance matrix used in previous approaches [11].

The proposed approach was executed during 600 seconds and was able to finish only one generation with 50 individuals providing a DT value of 0.001642 using 9 variables. Due to the high memory requirements, one of the computers with the

³ <http://research.ics.tkk.fi/eiml/datasets.shtml>

oldest GPU model was delaying the other two because of the synchronization step in the migration. Therefore, the experiments were repeated only with two nodes instead of three. The performance was increased significantly, allowing the algorithm to evolve a mean of 7.6 generations with a DT value of 0.001589 using only 4 variables.

Table 2. A) Delta test values for the large size data set ; B) Approximation errors (NRMSE) of the large data set with and without variable selection using an RBFNN with 15 neurons

Running Time		DT value (std)	# vars	# generations (std)
A)	600 secs (3 nodes, 4 GPUs)	0.001629 (2e-4)	9	1 (0)
	600 secs (2 nodes, 3 GPUs)	0.001592 (1e-4)	4	7.6 (0.5)
		Train Error	Test Error	
B)	with var. selec.	0.4846	0.5017	
	without var. selec.	1.3086	1.3197	

To test the validity of the selection provided, a model (Radial Basis Function Neural Network with 15 neurons) was designed using the methodology proposed in [7] without local search optimization. The experiments were done using all the variables and using the ones selected by the algorithm, obtaining the results shown in Table 2 B).

5 Conclusions

The problem of variable selection remains as an unsolved problem, being a crucial step before designing the models to classify or approximate. In this paper, a new algorithm that takes advantage of the new High Performance Computing technologies has been presented. Concretely, the main novelty is the use of a cluster where the nodes have graphical processing units to compute the k -Nearest Neighbours. The performance of the algorithm for small data sets is acceptable when compared with previous methods plus it is the only one of its kind that is able to provide good results in a reasonable time, when the size of the dataset becomes large.

Acknowledgments. This research has been supported by the projects by the Spanish CICYT Project TIN2007-60587 and TEC2008-04920 and Junta Andalucía Projects P08-TIC-03674 and P08-TIC03928 and PYR-2010-17 of CEI BioTIC GENIL (CEB09-0010) of the MICINN.

References

1. Alba, E., Nebro, A.J., Troya, J.M.: Heterogeneous computing and parallel genetic algorithms. *Journal of Parallel and Distributed Computing* 62, 1362–1385 (2002)
2. Alba, E., Tomassini, M.: Parallelism and evolutionary algorithms. *IEEE Trans. on Evolutionary Computation* 6(5), 443–462 (2002)

3. Bellman, R.E.: Adaptive control processes - A guided tour. Princeton University Press, Princeton (1961)
4. Cantú-Paz, E.: Efficient and Accurate Parallel Genetic Algorithms. Kluwer Academic Publishers, Massachusetts (2000)
5. Eirola, E., Liitiäinen, E., Lendasse, A., Corona, F., Verleysen, M.: Using the delta test for variable selection. In: ESANN 2008: European Symposium on Artificial Neural Networks, Bruges, Belgium, pp. 25–30 (April 2008)
6. Garcia, V., Debreuve, E., Barlaud, M.: Fast k nearest neighbor search using GPU. In: CVPR Workshop on Computer Vision on GPU (2008)
7. Guillén, A., González, J., Rojas, I., Pomares, H., Herrera, L.J., Valenzuela, O., Rojas, F.: Output Value-Based Initialization For Radial Basis Function Neural Networks. *Neural Processing Letters* (June 2007), doi:10.1007/s11063-007-9039-8
8. Guillén, A., Rojas, I., González, J., Pomares, H., Herrera, L.J., Paechter, B.: Improving the Performance of Multi-objective Genetic Algorithm for Function Approximation Through Parallel Islands Specialisation. In: Sattar, A., Kang, B.-h. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 1127–1132. Springer, Heidelberg (2006)
9. Guillén, A., Rojas, I., González, J., Pomares, H., Herrera, L.J., Paechter, B.: Boosting the performance of a multiobjective algorithm to design rBFNNs through parallelization. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 85–92. Springer, Heidelberg (2007)
10. Guillén, A., Pomares, H., González, J., Rojas, I., Valenzuela, O., Prieto, B.: Parallel multiobjective memetic rbfnn design and feature selection for function approximation problems. *Neurocomputing* 72(16-18), 3541–3555 (2009)
11. Guillen, A., Sovilj, D., Lendasse, A., Mateo, F., Rojas, I.: Minimising the delta test for variable selection in regression problems. *Int. J. High Perform. Syst. Archit.* 1, 269–281 (2008)
12. Herrera, F., Lozano, M.: Gradual distributed real-coded genetic algorithms. *IEEE Transactions on Evolutionary Computation* 4(1), 43 (2000)
13. Herrera, L.J., Pomares, H., Rojas, I., Verleysen, M., Guillén, A.: Effective input variable selection for function approximation. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4131, pp. 41–50. Springer, Heidelberg (2006)
14. Hiroyasu, T., Miki, M., Negami, M.: Distributed genetic algorithms with randomized migration rate. In: Proceedings of the IEEE Conf. Systems, Man and Cybernetics, pp. 689–694 (1999)
15. Pi, H., Peterson, C.: Finding the embedding dimension and variable dependencies in time series. *Neural Computation* 6(3), 509–520 (1994)

Towards ParadisEO-MO-GPU: A Framework for GPU-Based Local Search Metaheuristics

N. Melab, T.-V. Luong, K. Boufaras, and E.-G. Talbi

Dolphin Project

INRIA Lille Nord Europe - LIFL/CNRS UMR 8022 - Université de Lille1

40 avenue Halley, 59650 Villeneuve d'Ascq Cedex France

{Nouredine.Melab, The-Van.Luong, Karima.Boufaras, El-Ghazali.Talbi}@inria.fr

Abstract. This paper is a major step towards a pioneering software framework for the reusable design and implementation of parallel metaheuristics on Graphics Processing Units (GPU). The objective is to revisit the ParadisEO framework to allow its utilization on GPU accelerators. The focus is on local search metaheuristics and the parallel exploration of their neighborhood. The challenge is to make the GPU as transparent as possible for the user. The first release of the new GPU-based ParadisEO framework has been experimented on the Quadratic Assignment Problem (QAP). The preliminary results are convincing, both in terms of flexibility and easiness of reuse at implementation, and in terms of efficiency at execution on GPU.

Keywords: Software Framework, Local Search Meta-heuristics, Parallel Computing, GPU Computing, Neighborhood Exploration.

1 Introduction

Nowadays, parallel metaheuristics have grown to be a highly useful paradigm to solve large-scale CPU time-intensive and complex combinatorial problems. Metaheuristics are either single-solution namely S-Metaheuristics (local search metaheuristics) or population-based namely P-Metaheuristics (e.g. evolutionary algorithms). The focus in this paper is on S-Metaheuristics. Recently, GPU accelerators have emerged as a new powerful support for massively parallel computing. Last year, we came up with the pioneering work on GPU-based S-Metaheuristics [1]. Such experience has shown that parallel combinatorial optimization on GPU is not straightforward, and requires a huge effort at design as well as at implementation level.

Indeed, the design of GPU-aware S-Metaheuristics often involves the cost of a sometimes painful apprenticeship of parallelization techniques and GPU computing technologies. In order to free from such burden those who are unfamiliar with those advanced features, optimization frameworks must integrate the up-to-date parallelization techniques and allow their transparent exploitation and deployment on GPU accelerators. To the best of our knowledge, there does not exist any software framework for GPU-based metaheuristics. In [2], we have proposed

a framework called ParadisEO dedicated to the reusable design of parallel and distributed metaheuristics for only dedicated parallel hardware platforms. Later, we have extended the framework in [3] to dynamic and heterogeneous large-scale environments using Condor-MW middleware and in [4] to computational grids using Globus.

In this paper, we extend ParadisEO-MO (ParadisEO for S-Metaheuristics) to deal with GPU accelerators. The challenges and contributions consist in (1) rethinking the parallel models provided into the framework to manage efficiently the hierarchical organization of the memories (different latencies and sizes) of the GPU device as well as the interaction of this latter with the CPU ; (2) making the GPU as transparent as possible for the user minimizing his or her involvement in its management. In this paper, we propose solutions to this challenge as an extension of the ParadisEO framework. The focus is on the iteration-level parallel model of S-Metaheuristics which consists in exploring in parallel the neighborhood of a problem solution. The first release of the new GPU-based ParadisEO framework has been implemented using C++ and CUDA [5] and then experimented on the QAP.

The remainder of the paper is organized as follows. Section 2 highlights the principles of parallel iteration-level S-Metaheuristics and their challenges when using GPU computing. In Section 3, we describe the major design features and architecture of ParadisEO. Section 4 presents the design and implementation of ParadisEO-MO on top of GPU called ParadisEO-MO-GPU. Section 5 shows and comments some experimental results obtained with ParadisEO-MO-GPU on the QAP. In Section 6, we conclude the paper and draw some perspectives of the presented work.

2 Parallel GPU-Based S-Metaheuristics

An S-Metaheuristic is an iterative procedure which explores the neighborhood of a solution in order to improve its quality. The associated algorithm generates an initial (current) solution to the problem to be solved. This current solution is evaluated and its neighborhood is generated and evaluated. Based on the evaluation of the neighborhood, the best solution is selected to become the current solution. The process is repeated until a stopping criterion is found.

For large-scale combinatorial optimization problems, the neighborhood of a solution is often extremely large. Therefore, massively parallel computing is required to generate and evaluate it. The parallel generation and evaluation of the neighborhood is a master-worker and problem independent regular data-parallel application. GPU computing is very well-suited for this kind of parallel application. In the GPU (CUDA-based) model, the master is the CPU and the workers are threads executed by the processing cores of the GPU. Using GPU computing is not straightforward especially for non-experts in parallel computing. Indeed, a GPU accelerator provides a hierarchy of memories with different sizes and access latencies.

The challenge is to re-think the design of the parallel exploration and evaluation of the neighborhood taking into account the characteristics of GPU. Different issues have to be dealt with: (1) defining an efficient cooperation between CPU and GPU, which requires to share the work and to optimize the data transfer between the two components; (2) GPU computing is based on hyper-threading (massively parallel multi-threading) and the order in which the threads are executed is not known. Therefore, an efficient mapping has to be defined between each neighboring candidate solution and a thread designated by a unique identifier assigned by the GPU runtime; (3) the neighborhood has to be placed efficiently on the different memories taking into account their sizes and access latencies. From an implementation point of view, the challenge is to provide solutions to these issues in ParadisEO as transparent as possible way for the user.

3 ParadisEO-MO-Based Parallel S-Metaheuristics

3.1 Parallel Iteration-Level Model on GPU

The parallel iteration-level model is designed according to the data-parallel SPMD model of CUDA. In this model, a function code called the kernel is sent to the GPU to be executed by a large number of threads grouped into blocks. The task partitioning is such that the CPU hosts executes the whole serial part of the local search method. The GPU is in charge of the evaluation of the neighborhood of the current solution at each iteration. In order to minimize the cost of the data transfer from the CPU to GPU, the neighboring solutions are generated on GPU rather than on CPU. Indeed, only the current solution is sent to the GPU and each thread executes the same kernel. This is highly efficient for large neighborhoods. The kernel consists in generating and evaluating a neighbor. A defined mapping function allows each thread to find its corresponding neighboring solution. Once all the neighboring solutions are generated and evaluated they are sent back to the CPU where the best solution is selected. The process is iterated until a stopping criterion is satisfied.

3.2 The ParadisEO-MO Framework

ParadisEO-MO is part of ParadisEO dedicated to S-Metaheuristics such as Hill Climbing, Simulated Annealing, Tabu Search, ILS, etc. ParadisEO [2] is a framework dedicated to the reusable design of parallel hybrid metaheuristics by providing a broad range of features including evolutionary algorithms (ParadisEO-EO), local search methods (ParadisEO-MO), parallel and distributed models (ParadisEO-PEO), different hybridization mechanisms, etc. ParadisEO is a C++ LGPL extensible open source framework based on a clear conceptual separation of the metaheuristics. ParadisEO is one of the rare frameworks that provide the most common parallel and distributed models. These models are portable on distributed-memory machines and shared-memory multi-processors as they are implemented using standard libraries such as MPI, PVM and Pthreads.

4 GPU-Enabled ParadisEO

4.1 Architecture of ParadisEO-MO-GPU

ParadisEO-MO-GPU is a new framework which is a coupling between ParadisEO-MO and CUDA. It aims at deploying the S-Metaheuristics on GPU in a generic way. It is composed by a set of new C++ abstract and predefined classes that allows an easy and transparent development of S-metaheuristics on GPU accelerators. The architecture of ParadisEO-MO-GPU is layered as it is illustrated in Fig. 1.

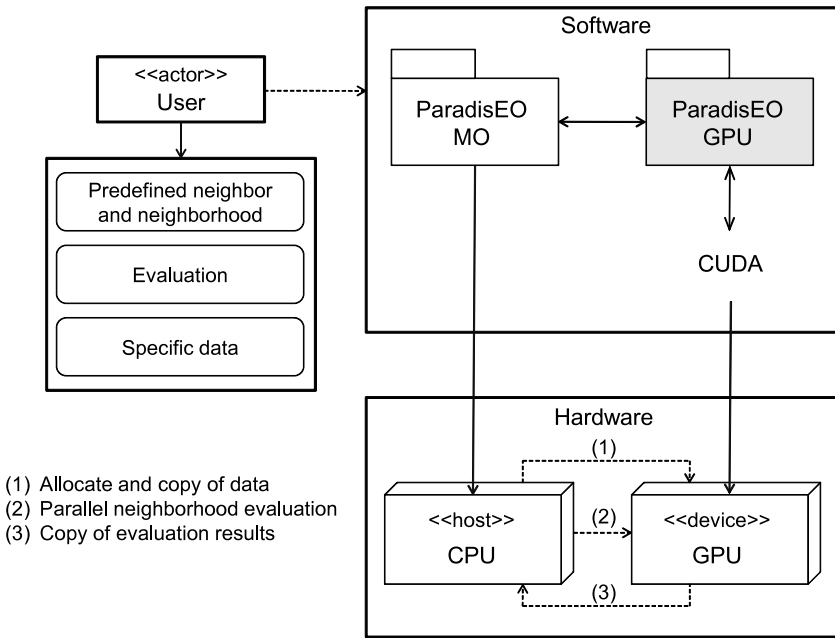


Fig. 1. A layered architecture of ParadisEO-MO-GPU

The user level indicates the different problem-dependent components which must be defined: input data, the evaluation problem function, neighbor and neighborhood representations. The second level presents the ParadisEO-MO framework including optimization solvers embedding S-metaheuristics. The interaction is done with the ParadisEO-GPU module which automatically pilots the CUDA programming interface. The hardware level supplies the different transparent tools provided by ParadisEO-GPU such as the allocation and copy of data or the parallel generation and evaluation of the considered neighborhood. In addition to this, the platform proposes predefined neighborhood and mapping wrappers adapted to hardware constraints to deal with binary and permutation problems.

4.2 A Case Study: Parallel Evaluation of a Neighborhood

ParadisEO-MO-GPU is illustrated in Fig. 2 through an UML sequence diagram. The scenario shows the design and implementation of parallel neighborhood evaluation of a local search on GPU. The different steps of the parallel evaluation process on GPU for each iteration are the following:

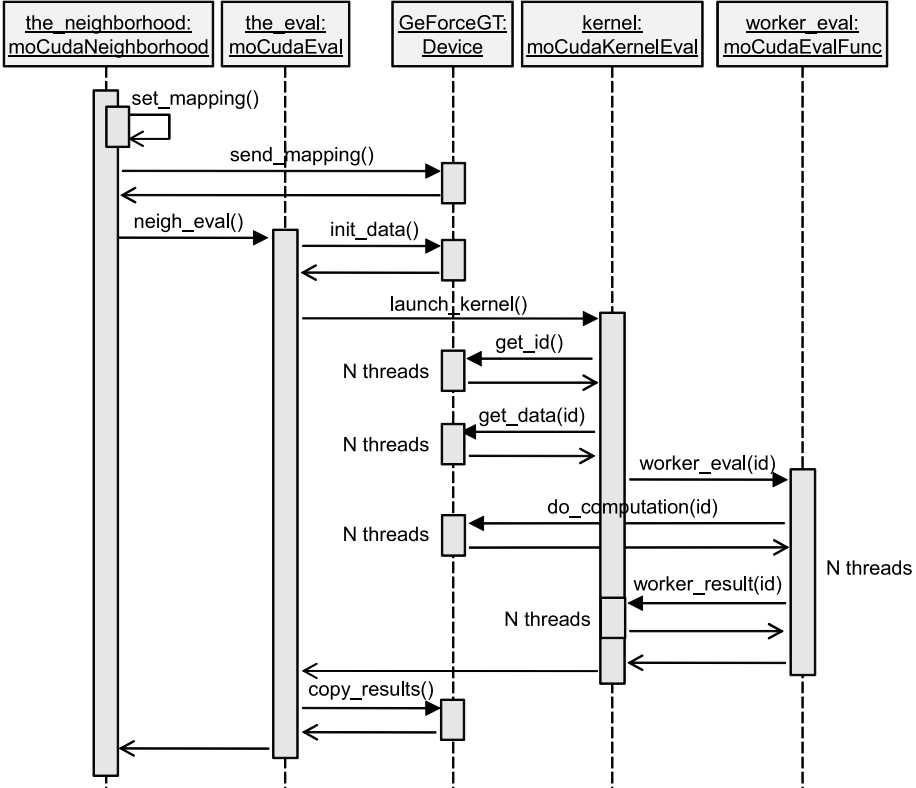


Fig. 2. The parallel generation and evaluation of a neighborhood provided in ParadisEO-MO-GPU

1. The neighborhood *moCudaNeighborhood* prepares all the steps for the parallel generation of the neighborhood on GPU. The initialization consists in setting a mapping table between GPU threads and neighbors. Then, the associated data are sent only once to the GPU since the mapping structure does not change during the execution process of local searches. The last step invokes the parallel evaluation and will be informed on its completion to retrieve the precomputed fitnesses structure.
2. Before proceeding to the parallel evaluation, the object *moCudaEval* configures a kernel with N threads such that each thread is associated exactly with one neighbor evaluation (N designates the neighborhood size). During

the first iteration, the object allocates the neighborhood fitnesses structure where the result of the evaluated neighbors will be stored. Otherwise, in any case, it only sends to the GPU device the candidate solution which generates the neighborhood.

3. The object *moCudaKernelEval* modelizes the main body which will be executed by N concurrent threads on different input data. A first step consists in getting the thread identifier then the set of its associated data. This mechanism is done thanks to the mapping table previously mentioned. The second step performs the evaluation computation of the corresponding neighbor. Finally, the resulting fitness is stored in the corresponding index of the fitnesses structure.
4. The worker *moCudaEvalFunc* is the specific object with computes on the GPU device the corresponding evaluation neighbor and returns the provided result.

Once the entire neighborhood has been performed in parallel on GPU, the precalculated fitness structure is copied back to the CPU and given as input to the ParadisEO-MO module. This way, the local search continues the neighborhood exploration (iteration) on the CPU side. Instead of evaluating again each neighbor in a sequential manner, the corresponding fitness value will only need to be retrieved from the precomputed fitnesses structure. Hence, this mechanism has the advantage of allowing both the deployment of any S-metaheuristics and the use of toolboxes provided in ParadisEO-MO (e.g. statistical or fitnesses landscape analysis, checkpoint monitors, etc.)

5 Experimentation

The QAP arises in many applications such as facility location or data analysis. Let $A = (a_{ij})$ and $B = (b_{ij})$ be $n \times n$ matrices of positive integers. Finding a solution of the QAP is equivalent to finding a permutation $\pi = (1, 2, \dots, n)$ that minimizes the objective function:

$$z(\pi) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{\pi(i)\pi(j)}$$

As the iteration-level parallel model does not change the semantics of the sequential algorithm, the effectiveness in terms of quality of solutions is not addressed here. The objective is to assess the impact in terms of efficiency of an implementation done with ParadisEO-MO-GPU compared with an optimized version made outside the platform. To achieve this, a tabu search has been implemented in 4 different versions: 1) a ParadisEO-MO implementation on CPU and its counterpart on GPU; 2) an optimized CPU implementation and its associated GPU version. ParadisEO versions are pure object-based implementations whereas the optimized ones are pointer-based.

The neighborhood is built by pair-wise exchange operations (known as the 2-exchange neighborhood) which is a standard way for permutation problems. The number of global iterations of the local search is set to 10000. The considered instances are the Taillard instances proposed in [6].

Experiments have been carried on top of an Intel Core i7 970 3.2Ghz with a GTX 480 card (15 multiprocessors with 32 cores). For measuring the acceleration factors, only a single core has been considered using the Intel i7 turbo mode (3.46Ghz). The average time has been measured in seconds for 30 runs. Results are reported in Table 1 for the different versions.

Table 1. Measures in terms of efficiency of a ParadisEO-MO-GPU implementation with an optimized version made outside the platform

Instance	ParadisEO-MO			Optimized version			Perf. gap	
	CPU	GPU	Acc.	CPU	GPU	Acc.	CPU	GPU
tai30a	0.8	0.9	×0.9	0.7	0.7	×1.0	13%	23%
tai40a	1.9	1.2	×1.6	1.7	1.0	×1.7	11%	17%
tai50a	3.6	1.6	×2.2	3.0	1.3	×2.3	17%	19%
tai60a	6.0	2.0	×3.0	5.0	1.6	×3.1	17%	20%
tai80a	15.1	2.8	×5.4	12.3	2.1	×5.8	19%	25%
tai100a	31.7	3.8	×8.3	26.0	2.8	×9.2	18%	27%
tai150b	119.7	8.9	×13.4	98.1	6.5	×15.1	18%	27%

From the instance tai40a, both GPU versions start to give positive accelerations (from ×1.6 to ×1.7). The poor performance for small instances can be explained by the fact that since the neighborhood is relatively small, the number of threads per block is not enough to fully cover the memory access latency. However, as long as the instance size increases, the acceleration factor grows accordingly (e.g. from ×5.4 to ×5.8 for tai80a). Finally, efficient speed-ups are obtained for the instance tai150b. They vary between ×13.4 and ×15.1.

A thorough examination of the acceleration factors highlights that they are quite similar for the two GPU versions. The performance difference which exists is certainly due to the CPU version provided by ParadisEO-MO. Indeed, regarding the two CPU versions, initially, there is already a slight performance gap regarding the execution time. It varies between 11% and 19% according to the instance. This gap can be explained by the overhead caused by the creation of generic objects in ParadisEO whereas the optimized version on CPU is a pure pointer-based implementation. This can also explain the performance difference between the two different GPU counterparts in which the same phenomenon occurs. However, for such transparent exploitation, the obtained results are really convincing. A conclusion of the experiments indicates that the performance results of the GPU version provided by ParadisEO are not so far from the GPU pointer-based one.

6 Conclusions and Future Work

In this paper, we have presented a step towards a ParadisEO framework for the reusable design and implementation of the GPU-based parallel metaheuristics. The focus is set on S-metaheuristics and the iteration-level parallel exploration of the neighborhood of a solution. We have revisited the design and implementation of this last model in ParadisEO-MO to allow its efficient execution and its transparent use on GPU.

The implementation in ParadisEO-MO using CUDA has been experimented on the QAP and compared to the same implementation realized outside ParadisEO. The preliminary results show that the performance gap which occurs between the two implementations is not really important. Indeed, for such flexibility and easiness of reuse at implementation, the obtained results with ParadisEO-MO-GPU are really promising (up to $\times 13$). We are strongly convinced that the overall performance provided by ParadisEO-MO-GPU will be much better for larger neighborhoods or other problems requiring more computational calculations (see [7] and [1] for previous test cases).

The first release of ParadisEO-MO on GPU is available on the ParadisEO website [1]. Tutorials and documentation are provided to facilitate its reuse. This release is dedicated to parallel S-metaheuristics based on the iteration-level parallel model. In the short run, this release will be first extended to the algorithmic (multi-start) and solution-level parallel models. Second, it will be extended to other problem representations such as discrete representation and other solution methods. Third, it will be validated on a wider range of problems. In the long run, ParadisEO will be revisited following the same roadmap for evolutionary algorithms on GPU.

References

1. Van Luong, T., Melab, N., Talbi, E.-G.: Local search algorithms on graphics processing units. A case study: The permutation perceptron problem. In: Cowling, P., Merz, P. (eds.) *EvoCOP 2010*. LNCS, vol. 6022, pp. 264–275. Springer, Heidelberg (2010)
2. Cahon, S., Melab, N., Talbi, E.-G.: ParadisEO: A Framework for the Reusable Design of Parallel and Distributed Metaheuristics. *J. of Heuristics* 10(3), 357–380 (2004)
3. Melab, N., Cahon, S., Talbi, E.-G.: Grid computing for parallel bioinspired algorithms. *J. Parallel Distributed Computing* 66(8), 1052–1061 (2006)
4. Tantar, A.A., Melab, N., Demarey, C., Talbi, E.G.: Building a Virtual Globus Grid in a Reconfigurable Environment - A case study: Grid5000 (2007)
5. Nickolls, J., Buck, I., Garland, M., Skadron, K.: Scalable Parallel Programming with CUDA. *ACM Queue* 6(2), 40–53 (2008)
6. Taillard, É.D.: Robust tabu search for the quadratic assignment problem. *Parallel Computing* 17(4-5), 443–455 (1991)
7. Luong, T.V., Melab, N., Talbi, E.G.: Parallel hybrid evolutionary algorithms on gpu. In: *IEEE Congress on Evolutionary Computation*, pp. 1–8 (2010)

¹ <http://paradiseo.gforge.inria.fr>

Efficient Simulation of Spatio-temporal Dynamics in Ultrasonic Resonators^{*}

Pedro Alonso-Jordá¹, Jesús Peinado-Pinilla¹,
Isabel Pérez-Arjona², and Victor J. Sánchez-Morcillo²

¹ Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Cno. Vera s/n, 46022 Valencia, Spain

{palonso,jpeinado}@dsic.upv.es

² Instituto de Investigación para la Gestión Integrada de Zonas Costeras, Universitat Politècnica de València, Crta. Nazaret-Oliva s/n, 46730 Grau de Gandia, Spain

{victorsm,iparjona}@fis.upv.es

Abstract. The simulation in computers of the evolution of pressure and temperature inside a cavity when acoustic energy is injected results in a very stiff and high time consuming application. The split-step method used for the integration of the spatial part of the physical model is based on intensive computation of two-dimensional FFTs. This operation is currently very optimized but its use through a Matlab implementation can not completely exploit this optimization. In addition, there exist other operations involved in the integration of the model well suited to be run efficiently on hardware accelerators like GPUs. Thus, we implemented CPU and GPU versions of the application which are much faster than the Matlab version. Moreover, in order to preserve the ease of use we kept the Matlab interface by using mex files.

1 Introduction

In this paper we present an efficient implementation of an application for the simulation of spatio-temporal dynamics of an acoustic field. A set of partial differential equations that describe the physics were proposed in [1]. Intensive study of these equations should be done to validate and tune the mathematical proposition by contrasting it with physical experiments. Due to the complexity of the model, the only way to approach the solution of the evolution equations is by simulation. Very different scales in time and space force to take an extremely small time step and, consequently, to perform a large number of iterations to simulate the experiment resulting in a high consumption of computation time.

We use in the simulation the split-step and Euler methods for integration. The computational cost of the application based on the chosen methods relies on two-dimensional FFTs (Fast Fourier Transforms) and matrix point-wise operations. The application is characterized by many, dependent and fast iterations

^{*} This work was financially supported by Spanish Ministerio de Ciencia e Innovación, European Union FEDER (FIS2008-06024-C03-02).

(since matrices are small) so there exists little opportunity to make the computation of each iteration in parallel using, i.e., multicores.

We solved the problem by implementing the application both in CPU and Graphics Processor Units (GPUs). The good cost–performance ratio and the availability of a high level language like CUDA [2] have led GPUs out beyond the graphics field. Thus, GPUs have strongly consolidated as a serious tool to accelerate high performance scientific computations (“General–Purpose Computation on Graphics Hardware” [3]). Actually, great efforts are being made to move scientific applications to GPUs to accelerate computations. For example, a survey of discretization of ODEs and PDEs on GPUs can be found in [4]. A similar work can be found in [5] but discretizing PDEs to integrate the equations.

However, not all the applications well fit the requirements to exploit this hardware. The FFT is a very optimized operation which runs very fast on CPUs. On the contrary, other application operations run very fast on GPUs due to its high degree of intrinsic parallelism. The implementations on both CPU and GPU we present far outweigh Matlab, being the GPU version the most competitive. Furthermore, we modified the GPU version by incorporating a new computation of the two–dimensional FFT based on V. Volkov work [6].

We also provide Matlab interfaces based on MEX files [7,8,9]. MEX files are external interface functions which allow to call custom routines directly from Matlab as it is done with Matlab built-in functions. This tool is very useful for researches not skillful on programming issues. Experiments carried out with the implementations presented in this paper led to important contributions to understand the self–organization of ultrasounds in viscous fluids [10].

The paper is organized as follows. Section 2 summarizes the physical problem. Next we describe the simulation algorithm. Section 4 and Section 5 describe, respectively, both CPU and GPU application versions. The experimental results are analyzed in Section 6. The paper ends with some conclusions and remarks.

2 The Physical Model

Many systems in nature, when driven far from equilibrium, can self–organize, giving rise to a large variety of patterns or structures. Although studied intensively for most of the last century, it has only been during the past thirty years that pattern formation has emerged as branch of science in its own right [11]. A model to study the spatio–temporal dynamics of an acoustic field has been proposed in a recent work [1]. The physical system consists of two solid walls, containing a viscous fluid, where acoustic energy is injected by vibrating one of the walls. In a viscous medium sound propagates with a speed c that depends significantly on temperature. The propagation of sound in such a medium can be described by two coupled equations for pressure, $p(r, t) = P(x, y, t) \cos(kz)$, and temperature, $T(r, t) = T_0(x, y, t) + T_1(x, y, t) \cos(2kz)$. In the case of a fluid confined in a driven resonator, it was shown in [1] that the pressure field P and temperature fields T_0 and T_1 inside the cavity, which are proportional to the amplitudes of pressure field and the homogeneous and grating (modulated)

temperature components, obey the evolution equations,

$$\begin{aligned}
 \tau_P \frac{\partial P}{\partial \tau} &= -(1 + i\Delta)P + P_{in}I + i\nabla^2 P + i(T_0 + T_1)P, \\
 \frac{\partial T_0}{\partial \tau} &= -T_0 + D\nabla^2 T_0 + 2|P|^2, \\
 \frac{\partial T_1}{\partial \tau} &= -\tau_g^{-1}T_1 + D\nabla^2 T_1 + |P|^2.
 \end{aligned}
 \tag{1}$$

The constants τ_P and τ_g are the normalized relaxation times of the pressure field and the temperature grating component, respectively. Other parameters are the detuning Δ between the driving frequency ω and the closest cavity mode, P_{in} the injected pressure and D the diffusion coefficient. Finally $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$ is the transverse Laplacian operator, necessary for pattern formation.

The model parameters can be estimated for a typical experimental situation [1]. It follows that the normalized decay times are $\tau_P \sim 10^{-6}$, and $\tau_g \sim 10^{-2}$ under usual conditions. So the problem is typically very stiff, since $0 < \tau_P \ll \tau_g \ll 1$. Stiff systems of differential equations, where the different variables evolve along very different time scales, are usually problematic for numerical solving, and certain numerical methods are unstable, unless the step size is taken to be extremely small yielding thus very high time consumption.

3 The Simulation Algorithm

The simulation performs some previous initializations to call latter the main process of the algorithm which is summarized in the following:

Algorithm termo2D:

```

for iteration = 1:2000000
    (P, T0, T1) ← terms(P, T0, T1, n, τP, τg, Δ, Pin, δ)
    P          ← M · (CP * P)
    T0        ← M · (CT0 * T0)
    T1        ← M · (CT1 * T1)
end for
    
```

where symbol $*$ denotes the convolution operation and symbol \cdot the matrix-wise product of matrices. Matrices P , T_0 and T_1 are input/output matrices; CP , CT_0 and CT_1 are input matrices already transformed into the spacial Fourier space and M is also an input matrix that contains the acoustic energy injected by one of the walls (resonator). All of matrices are of size $n \times n$, where $n = 64$. Value $\delta = 10^{-6}$ is the time step size. The number of iterations might vary but the given value of 2 million is representative of the amount of iterations required for the particular physical system in the experiments to be stable.

The integration method is divided into two steps: 1) the linear and non-linear blocks; and 2) the spatial part, which corresponds to the Laplacian operators in equations (1). The linear block is exactly integrated from the linear matrix system in equations (1) when spatial and nonlinearities are not considered. The

function $(P, T_0, T_1) = \mathbf{terms}(P, T_0, T_1, n, \tau_p, \tau_g, \Delta, P_{in}, \delta)$ is

$$\begin{aligned}
 P' &= P + (0.5\delta/\tau_p) (-1 + i\Delta)P + P_{in}I + i(T_0 + T_1) \cdot P \\
 T'_0 &= T_0 + 0.5\delta (-T_0 + 2P \cdot P^*) \\
 T'_1 &= T_1 + 0.5\delta (-1/\tau_g)T_1 + P \cdot P^* \\
 P &\leftarrow P + (\delta/\tau_p) (-1 + i\Delta)P' + P_{in}I + i(T'_0 + T'_1) \cdot P' \\
 T_0 &\leftarrow T_0 + \delta (-T'_0 + 2P' \cdot P'^*) \\
 T_1 &\leftarrow T_1 + \delta (-1/\tau_g)T'_1 + P' \cdot P'^*
 \end{aligned}$$

Fig. 1. Function **terms** (superscript * denotes complex conjugated, I is the identity matrix and i is the complex variable ($i = \sqrt{-1}$))

nonlinear part is integrated via a second order Runge–Kutta scheme. The computation of these blocks is carried out in function **terms** (Fig. 1).

The spatial part is integrated using an exact split–step algorithm, which consists of transforming the spatial part of the equation into the spatial Fourier space and obtaining a differential equation with exact solution. The equation is transformed into the Fourier space, propagated multiplying by the temporal step and re–transformed into the real space (convolution). The advantage of the split–step method (and of spectral and pseudospectral methods, in general) is that it provides high accuracy in comparison with the finite elements technique for a similar number of discretization points [12].

4 The CPU Algorithm

Algorithm **termo2D** implemented in C is shown in Fig. 2. A **convolution** routine involves one forward and one backward two–dimensional FFT plus two point–wise matrix–matrix multiplications, one for the multiplication of matrices in the spatial Fourier space and one more for the multiplication with the acoustic energy matrix M . The corresponding Matlab code for the first convolution is

```
P = M .* ifft2( CP .* fft2(P) );
```

These two–dimensional FFT (FFT_2) are carried out in the C code by using the Intel MKL [13] FFT functions called as **DftiX**. The two point–wise matrix–matrix multiplications are performed straightforwardly by using standard C loops ranging matrices and taking into account that they are complex.

For the CPU code it was chosen to implement function **terms** (Fig. 1) in Fortran 90 and to use the ISO C binding package of Fortran 2003 to mix both Fortran and C since operators $+$ and $*$ are overloaded for arrays and scalar–array

```
for (iteration = 1; iteration<=2000000; iteration++) {
    terms( dP, dT0, dT1, p1, p2, p3, p4, p5, p6, p7, N );
    convolution( M, dP, dCP );
    convolution( M, dT0, dCT0 );
    convolution( M, dT1, dCT1 );
}
```

Fig. 2. Implementation in C of Algorithm **termo2D**

```

void convolution( int n, float2 *A, float2 *B, float2 *C ) {
    cufftExecC2C(fftwPlan, (cufftComplex*)A, (cufftComplex*)A, CUFFT_FORWARD);
    cudaMatrixMatrixWiseProduct( n, A, B, 1.0/(float) (n*n) );
    cufftExecC2C(fftwPlan, (cufftComplex*)A, (cufftComplex*)A, CUFFT_INVERSE);
    cudaMatrixMatrixWiseProduct( n, A, C, 1.0 );
}

```

Fig. 3. Convolution operation in GPU with CUDA

operations in Fortran 90. Furthermore, Intel Fortran Compiler (`ifort`) vectorizes these operations. The routine receives arguments `p1` to `p7` which are pre-computations involving values τ_p , τ_g , Δ and δ carried out before calling `terms`.

5 The GPU Algorithm

Algorithm **termo2D** is also implemented in CUDA as shown in Fig. 2. Function `convolution` in CUDA implementation performs the same four operations as in C implementation (Fig. 3). The first and third steps are calls to cuFFT routine `cufftExecC2C` to perform forward and backward FFT₂, respectively, on the first matrix argument `A` in-place. The second and fourth steps are matrix-matrix point-wise products of matrices.

Function `terms` in Fig. 2 is a C function which sets up the threads block and grid size to call afterwards kernel `kterms` (Fig. 4). The kernel receives arguments `p1` to `p7` from routine `terms`. We used auxiliary device functions (`complex_mult`, `conjugate`) to perform the denoted operations. We exploited the fact that all the operations can be carried out independently on each matrix element yielding a simple kernel implementation with a very high degree of low grain concurrency. Matrices P' , T'_0 and T'_1 in function `terms` (Fig. 1) are represented in the kernel through local complex (`float2`) variables `Paux`, `T0aux` and `T1aux`, respectively.

We performed an improvement in the computation of the two-dimensional FFT (FFT₂) using Volkov routines for the computation of one-dimensional FFT [6]. Volkov solution is based on a special storage of matrices in the device memory. Before computing the FFT a data movement must be done in CPU to change the standard two-dimensional matrix storage by columns in a *block* storage. In the *block* storage we used, each 64 element column of the given matrix where the FFT will be applied is saved in a 8×8 block of a matrix of 8×8 blocks. Let M be a 64×64 elements matrix and A a 8×8 blocks matrix where each block is of size 8×8 . Element (i, j) , $i, j = 0, \dots, 63$, of M will be stored in block $(j\%8, j/8)$ of matrix A . The same element (i, j) will be stored in position $(i\%8, i/8)$ inside the given block. Thus, element (i, j) of matrix M will be found in position $(j\%8 + i\%8, j/8 + i/8)$ of matrix A . In addition, we used the fact that the transformation from a standard transposed matrix to a block matrix is equivalent to the transposition of the corresponding *block* matrix.

To compute the FFT₂ of a matrix, a one-dimensional FFT must be computed of the columns followed by a one-dimensional FFT computation on the rows, so $\text{FFT}_2(M) = (\text{FFT}(\text{FFT}(M)^T))^T$, where $\text{FFT}(M)$ is the FFT of all the columns of M (note that T means “transpose” but not conjugation).

```

__global__ void kterms( float2 *P, float2 *T0, float2 *T1,
                      float2 p1, float p2, float p3, float p4, float p5,
                      float p6, float P7, unsigned int N ) {
int i = blockIdx.x * blockDim.x + threadIdx.x;
int j = blockIdx.y * blockDim.y + threadIdx.y;
int k = i + j*N;
float2 Paux, T0aux, T1aux, a, b;
if ( i < N && j < N ) {
    a      = complex_mult( p1, P[k] );
    a.x    += P7;
    b      = complex_mult( make_float2( -(T0[k].y + T1[k].y),
                                         T0[k].x + T1[k].x ), P[k] );
    Paux.x = P[k].x + ( a.x + b.x ) * p5;
    Paux.y = P[k].y + ( a.y + b.y ) * p5;
    a      = complex_mult( P[k], conjugate( P[k] ) );
    T0aux.x = T0[k].x + ( -T0[k].x + 2.0*a.x ) * p4;
    T0aux.y = T0[k].y + ( -T0[k].y + 2.0*a.y ) * p4;
    T1aux.x = T1[k].x + ( p6 * T1[k].x + a.x ) * p4;
    T1aux.y = T1[k].y + ( p6 * T1[k].y + a.y ) * p4;
    a      = complex_mult( p1, Paux );
    a.x    += P7;
    b      = complex_mult( make_float2( -(T0aux.y + T1aux.y),
                                         T0aux.x + T1aux.x ), Paux );
    P[k].x = P[k].x + ( a.x + b.x ) * p3;
    P[k].y = P[k].y + ( a.y + b.y ) * p3;
    a      = complex_mult( Paux, conjugate( Paux ) );
    T0[k].x = T0[k].x + ( -T0aux.x + 2.0*a.x ) * p2;
    T0[k].y = T0[k].y + ( -T0aux.y + 2.0*a.y ) * p2;
    T1[k].x = T1[k].x + ( p6 * T1aux.x + a.x ) * p2;
    T1[k].y = T1[k].y + ( p6 * T1aux.y + a.y ) * p2;
}
}

```

Fig. 4. Kernel for the linear and non-linear terms calculation

Volkov routines compute a batch of FFT's on the columns of a given matrix. In our case, we compute the FFT_2 of a 64 order matrix by using Volkov routine $\text{FFT64}(A, 64)$ (and the corresponding inverse $\text{IFFT64}(A, 64)$) which computes the one-dimensional 64-FFT on each of the 64 columns of matrix A to perform a convolution operation $A \leftarrow \text{IFFT}_2(B * \text{FFT}_2(A))$ as shown in Fig. 5 a), which corresponds to the first three operations shown in the code of Fig. 3.

In the computation of the FFT_2 , two transpositions of code in Fig. 5 a) can be avoided since the FFT can be performed on the columns firstly and on the rows secondly. This property leads to the code in Fig. 5 b). Note that now input matrix B must be a *block* matrix and transposed.

We used a kernel contained in CUDA SDK [2] for the transpositions. Sums and products-wise of matrices as scalar-matrix products can be applied on *block* matrices, making it possible to use directly *block* matrices in the `kterms` kernel.

$A_1 \leftarrow \text{FFT64}(A)$	$A_1 \leftarrow \text{FFT64}(A)$
$A_2 \leftarrow \text{transpose}(A_1)$	$A_2 \leftarrow \text{transpose}(A_1)$
$A_2 \leftarrow \text{FFT64}(A_2)$	$A_2 \leftarrow \text{FFT64}(A_2)$
$A \leftarrow \text{transpose}(A_2)$	$A_1 \leftarrow B * A_2$
$A \leftarrow B * A$	$A_2 \leftarrow \text{IFFT64}(A_1)$
$A_1 \leftarrow \text{IFFT64}(A)$	$A_1 \leftarrow \text{transpose}(A_2)$
$A_2 \leftarrow \text{transpose}(A_1)$	$A \leftarrow \text{IFFT64}(A_1)$
$A_2 \leftarrow \text{IFFT64}(A_2)$	
$A \leftarrow \text{transpose}(A_2)$	
a)	b)

Fig. 5. Convolutions using Volkov FFT64 routine

6 Experimental Results

We used in our experiments an Intel Quad Core Xeon E5430 2.66 GHz. running Linux and a NVidia Quadro FX 5800 with 240 parallel processor cores and 4GB global memory. We used Intel developing tools (Compiler Version 11.1 including MKL), CUDA distributions 2.2 and 3.1 and Matlab version R2009b.

Results in Table 1 were obtained using “Matlab” application, “CPU” implementation, “GPU” implementation using cuFFT of the two different CUDA distributions, and “GPU Improved” version using our FFT₂ implementation. The table shows execution time and percentage of time used for the two main steps: function `terms` and the three `convolution` functions. All tests have been launched from Matlab command interface.

Total time results show that CPU version is more than two times faster than Matlab one. However, we remark on that the weight (percentage) of the two steps are similar in both versions since the underlying hardware is the same.

The most different behaviour occurs between CPU and GPU, being GPU time less than 20% of CPU time (cuda 3.1). As it can be seen in the first data column, the reduction in time is mainly due to the computation of `terms`. This is a $O(n^2)$ flops function but it becomes like if it were a $O(1)$ flops in GPU due to the high degree of intrinsic parallelism, that is, all the n^2 elements of the involved matrices can be computed concurrently. This fact is also reflected in the relative weight of the algorithm parts. The most weighty part is completely different in both type of platforms meaning that computation of `terms` is far suitable to be carried out on GPU.

Table 1. Results of Algorithm `termo2D` in different platforms

	terms function			Convolutions			Total time sec.	Gflops
	sec.	%	Gflops	sec.	%	Gflops		
Matlab	3118	64.6	0.53	1701	35.3	1.5	4820	0.87
CPU	1461	67.9	1.12	687	31.9	3.7	2150	1.95
GPU (cuda 2.2)	27	4.8	60.7	541	95.2	4.7	568	7.38
GPU (cuda 3.1)	27	6.4	60.7	397	93.6	6.4	424	9.89
GPU improved	27	6.7	60.7	379	93.3	6.7	406	10.3

Otherwise, FFTs are more suitable for CPU. Convolutions part spends close to 95% of total execution time in GPU and a third of the execution time in the CPU. Two factors are involved, the hardware is more suitable to perform this operation and Intel MKL routines for the computation of the FFT are very optimized. On the contrary, GPU hardware is more difficult to be exploited with this operation type due to data dependences. Actually, noticeable effort has been done between cuFFT of CUDA distributions 2.2 and 3.1. Nevertheless, the convolutions time is lower in GPU than in CPU. The computation of convolutions part is 27% (cuda 2.2) and 73% (cuda 3.1), respectively, faster in GPU using cuFFT than in CPU. In the case of our “Improved GPU” version, this computation is 81.3% faster in GPU than in CPU.

7 Conclusions

In this work we have shown an application consisting in the simulation of a physical system with considerable research interest. The application is very suitable to be run in GPUs where we obtained a significant reduction in time. Hence, we can conclude that it is worth to move this type of applications containing both operation types from CPU to GPU. In addition, we offered researchers in this field an easy of use tool since it can be managed in a Matlab environment [14].

Although the speed up obtained with our improved computation of two-dimensional FFT is not very large we foresee that, as it is shown in [6], other applications based on the split–step integration method working on matrices of larger orders than 64 will get a larger improvements too.

References

1. Perez–Arjona, I., Sánchez–Morcillo, V.J., de Valcárcel, G.J.: Ultrasonic Cavity Solitons. *Europhys. Lett.* 82, 10002 (2008)
2. NVidia Corp. CUDA Version 3.1 (2010)
3. General–Purpose Computation on Graphics Hardware., <http://gggpu.org>
4. Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.E., Purcell, T.J.: A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum* 26(1), 80–113 (2007)
5. Januszewski, M., Kostur, M.: Accelerating numerical solution of Stochastic Differential Equations with CUDA (2009), <http://arxiv.org/abs/0903.3852>
6. Volkov, V., Kazian, B.: Fitting the FFT onto the G80 architecture, UC Berkeley CS258 Project Report (May 2008)
7. MathWorks: MATLAB MEX Files., <http://www.mathworks.com>
8. NVIDIA. MATLAB plugin, http://developer.nvidia.com/object/matlab_cuda.html
9. NVIDIA. Accelerating MATLAB with CUDA Using MEX Files (2007)
10. Sánchez–Morcillo, V.J., et al.: Self–organization of ultrasound in viscous fluids. *Europhys. Lett.* 92, 10003 (2010)
11. Cross, M.C., Hohenberg, P.C.: Pattern formation out of equilibrium, *Rev. Mod. Phys.* 65, 851 (1993)
12. Trefethen, L.N.: *Spectral methods in MATLAB*. SIAM, Philadelphia (2000)
13. Intel®: Intel Math Kernel Library for the Linux* OS 10.1. User’s Guide (2008)
14. termo2D package: <http://www.inco2.upv.es/software.php>

GPU Implementation of a Bio-inspired Vision Model

Raquel Ureña, Christian Morillas, Samuel Romero, and Francisco J. Pelayo

Dept. of Computer Architecture and Technology, University of Granada, Spain
{ruperez, cmorillas, sromero}@atc.ugr.es, fpelayo@ugr.es

Abstract. This paper describes the implementation of a bio-inspired visual processing model which aims to remark the relevant information from a scene to generate different kinds of stimuli such as visual, haptic or audio stimuli, useful in aid systems for visually impaired persons. Desirable features for such system are real-time operation, minimum latency and portability. To meet all these requirements we have implemented the whole model in the GPU NVIDIA ION2 which is integrated in a light-weight netbook and performs at a maximum rate of 35 frames per second.

Keywords: Visually impaired, GPU, CUDA, GPGPU, Bio-inspired image Processing.

1 Introduction

The purpose of this paper is to present an implementation of a bio-inspired vision model able to enhance the main features of an image using GPUs. This has special interest in the development of aid systems for low vision impaired and also for the blind.

Vision is regarded as one of the main sensorial organs of the human being, given that around a 40% of the information arriving to the brain is related to the visual system. However, the World Health Organization estimates that the number of people with visual impairments (with visual remains) is 285 million (65% of whom are aged over 50 years). Of these, 246 million have low vision (63% over 50) and 39 million are estimated to be blind (82% over 50) [1].

In this sense there are several systems which try to improve the visual capabilities taking advantage of residual vision, aimed to low vision affected persons. Some of these devices employ an opaque and immersive Head Mounted Display (HMD) to project the enhanced images. For example the System LVES [2], the system JORDY by Enhanced Vision or an aid system which combines several image enhancements adapted to the visual field of each user, using a GPU to achieve real time performance [3]. Most of these aid systems perform transformations of the input image, amplifying it in size, intensity or contrast. These transformations are mainly based on digital zooming and edge overlaying and are applied to the whole image or, in the case of [3], to different regions of interest previously defined, but are unable to detect the main features of a scene and especially enhance them.

On the other hand, there is a wide range of devices specifically oriented to the Blind. Some of them are automatic sensorial transduction systems, so that visual

inputs that cannot be processed by the natural system due to pathology are forwarded to a different sense which fully or partially preserves its function, as, for instance, the hearing.

Some examples of these systems are VIS2SOUND [4], The vOICe [5] an EAV [6]. The system VIS2SOUND uses an FPGA to detect the main features of a scene and a 3D sound generator to generate audio stimuli. The software of The vOICe performs a sequential scan of the incoming image, encoding every pixel into a sound, which turns out to make patient training quite complex, and imposes a severe limitation on the frame rate. EAV performs computations that are exclusively based on stereo-vision, discarding some other relevant features of the image, as color.

Moreover there are other kinds of systems based on GPS localization such as Trekker [9] developed by HP which continuously pinpoints the user position, offering talking digital maps and providing diverse information in the context of the user surroundings. The main drawback that GPS systems present is that they are not able to help the users in real environments with people and obstacles since they are not able to prevent the user from colliding with them.

The next section focuses on explaining the main characteristics of the bio-inspired vision model implemented. Section 3 exposes details related to its implementation on a GPU. In the last point, we discuss the outcomes of our system, concluding with some comments about the work described in this paper.

2 The Bio-inspired Vision Model

Color vision in humans is based on three different cone pigment types. Each pigment type is most sensitive to light of a particular wavelength, approximately 420, 530 and 560 nanometers respectively. Because of their peak sensitivities these three cone types are referred to as short-wavelength sensitive (S cones) which make possible the perception of what we identify as the blue color, medium wavelength sensitive (M cones) related with the perception of green and long wave-length sensitive (L cones) which enable the perception of the red color. Each cone pigment absorbs a broad range of wavelengths. As a result light stimulates more than just one class of cones. The retina performs a processing of visual information, encoding color as separate color and spatial opponent channels [10, 11].

The implemented model is based on the application of spatial and temporal filtering over the input images performed in order to remark a number of features of the image, recalling the way in which the human visual system works. A similar model is used in systems Retiner[7,8] and Vis2Sound[4]. But in these cases, FPGA based devices are used to achieve real time performance.

Fig. 1 shows the reference architecture for the processing carried out by the model. As we can observe, the system performs a linear combination between the output from the spatial and temporal filters to obtain a perceptual image in which the main features of the scene are remarked as well as those parts where changes are taking place. Other models use similar operations to generate saliency maps [12].

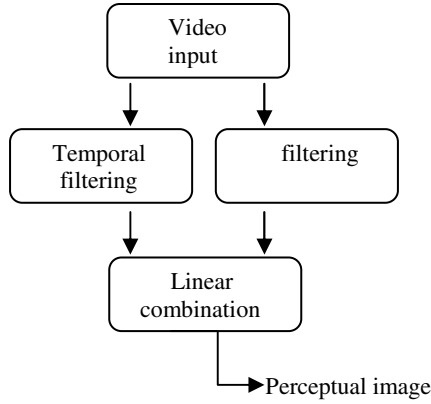


Fig. 1. Computation flow for the input images

2.1 Spatial Filtering

In order to carry out the spatial filtering, the model separates the video input into three-color planes (Red, Green, Blue), whose gain can be selected independently in order to enhance some kinds of chromatic characteristics of the scene. Over these color channels and over the intensity channel, obtained as a linear combination of the R, G and B channels, we can design a set of spatially opponent filters to simulate the function of bipolar cells.

This opposition between the center and the periphery of the receptive field of these cells is modeled with a Difference-of-Gaussians (DoG), according to Eq. (1),

$$DoG = G_{\sigma_1} - G_{\sigma_2} = \frac{1}{\sqrt{2\pi}} \left[\frac{1}{\sigma_1} \exp \left[-\frac{x^2+y^2}{2\sigma_1^2} \right] - \frac{1}{\sigma_2} \exp \left[-\frac{x^2+y^2}{2\sigma_2^2} \right] \right] \quad (1)$$

where Gaussians G_{σ_1} and G_{σ_2} are applied to different color channels.

This vision model performs two filtering operations: enhancing color contrast (magenta vs. green, and yellow vs. blue), and an additional intensity contrast filtering enhancing the edges of the scene. To be able to extract this information, we compute a convolution between the image and specific Difference-of-Gaussians masks, which can be defined according to Eq. (2),

$$Dog(A, \sigma_1, B, \sigma_2) = Gauss(A, \sigma_1) - Gauss(B, \sigma_2) \quad (2)$$

where A is the stream of information that will be convolved with a Gaussian mask with σ_1 , being B the stream operated with the kernel defined by means of the σ_2 parameter. In the present approach, both A and B are set to linear combinations of color channels. The spatial filtering implemented is defined mathematically in Eqns. (3) to (7).

$$LS_vs_M = Dog\left(\frac{R+B}{2}, \sigma_1, G, \sigma_2\right) \quad (3)$$

$$LM_vs_S = Dog\left(\frac{R+G}{2}, \sigma_1, B, \sigma_2\right) \quad (4)$$

$$LMS = Dog(I, \sigma_1, I, \sigma_2) \quad (5)$$

$$Spatial\ filtering = \alpha(LS_vs_M) + \beta(LM_vs_S) + \gamma(LMS) \quad (6)$$

$$\alpha + \beta + \gamma = 1 \quad (7)$$

where L stands for the red channel, M stands for the green channel, S stands for the blue channel and I stands for the intensity channel. α, β, γ are weighting constants that can be set by the user depending on the specifically features to be enhanced.

We have chosen this combination of the color input channels according the way the human brain combines the signals from the three cone types, two chromatic and one achromatic system [10].

2.2 Temporal Filtering

In some cases, it might be interesting to enhance image changes between consecutive frames, thus stressing those parts of the scene where changes are taking place (due to external movement, ego-motion or both). The temporal enhancement takes as inputs two or more consecutive frames. In order to carry out the temporal filtering, the system performs the difference between the actual frame and the average of the n previous ones.

3 GPU Implementation

The requirements for this application as portability, constrained power consumption, real-time performance and customization, lead us to consider the GPU NVIDIA ION2 as the best option to design and develop the system. Some others aid systems like Vis2Sound use FPGA or DSP devices since they provide a high computation capability in a small and low power device. However, the GPU used in this system has 16 processors integrated on a platform with sufficient battery autonomy (about 4 hours). Moreover, the system takes advantage of the Intel ATOM N450 processor, integrated in the used netbook, which is faster than other FPGA built-in processors, such as PowerPC. Furthermore the cutting-edge GPU technology used provides more flexibility to develop and customize dynamically the application and more competitive time-to-market than other devices.

Our target GPU consists of two streaming multiprocessors. Each streaming multiprocessor has one instruction unit, eight stream processors (SPs) and one local memory (16KB), so it has 16 SPs in total. The eight SPs in the same streaming multiprocessor are connected to one instruction unit. This means that the eight SPs execute the same instruction stream on different data (called *thread*). In order to extract the maximum performance of SPs by hiding memory access delay, we need to provide four threads for each SP, which are interleaved on the SP. Therefore, we have to provide at least 32 threads for each streaming multiprocessor.

To optimize the use of the available multiprocessors, the parameters to be determined are the number of threads/block and the shared memory space between the threads of each block.

To accurately size the modules we have used the CUDA Occupancy Calculator tool that shows the occupation of the multiprocessor's cache and its percentage of utilization [13]. The thread block size is chosen in all cases so that multiprocessor occupancy is 100%. The size of the GRID (number of processing blocks to be executed by the kernel) is dynamically set according to the size of the image.

3.1 Spatial Filtering Implementation

In order to optimize the spatial filtering process we take advantage of the linear property of the convolution. Therefore, we can reduce the processing only to the convolution of each color channel with two different Gaussian masks. Then we linearly combine these filtered channels. Eqns. (8) to (10) describe mathematically this simplification [14].

$$LS_vs_M = \frac{1}{2}(Gauss(R, \sigma_1) + Gauss(B, \sigma_1)) - Gauss(G, \sigma_2) \quad (8)$$

$$LM_vs_S = \frac{1}{2}(Gauss(R, \sigma_1) + Gauss(G, \sigma_1)) - Gauss(B, \sigma_2) \quad (9)$$

$$LMS = \frac{1}{3}(Gauss(R, \sigma_1) + Gauss(G, \sigma_1) + Gauss(B, \sigma_1)) - \frac{1}{3}(Gauss(R, \sigma_2) + Gauss(G, \sigma_2) + Gauss(B, \sigma_2)) \quad (10)$$

Moreover the Gaussian mask is separable, so each mask can be decomposed into two uni-dimensional filters, one for the rows and one for the columns. Therefore we can reduce the number of multiplications needed to calculate each filtered pixel from n^2 to $2n$ being $n \times n$ the mask dimensions.

In order to carry out these convolutions in real time we have developed two CUDA filtering kernels, one for the rows and another one for the columns. Both modules are very similar so we detail only the one for rows.

First of all we transfer the image from the CPU global memory to the GPU global memory. Each datum is stored as an 8-bit unsigned int. Also we store the coefficients of the two filters in the constants memory.

The convolution operation requires a neighborhood with the same width of the filter mask to calculate each pixel. So each thread transfers one datum from the global memory to the shared memory. In order to get the maximum precision and to avoid bank conflicts, these data are stored in the shared memory as floating point data. Therefore there are $n - 1$ threads per block that only load data, but do not calculate any filtered pixel. This distribution of the workload is depicted in Fig. 2. So as not to waste too many threads in the loading stage, the block size must be large enough compared to the filter mask width. In this case the block size is set to 1×128 , and the filter width is 7, so only 6 threads are wasted per processing block. The block width is set to 128 to achieve the required alignment when accessing to global memory, and also to optimize the multiprocessor utilization.

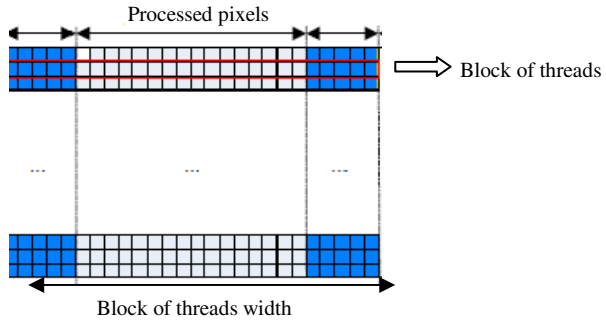


Fig. 2. Distribution of the image pixels between the blocks of threads

When all the data are stored in the global memory each thread multiplies the filter coefficients, stored in the constants memory, with the corresponding pixel and its neighborhood. Then the result is stored in the global memory. Each thread repeats this procedure twice, once for each mask. Therefore we carry out the two rows filtering with just one read access to the global memory.

The columns filters are computed in a similar way, taking as input the previously filtered images.

Once the filtering process is finished we just need to linearly combine partial results as we have explained earlier, exploiting the parallelism provided by the GPU. Finally we convert the resulting image to an 8-bit integer representation and we transfer it to the CPU.

3.2 Temporal Filtering Implementation

The inputs for this module are $n+1$ gray scale images previously filtered with a Gaussian mask to reduce the noise. As the filtered intensity channel is calculated in the spatial filtering stage, this frame is stored in the GPU global memory employing a circular buffer. So in this stage the difference between the current intensity channel and the mean of the n previous ones is performed, where n is between one and three. This difference is carried out using a floating point representation.

Finally, a linear combination between the spatial filtered image and the temporal filtered one is computed obtaining the Perceptual Image. The weighting factors in this linear combination can be customized according to the particular needs.

This final image is converted to an 8-bit integer representation and transferred to the CPU.

It should be notice that when doing transferences between the CPU global memory and GPU global memory an 8-bit representation is employed in order to minimize the transference delay which could be the bottle-neck of the application.

4 Results

As we have mentioned before the complete model is implemented in a GPU NVIDIA ION2 and it is able to perform at a maximum rate of 35 frames per second when working with 640x480 RGB images. Moreover the spatial filtering stage only can be computed at 40 frames per second. Nevertheless the developed GPU kernels can be run in any NVIDIA GPU compatible with CUDA and are scalable to the number of multiprocessor of each GPU.

Fig. 3 shows the result of applying the spatial filtering model, explained in section 2.1, to an image. As we can see in the original image (Fig. 3a) we can hardly distinguish the numbers. In Fig. 3(b) we can observe how the LS_vs_M combination clarifies red pixels while it obscures the green ones. The LM_vs_S combination enhances the orange, green and yellow pixels while it obscures the blue ones. Finally the achromatic combination enhances the edges. A linear combination of these three images enables us appreciate properly all the numbers, therefore we can observe how the system is able to remark the main features of the scene.

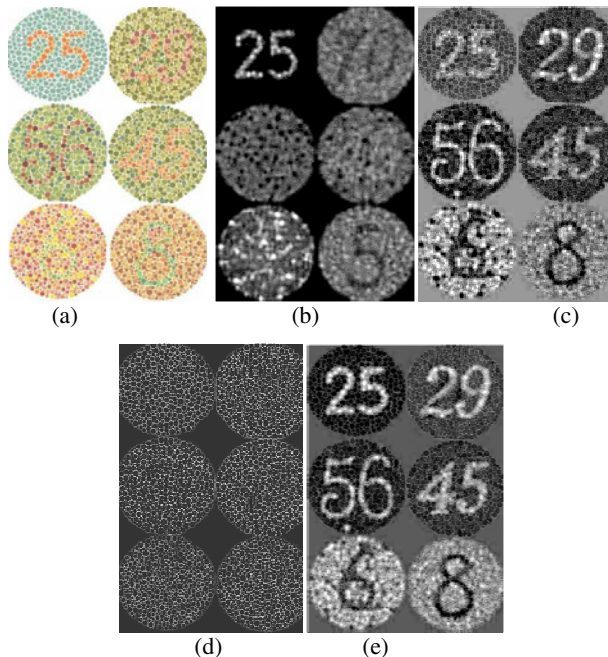


Fig. 3. (a) Original image. (b)LS_vs_M. (c) LM_vs_S (d)LMS. (e) Image resulted from the linear combination of the chromatic and the achromatic channels.

5 Conclusion and Future Work

We have presented a bio-inspired vision model able to remark the relevant information from a scene such as persons and obstacles. In order to carry out on-line real-time image processing we have implemented the system on a GPU NVIDIA ION2 which performs at a maximum rate of 35 frames per second.

Since the used GPU is integrated in a light-weight and battery operated netbook, the developed system presents several fields of application, such as aid systems for the visually impaired or autonomous navigation systems.

The system can be applied in sensorial transduction systems in which the main features of the scene are used to generate non visual stimuli such as haptic or audio stimuli.

Regarding to the aid systems, the output image of the developed model will be used to modulate the degree of enhancement of the different regions of the image. Therefore the system could present to the user, through an HMD device, a real color image in which the main features of the scene appear remarked and the glares appear attenuated. We are going to conduct a series of tests with a group of Retinitis Pigmentosa affected to demonstrate the usefulness of this model. Specifically for this group, the system is going to be used to enhance image features in low contrast environments where those affected experience several difficulties.

Moreover we are planning to implement some of the most useful visual enhancements using others embedded devices based on ARM processor in order to obtain real-time processing in smaller and lighter devices than heretofore employed net-book.

Acknowledgments. This work has been supported by the Junta de Andalucía Project P06-TIC-02007, the Spanish National Grants RECVIS (TIN2008-06893-C03-02) and DINAM-VISION (DPI2007-61683), and the project GENIL-PYR-2010-19.

References

1. World Health Organization (2011), <http://www.who.int/blindness/en/index.html>
2. Massof, R.W., Rickman, D.L.: Obstacles encountered in the development of the low vision enhancement system. *Optom. Vis. Sci.* 69, 32–41 (1992)
3. Ureña, R., Martínez-Cañada, P., Gómez-López, J.M., Morillas, C., Pelayo, F.: A portable low vision aid based on GPU. In: First International Conference on Pervasive and Embedded Computing and Communication Systems, PECCS 2011, Vilamoura, Portugal, March 5-7, pp. 201–206 (2011); ISBN: 978-989-8425-48-5
4. Morillas, C., Cobos, J., Pelayo, F., Prieto, A., Romero, S.: VIS2SOUND on Reconfigurable Hardware. In: International Conference on Reconfigurable Computing and FPGAs (2008)
5. The vOICe System, <http://www.seeingwithsound.com>
6. EAV System, <http://www.iac.es/proyect/eavi/index.html>
7. Pelayo, F.J., Romero, S., Morillas, C., Martínez, A., Ros, E., Fernández, E.: Translating Image Sequences into Spike Patterns for Cortical Neuro-stimulation. *Neurocomputing* 58-60, 885–892 (2004)
8. Morillas, C., Romero, S., Martínez, A., Pelayo, F., Reyneri, L., Bongard, M., Fernández, E.: A Neuroengineering suite of Computational Tools for Visual Prostheses. *Neurocomputing* 70(16-18), 2817–2827 (2007)
9. Trekker System, <http://www.ulva.com/Online-Store/GPS/trekker.htm#more>
10. Sekuler, R., Blake, R.: Perception. McGraw-Hill, New York (1994)
11. Spillmann, L., Werner, J.: Visual Perception. Academic Press, Inc., London (1990)
12. Itti, L., Koch, C.: Computational modelling of visual attention. *Nature Reviews Neuroscience* 2, 194–203 (2001)
13. NVIDIA Corporation, NVIDIA CUDA C Programming Best Practices Guide 2.3 (2009)
14. Martínez, A., Reyneri, L.M., Pelayo, F.J., Romero, S.F., Morillas, C.A., Pino, B.: Automatic generation of bio-inspired retina-like processing hardware. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 527–533. Springer, Heidelberg (2005)

Bipartite Graph Matching on GPU over Complete or Local Grid Neighborhoods

Cristina Nader Vasconcelos¹ and Bodo Rosenhahn²

¹ Universidade Federal Fluminense, Instituto de Computação, Brazil

² Leibniz Universität Hannover, Institut für Informationsverarbeitung, Germany

crisnv@ic.uff.br,
rosenhahn@tnt.uni-hannover.de
www.ic.uff.br/~crisnv/

Abstract. Several schedule and assignment tasks can be modeled as a bipartite graph matching optimization, aiming to retrieve an optimal set of pairs connecting elements from two distinct sets. In this paper we investigate how to compute a weighted bipartite graph matching on Graphics Processing Units (GPUs) inspired by its low cost and increasing parallel processing power. We propose a data-parallel approach to be computed using GPUs processing kernels based on *The Auction Algorithm*, and data structures that allow it to be applied to problems modeled over complete bipartite graphs and also over huge bipartite graphs with connections across the neighborhood systems from two sets of 1D, 2D or 3D data grids.

1 Introduction

In this paper we are interested in efficiently computing optimal matchings over weighted bipartite graphs. The efficiency of our proposal comes from a reformulation of *The Auction Algorithm* [1,2] and the creation of special data structures in order to compute the matching in parallel on GPU.

Modern GPUs programming schemes require the desired algorithm to be formulated in what is called a data-parallel approach. It induces the algorithm decomposition into similar operations, grouped into kernels, to be applied independently and in parallel to data streams. The GPU parallelism occurs by processing concurrently the same kernel over different data stream elements.

The classical algorithm used to compute the bipartite graph matching is *The Hungarian Algorithm* [4]. Some of its steps can be naturally computed concurrently in such data-parallel approach, but others intrinsically demand the introduction of a non GPU-friendly message pass scheme or of global synchronization points, as they provoke ambiguities if computed concurrently or locally. Because of this reason, its reformulation into independent processing kernels, as required for a GPU computation, is not straightforward. On the other hand, *The Auction Algorithm* was originally proposed as a distributed method [1,2], allowing its natural extension into a data-parallel formulation.

In our former proposal [6], we considered the computation on GPUs over complete bipartite graphs (Figure 1a), where every vertex of the first set is connected

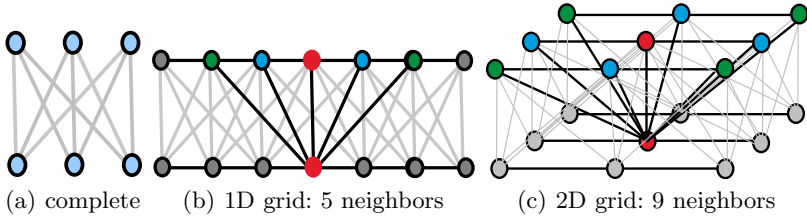


Fig. 1. Bipartite graphs

to every vertex of the second set. This scenario models applications where every instance from one set is a potential candidate to be matched to every instance from the second set.

Aiming to reduce the number of operations and GPU’s memory usage requirements of our former model [6], in this paper we propose the creation of new data structures and the algorithm adaptations to them. More specifically, we focus on the efficient representation of (and access to) the matching candidates for scenarios where those candidates belong to one of two distinct data grids (Figure 1b). This scenario models applications where each instance from one set is a potential candidate to be matched only to those instances from the second set that are within a local neighborhood defined by its grid position. Interesting applications can be modeled in that sense, like the correspondence between samples from two distinct audio signals (1D grids), pixels from two distinct images (2D grids), or between two sets of volumetric data (3D grids).

This paper is organized as follows. The Section 2 presents *The Auction Algorithm* fundamentals, Section 3 presents our GPU formulation and Section 4 presents the neighborhood treatments, data structures and algorithm adaptations proposed. Finally, Sections 5 and 6 presents our results and conclusions.

2 The Auction Algorithm

The Auction Algorithm was originally proposed as a distributed relaxation method [1,2] and is semantically described as a real auction where persons compete for objects by raising their prices through competitive bidding.

Suppose that there are $|S1|$ persons and $|S2|$ objects (where $(|S1| \leq |S2|)$). Its goal is to match them in a way such that each person should be assigned to a single object and each object should be assigned to at most a single person. The matching should respect the restriction that each person i can only be assigned to object j if the pair $(i; j)$ belongs to a given set $Ob(i)$ of possible matching pairs. Analogously, for each object j it is possible to define $Pe(j)$ as the set of persons that can be matched with j . There is a benefit matrix B_{ij} for matching a person i with an object j , and the goal of the auction is to assign persons to objects and maximize the total benefit, defined as:

$$\sum_{i=0}^{s1-1} b_{ij_i} \tag{1}$$

The auction algorithm introduces an economic equilibrium problem that can be seen as a dual problem. It supposes that an object j has a price p_j and that the person who receives the object must pay the price p_j . As each person associates an benefit b_{ij} with each object, then the object j net value of for person i is related with the difference between the corresponding benefit the object price. Each person i would logically want to be assigned to an object j_i with maximal value, that is, with

$$b_{ij_i} - p_{j_i} = \max \{b_{ij} - p_j\}. \quad (2)$$

The Auction Algorithm goal is to find an equilibrium assignment and its corresponding price vector. The algorithm iterates between two steps: a bidding phase and an assignment phase. During the bidding phase, each unassigned person finds an object j which offers maximal value (according to equation 2) and makes a bid for that object offering a bidding increment γ_i calculated as:

$$\gamma_i = v_i - w_i + \epsilon \quad (3)$$

where (v_i) and (w_i) are respectively the maximal and second maximal net values of objects that the person i is interested in.

After the bidding phase, the algorithm turns into the assignment phase. Then, each object j , if it was selected as a best object by any nonempty set of people $Pe(j)$, determines the highest bidder by:

$$i_j = \arg \max_{i \in Pe(j)} \gamma_i \quad (4)$$

Using the highest bidding increment the object raises its price and gets assigned to the person i , considered as highest bidder i_j . If the object was previously assigned to other person, that person becomes unassigned.

Iterating between those two phases, the algorithm continues until all persons have an assigned object. The termination with a feasible assignment (if it exists) is assured by noting that once an object is assigned to any person, it will never be turned into an unassigned object again. Besides, if an object receives a bid in k iterations, its price must exceed its initial price by at least $k\epsilon$, thus, at some point of the iteration, an assigned object will become expensive enough to be judged less valuable (according to equation 2) than some other object that has not received a bid so far. It follows an object can receive a bid in a limited number of iterations while some other object still has not yet received any bid. Once all objects have received at least one bid, the auction terminates.

3 Parallel Auction Computation on GPU

Our proposal reformulates *The Auction Algorithm* for a GPU computation using an iteration cycle with one kernel for the bidding phase, one kernel for the assignment phase and a loop that iterates between the phases triggering the GPU threads until convergence is achieved (Figure 2).

Before the matching computation begins, data streams have to be created on GPU representing some of the algorithm's temporary data and the bipartite

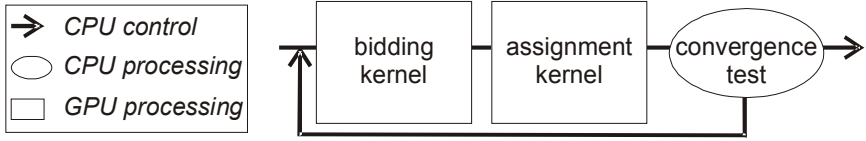


Fig. 2. Proposed iteration cycle: independent kernels

graph structure. The existing edges and its weights are represented according to the graph model adopted (complete or local) and are discussed in Section 4.1.

The other structures adopted are represented as unidimensional streams. There are dynamic value streams created to represent: the objects prices (initially set as zero); the objects indexes associated with each person (or a sentinel value if not associated); the persons indexes associated with each object (or a sentinel value if not associated); the bids values suggested by each person in last iteration; the bids targets from each person in last iteration (an object index, if any).

For efficiently accessing and storing data, we propose to split some structures that are naturally represented with a 2D sparse matrix of $n*n$ elements into two 1D dense streams, of n elements each. For instance, we store the persons bids values to the objects, into two unidimensional streams: one for storing the value and one for the target index. Such separation was possible because a person can only bid for a single object in each iteration. In our formulation, each person can still write its bidding concurrently to all the others with no communication between them, but requiring much less storage space.

Once the iteration cycle starts, the bidding kernel (bk) is executed by every person concurrently. During its execution, a person decides if he is going to suggest a bid and which object will be the target, or if he is already associated to an object(does not ask for another).

Next, the assignment kernel (ak) is executed by every object concurrently, for checking if they have received any bid recently. If true, they are responsible for updating their own prices and the current bidder. They are also responsible for setting a previous bidder free (if any) to let them restart bidding.

Once that GPU can not trigger its own processes and threads, the convergence decision is taken on CPU. It evaluates if all persons have already been assigned to an object, and, if not, it triggers the bidding kernel (bk) and the assignment kernel (ak) once more.

Data transfers of GPU streams into CPU memory space are one of the most expensive operations in CPU-GPU computing and such cost is directly proportional to the amount of data retrieved. Thus, our proposal reduces the amount of data consulted on CPU for the convergence decision into a minimum.

Our proposal tracks (from within the GPU kernels), the algorithm cases when the number of assigned persons changes. A single value representing this counter is transferred between the GPU and CPU at the end of each iteration cycle. Those changes occur when a free object receives the first bid. In cases when and assigned

object changes its corresponding bidder, the number of assigned persons remains unchanged. These observations are assured by the algorithm property that once an object is assigned for any person it never turns to unassigned again.

4 Neighborhood and Grids

Checking (and representing) edges connecting every sample from a first set against every sample from a second set induces unnecessary calculations and waste of storage space in applications where only a subset of those connections are actual real matching candidates. This observation inspires our approach for constructing a bipartite graph with edges based on grid adjacency, in which a node from a first grid will only be connected to those nodes within a neighborhood range in the second grid (Figure 1b).

4.1 Memory Requirements

Traditionally, the existent edges in a graph are represented into an adjacency matrix that is a $N \times N$ matrix with rows and columns labeled by the nodes indexes, with a 1 or 0 in its positions according to whether the corresponding nodes are adjacent or not. In our previous solution [6], assuming bipartite graphs ($N = S1 \cup S2$, $S1 \cap S2 = \emptyset$) the adjacent matrix was replaced with the auction benefit matrix ($S1 \times S2 \leq N \times N$), where its rows/columns indexes represent nodes from the first/second set respectively and the values stored represent the edges weights as the benefit for matching the corresponding pair of nodes.

Some applications have intrinsic assumptions about the maximum distance between the matching candidates, as if observing some physical effect or if based on chain alignment criteria. Consider, for instance, the matching problem over a pair of 2D grids composed by $S1 = S2 = N \times N$ samples each. Modeling the complete bipartite graph connecting samples from one grid versus the other would require $O((N \times N)^2)$ edges. These facts motivated the grid's formulation proposed here in which the edges are taken only connecting elements among the grids natural neighborhood. We suggest two different approaches, depending on the matching candidates criteria connecting the input sets $S1$ and $S2$:

- Complete candidature: any element of $S1$ can be matched with any element of $S2$. The auction benefit matrix is a dense matrix.
- Neighborhood candidature: each element i of $S1$ can only be matched with elements of $S2$ that belong to a local neighborhood defined around i 's position transfered to the second grid. In such case, the auction benefit matrix is a sparse matrix.

Storing the auction benefits matrix on GPU memory space is the largest memory requirement of our former algorithm ($S1 \times S2$) [6], as the other data structures can be kept with the dimension of only one of the input distinct sets ($S1$ or $S2$). Even though such solution, based on a complete bipartite representation, can be easily adapted to any kind of bipartite graph by setting negative

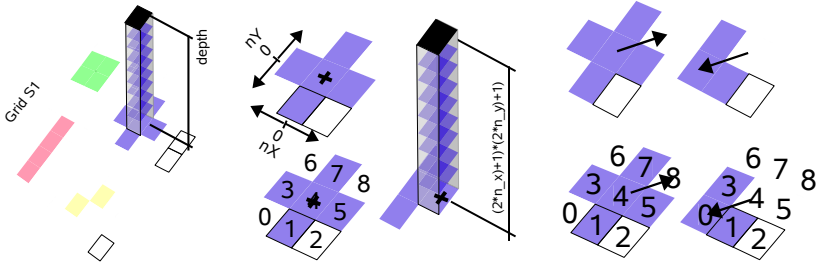


Fig. 3. Matching a pair of 2D grids(from left to right): Depth-Grid; Neighbors to Depth Mapping; Invertible Person-Object Mapping

infinity benefits to non-existent edges, in practical such representation imposes a limitation over the graph’s size bounded by GPU memory available. Besides, in cases were $S1$ and $S2$ are huge, this model may turn not only memory requirements very high but also the optimization computing time unacceptable.

The ideal memory requirements for the representation of the neighborhood candidature can be formally calculated as follows. For 1D grids, consider a node $i \in S1$, and a neighborhood maximum distance of md . There is an edge to every node $j \in S2$ where $i - md \leq j \leq i + md$. In such cases, the benefit matrix is a sparse matrix, having at most $2 * md + 1$ non-zero values in each row. For 2D, 3D or d -dimensions grids, and a neighborhood maximum distance of md , there is an edge to every node $j \in S2$ where $i - md \leq j \leq i + md$ in each of the dimensions of the grid, thus, the benefit matrix is a sparse matrix, having $(2 * md + 1)^d$ non-zero values for each row. Inspired by applications where $md \ll S2$, we seek for replacing the complete graph representation $(S1 \times S2)$ with a storage and algorithm solution with the ideal memory space requirement of $S1 \times (2 * md + 1)^d$.

4.2 Data Structures and Kernels Adaptation

In this section it is shown how to decrease the memory usage and number of computations to the ideal presented for neighborhood candidatures.

In a new kernel, before the auction starts, we evaluate the benefits for matching a node in $S1$ with each candidate from $S2$ within a neighborhood maximum distance md . Then, these benefits are stored in a memory called here as depth-grid with $(S1 \times (2 * md + 1)^d)$ entries, of the same base resolution of the grid $S1$, but with one additional dimension, we called here as depth. The elements that vary only in depth coordinate, represent the set of edges (and corresponding benefits) connecting a single element in $S1$ to all its candidate neighbors in $S2$. This structure stores only the auction benefits for the existent edges. The list of edges itself is obtained by dynamically calculating (no storage space) the first node ($\in S1$) from the entry grid’s coordinates and the second node ($\in S2$) from the entry depth coordinate. The 2D case is illustrated in Figure 3.

The auction processing kernels have to be adapted to this new representation of the benefits in a depth-grid in order to retrieve $Pe(j)$ and $Ob(i)$ dynamically

and to register the biddings and assignments correctly. Originally, the bidding kernel had a loop over a row of the benefits matrix. In the depth-grid representation, each instance of this kernel loops in depth from the same grid position.

Dynamical conversions between the local index (ranging from 0 to the maximum number of neighbors -1) have to be taken to a global reference, in order to discover which person is bidding for which object, and vice-versa, and also between local indexes (person to object local indexes and vice-versa). Mappings are calculated dynamically that convert a depth in the matrix to the corresponding neighbor position (and vice-versa); to recover from (and to) a local index (indexes from 0 to $((2 * md) + 1)^d - 1$) to the global index (in $S1$ or $S2$ coordinate systems); or to local index based on different grid positions (see figure 3).

For performance reasons, another level of indirection was introduced. As GPU’s processing kernels are grouped into blocks, induced by our proposal to group those kernels processing neighboring elements (neighboring persons during bidding, and neighboring objects during assignment), the kernels within the same block tend to read data from a well defined memory region. As a starting point in each block computation, these memory targets are dynamically retrieved from global memory and the corresponding data are preloaded into Shared Memory, offering fast data accesses through Shared Memory local mappings.

5 Results

The proposed Depth-Grid representation has its memory requirement reduction warranted if implemented in CPU or GPU (as long as $(2 * md + 1)^d < S2$). As we are interested in the parallel computation, the results were obtained testing the auction in two scenarios with CUDA 3 support: the first one, running in a notebook with a Nvidia GeForce 9600 GT with 512MB of VRAM, and the second with a stronger desktop machine with a Nvidia GeForce with 1536MB.

The table presented in Figure 4 illustrates the obtained results. It shows that when adopting the complete graph representation even a small grid of $S1 = S2 = 256^2$ elements became impracticable in both GPUs tested, but with the Depth-Grid representation proposed even grids with $S1 = S2 = 1024^2$ and 289 neighbors could be processed in the weakest GPU, and with 1089 neighbors

2D grid size	Depth grid representation in Mb								Complete Representation in Mb	
	<i>md</i>	1	2	4	8	16	32	64		
	<i>1D</i>	3	5	9	17	33	65	129		
<i>depth</i>	9	25	81	289	1089	4225	16641			
16 ^ 2		0.002	0.006	0.02*	*	*	*	*	0.06	
32 ^ 2		0.01	0.02	0.08	0.3*	*	*	*	1	
64 ^ 2		0.04	0.10	0.3	1.1	4.3*	*	*	16	
128 ^ 2		0.14	0.39	1.3	4.5	17	66*		256	
256 ^ 2		0.56	1.6	5.1	18	68	264	1040	4096	
512 ^ 2		2.3	6.3	20	72	272	1056	4160	65536	
1024 ^ 2			9	25	81	289	1089	4225	16641	1048576

* Out of grids bounds

- Supported by both GPU tested
- Supported by the GPU with 1.5 Gb
- Demand more memory then both GPU tested

Fig. 4. Depth-Grid versus Complete Representation memory usage

in the strongest tested. This impacts directly on processing timing by reducing the number of memory accesses realized during the auction computation. Aiming to let the scientific community to be able to analyze their own experiments on complete graphs or graphs over data grids, the source codes for both representation developed in CUDA [3] will be available for download from the author's homepage [5].

6 Conclusions

The optimization of a bipartite graph matching considering the possibility of matching between every element from a first set against every element from a second set (complete bipartite graphs) can be impracticable when dealing with huge sets. This paper presented a GPU-friendly formulation for a bipartite graph matching and extended our previous work [6] by dealing with huge data sets and with the special cases when input data have adjacency relations.

Our main contribution is to show that in cases where the neighborhood size is much smaller than the number of elements in the second grid, a neighborhood candidature represented as a complete bipartite graph is a waste of resources. We propose new data structures and modifications to our former algorithm parallel kernels in order to consider neighborhood constraints over the matching candidate pairs of nodes, thus, inducing bipartite graphs with more reasonable memory usage and lower computational requirements (from $(S1 \times S2)$ to $(S1 \times (2 * md + 1)^d)$). The number of computations within algorithm's kernels is also reduced by dynamically mapping the neighboring edges and the candidate sets ($Pe(j)$ and $Ob(i)$) instead of checking every element in the opposite grid.

Spacial related data are common in several applications including finding correspondences between 1D, 2D or 3D signals. The results presented show that the extensions proposed here allow the computation of the matching over huge data grids, opening the possibility of considering such graph and technique as a model in applications using huge sets of data and demanding fast results. The source code will be available for download from the author's homepage [5].

References

1. Bertsekas, D.P.: Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications* 1, 7–66 (1992)
2. Bertsekas, D.P., Castanon, D.A.: Parallel asynchronous hungarian methods for the assignment problem. *ACM Trans. on Database Syst.* 5, 212–230 (1993)
3. Cuda programming guide (2007), <http://developer.download.nvidia.com/>
4. Kuhn, H.W.: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 83–97 (1955)
5. Vasconcelos, C.N.: Bgm code, <http://www.ic.uff.br/~crisnv/bgm/bgm.html>
6. Vasconcelos, C.N., Rosenhahn, B.: Bipartite graph matching computation on GPU. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) *EMMCVPR 2009*. LNCS, vol. 5681, pp. 42–55. Springer, Heidelberg (2009)

GPU Computation in Bioinspired Algorithms: A Review

M.G. Arenas, A.M. Mora, G. Romero, and P.A. Castillo

Department of Architecture and Computer Technology. CITIC
University of Granada (Spain)
{mgarenas, amorag, gustavo, pedro}@atc.ugr.es

Abstract. Bioinspired methods usually need a high amount of computational resources. For this reason, parallelization is an interesting alternative in order to decrease the execution time and to provide accurate results. In this sense, recently there has been a growing interest in developing parallel algorithms using graphic processing units (GPU) also referred as GPU computation. Advances in the video gaming industry have led to the production of low-cost, high-performance graphics processing units (GPUs) that possess more memory bandwidth and computational capability than central processing units (CPUs). As GPUs are available in personal computers, and they are easy to use and manage through several GPU programming languages (CUDA, OpenCL, etc.), graphics engines are being adopted widely in scientific computing applications, particularly in the fields of computational biology and bioinformatics. This paper reviews the use of GPUs to solve scientific problems, giving an overview of current software systems.

1 Introduction

Recently there has been a growing interest in Graphics Processing Unit (GPU) computation. The fact that GPUs have the ability to perform restricted parallel processing has elicited considerable interest among researchers with applications requiring intensive parallel computation.

GPUs are specialized stream processors, initially useful for rendering graphics applications. Typically, a GPU is able to perform graphics manipulations at a much higher speed than a general purpose CPU, since the graphics processor is specifically designed to handle certain primitive operations which occur frequently in graphics applications. Internally, the GPU contains a number of small processors that are used to perform calculations. Depending on the power of a GPU, the number of threads that can be executed in parallel on such devices is currently in the order of hundreds and it is expected to multiply in a few months. Nowadays, developers can write (easily) their own high-level programs on GPU. Due to the wide availability, programmability, and high-performance of these consumer-level GPUs, they are cost-effective for, not just game playing, but also scientific computing. Now, GPUs are exposed to the programmer as a

set of general-purpose shared-memory SIMD (Single Instruction Multiple Data) multi-core processors. This makes these architectures well suited to run large computational problems, such as those from bioinformatics area.

Then, the goal of this article is to review the use of GPUs to solve bioinformatic problems, explaining the general approach to using a GPU and given an overview of currently usable software systems.

To this end, the rest of this paper is structured as follows: Section 2 presents GPUs as highly parallel devices architectures. Section 3 gives a background on the different higher level programming languages used to profit GPUs. Finally, Section 4 reviews the related works in bioinformatic applications on GPUs, followed by a brief conclusion (Section 5).

2 Throughput, Parallelism and GPUs

Moore's Law describes a long-term trend in the history of computing hardware: the number of transistors that can be placed inexpensively on an integrated circuit has doubled approximately every two years. The trend has continued for more than half a century and is not expected to stop (theoretically until not too many years above 2015). In 2005 Gordon Moore stated in an interview that the law cannot be sustained indefinitely because transistors would eventually reach the limits of miniaturization at atomic levels.

Parallel computation has recently become necessary to take full advantage of the gains allowed by Moore's law. For years, processor makers consistently delivered increases in clock rates and instruction-level parallelism, so that single-threaded code executed faster on newer processors with no modification. Now, to manage CPU power dissipation, processor makers favor multi-core chip designs, and software has to be written in a multi-threaded or multi-process manner to take full advantage of the hardware. Graphics processors have rapidly matured over the last years, leaving behind their roots as fixed function accelerators and growing into general purpose computational devices for highly parallel workloads. Some of the earliest academic work about GPUs as computational devices date back to University of Washington in 2002 [1] and Stanford in 2004 [2].

3 GPUs Programming

3.1 Programming Model

In respect to the programming tools which a developer can use to exploit a GPU, most of the Application Program Interfaces (APIs) are based on C-like languages, but having some restrictions to improve the parallel execution, such as no recursion or limited pointers. Some of them use the open source compiler LLVM [3] from University of Illinois.

From 2003 the two main GPU developers, ATI and NVIDIA, started selling hardware solutions that need to be programmed with proprietary APIs. Despite previous work, the first widely supported GPUs were DX10 generation GeForce

8 series from NVIDIA, using the famous API *CUDA*. On the other hand, the Radeon HD2xxx series from ATI, applied the API *Close To Metal*.

In addition, some people at Apple betted on the potential of GPUs and developed another tool, known as *OpenCL*. While, at the same time, Microsoft created *DirectCompute* (for Windows).

OpenCL aimed to become the OpenGL of heterogeneous computing for parallel applications. It is a cross-platform API with a broad and inclusive approach to parallelism, both in software and in hardware. While explicitly targeting GPUs, it also considers multi-core CPUs and FPGAs. The applications are portable across different hardware platforms, varying performance while keeping functionality and correctness. The first software implementations date back to 2009.

3.2 Execution Model

OpenCL, DirectCompute and CUDA are APIs designed for heterogeneous computing with both a host CPU and an optional GPU device. The applications have serial portions, that are executed on the host CPU, and parallel portions, known as *kernels*. The parallel kernels may execute on an OpenCL compatible device (CPU or GPU), but synchronization is enforced between kernels and serial code. OpenCL is distinctly intended to handle both task and data parallel workloads, while CUDA and DirectCompute are primarily focused on data parallelism.

A kernel applies a single stream of instructions to vast quantities of data that are organized as a 1-3 dimensional array. Each piece of data is known as a work-item in OpenCL terminology, and kernels may have hundreds or thousands of work-items. The kernel itself is organized into many work-groups that are relatively limited in size; for example a kernel could have 32K work-items, but 64 work-groups of 512 items each.

Unlike traditional computation, arbitrary communication within a kernel is strongly limited. However, communication and synchronization is generally allowed locally within a work-group. So work-groups serve two purposes: first, they break up a kernel into manageable chunks, and second, they define a limited scope for communication.

3.3 Memory Model

The memory model defines how data is stored and communicated within a device and between the device and the CPU. DirectCompute, CUDA and OpenCL share the same four memory types (with different terminology):

- Global memory: it is available for both read and write access to any work-item and the host CPU.
- Constant memory: is a read-only region for work-items on the GPU device, but the host CPU has full read and write access. Since the region is read-only, it is freely accessible to any work-item.

- Private memory: is accessible to a single work-item for reads and writes and inaccessible for the CPU host. The vast majority of computation is done using private memory, thus in many ways it is the most critical term of performance.
- Local memory: is accessible to a single work-group for reads and writes and is inaccessible for the CPU host. It is intended for shared variables and communication between work-items and is shared between a limited number of work-items.

4 Bioinspired Methods on GPUs

This section reviews different Evolutionary Computation (EC) approaches using GPUs found in bibliography. The main EC paradigms are: Evolutionary Strategies (ES) [4], Evolutionary Programming (EP) [5], Genetic Algorithms (GA) [6] and Genetic Programming (GP) [7,8,9] with hybridations and variations on each one. Moreover, parallel EC approaches can be classified in master-slave model, fine-grained, or coarse-grained. All the EC approaches on GPUs are parallel, thus a classification depending on the parallel model used is presented in this section. We will focus on master-slave, fine-grained (cellular EAs), coarse-grained (Island Model or Deme Model) approaches; and a hierarchical model [10].

Master-Slave Approaches. Wong et al. [11] proposed an EP algorithm for solving five simple test functions, called Fast Evolutionary Programming (FEP). In this master-slave approach, some actions are executed in the CPU (main loop of the algorithm and crossover operator), while evaluation and mutation are run in the GPU (no need of external information). The competition and selection of the individuals are performed on the CPU, while mutation, reproduction and evaluation are performed on the GPU. In this case, the reproduction step implies interaction among, at least, two individuals. A maximum speedup of x5.02 is obtained when the population size increases. This is the most common organization in GPU implementations, since no interaction among individuals is required during the evaluation, so this step can be fully parallelizable.

A GP method proposed by Harding and Banzhaf [12] uses the GPU only for performing the evaluation, while the rest of the steps of the algorithm are run on the CPU. The authors tested real-coded expressions until 10000 nodes, boolean expressions until 1500 nodes, and some real world problem where they evaluate expressions until 10000 nodes. In some cases, the results yielded speedup of thousand times.

Zhang et al. [10] adapt different EAs to a GPU using CUDA. The authors implemented an hierarchical parallel genetic algorithm using a deme model at the high level, and a master-slave schema at the low level. In this implementation, the CPU initializes the populations and distributes them to thread blocks in shared memory. Then, GPU threads within each block run a GA independently (selection, crossover, mutation and evaluation), and migrates individuals to other thread blocks in its neighborhood. In this case, no speedup results were reported.

Fine-grained Approaches. In this scheme, Wong et al. [13,14] proposed a parallel hybrid GA (HGA) where the whole evolutionary process is run on the GPU, and only the random number generation is done in CPU. Each GA individual is set to each GPU, and each one selects probabilistically an individual in its neighborhood to mate with. Just one offspring individual is generated, and replaces the old one in that GPU. The authors compare HGA with a standard GA run in a CPU and the FEP [11] algorithm. Using a new pseudo-deterministic selection method, the amount of random numbers transferred from the CPU is reduced. HGA reaches speedup of 5.30 when compared against the sequential version.

Yu et al. [15] implemented the first real cellular EA using GPU, for solving the Colville minimization problem. They place the population in a toroidal 2D grid and use the classical Von Neumann neighborhood structure with five cells. They store chromosomes and their fitness values in texture memory on the graphic card, and both, fitness evaluation and genetic operations, are implemented entirely with fragment programs executed in parallel on GPU. Real-coded individuals of a population are represented as a set of 2D texture maps. $BLX - \alpha$ crossover and non-uniform mutation are run as tiny programs on every pixel at each step in a SIMD-like fashion, solving some function optimization problems and reaching a speedup of x15 with a population of 512x512 individuals. They store a set of random numbers at the beginning of the evolution process to solve the random number generation problem when using GPU processors.

Luo et al. [16] implemented a cellular algorithm on GPU to solve three different SAT problems using a greedy local search (GSAT) [17] and a cellular GA (cGA). They saved local minimums using a random walk strategy, jumping to other search space location. The cellular GA adopts a 2D toroidal grid, using the Moore neighborhood, stored on texture GPU memory. This implementation generates the random numbers in the GPU (using a generated seed on the CPU at the beginning of the process). The GPU version reduces in about 6 times the running time.

Li et al. [18] proposed a cellular algorithm on GPU for solving some common approximation functions. The authors reported experiments using big populations (up to 10000 individuals) reaching speedups of x73.6 for some implementations.

In [19] the authors propose a fine-grained parallel immune algorithm (FGIA) based on GPU acceleration, which maps parallel IA algorithm to GPU using CUDA. The results show that the proposed method (even increasing the population size) reduces running time.

Alba et al. [20] use CUDA and store individuals and their fitness values in the GPU global memory. Both, fitness evaluation and genetic operators, are run on GPU (no CPU is used). They use a pseudo random number generator provided by the SDK of CUDA named Merseinne Twister. Their experiments include some general discrete and continuous optimization problems, and they compare physical efficiency and numerical efficacy with respect to CPU implementation.

Coarse-grained Approaches (island model). With regard to the last topology, one of the first island models on GPU approaches was published on the GPU competition of GECCO 2009 [21]. It presents some technical details of an island model entirely hard-coded on GPU, with a ring-like topology. Nevertheless, the evolutionary operators implemented on GPU are only specific to the GECCO competition, and the validity of the experiments just works on a small number of problems.

Tsutsui et al. [22] propose run a coarse-grained GA on GPU to solve the quadratic assignment problem (QAP) using CUDA. This is one of the hardest optimization problems in permutation domains. Their model generates the initial population on CPU and copied it to the GPU VRAM; then, each subpopulation in a GPU (NVIDIA GeForce GTX285) is evolved. At some generations, individuals in subpopulations are shuffled via the GPU VRAM. Results showed a speedup from x3 to x12 (using eight QAP instances), compared to the Intel i7 965 processor.

The model by Luong et al. [23] is based on a re-design of the island model. Three different schemes are proposed: The first one implements a coarse-grained EA using a master-slave model to run the evaluation step on GPU. The second one distributes the EA population on GPUs, while the third proposal extends the second one using fast on-chip memory. Second and third approaches reduce the CPU/GPU memory latency, although their parameters (number of islands, migration topology, frequency and number of migrants) must be adapted to the GPU features. Sequential and parallel implementations are compared, obtaining a speedup of x1757 using the third approach.

Pospíchal et al. [24,25] propose a parallel GA with island model running on GPU. The authors map threads to individuals, thus, threads-individuals can be synchronized easily in order to maintain data consistency, and on-chip hardware scheduler can swiftly swap existing islands between multiprocessors to hide memory latency. Fast, shared memory within the multiprocessor is used to maintain populations. Since the population size is limited to 16KB per island on most GPUs, if the population is larger, slower main memory has to be used. The migration process is based on an asynchronous unidirectional ring, that requires an inter-island communication (slower main memory has to be used). The authors report speedups up to 7000 times higher on GPU compared to CPU sequential version of the algorithm.

5 Conclusions

In this paper we have reviewed the use of GPUs to implement bioinspired algorithms to solve optimization problems. We have commented the GPU computing general approach, and given an overview of currently usable programming languages and software tools.

Most of the bio-inspired methods use the GPU mainly to speed up just the fitness evaluation (usually the most time-expensive process). In most of the EC approaches, competition and selection are performed by CPU, while fitness

evaluation, mutation and reproduction are performed on GPU (which is a massively parallel machine with shared memory). GPU allows processors to communicate with any other processors directly, thus more flexible fine-grained algorithms can be implemented on GPU.

In general, approaches found in literature obtain speedups up to several thousands times higher on GPU compared to CPU sequential versions of the same algorithms.

However, as the programming tools improve, newer EC approaches run the whole optimization algorithm on the GPU side, with no need of CPU interaction.

Acknowledgements

This work has been supported in part by the CEI BioTIC GENIL (CEB09-0010) Programa CEI del MICINN (PYR-2010-13) project, the Junta de Andalucía TIC-3903 and P08-TIC-03928 projects, and the Jaén University UJA-08-16-30 project.

References

1. Thompson, C.J., Hahn, S., Oskin, M.: Using modern graphics architectures for general-purpose computing: a framework and analysis. In: Proceedings of the 35th Annual ACM/IEEE International Symposium on Microarchitecture. MICRO 35, pp. 306–317. IEEE Computer Society Press, Los Alamitos (2002)
2. Buck, I., Foley, T., Horn, D., Sugerman, J., Fatahalian, K., Houston, M., Hanrahan, P.: Brook for gpus: stream computing on graphics hardware. ACM Trans. Graph. 23, 777–786 (2004)
3. Illinois, U.: The LLVM Compiler Infrastructure. University of Illinois at Urbana-Champaign (2011), <http://llvm.org>
4. Rechenberg, I.: Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution. Frommann-Hozboog, Stuttgart (1973)
5. Fogel, L.: Artificial Intelligence Through Simulated Evolution. John Wiley & Sons, Chichester (1966)
6. Holland, J.H.: Adaptation in natural and artificial systems. University of Michigan, Boston (1975)
7. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
8. Koza, J.R.: Genetic Programming II: Automatic Discovery of Reusable Programs. MIT Press, Cambridge (1994)
9. Koza, J.R., Andre, D., Bennett III, F.H., Keane, M.: Genetic Programming 3: Darwinian Invention and Problem Solving. Morgan Kaufman, San Francisco (1999)
10. Zhang, S., He, Z.: Implementation of parallel genetic algorithm based on CUDA. In: Cai, Z., Li, Z., Kang, Z., Liu, Y. (eds.) ISICA 2009. LNCS, vol. 5821, pp. 24–30. Springer, Heidelberg (2009)
11. Wong, M., Wong, T., Fok, K.: Parallel evolutionary algorithms on graphics processing unit. In: The 2005 IEEE Congress on Evolutionary Computation, vol. 3, pp. 2286–2293 (2005)

12. Harding, S., Banzhaf, W.: Fast genetic programming and artificial developmental systems on gpus. In: 21st International Symposium on High Performance Computing Systems and Applications, HPCS 2007, vol. 2 (2007)
13. Wong, M., Wong, T.: Parallel hybrid genetic algorithms on Consumer-Level graphics hardware. In: IEEE Congress on Evolutionary Computation, CEC 2006, pp. 2973–2980 (2006)
14. Wong, M., Wong, T.: Implementation of parallel genetic algorithms on graphics processing units. In: et al., M.G., ed.: Intelligent and Evolutionary Systems. SCI, vol. 187, pp. 197–216. Springer, Heidelberg (2009)
15. Yu, Q., Chen, C., Pan, Z.: Parallel genetic algorithms on programmable graphics hardware. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 1051–1059. Springer, Heidelberg (2005)
16. Luo, Z., Liu, H.: Cellular genetic algorithms and local search for 3-SAT problem on graphic hardware. In: IEEE Congress on Evolutionary Computation, CEC 2006, pp. 2988–2992 (2006)
17. Selman, B., Kautz, H.: Domain-independent extensions to gsat: Solving large structured satisfiability problems. In: PROC. IJCAI 1993, vol. 93, pp. 290–295 (1993)
18. Li, J., Wang, X., He, R., Chi, Z.: An efficient fine-grained parallel genetic algorithm based on GPU-Accelerated. In: IFIP International Conference on Network and Parallel Computing Workshops, NPC 2007, pp. 855–862 (2007)
19. Li, J., Zhang, L., Liu, L.: A parallel immune algorithm based on fine-grained model with gpu-acceleration. In: Proceedings of the 2009 Fourth International Conference on Innovative Computing, Information and Control, ICICIC 2009, pp. 683–686. IEEE Computer Society, Los Alamitos (2009)
20. Vidal, P., Alba, E.: Cellular genetic algorithm on graphic processing units. In: et al., J.G., ed.: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). SCI, vol. 284, pp. 223–232. Springer, Heidelberg (2010)
21. Pospichal, P., Jaros., J.: Gpu-based acceleration of the genetic algorithm. Technical report, GECCO competition (2009)
22. Tsutsui, S., Fujimoto, N.: Solving quadratic assignment problems by genetic algorithms with gpu computation: a case study. In: GECCO 2009: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference, pp. 2523–2530. ACM, New York (2009)
23. Luong, T.V., Melab, N., Talbi, E.G.: GPU-based Island Model for Evolutionary Algorithms. In: Genetic and Evolutionary Computation Conference (GECCO), Portland United States (2010)
24. Pospíchal, P., Jaros, J., Schwarz, J.: Parallel genetic algorithm on the CUDA architecture. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcazar, A.I., Goh, C.-K., Merelo, J.J., Neri, F., Preuß, M., Togelius, J., Yannakakis, G.N. (eds.) EvoApplications 2010. LNCS, vol. 6024, pp. 442–451. Springer, Heidelberg (2010)
25. Pospíchal, P., Schwarz, J., Jaroš, J.: Parallel genetic algorithm solving 0/1 knapsack problem running on the gpu. In: 16th International Conference on Soft Computing MENDEL 2010, Brno University of Technology, pp. 64–70 (2010)

About Complete Obligatory Hybrid Networks of Evolutionary Processors without Substitution

Artiom Alhazov¹, Gemma Bel-Enguix^{2,*}, Alexander Krassovitskiy^{2,*},
and Yurii Rogozhin¹

¹ Institute of Mathematics and Computer Science
Academy of Sciences of Moldova
Academiei 5, Chişinău MD-2028, Moldova
{artiom, rogozhin}@math.md

² Rovira i Virgili University Research Group on Mathematical Linguistics
Catalunya 35, Tarragona 43002 Spain
{gemma.bel@, alexander.krassovitskiy@estudiants.}urv.cat

Abstract. In this paper we consider complete obligatory hybrid networks of evolutionary processors (OHNEPs) with insertion and deletion operations (without substitution). Such networks are not computationally complete and we modify the notion of obligatory operation introduced in [1] in order to reach universality. We use very simple evolutionary processors with one modified operation of insertion or deletion per node (without substitution). Using techniques presented in the paper a universal complete OHNEP with 182 nodes can be constructed.

1 Introduction

Insertion, deletion, and substitution are fundamental operations in formal language theory. Networks of evolutionary processors (NEPs) [6] are systems built of elementary agents connected in a graph, each one specializing in one of these three operations. Such teams of agents with limited capabilities have been shown to be very powerful and flexible, taking advantage of their ability to communicate with each other inside the network. They operate on a set of words by rewriting them and redistributing them according to the system's protocol. The usual associated result is the set of words ever appearing in a specific node. NEPs are inspired by cell biology: each processor can represent a cell with DNA point mutations and a filtering mechanism. NEPs with a very small number of nodes are known [4] to be computationally complete.

Particularly interesting variants of these devices are the so-called *hybrid networks of evolutionary processors* (HNEPs) [10], where each (string rewriting) processor performs only one of the above operations in a certain position of the words in that node. Furthermore, the filters are random-context conditions, i.e., they check presence/absence of certain symbols in the words. It is known that HNEPs with 7 nodes are universal [3] and Accepting HNEPs with 6 nodes are universal [9], while HNEPs with 2 nodes are not computationally complete [3].

* Supported by the Spanish Ministerio de Ciencia y Tecnología, project MTM2007-63422.

In this paper we consider a variant of HNEPs, called Obligatory Hybrid Networks of Evolutionary Processors (OHNEP for short) introduced in [1]. The differences between HNEP and OHNEP are the following:

1. in deletion and substitution: a node discards a string if no operations in the node are applicable to the string (in HNEP case, this string remains in the node),
2. the underlying graph is a directed graph (in HNEP case, this graph is undirected).

These differences make OHNEPs universal [1] with 1 operation per node, no filters and only left insertion and right deletion. (The second difference is not important for this paper, since the graph is complete).

In [2] one considered complete OHNEPs, i.e., OHNEPs with complete underlying graph. One may now regard complete OHNEP as a set of very simple evolutionary processors “swimming in the environment”. In [2] it is proved that the complete OHNEPs with very simple evolutionary processors, i.e., evolutionary processors with only one operation (obligatory deletion, obligatory substitution and insertion) and filters containing not more than 3 symbols are computationally complete. An open problem posed in the paper is as follows: whether one can avoid substitution in order to reach universality using very simple evolutionary processors? First of all, we notice that in complete OHNEPs without substitution there is no control on the number of insertion or deletion of terminal symbols (i.e., those symbols which appear in output words). Therefore, we need to modify the definition of OHNEPs in order to increase their computational power. We show that it is possible to avoid substitution using modified operations of insertion and deletion in evolutionary processors similar to “matrix” rules in formal language theory. By using the techniques presented in the paper a universal complete OHNEP with 182 nodes without substitution can be constructed. Several questions are posed, in particular the question about the minimal total power of filters of evolutionary processor in computationally complete OHNEPs and the question about universal complete OHNEP without substitution with the minimal number of nodes.

2 Definitions

We recall some notions we shall use throughout the paper. An *alphabet* is a finite and nonempty set of symbols. The cardinality of a finite set A is written as $card(A)$. A sequence of symbols from an alphabet V is called a word over V . The set of all words over V is denoted by V^* , and the empty word is denoted by ε ; we use $V^+ = V^* \setminus \{\varepsilon\}$. The length of a word x is denoted by $|x|$, while we denote the number of occurrences of a symbol a in a word x by $|x|_a$. For each nonempty word x , $alph(x)$ is the minimal alphabet W such that $x \in W^*$.

Circular Post Machines (CPMs) were introduced in [8], where it was shown that all introduced variants of CPMs are computationally complete, and moreover, the same statement holds for CPMs with two symbols. In this article we use the deterministic CPMs.

A *Circular Post Machine* is a quintuple $(\Sigma, Q, \mathbf{q}_1, \mathbf{q}_f, R)$ with a finite alphabet Σ where $0 \in \Sigma$ is the blank, a finite set of states Q , the initial state $\mathbf{q}_1 \in Q$, the final state $\mathbf{q}_f \in Q$, and a finite set of instructions R with all instructions having one of the

forms $\mathbf{p}x \rightarrow \mathbf{q}$ (erasing the symbol read by deleting a cell), $\mathbf{p}x \rightarrow \mathbf{yq}$ (overwriting and moving to the right), $\mathbf{p}0 \rightarrow \mathbf{yq}0$ (overwriting and creating a blank cell), where $x, y \in \Sigma$ and $\mathbf{p}, \mathbf{q} \in Q$, $\mathbf{p} \neq \mathbf{q}_f$. We also refer to all instructions with \mathbf{q}_f in the right hand side as halt instructions. The storage of this machine is a circular tape, the read and write head moves only in one direction (to the right), and with the possibility to delete a cell or to create and insert a new cell with a blank.

We now summarize the necessary notions concerning *obligatory evolutionary operations*. For an alphabet V , we say that a rule $a \rightarrow b$, with $a, b \in V \cup \{\varepsilon\}$ is an *obligatory substitution operation* if $a \neq \varepsilon$ and $b \neq \varepsilon$; it is an *obligatory deletion operation* if $a \neq \varepsilon$ and $b = \varepsilon$; and it is an (*obligatory*) *insertion operation* if $a = \varepsilon$ and $b \neq \varepsilon$. The set of all obligatory substitution, deletion, and insertion operations over an alphabet V are denoted by Sub_V, Del_V , and Ins_V , respectively. Given such rules π, ρ, σ , and a word $w \in V^*$, we define the following *obligatory evolutionary actions* of π, ρ, σ on w if $\pi \equiv a \rightarrow b \in Sub_V, \rho \equiv a \rightarrow \varepsilon \in Del_V$, and $\sigma \equiv \varepsilon \rightarrow a \in Ins_V$:

$$\pi^*(w) = \{ubv \mid w = uav, u, v \in V^*\}, \rho^*(w) = \{uv \mid w = uav, u, v \in V^*\} \quad (1)$$

$$\rho^r(w) = \{u \mid w = ua\}, \rho^l(w) = \{v \mid w = av\} \quad (2)$$

$$\sigma^*(w) = \{uav \mid w = uv, u, v \in V^*\}, \quad (3)$$

$$\sigma^r(w) = \{wa\}, \sigma^l(w) = \{aw\}. \quad (4)$$

We enhance operations (2) and (4) to matrix operations of deletion and insertion as follows:

$$\rho_{1,2}^r(w) = \rho_1^r(\rho_2^r(w)), \rho_{1,2}^l(w) = \rho_1^l(\rho_2^l(w)), \text{ where } \rho_1^r, \rho_2^r, \rho_1^l, \rho_2^l \in Del_V \quad (5)$$

$$\sigma_{1,2}^r(w) = \sigma_1^r(\sigma_2^r(w)), \sigma_{1,2}^l(w) = \sigma_1^l(\sigma_2^l(w)), \text{ where } \sigma_1^r, \sigma_2^r, \sigma_1^l, \sigma_2^l \in Ins_V \quad (6)$$

This extension of operation insertion and deletion is correct taking into account that context-free insertion-deletion systems of size $(2, 0, 0; 2, 0, 0)$ (insertion of two symbols without context or deletion of two symbols without context) are not computationally complete [11]. Notice that in (1), (2) and (5) the result of obligatory evolution operation may be the empty set (this is the main difference between obligatory hybrid networks of evolutionary processors and hybrid networks of evolutionary processors).

Symbol $\alpha \in \{*, l, r\}$ denotes the way of applying an insertion or a deletion rule to a word, namely, at any position ($\alpha = *$), in the left end ($\alpha = l$), or in the right end ($\alpha = r$) of the word, respectively. Note that a substitution rule can be applied at any position. For a rule σ , an action $\alpha \in \{*, l, r\}$ and a language $L \subseteq V^*$, we define the α -action of σ on L by $\sigma^\alpha(L) = \bigcup_{w \in L} \sigma^\alpha(w)$. For a given finite set of rules M , we define the α -action of M on a word w and on a language L by $M^\alpha(w) = \bigcup_{\sigma \in M} \sigma^\alpha(w)$ and $M^\alpha(L) = \bigcup_{w \in L} M^\alpha(w)$, respectively.

Before defining an evolutionary processor, we define the filtering mechanism.

For disjoint subsets $P, F \subseteq V$ and a word $w \in V^*$, we define the predicate φ (see [7] and [3]) as $\varphi(w; P, F) \equiv (P = \emptyset \vee alph(w) \cap P \neq \emptyset) \wedge (F \cap alph(w) = \emptyset)$. This corresponds to predicate $\varphi^{(\beta)} = \varphi^{(2)}$ in the definitions from [7], i.e., when the filtering parameter is $\beta = 2$. Other values of β correspond to different filtering conditions, but we do not need them in this paper. The construction of this predicate is based on *random-context conditions* defined by the two sets P (*permitting contexts*) and F (*forbidding contexts*). For every language $L \subseteq V^*$ we define $\varphi(L, P, F) = \{w \in L \mid \varphi(w; P, F)\}$.

An *obligatory evolutionary processor (with matrix operations)* over alphabet V is a 5-tuple $EP = (M, PI, FI, PO, FO)$, where:

- Either $M \subseteq Sub_V$ or $M \subseteq Del_V$ or $M \subseteq Ins_V$. The set M represents the set of obligatory evolutionary operations (1) – (4) of the processor ((1) – (6) in the case with matrix operations). Note that every processor is dedicated to only one type of the above obligatory evolutionary operations.

- $PI, FI \subseteq V$ are the *input* permitting/forbidding contexts of the processor, while $PO, FO \subseteq V$ are the *output* permitting/forbidding contexts of the processor.

We denote the set of obligatory evolutionary processors over V by OEP_V .

Definition 1. An *obligatory hybrid network of evolutionary processors (an OHNEP for short)* is a 6-tuple $\Gamma = (V, G, N, C_0, \alpha, i_0)$, where the following conditions hold:

- V is a finite set (the alphabet).

- $G = (X_G, E_G)$ is a directed graph where the set of nodes is X_G and the set of edges is E_G . G is called the *underlying graph of the network*.

- $N : X_G \longrightarrow OEP_V$ is a mapping which associates with each node $x \in X_G$ the obligatory evolutionary processor $N(x) = (M_x, PI_x, FI_x, PO_x, FO_x)$.

- $C_0 : X_G \longrightarrow 2^{V^*}$ is a mapping which identifies the initial configuration of the network. It associates a finite set of words with each node of the graph G .

- $\alpha : X_G \longrightarrow \{*, l, r\}$; $\alpha(x)$ defines the action mode of the rules performed in node x on the words occurring in that node. We indicate α as a superscript of M_x .

- For every node, $x \in X_G$, we define the following filters: the input filter is given as $\rho_x(\cdot) = \varphi(\cdot; PI_x, FI_x)$, and the output filter is defined as $\tau_x(\cdot) = \varphi(\cdot, PO_x, FO_x)$. That is, $\rho_x(w)$ (resp. τ_x) indicates whether or not the word w can pass the input (resp. output) filter of x . More generally, $\rho_x(L)$ (resp. $\tau_x(L)$) is the set of words of L that pass the input (resp. output) filter of x .

- $i_0 \in X_G$ is the output node of the OHNEP.

An OHNEP is said to be a *complete* OHNEP, if its underlying graph is a complete graph. A configuration of an OHNEP Γ , as above, is a mapping $C : X_G \longrightarrow 2^{V^*}$ which associates a set of words with each node of the graph. A component $C(x)$ of a configuration C is the set of words that can be found in the node x in this configuration, hence a configuration can be considered as the sets of words which are present in the nodes of the network at a given moment. A configuration can change either by an *evolutionary step* or by a *communication step*. When it changes by an evolutionary step, then each component $C(x)$ of the configuration C is changed in accordance with the set of evolutionary rules M_x associated with the node x and the way of applying these rules $\alpha(x)$. Formally, configuration C' is obtained in *one evolutionary step* from the configuration C , written as $C \Longrightarrow C'$, iff $C'(x) = M_x^{\alpha(x)}(C(x))$ for all $x \in X_G$.

When the system evolves by a communication step, each evolutionary processor $N(x)$, where $x \in X_G$, sends all its words passing the output filter of x to the processors in all nodes connected with x ; from all the words sent by processors to $N(y)$, this processor $N(y)$ receives those words that pass the input filter of x . Formally said, configuration C' is obtained in *one communication step* from configuration C , denoted $C \vdash C'$, iff $C'(x) = (C(x) - \tau_x(C(x))) \cup \bigcup_{(y,x) \in E_G} (\tau_y(C(y)) \cap \rho_x(C(y)))$ for all $x \in X_G$.

For an OHNEP Γ , a computation in Γ is a sequence of configurations C_0, C_1, C_2, \dots , where C_0 is the initial configuration of Γ , $C_{2i} \implies C_{2i+1}$ and $C_{2i+1} \vdash C_{2i+2}$, for all $i \geq 0$. If we use OHNEPs as language generating devices, then the generated language is the set of all words which appear in the output node at some step of the computation. Formally, the language generated by Γ is $L(\Gamma) = \bigcup_{s \geq 0} C_s(i_0)$.

3 Obligatory Hybrid Networks of Evolutionary Processors with Matrix Operations

Lemma 1. *There are regular languages that cannot be generated by any complete OHNEP where evolutionary processors have only one operation per node (either obligatory deletion or obligatory insertion and with no substitution).*

Proof. Indeed, consider the regular language $(ab)^*$. Suppose Π is a complete OHNEP where evolutionary processors have only one operation per node and with no substitution rules. Consider a computation that produces a word $(ab)^n, n > 0$, of sufficient length. Clearly, on this computation system Π uses a node that inserts symbol a at some point. But then, this insertion can be repeated arbitrary amount of times. Indeed, such word may pass arbitrary amount of times the same input/output filters of the node that inserts a . Hence, Π generates also words with more then one consecutive letters a . This is a contradiction.

Theorem 1. *Any CPMO P can be simulated by a complete OHNEP P' where evolutionary processors are with only one insertion (matrix insertion) or deletion (matrix deletion) operation per node.*

Proof. Let us consider a CPMO P with symbols $a_j \in \Sigma, j \in \{0, 1 \dots, n\}$, $a_0 = 0$ is the blank symbol, and states, $q_i \in Q, i \in \{1, 2, \dots, f\}$, where q_1 is the initial state and the only terminal state is $q_f \in Q$. Denote by $Lab(R)$ set of labels corresponding to the rules R of P .

A configuration $v = q_i a_j W$ of CPMO P describes that P in state $q_i \in Q$ considers symbol $a_j \in \Sigma$ to the left of $W \in \Sigma^*$. This configuration corresponds to the string v in a node of OHNEP P' . The final configuration $q_f a_j W$ of P corresponds to the string $q_f a_j W$ in the output node $\langle out \rangle$ of P' .

Now we construct a complete OHNEP P' with matrix operations that simulates P .

$$\begin{aligned}
 P' &= (V, G, N, C_0, \alpha, i_0), V = \{S\} \cup Q \cup \Sigma \cup \Delta, \text{ where} \\
 \Delta &= \{\delta_p, \delta'_p \mid p \in Lab(R)\}, \\
 G &= (X_G, E_G), \text{ is a complete graph : } E_G = X_G \times X_G, \\
 X_G &= \{\langle init \rangle, \langle out \rangle = i_0\} \cup \{\langle p \rangle_t^a \mid p : q_i a_j \rightarrow q_l \in R, 1 \leq t \leq 5\} \\
 &\quad \cup \{\langle p \rangle_t^b \mid p : q_i a_j \rightarrow a_k q_l \in R, 1 \leq t \leq 5\} \\
 &\quad \cup \{\langle p \rangle_t^c \mid p : q_i 0 \rightarrow a_k q_l 0 \in R, 1 \leq t \leq 5\}.
 \end{aligned}$$

$$C_0(x) = \{Sq_1W\}, \text{ if } x = \langle init \rangle, \text{ where } q_1W \text{ is the input of } P,$$

$$C_0(x) = \emptyset, x \in X_G \setminus \{\langle init \rangle\}, N(\langle init \rangle) = (\{S \rightarrow \varepsilon\}^l, \emptyset, \emptyset, \emptyset, \emptyset),$$

$$N(\langle out \rangle) = (\emptyset, \{q_f\}, \Delta, \emptyset, \emptyset) \text{ and we define the other nodes below.}$$

CPM0 P starts a computation from a configuration q_1a_jW and OHNEP P' starts computation from a string Sq_1a_jW in the input node $\langle init \rangle$ accordingly (other nodes of P' contain empty sets of strings): $Sq_1a_jW \xrightarrow{N(\langle init \rangle)} q_1a_jW$ (it means that evolutionary processor $N(\langle init \rangle)$ is applied to string Sq_1a_jW and resulting string q_1a_jW is sent to all nodes of P' , including node $\langle init \rangle$). Clearly, string q_1a_jW will be rejected by $N(\langle init \rangle)$ as there is no operation applicable to this string. Further we continue to construct P' and describe how P' simulates three types of rules of CPM0 P .

Case (A). Rule $p : q_ia_j \rightarrow q_l \in R$ is simulated by the following nodes.

$$\begin{aligned} N(\langle p \rangle_1^a) &= (\{\varepsilon \rightarrow \delta_p\}^r, \{q_i\}, \Delta, \mathbf{0}, \mathbf{0}), \\ N(\langle p \rangle_2^a) &= (\{(q_i \rightarrow \varepsilon, a_j \rightarrow \varepsilon)\}^l, \{\delta_p\}, \{\delta'_p\}, \mathbf{0}, \mathbf{0}), \\ N(\langle p \rangle_3^a) &= (\{\varepsilon \rightarrow \delta'_p\}^r, \{\delta_p\}, \{\delta'_p\}, \mathbf{0}, \{q_i\}), \\ N(\langle p \rangle_4^a) &= (\{\varepsilon \rightarrow q_l\}^l, \{\delta'_p\}, \{q_l\}, \mathbf{0}, \mathbf{0}), \\ N(\langle p \rangle_5^a) &= (\{\delta'_p \rightarrow \varepsilon, \delta_p \rightarrow \varepsilon\}^r, \{q_l\}, \mathbf{0}, \mathbf{0}, \mathbf{0}). \end{aligned}$$

Let $q_ia_jW \xrightarrow{q_ia_j \rightarrow q_l} q_lW$ be a computation step in P , i.e., rule $q_ia_j \rightarrow q_l$ is applied to configuration q_ia_jW and q_lW is the next configuration.

In P' the string q_ia_jW (copies of this string) is distributed among all nodes $N(\langle p \rangle_1^a)$, where p is a label of instruction $q_ia_t \rightarrow q_l$, $a_t \in \Sigma$. It is clear that this word will be rejected by other evolutionary processors $N(\langle p \rangle_1^a)$ of P' , where p is a label of instruction of P with the left part $\langle q_s a_r \rangle$, $q_s \neq q_i, a_r \in \Sigma$. Thus for nodes $N(\langle p \rangle_t^a)$, $1 \leq t \leq 5$ evolution of string q_ia_jW leads to the correct result q_lW :

$$q_ia_jW \xrightarrow{N(\langle p \rangle_1^a)} q_ia_jW\delta_p \xrightarrow{N(\langle p \rangle_2^a)} W\delta_p \xrightarrow{N(\langle p \rangle_3^a)} W\delta_p\delta'_p \xrightarrow{N(\langle p \rangle_4^a)} q_lW\delta_p\delta'_p \xrightarrow{N(\langle p \rangle_5^a)} q_lW$$

Assume that p is a label of instruction of P with the left part $\langle q_ia_t \rangle$, $a_t \neq a_j$ or $\langle q_s a_r \rangle$, $q_s \neq q_i, a_r \in \Sigma$. It is easy to check that during an evolution of string q_ia_jW ($q_s a_r W$) the pair $\langle q_ia_t \rangle$ ($\langle q_r a_s \rangle$) cannot be deleted by node $N(\langle p \rangle_2^a)$ and this string will be lost.

Thus OHNEP P' correctly simulates rule $q_ia_j \rightarrow q_l$ of CPM0 P .

Case (B). Rule $p : q_ia_j \rightarrow a_kq_l \in R$ is simulated by the following nodes.

$$\begin{aligned} N(\langle p \rangle_1^b) &= (\{(\varepsilon \rightarrow a_k, \varepsilon \rightarrow \delta_p)\}^r, \{q_i\}, \Delta, \mathbf{0}, \mathbf{0}), \\ N(\langle p \rangle_t^b) &= N(\langle p \rangle_t^a), t = 2, 3, 4, 5 \end{aligned}$$

We use the same techniques as above and show that OHNEP P' correctly simulates rule $q_ia_j \rightarrow a_kq_l$ of CPM0 P .

$$\begin{aligned} q_ia_jW \xrightarrow{N(\langle p \rangle_1^b)} q_ia_jWa_k\delta_p \xrightarrow{N(\langle p \rangle_2^b)} Wa_k\delta_p \xrightarrow{N(\langle p \rangle_3^b)} \\ Wa_k\delta_p\delta'_p \xrightarrow{N(\langle p \rangle_4^b)} q_lWa_k\delta_p\delta'_p \xrightarrow{N(\langle p \rangle_5^b)} q_lWa_k \end{aligned}$$

Case (C). Rule $p : q_i0 \rightarrow a_kq_l0 \in R$ is simulated by the following nodes.

$$\begin{aligned}
 N(\langle p \rangle_t^a) &= N(\langle p \rangle_t^b), t = 1, 3, 5, \\
 N(\langle p \rangle_2^c) &= (\{(q_i \rightarrow \varepsilon, 0 \rightarrow \varepsilon)\}^l, \{\delta_p\}, \{\delta'_p\}, \emptyset, \emptyset), \\
 N(\langle p \rangle_4^c) &= (\{\varepsilon \rightarrow 0, \varepsilon \rightarrow q_l\}^l, \{\delta'_p\}, \{q_l\}, \emptyset, \emptyset).
 \end{aligned}$$

We use the same techniques as above and show that OHNEP P' correctly simulates rule $q_i0 \rightarrow a_kq_l0$ of CPM0 P .

$$\begin{aligned}
 q_i0W \xrightarrow{N(\langle p \rangle_1^a)} q_i0Wa_k\delta_p \xrightarrow{N(\langle p \rangle_2^c)} Wa_k\delta_p \xrightarrow{N(\langle p \rangle_3^c)} \\
 Wa_k\delta_p\delta'_p \xrightarrow{N(\langle p \rangle_4^c)} q_l0Wa_k\delta_p\delta'_p \xrightarrow{N(\langle p \rangle_5^c)} q_l0Wa_k
 \end{aligned}$$

Now we can conclude that if a computation in CPM0 P starts with configuration q_1W and ends with the final configuration q_fW' then OHNEP P' starting with string Sq_1W in the node $\langle init \rangle$ will obtain string q_fW' in the output node $\langle out \rangle$ and if P does not stop then in the node $\langle out \rangle$ of P' never appears some nonempty string. Thus P' correctly simulates P .

Corollary 1. *The family of complete OHNEPs with obligatory evolutionary processors with only one operation (matrix) insertion or (matrix) deletion per node (without substitution) is computationally complete.*

Corollary 2. *There exists a universal complete OHNEP with obligatory evolutionary processors with only one operation of (matrix) insertion or (matrix) deletion per node (without substitution) having 182 nodes.*

Proof. Let us consider the smallest known universal CPM0 P with 6 states and 6 symbols [5]. Now we construct OHNEP P' according to the algorithm in Theorem 1 above and we get 182 nodes (five nodes per CPM0 rule plus input and output nodes).

4 Conclusions

The general theory of Networks of Evolutionary Processors brings together formal languages and multi-agent systems, in a research field that has shown to share the consistency of the first and the flexibility of the latter.

Taking advantage of these features, we continue developing the the model of Hybrid Network of Evolutionary Processors. In previous works [12], we introduced and considered a new variant of HNEPs, Obligatory HNEPs. OHNEPs have two characteristics that are different from HNEPs: a) they have a directed underlying graph, and b) they discard a string if operations at the node are not applicable to such string. This paper is a step forward in the same line of research. We consider here a special case of OHNEPs with two properties: a) the underlying graph is complete and, b) the nodes do not have the rule of substitution. We name such mechanisms COHNEPs without substitution.

Such networks are not computationally complete and, to reach universality, we modified the notion of obligatory operation. We used very simple evolutionary processors with one modified operation of insertion or deletion per node.

We demonstrated that very simple OHNEPs without substitution are able to simulate any deterministic Circular Post Machine CPM0. Therefore, paper proves, once more,

the flexibility of OHNEPs, and their suitability to simulate different systems. Not only Circular Post Machines, but other complex mechanisms can be successfully approached by OHNEPs.

Using techniques presented in our paper a universal complete OHNEP with 182 nodes can be constructed. Several open questions are posed, in particular the question about the minimal total power of filters of evolutionary processor in computationally complete OHNEPs and the question about universal complete OHNEP without substitution and having the minimal number of nodes.

The efficiency of our model has been demonstrated by simulations. This shows that OHNEPs, being devices coming from formal language theory, have also a their role in multi-agent systems. OHNEPs can also be easily applied to artificial intelligence, because they are able to build complexity from extremely simple units that, additionally, are able to communicate and collaborate in a very simple way. Therefore, the development of OHNEPs and COHNEPs is very attractive from the mathematical point of view and very useful and promising for artificial intelligence and applied research.

References

1. Alhazov, A., Bel-Enguix, G., Rogozhin, Y.: Obligatory Hybrid Networks of Evolutionary Processors. In: Proc. of the First International Conference on Agents and Artificial Intelligents, ICAART 2009, Porto, Portugal, January 19-21, pp. 613 – 618 (2009)
2. Alhazov, A., Bel-Enguix, G., Krassovitskiy, A., Rogozhin, Y.: Complete Obligatory Hybrid Networks of Evolutionary Processors. In: 9th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2011), April 6 - 8, University of Salamanca, Spain (2011) (accepted)
3. Alhazov, A., Csuhaĵ-Varjú, E., Martín-Vide, C., Rogozhin, Y.: On the Size of Computationally Complete Hybrid Networks of Evolutionary Processors. *Theoretical Computer Science* 410, 3188–3197 (2009)
4. Alhazov, A., Dassow, J., Martín-Vide, C.: On Networks of Evolutionary Processors with Nodes of Two Types. *Fundamenta Informaticae* 91(1), 1–15 (2009)
5. Alhazov, A., Kudlek, M., Rogozhin, Y.: Nine Universal Circular Post Machines. *Computer Science Journal of Moldova* 10(3), 247–262 (2002)
6. Castellanos, J., Martín-Vide, C., Mitrana, V., Sempere, J.M.: Solving NP-complete problems with networks of evolutionary processors. In: Mira, J., Prieto, A.G. (eds.) *IWANN 2001*. LNCS, vol. 2084, pp. 521–528. Springer, Heidelberg (2001)
7. Csuhaĵ-Varjú, E., Martín-Vide, C., Mitrana, V.: Hybrid Networks of Evolutionary Processors are Computationally Complete. *Acta Informatica* 41(4-5), 257–272 (2005)
8. Kudlek, M., Rogozhin, Y.: Small Universal Circular Post Machines. *Computer Science Journal of Moldova* 9(1), 34–52 (2001)
9. Loos, R., Manea, F., Mitrana, V.: Small universal accepting hybrid networks of evolutionary processors. *Acta Informatica* 47, 133–146 (2010)
10. Martín-Vide, C., Mitrana, V., Pérez-Jiménez, M., Sancho-Caparrini, F.: Hybrid Networks of Evolutionary Processors. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) *GECCO 2003*. LNCS, vol. 2723, pp. 401–412. Springer, Heidelberg (2003)
11. Verlan, S.: On minimal context-free insertion-deletion systems. *Journal of Automata, Languages and Combinatorics* 12(1-2), 317–328 (2007)

Chemical Signaling as a Useful Metaphor for Resource Management

Evgeny Skvortsov^{1,*}, Nima Kaviani², and Veronica Dahl¹

¹ Simon Fraser University, Canada

² University of British Columbia, Canada
{evgenys,vdahl}@sfu.ca, nkaviani@cs.ubc.ca

Abstract. Declarative logic programs have proved useful for resource management problems since the early 80's. However the complexity of such programs is in a direct exponential relationship with the growth in the number of resources and users. We provide a simple, easy to implement, methodology for mathematically representing requests over resources inspired by the chemical signaling model of neural networks. Our resource management model uses substructural logic in its novel incarnation, HYPROLOG, to provide a connectionist neural network representation in which requests for resources are mapped to signals triggered and consumed by resource requesters and resource consumers respectively. Through this connectionist representation model, we achieve high level of expressivity while making the model directly executable. We exemplify the power of our model through representing a concrete temporal resource scheduling information system and then apply it to some real world mathematical problems.

Keywords: Substructural Logic, HYPROLOG, Constraint Handling Rules, Assumptions, Resource Allocation.

1 Introduction

Logic based Natural Language (NL) processing and intelligent information systems have a long history of interesting cross-fertilizations. Two notable examples are Prolog itself, which (as is not too widely known) evolved from Alain Colmerauer's Q-systems for processing natural language, and memoization, a technique first developed by David H. D. Warren for natural language processing applications, which became widely used for information systems. On the other hand there have been efforts to integrate logical reasoning with connectionist models similar to those represented in neural networks [10] [18] [19]. The main reason in doing so is to extend the binary model of logical inference to a more fuzzy inference model capable of handling infinite information [14].

Connectionist models of neural networks can as well serve as an inspiration to conceptually represent classic problems of intelligent information systems. The class of intelligent information systems which we address here, is that of resource

* Current affiliation of Evgeny Skvortsov is Google Inc.

allocation systems. These have been long addressed through Logic Programs, as from the first expert systems to dynamically built computer system specifications from the hardware and software resources given by the manufacturing companies [7]. In this paper, we focus on those resource allocation problems which involve temporal boundaries, e.g. scheduling problems. Generally, the diversity of the resources and constraints that should be taken into consideration while addressing such problems (e.g. the order of appearance for the activities [5], or reservation of some special resources for some activities [12]), hinders the possibility of providing efficient resource allocation algorithms.

Current techniques for resource management are mostly based on *finite domain constraint satisfaction* approaches as a sub-branch of Artificial Intelligence. However, mapping the concepts of a domain to a set of machine-understandable constraints is not a straightforward task, and consequently, even for simple scheduling problems, complicated constraints and relations must be defined. A simplification to such complications can happen through using the connectionist model of neural networks in which resources are introduced as neurons connected to one another through axons representing resource dependencies. A resource dependency can be considered as a request for consuming one resource and generating a second potentially nonidentical resource.

This can well represent resource allocation problems in which *available time-slot* resources are consumed to collectively produce scheduling resources. This can be seen as a neural network, where initially available resources are represented by activated neurons. To solve the problem axons trigger and thusly activate and deactivate resource-neurons. The problem is solved when all target neurons happen to be finally activated. Activation and deactivation is modeled by presence or absence of certain chemicals. Here we provide a user friendly while easy-to-implement methodology for resource allocation problems that follows the connectionist model of neural networks. We also show a directly executable rendition of it in terms of substructural logic (in particular, a version of linear logic [13] called *affine logic* [8]). Affine Logic is embodied in HYPROLOG [6] which provides an extension to Prolog with abduction, assumptions and the full power of *Constraint Handling Rules (CHR)* [11]. We exemplify our proposed approach within a temporal meeting-scheduling problem - parent-teacher interview scheduling - as a well-known resource allocation problem, and describe its advantages. Our choice of the connectionist approach to solve resource allocation problems, other than lending itself closely to a direct identification between resources and requests as neurons, and actions as axons, has the further advantage of direct executability as provided by our implementation.

Section 2 provides an overview of the approaches in the area of resource allocation, introduces HYPROLOG, and motivates its use to mimic resource management problems along the neural net metaphor. Section 3 describes our approach in using HYPROLOG to solve the scheduling problem, and Section 4 presents our conclusion.

2 Background and Motivation

Scheduling problems are combinatorial problems defined over allocating a set of resources to a group of activities. For instance, people in an organization may need to allocate their free time resources so as to materialize required meeting activities into some feasible overall schedule. This process is often complicated by some additional constraints. The two main approaches respectively focus on a) enforcing the required constraints, or b) omitting constraints with low priority. These approaches are mostly referred to as *constraint propagation* [1][2][4][12] and *local search* [9][16][17]. A *constraint propagation* approach reduces the set of possible values for the variables intervening in the constraints [3]. A *local search* starts from an initial partial solution and then improves the results by navigating from one partial solution to another, trying to find the optimal solution. A partial solution's degree of optimality is measured based on the number of violated constraints, i.e. *distance to feasibility*, where the solution with less distance to feasibility is chosen as the best one.

Although there has been a lot of research in both these areas, neither approach is easily portable to other domains. A higher level and more intuitive approach would be to allow the user to state what the resources are (e.g., Dumbledore has a free hour at noon, and so does Hermione), what the requests are (e.g., Hermione needs to meet with Dumbledore) and what the possible actions are (e.g. remove Dumbledore's free hour at noon, Hermione's free hour at noon, and Hermione's request to meet Dumbledore, while noting that Hermione and Dumbledore are to meet at noon), and let an intelligent solver choose applicable actions as needed to solve the problem. This is precisely what our proposed methodology provides: users can formulate a particular problem's resources and requests as linear affine implications, also called *assumptions* [8]; and describe possible actions in terms of consuming resources and requests, while noting through assumptions those requests that have been satisfied. Our solver manages the process of trying out applicable actions and uses abduction [15] to ensure that no unsatisfied requests remain at the end of the solving process. Our methodology also borrows from the connectionist model of neural networks in which assumptions resemble neurons with chemicals that lead to triggering the abductive rules in the network and consuming the neurons' chemicals, which in turn lead to propagating the neurons or morphing the neurons to other neurons capable of generating new chemicals that trigger other abductive rules in the network. This behavior transitively continues until the network becomes stable. The eventual state of the network describes a solutions sought for the given problem.

The entire system is formulated in HYPROLOG [6], an extension to Prolog with assumptions and abduction which provides one of the most efficient implementations of abductive logic programming, is implemented directly on top of SICStus Prolog [1] and CHR [11] and can use the full power of these languages. CHR is a declarative, rule-based language for writing constraint solvers which is now included as an extension of several versions of Prolog. Operationally

¹ <http://www.sics.se/sicstus/>

Table 1. The logical meaning of the HYPROLOG rules inherited from CHR

	Propagation Rule	Simplification Rule
HYPROLOG Rule	$H ::> G B$	$H <:> G B$
Logical meaning:	$\forall \bar{x}((\exists \bar{y}G) \rightarrow (H \rightarrow \exists \bar{z}B))$	$\forall \bar{x}((\exists \bar{y}G) \rightarrow (H \leftrightarrow \exists \bar{z}B))$

and implementation-wise, CHR extends Prolog with a constraint store, and the rules of a CHR program serve as rewriting rules over constraint stores. CHR is declarative in the sense that its rules can be understood as logical formulas with constraint predicates that must be declared as such, and can be called from a Prolog program. Integrity constraints in HYPROLOG are written as any sort of CHR rules with abducibles and assumptions in the head. Similar to CHR, HYPROLOG provides two main rules, *Propagation* and *Simplification*, each representing a logical meaning shown in Table 1. Assumptions in HYPROLOG are divided into two parts, *timeless* and *regular*, where timeless assumptions allow the consumption of concepts either prior or after their assumption while regular assumptions only allow the consumption of concepts after their assumption. Furthermore, both timeless and regular assumptions can be declared as intuitionistic or linear, taken from intuitionistic or linear logic, according to whether the availability of resources is infinite or limited respectively (see Table 2). Concepts in HYPROLOG must be identified as either regular or timeless through declarations such as, `assumption a/1` or `timeless_assumption b/1`.

Table 2. The notation for linear and intuitionistic assumptions in HYPROLOG

<code>+h(a)</code> , <code>*h(a)</code>	linear and intuitionistic definition of concept a in regular assumption
<code>=+h(b)</code> , <code>=*h(b)</code>	linear and intuitionistic definition of concept b in timeless assumption
<code>-h(a)</code> , <code>=-h(b)</code>	regular or timeless consumption of concepts a and b

Our HYPROLOG-based model of resource management takes advantage of assumptions and consumptions to represent the degradation of the resources.

3 Resource Management in Substructural Logic

3.1 Our Proposed Model

We are given a set R of resources, a set Q of requests, and a set of allowed *actions*. Our aim is to satisfy the requests and to acquire a new set R' of resources that might be needed for some future tasks (the set can be empty). Completion of an action consumes some resources, satisfies some requests and may produce additional resources. For example, we have a set of people, each person has some free time slots and some people need to meet each other. In this case resources are the free time slots, and requests are the given requests for people to meet. To solve the problem we build the following neural network. We use neuron to model each resource and request. Each action is modeled by an axon. Resources which the action consumes are marked as inputs of the axon, resources it produces and the requests are marked at outputs of the axon. Each neuron at each

moment is either active or passive. Activity of the neuron can be implemented by the neuron containing a certain chemical if it is active and not containing it if it is not active. Each axon connects several neurons, some of them are inputs and some are outputs for the axon. For the axon to trigger all inputs must be active. When the axon triggers it deactivates all inputs and activates all outputs. That is the axon consumes the activation chemical in its inputs and produces it at outputs. Thus we can use the neural network to model resource allocation problems. Various methods can be used for the network evolution. We find that encoding the network in HYPROLOG is a straightforward and elegant approach. Let us consider a meeting scheduling problem as an example. We have a collection of people, and a collection of slots. Some of the people request to have meetings with each other and report their time slots availability. The aim is to schedule all requested meetings within available slots. Our HYPROLOG formalization of this model is as follows. Neurons are represented by assumptions. So, for each request $r \in R$ and each slot $q \in Q$ we have assumptions $+r, +q$. Note that goals and resources might have some parameters. So, in the example above we have assumptions $+free(Person, Time)$ that specify that Person is free at time Time and $+request(PersonA, PersonB)$ that specifies that PersonA needs to meet PersonB. Using such parameters allows us to define a neural network parametrically and thus avoid redundancy in the description. All initially available resources r_i and all problem requests q_j are expressed as assumptions in the right-hand side of an initialization predicate problem/0, of the form:

$$problem :- +r_1, \dots, +r_n, +q_1, \dots, +q_m. \tag{1}$$

Possible actions are described as predicates act/0 which consume some resources, satisfy requests, produce some resources and place assumptions of the form $+o(\cdot)$ to record output. Thus we have predicates

$$act :- -r_1^1, \dots, -r_1^{n_1}, +r_{n_1+1}^1, \dots, +r_1^{n_1+l_1}, -q_1^1, \dots, q_1^{m_1}, +o. \tag{2}$$

⋮

$$act :- -r_k^1, \dots, -r_k^{n_k}, +r_{n_k+1}^1, \dots, +r_1^{n_l+k_l}, -q_k^1, \dots, q_k^{m_k}, +o. \tag{3}$$

A sample problem might be

$$problem :- +free(potter, 1), +free(potter, 2), \tag{4}$$

$$+free(dumbledore, 1), +free(dumbledore, 2), \tag{5}$$

$$+free(hermione, 1), +request(potter, dumbledore), \tag{6}$$

$$+request(hermione, dumbledore). \tag{7}$$

and the only possible action is assigning two persons to meet at a certain time. The action consumes available time slots and fulfills the request of the meeting:

$$act :- -free(A, Time), -free(B, Time), -request(A, B), +o(meet(A, B, Time)).$$

To guarantee that no requests remain unsatisfied at the end of the process, we use a special abducible *done* which will be called after the problem has been “solved”

to test that all problem requests have been granted, and we add for each request assumption $q \in Q$ a constraint rule: $q, done \Rightarrow fail$. For our example, we have $+request(A, B), done \Rightarrow fail$. Actions are called recursively through our solver:

$$solve. \tag{8}$$

$$solve:-act, solve. \tag{9}$$

The top querying predicate is defined as

$$go:-problem, solve, done,$$

where the *problem* predicate initiates the constraint store with resources and requests, *solve* runs the solution procedure, and *done* makes sure that all requests are satisfied and if needed consumes resources that must be available for eventual further use. A variety of resource management problems can be described using this methodology. The Prolog code for our sample problem and some other examples is shown at the project homepage:

<http://www.cs.sfu.ca/~evgenys/personal/SLRM/SLRM.html>

3.2 Completeness of the Model

A Prolog program is called *complete* if for any solvable query the system outputs a result in a finite number of steps. In general the proposed resource management model is obviously not complete. For example if the first possible action doesn't consume any resources and produces some, then the system will loop forever. There might be more complicated reasons to loop. See the Wolf, Goat, Cabbage program on the project website for an example of how one can avoid this problem, within our model. On the other hand, in some naturally restricted cases the system is complete; for example if any allowed action consumes more resources than it produces. We next give necessary and sufficient conditions for the completeness of a specific model where there are no CHR rules which change the set of available resources. For any action A , we define R_c^A as the set of resources consumed by the action and R_p^A as the set of resources produced by the action. Let R be the set of all resources.

Theorem 1. *The model containing actions \mathcal{A} is complete if and only if there is a function $\mathcal{V} : R \mapsto \mathbb{N}$ such that for any action $A \in \mathcal{A}$ we have inequality*

$$\sum_{r \in R_c^A} \mathcal{V}(r) > \sum_{r \in R_p^A} \mathcal{V}(r). \tag{10}$$

Proof. Let m be the number of resources considered in the model and n be number of actions. We will suppose that actions and resources are ordered. Let C be a matrix such that c_{ij} equals to number of units of resource i produced ($c_{ij} < 0$ if the resource is consumed) by application of j -th action. Problem of incompleteness of the model can be reduced to the following system of linear inequalities:

$$\begin{cases} C\mathbf{x} \geq 0 \\ \sum x_i > 0 \\ \mathbf{x} \geq 0 \end{cases}$$

I. e. the model will be incomplete if there is a non-empty sequence of application of actions such that after it no resources are consumed. By Farkas lemma it follows that the system has solution if and only if there is no solution to

$$\begin{cases} C^T \mathbf{v} \geq \mathbf{1} \\ \mathbf{v} \geq 0. \end{cases} \quad (11)$$

It's easy to see that system [\(11\)](#) is equivalent to finding the required values of resources. \square

Corollary 1. *The meeting scheduling model here proposed is complete.*

Proof. The only available resource in the model is a free slot while the only possible action consumes one slot, and doesn't produce any resources. Thus by [Theorem 1](#) we obtain the corollary. \square

4 Discussion

We have proposed and formalized a directly executable high level model of resource management in terms of substructural logic as embedded in HYPROLOG. We have demonstrated the completeness of our model in certain special but useful cases. We have exemplified our model with a running HYPROLOG implementation of a system for scheduling meetings. Readers interested in alternative applications were referred to the project website. The CLP family of languages, in which unification in the Herbrand universe is supplemented with constraint processing of linear equations or inequalities over the numbers, provides perhaps the closest approaches to our own. Previous work on CHR itself has explored temporal applications such as scheduling and working with time points, but continues the CLP tradition of thinking in terms of equations, or using techniques such as path consistency. Our approach is novel in that it does not express the problem constraints equationally, but in the resource-based terms of substructural logic, within a novel incarnation of it, HYPROLOG. This results in elegant and efficient formulations where resources can be represented as linear resources. We have not focused on efficiency but on direct executability within user-friendly formulations. With this work we hope to stimulate further work on the potential of substructural logics for executable models of resource management.

Acknowledgement

We gratefully acknowledge support from V. Dahl's NSERC Discovery Grant 31611024.

References


1. Abdennadher, S.: University Course Timetabling Using Constraint Handling Rules. *Applied Artificial Intelligence* 14(4), 311–325 (2000)
2. Azevedo, F., Barahona, P.M., Frangouli, H., Harmandas, V., Lajos, G., Burke, E., Ross, P., Meyer, M., Burke, E., Ross, P., et al.: Timetabling in Constraint Logic Programming. *Practice* 1, 22–45 (1994)
3. Baptiste, P., Le Pape, C.: A theoretical and experimental comparison of constraint propagation techniques for disjunctive scheduling. In: *Proc. IJCAI*, vol. 1, pp. 600–606 (1995)
4. Cambazard, H., Demazeau, F., Jussien, N., David, P.: Interactively solving school timetabling problems using extensions of constraint programming. *PATAT*, 107–124 (2004)
5. Carlier, J., Pinson, E.: A practical use of Jacksons preemptive schedule for solving the job-shop problem. *Annals of Operations Research* 26(269-287), 47 (1990)
6. Christiansen, H., Dahl, V.: HYPROLOG: A New Logic Programming Language with Assumptions and Abduction. In: Gabbrielli, M., Gupta, G. (eds.) *ICLP 2005*. LNCS, vol. 3668, pp. 159–173. Springer, Heidelberg (2005)
7. Dahl, V., Sambuc, R.: Un système de banque de données en logique du premier ordre, en vue de sa consultation en langue naturelle. *Rapport de DEA, université d'aix-marseille ii*. pp. 1–29 (1976)
8. Dahl, V., Tarau, P., Li, R.: Assumption Grammars for Processing Natural Language. In: *Proceedings of the Fourteenth International Conference on Logic Programming*, pp. 256–270 (1997)
9. Di Gaspero, L., Schaerf, A.: Multi-neighbourhood local search with application to course timetabling. In: Burke, E.K., De Causmaecker, P. (eds.) *PATAT 2002*. LNCS, vol. 2740, pp. 262–275. Springer, Heidelberg (2003)
10. Ding, L., Teh, H.H., Wang, P., Lui, H.C.: A prolog-like inference system based on neural logic – an attempt towards fuzzy neural logic programming. *Fuzzy Sets and Systems* 82(2), 235–251 (1996); *Connectionist and Hybrid Connectionist Systems for Approximate Reasoning*
11. Fruhwirth, T., et al.: Constraint handling rules. *Constraint Programming: Basics and Trends* 910, 90–107 (1995)
12. Gavanelli, M.: University Timetabling in ECL^iPS^e . *Association for Logic Programming (ALP) News Letter* 19(3) (2006)
13. Girard, J.-Y.: Linear logic. *Theoretical Computer Science* 50, 1–102 (1987)
14. Imanaka, T., Soga, M., Uehara, K., Toyoda, J.: An Integration of Prolog and Neural Networks to Deal with Sensibility in Logic Programs.. In: *ICSI*, pp. 738–746 (1990)
15. Kakas, A.C., Kowalski, R.A., Toni, F.: The role of abduction in logic programming. *Handbook of Logic in Artificial Intelligence and Logic Programming* 5, 235–324 (1998)
16. Kostuch, P.: The university course timetabling problem with a three-phase approach. LNCS, pp. 109–125.
17. Schaerf, A.: Combining local search and look-ahead for scheduling and constraint satisfaction problems. In: *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pp. 1254–1259 (1997)
18. Shen, Z., Ding, L., Mukaidono, M.: A theoretical framework of fuzzy prolog machine. *Fuzzy Computing: Theory, Hardware and Appl.*, 139–153 (1988)
19. Shen, Z., Ding, L., Mukaidono, M.: Fuzzy resolution principle. In: *Proceedings of the Eighteenth International Symposium on Multiple-Valued Logic*, pp. 210–215 (May 1988)

Distributed Simulation of P Systems by Means of Map-Reduce: First Steps with Hadoop and P-Lingua

L. Diez Dolinski, R. Núñez Hervás, M. Cruz Echeandía, and A. Ortega

Departamento de Ingeniería Informática, Escuela Politécnica Superior, Universidad Autónoma de Madrid

{leonardo.diez,marina.cruz,alfonso.ortega}@uam.es, rafaups@yahoo.es

Abstract.  In this paper we face the spatial difficulties inherent to the simulation of P systems on conventional (von Neumann) computers when they are applied to solve real problems. We have extended P Lingua (a well known textual programming language for P systems) to access distributed resources by means of Hadoop (a freely available implementation of the MapReduce paradigm). The current work shows the way that we propose to develop distributed, general purpose simulators for P systems.

Keywords: Natural computing, P systems, distributed simulation.

1 Motivation

This paper is phocused on *Membrane Computing* (also called *P systems*, from its main author Gheorge Păun) a new model of computation in the realm of *Natural Computing*. P systems abstract the processes taking place in the compartmental structures of the living cells to consider them as computations. This compartmental structure is formalized as an external membrane (called *skin*) that contains one or more sibling membranes each of which has the same structure (they contain one or more membranes with, again, the same structure). The biochemical contents of the cells in the living beings are represented by means of multisets of symbols (set of symbols in which more than one copy of each symbol is allowed) and a set of rules that consume some symbols of the multiset to produce others. Different families of P systems allow the creation and dissolution of membranes as well as different mechanisms for carrying symbols across the membranes. P systems are inherently parallel, both in the selection of symbols consumed and in the application of the rules. [\[1,2\]](#) show different results on their expressive power and the performance of their algorithms.

There are almost none kind of P systems actually supported by specific hardware implementations. [\[3\]](#). Under these circumstances the only available platform for natural computers implies the simulation of the bio-inspired models in *conventional computers*.

¹ This work was partially supported by the R&D program of the Community of Madrid (S2009/TIC-1650, project “e-Madrid”).

When P systems are simulated in conventional computers, the *exponential* amount of temporal resources needed to solve NP-problems could become an *exponential* amount of spacial resources. For example, let us consider a NP-problem whose solution implies to check 2^n elements to determine which one is the solution. A sequential algorithm would take an exponential number of steps (2^n) to check all the possibilities and would need to store only the current one in each moment. P systems, theoretically, have no spacial limit for the number of membranes they contain. Let us consider a P system with just one membrane in the skin. Let us suppose that this membrane is able to create another one in each step. It is easy to see that we will have 2^n membranes after n steps. P systems process all their membranes in parallel. If we use one of them for each possible solution and are able to check if any of them is a solution in linear time; we could finally find the solution of the problem after a linear number of steps but we will need an exponential amount of membranes. This example shows the reason why the size of problems we are able to actually solve by general simulators of P systems on conventional computers is small. If we are interested in developing a general purpose programming platform based on P systems we have to take into account the spatial complexity of the problem. Clusters of computers and the access to distributed resources (grid and cloud computing) could offer an alternative to manage as much resources as necessary to solve bigger and more interesting problems. This approach is focused rather on the spatial limitations than on the temporal performance.

This paper explores for the first time the *map-reduce* programming paradigm for simulating P systems with distributed resources. We have used *hadoop* (a well known implemetation of map-reduce freely available at <http://hadoop.apache.org/>) Nevertheless it is not the first approach to the distributed simulation of P-systems. In [4] a Java distributed simulator of transition P-system is described. It uses the RMI protocol. We have previously explored this same approach to the parallel simulators of other natural computing models (NEPs [5]) It is difficult to properly handle parallel and distributed resources of clusters of computers by means of Java because it is mandatory to add a non standard distributed Java Virtual Machine. The RMI protocol, besides, seems a less efficient and natural approach than, for example, the message passing interface (MPI [6]) extensions for C++. Other works like [7] actually shows parallel implementations of a specific P system. It uses GPUs as a parallel computing platform. Although they get an excellent performance we are more interested in general purpose approaches that provides the researchers with easy to use parallel and distributed simulators for more general families of models.

There are several research groups interested in programming tools for natural computers. P-Lingua (developed by the Research Group on Natural Computing of the University of Sevilla) is a programming language for membrane computing which aims to be a standard to define P systems. One of its main features is to remain as close as possible to the formal notation used in the literature to define P systems. The programmer will not have to do any additional effort to describe his P systems with P-Lingua once he has formalized them. P-Lingua

is also the name of a software package that includes several built-in simulators for each supported model as well as the needed compilers to simulate P-Lingua programs. More details can be found at <http://www.p-lingua.org> and [8].

We have used some of the P-Lingua modules freely available to develop our hadoop simulator. We have also used a notation similar to that of P-Lingua for some configuration files.

In the following paragraphs we first introduce the map-reduce programming paradigm, and hadoop, one of its Java implementations; then we will explain how we propose to use distributed resources to simulate P systems. Finally we will show some conclusions and highlight our further research lines.

2 Introduction to Map Reduce

MapReduce [9] is a programming model designed to address distributed processing of large amounts of information with a simple and easy-to-use approach. When it was conceived at Google labs, they were spending too much time developing ad-hoc solutions to conceptually straightforward problems because they need to handle details of parallelization, fault-tolerance, data distribution and load balancing. An analysis of these tasks shown that most of them could be solved handling the input data as a list of records and using *map* and *reduce* primitives to treat it and to generate another list of records as output.

A MapReduce implementation performs the following steps. First, the *Map* function takes each record in the input data, a key/value pair, and produces a set of intermediate key/value pairs. Next, after all intermediate pairs are generated, a second intermediate set is created with each different intermediate key and the list of all its associated intermediate values. Finally, the *Reduce* function takes each key and its values from this second intermediate data and produces a list with output values, which is usually smaller than the received one.

In this process, the programmer is just responsible for writing the *Map* and *Reduce* functions. The only restriction in this task is to maintain the consistency of the types, i.e. *Map* output key and value types must match *Reduce* input types. Additionally, the developer must be careful with the access to the external resources, in order to avoid multithread related issues, as concurrent access to data. *Counters* are a facility defined with the model that can help in this task, allowing to create a global counter without using mutually-exclusive operations. This technique consists in keeping a local counter on each node and in calculating the total sum at the end of each processing phase.

A third optional function that can be implemented is the *Combiner* function. It allows to increase the system performance through a local reduction of the data before starting the communication between the system nodes. The *Reduce* function can also be sometimes used for this purpose.

A simple way to understand this programming model is trying to implement SQL grouping and aggregate functions as MapReduce programs. For example, the SUM function could be done straightforward. First, the *Map* function emits for each input record a pair with the grouping fields values and the value to sum. Then, the *Reduce* function sums all the received values for each key.

3 Introduction to Hadoop

One of the most popular implementation of the MapReduce model is included in the Apache Hadoop project. This is an open-source framework for creating distributed applications in Java. It is based on two main components: a distributed file system (Hadoop Distributed File System, or HDFS) and a MapReduce engine.

Typically, Hadoop MapReduce engine works with HDFS data, where each HDFS node runs a MapReduce node. It consists of a single master *Job Tracker* and a *Task Tracker* per node. The first receives MapReduce jobs from clients and distributes smartly to the *Task Trackers*. These are the responsible of running the received jobs. This configuration allows the framework to exploit the data locality in order to reduce bandwidth consumption.

Another important task performed by the *Job Tracker* is the fail recovery. It monitors the jobs running in the system and re-executes tasks when a *Task Tracker* fails.

So simple as the MapReduce model itself, developing an application with Hadoop MapReduce consists in implementing two classes, one for each function, and a main function that configures the environment. Besides, as the theoretical model specifies, a Combiner class can also be defined.

In addition to the basic functionality, Hadoop MapReduce includes debugging and profiling development tools and advanced features as Counters, distributed read-only files cache, as well as data compression facilities.

One major advantage from choosing Hadoop MapReduce implementation is the wide availability of commercial services that runs it as a cloud computing tool. A list of these services can be found at <http://wiki.apache.org/hadoop/Distributions%20and%20Commercial%20Support>. This fact extends the range of applications of the systems which are developed with this framework.

4 Distributed Simulation of P Systems

The goal of this work is to distribute the P system simulation between several simultaneous processes by using the MapReduce programming model. In this procedure, each MapReduce cycle represents a step in the evolution of the P system, where there are as many calls to the Map function as membranes in the system before the simulation step is executed. After the simulation step, there are as many Reduce function executions as membranes in the system. Each execution of the function Map simulates a single membrane in its context and emits its results to itself and to the membranes to which it is directly connected (its parents and sons). The Reduce function simply joins all the data received for one membrane and writes its new configuration in the distributed filesystem.

In our approach, it is mandatory to use unique identifiers for the membranes in the system. This is unnecessary when the information of the complete structure is kept together, but becomes crucial when it is split. A Global Unique Identifier, also called GUID, is assigned to each membrane, with a numeric value greater

than all labels in the system. The library used to simulate the behaviour of the membranes is not designed to support this need, but it has a rarely used feature called `EnvironmentID` which has been used to solve it temporarily.

A New Input Format Based on P-Lingua

We had to define a syntax for describing the P systems under consideration. We call it ‘Distributed P-Lingua’ and use the file extension ‘`dpli`’ for the input files to our system. As we have explained before, an important issue to take into account is the distributed storage of the P system structure. The goal is to define a special version of the P-lingua syntax with the fewest possible changes which are required to support the distributed execution. Each membrane is stored independently with all the associated data that would be needed in order to execute its related rules. With all this information, each line in a distributed P-Lingua file has the following format:

```
GUID:label:structure:symbols
```

The *structure* field includes the membrane itself surrounded, when it is not the system skin, by its parent membrane labelled with its GUID. Besides, it contains its children membranes, including its GUIDs as `EnvironmentID` with a comma after its label. Finally, the symbols could be included in the structure or in a separated field. For example, the following P-Lingua fragment:

```
@mu = [[[]'3 []'4]'2]'1;
@ms(3) = a,f;
```

It could be written in Distributed P-Lingua as:

```
101:1: [[[]'2,102]'1
102:2: [[[]'3,103 []'4,104]'2]'101
103:3: [[a,f]'3]'102
104:4: [[[]'4]'102
```

Regarding rules, on the other hand, it is pointless to copy them on every membrane line. Therefore, they are kept in the original P-Lingua file. This file is made available to all nodes in the Hadoop cluster using the distributed cache feature.

Supported Rules

The development described in this document is focus on simple rules, i.e. evolution, communication, creation and dissolution rules. Consequently, it can be used to simulate P systems with transitions and creation of new membranes. In addition, it seems easy to follow a similar approach to add support to P systems with division.

Pseudocode

Finally, the pseudocode from the important program functions is included. As already mentioned, the key work is done in the Map function, while the Reduce function just joins the information and writes it to disk. The Main function creates a Map/Reduce task for each simulation step.

```
function Map(key, value):
    // create a small P sys, with the membrane context structure
    psystem = create_psystem(value)
    // simulates a step on the created system
    sim = create_simulator(psystem)
    stopped = sim.step()
    // locate membranes in the resultant structure
    parent = get_parent(sim)
    current = get_current(sim)
    children = get_children(sim)
    // emits the resultant structure to the related membranes
    if parent != null: emit(parent.id, parent.symbols)
    if current == null: emit(parent.id, current dissolved)
    if current != null: emit(current.id, current)
    for children as child: emit(child.id, child.symbols)
    // increment counters to determine if the simulation is over
    mem_counter++
    if stopped: stop_counter++
function Reduce(key, values):
    parse values
    join structure + symbols received + dissolved children
    write to distributed filesystem
function Main():
    // run map and reduce while simulation is not over
    do
        mem_counter = stop_counter = 0
        configure enviroment
        execute map/reduce
    while mem_counter > stop_counter
```

5 The Simulation of an Example P System

To verify that the designed solution is feasible, we have simulated a P system taken from the literature with the implemented system. This simple example generates a random number and calculates its square, and is defined as follows in the P-lingua language:

```
@model<transition>
def main() {
    call n_cuadrados(); }
def n_cuadrados() {
    @mu = [[[]'3 []'4]'2]'1;
```



```

@ms(3) = a,f;
[a --> a,bp]'3;
[a --> bp,@d]'3;
[f --> f*2]'3;
[bp --> b]'2;
[b []'4 --> b [c]'4]'2;
(1) [f*2 --> f ]'2;
(2) [f --> a,d]'2; }

```

Since the initial structure of the P system consists of four membranes, the first step will be simulated using four calls to the Map function, which will generate the results of the first step of evolution of each membrane separately using the P-lingua simulation framework, and an equivalent number of calls to the Reduce function, to reassemble the new structure of each membrane. It can be seen that, at some point, the application of the unique dissolution rule will affect the membrane labeled with 3. In the simulation of this step, the Reduce function call for that membrane will receive the dissolve order and will not write to disk the resultant structure. From that time, the system will not apply more dissolution rules, so the rest of steps will imply three calls of each function per step.

We have successfully simulated this P system. In addition, we have not used any specific technique that, according to the documentation of Hadoop, could prevent our implementation from being run in a real cluster.

6 Conclusions

P systems are one of the better known new bio-inspired models of computation. Their intrinsic parallelism and their unbound spatial capacity are two of their main characteristics that allow to define new algorithms to solve NP problems with better temporal performance than the corresponding versions for conventional (von Neumann) architectures. Therefore, they could be considered an alternative architecture to develop new computers. Nevertheless the specific hardware to effectively support this model is currently far from being considered a real architecture.

When simulating P systems on conventional computers, the potentially exponential space complexity of some of the algorithms makes it impossible to tackle instances with a size big enough to be considered interesting. This is one of the reasons why there are not computers actually based on P systems and the main motivation of this work. In this paper we take the first steps to access to distributed resources (usually by means of internet) while simulating P systems.

We have chosen the MapReduce programming paradigm, and hadoop, one of its Java implementation freely available.

For describing the P systems being simulated, we have defined ‘Distributed P Lingua’, an extension to a subset of P Lingua (a textual programming language that can be considered a ‘de facto’ standard). We have found a rather straightforward way to develop these kind of distributed P systems simulators hiding low level details and keeping them as general as the theoretical models.

7 Further Research

With respect to our current implementation, in the future we plan to improve some of its aspects (for example, the unique identification of membranes and the reuse of the rules in different membranes). We have, also, to increase its robustness and improve its performance. We also plan to actually test this implementation on a cloud computing environment.

With respect to the families of P systems supported, we have to incorporate other features to completely cover the theoretical definition of them. These features include advanced rules, as division, symmetric communication or probabilistic rules, but also environment identifiers, tissue-like membranes or parametric sentences.

Last but not least, it must be considered the worst-case scenario. Since the structure of each membrane is held together in a single cluster node, it is possible that it could be overwhelmed by a membrane containing a large amount of daughter membranes. In this situation, it would be necessary to propose a solution that allows the simulation of a membrane breaking its structure into smaller parts, while maintaining the proper functioning of the application.

References

1. Păun, G.: Computing with membranes. *Journal of Computer and System Sciences*, 61(1) (2000), 108-143, and *Turku Center for Computer Science-TUCS Report Nr. 208* (1998)
2. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, Riscos-Núñez, A.: Towards a programming language in cellular computing. *Electronic Notes in Theoretical Computer Science* 123, 93-110 (2005)
3. Fernández, L., Martínez, V.J., Mingo, L.F.: A hardware circuit for selecting active rules in transition P systems. In: *Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2005* (2005)
4. Syropoulos, A., Mamatras, E.G., Allilomes, P.C., Sotiriades, K.T.: A distributed simulation of transition P systems. In: *Martín-Vide, C., Mauri, G., Păun, G., Rozenberg, G., Salomaa, A. (eds.) WMC 2003. LNCS, vol. 2933, pp. 133-145. Springer, Heidelberg* (2004)
5. del Rosal, E., del, N.R., Castañeda, C., Ortega, A.: Simulating NEPs in a cluster with jNEP. *International Journal of Computers, Communications & Control*, 480-485 (2008); *Supplementary Issue: Proceedings of ICCCC vol. III, 480-485* (2008)
6. *Message Passing Interface Forum, MPI: A Message-Passing Interface Standard, University of Tennessee, UT-CS-94-230* (1994)
7. Cecilia, J.M., Guerrero, G.D., García, J.M., Martínez-del-Amor, M.A., Pérez-Hurtado, I., Pérez-Jimenez, M.J.: Simulation of P Systems with Active Membranes on CUDA. In: *Proceedings of HIBI 2009 International Workshop on High Performance Computational Systems Biology* (2009)
8. García-Quismondo, M., Gutiérrez-Escudero, R., Pérez-Hurtado, I., Pérez-Jiménez, M.J., Riscos-Núñez, A.: An overview of P-lingua 2.0. In: Păun, G., Pérez-Jiménez, M.J., Riscos-Núñez, A., Rozenberg, G., Salomaa, A. (eds.) *WMC 2009. LNCS, vol. 5957, pp. 264-288. Springer, Heidelberg* (2010)
9. Dean, J., Ghemawat, S.: *MapReduce: Simplified Data Processing on Large Clusters. Google, Inc.* (2004)

Hierarchy Results on Stateless Multicounter $5' \rightarrow 3'$ Watson-Crick Automata

Benedek Nagy¹, László Hegedüs¹, and Ömer Egecioğlu²

¹ Department of Computer Science, Faculty of Informatics,
University of Debrecen, Debrecen, 4032 Hungary
nbenedek@inf.unideb.hu, hegedus.laszlo@inbox.com

² Department of Computer Science, University of California,
Santa Barbara, CA 93106, USA
omer@cs.ucsb.edu

Abstract. We consider stateless counter machines which mix the features of one-head counter machines and a special type of two-head Watson-Crick automata (WK-automata). Our biologically motivated machines have heads that read the input starting from the two extremes. The reading process is finished when there are no more symbols between the heads. Depending on whether the heads are required to advance at each move, we distinguish between realtime and non-realtime machines. If every counter makes at most k alternations between nondecreasing and decreasing modes in every computation, then the machine is k -reversal. In this paper we concentrate on the properties of nondeterministic stateless WK-automata with counters. Results about deterministic versions can be found in (Egecioğlu et al.: Stateless multicounter $5' \rightarrow 3'$ Watson-Crick Automata, BIC-TA 2010).

1 Introduction

A well investigated branch of DNA computing is the theory of Watson-Crick automata ([3,13]). These are finite state machines equipped with two read-only heads. They operate on DNA molecules, i.e., on double stranded sequences of bases. The strands of a DNA molecule have directions as a result of the underlying chemical bonds, determining the $5'$ and $3'$ ends of a strand. The two strands have opposite biochemical directions. Between the two strands, there is a one-to-one correspondence of the bases given by the so-called Watson-Crick complementarity relation. In this way a strand of the molecule uniquely defines the other, and therefore the DNA molecules can be described by ordinary strings (as, for instance, in [6,7]). In biology several enzymes are known to act on a DNA strand in direction from $5'$ to $3'$. Consequently, in the case of $5' \rightarrow 3'$ Watson-Crick automata, at the beginning of a computation, the reading heads start from opposite ends of the input and they can move opposite direction in computational point of view (but the same direction biochemically). These automata have been used to characterize linear context-free languages in [10]. In this paper we consider only $5' \rightarrow 3'$ Watson-Crick automata, and consequently use the terminology *WK-automata* and omit the qualification $5' \rightarrow 3'$.

Stateless machines (i.e. machines with only one state) have recently been connected to certain aspects of membrane computing [11,12]. As the name shows, a stateless machine (without additional storage) cannot store any information. Thus other methods are used, e.g., the automaton is equipped with some number of counters which are zero at the beginning and again zero at the end of a computation. In [5,14] the computing power of stateless multihead automata with respect to decision problems and head hierarchies were investigated.

If the machine is not allowed to make transitions without moving a head, then the model is called *realtime*. Otherwise the machine is *non-realtime* and can make transitions without moving the heads. We say that a counter is in increasing/decreasing mode if its value is increased/decreased at the last transition that changed its value. A counter machine is *k-reversal* if each counter makes at most k “full” alternations between increasing mode and decreasing mode and vice-versa on any computation (accepting or not), and is *reversal bounded* if it is k -reversal for some k .

Deterministic stateless (one-way) m -counter machines were investigated in [1], where hierarchies with respect to the number of counters and number of reversals were studied. Similar hierarchy results and characterizations are reported in [4] for the non-realtime versions. Hierarchies of the accepted language families by WK-automata are presented in [9] including stateless versions without counters. In this paper we concentrate on nondeterministic stateless realtime WK-automata with counters. Some results about the deterministic case can be found in [2]. We give examples and establish hierarchies of WK-automata with respect to counters and reversals.

2 Stateless Multicounter WK-Automata

We start with the description of the components of a stateless multicounter WK-automata. The input is of the form $\phi w \$$ with $w \in \Sigma^*$ and ϕ and $\$$ are endmarkers that are not in Σ . The machine has two read-only heads H_1, H_2 . Head H_1 moves from left to right and H_2 moves from right to left. Originally, H_1 is on ϕ and H_2 is on $\$$. The machine is equipped with m counters, that are initially all zero. A move of the machine depends on the symbols under the heads and the signs of the counters (the automata can distinguish two cases: zero or positive). It consists of moving the heads and at the same time incrementing, decrementing, or leaving unchanged each counter. The input w is accepted by M if the counters are again zero when the heads *meet*.

The essence of when the heads H_1 and H_2 meet is captured best by making use of a function φ which indicates whether the heads are close or far apart in processing the input. This locality requirement can be justified in part by biological properties that give rise to WK-automata. For the model it suffices to know if there are zero, one, two, or more than two letters between the heads. Define

$$\varphi(M) = \begin{cases} p & \text{if there are } p \text{ letters between the two heads of } M \text{ and } p \leq 2, \\ \infty & \text{if there are more than two letters between the heads of } M. \end{cases}$$

We use the notation $\varphi(M)$ although φ is actually a function of the current positions of the heads of M , i.e., function of the actual configuration.

For a deterministic stateless multicounter WK-automaton M , a transition (a move)

$$((x, y; s_1, s_2, \dots, s_m), p) \rightarrow (d_1, d_2; e_1, e_2, \dots, e_m) \tag{1}$$

has the following parameters: $x, y \in \Sigma \cup \{\$, \#\}$ are the symbols under the heads H_1 and H_2 , respectively; s_i is the sign of counter C_i : $s_i = 0$ if the i -th counter is zero, $s_i = 1$ if it is positive. $s_1 s_2 \dots s_m$ is referred to as a *sign vector*; $d_1, d_2 \in \{0, 1\}$ indicate the direction of move of the heads with $d_1 + d_2 \leq p$. A value 0 signifies that the head stays where it is. $d_1 = 1$ means that H_1 moves one cell to the right, and $d_2 = 1$ means that H_2 moves one cell to the left; $e_i = +, -, \text{ or } 0$, corresponding to the operations of increment, decrement, or leave unchanged the contents of the i -th counter. Here $e_i = -$ applicable only if $s_i = 1$. A move (\square) is possible if and only if $\varphi(M) = p$. It should be noted that $\varphi(M)$ is not part of the system, nor it is a counter, just a technical parameter. M is *nondeterministic* if multiple choices are allowed for the right hand side of (\square) .

The machine is *realtime* if not both d_1 and d_2 are zero for any move of the machine. Otherwise it is *non-realtime*.

The machine is *k-reversal* if for a specified k , no counter makes more than k alternations between increasing mode and decreasing mode (i.e. k pairs of increase followed by decrease stages) in any computation, accepting or not. The machine is *reversal bounded* if it is k -reversal for some k .

We denote the set of all nondeterministic realtime k -reversal m counter non-realtime WK-automata by WKC_m^k , and the realtime versions by RWKC_m^k . The reversal bounded versions are given by

$$\text{WKC}_m^* = \bigcup_{k=0}^{\infty} \text{WKC}_m^k, \quad \text{RWKC}_m^* = \bigcup_{k=0}^{\infty} \text{RWKC}_m^k;$$

while WKC_m^∞ and RWKC_m^∞ denote the unbounded reversal versions. We use a “d” prefix to refer to the deterministic versions of these machines. So dRWKC_m^k denotes all deterministic realtime k -reversal m -counter machines. This notation is also used for the corresponding language classes.

The formal definition of a nondeterministic stateless multicounter WK-automaton is as follows.

Definition 1. *A nondeterministic stateless multicounter WK-automaton is a quadruple $M = (\Sigma, \delta, \$, \#)$ where Σ is a nonempty alphabet, δ is a mapping from $(\Sigma \cup \{\$, \#\})^2 \times \{1, 0\}^m \times \{0, 1, 2, \infty\}$ to $2^{\{0,1\}^2 \times \{0,+,-\}^m}$ and $\$, \# \notin \Sigma$ are two special symbols called endmarkers.*

3 Closure Properties of Nondeterministic Stateless Multicounter WK-Automata Languages

Some closure properties of languages accepted by nondeterministic stateless multicounter WK-automata follow.

Proposition 1. *The language family accepted by nondeterministic stateless realtime multicounter WK-automata is closed under the union operation.*

Proof. If M_1 is a k_1 -reversal m_1 -counter machine and M_2 is a k_2 -reversal m_2 -counter machine, then we can construct a $\max\{k_1, k_2\}$ -reversal, $\max\{m_1 + m_2\} + 2$ -counter machine which can simulate both M_1 and M_2 . The two additional counters keep track of which machine is being simulated. \square

Proposition 2. *Language families $\text{WKC}_m^x, \text{RWKC}_m^x, \text{dWKC}_m^x, \text{dRWKC}_m^x$ ($m \in \mathbb{N}, x \in \mathbb{N} \cup \{*, \infty\}$) are closed under reversal (mirror image) of words.*

Proof. Suppose a machine accepts w with k -counters and m -reversals. Then w^R can also be accepted with the same parameters, just the behaviour of the heads have to be exchanged. \square

Proposition 3. *For $|\Sigma| \geq 2$, language families accepted by nondeterministic stateless realtime multicounter WK-automata are not closed under concatenation operation.*

Proof. The idea of the proof is as follows. Let us consider the language of even palindromes $L = \{ww^R \mid w \in \{a, b\}^*\}$. It can be accepted by a realtime deterministic WK-automaton without counters [9]. Its concatenation with itself is $L \cdot L = \{ww^Ruu^R \mid w, u \in \{a, b\}^*\}$. Since an RWKC_m^∞ machine has only memory by counters, there is no way to store a word of arbitrary length as w or u^R could be. Without storing any (or both) of these words our stateless machine is unable to check their reverse. \square

Corollary 1. *Languages accepted by nondeterministic realtime stateless multicounter WK-automata are not closed under Kleene-closure.*

4 Hierarchies Regarding Nondeterministic Stateless Multicounter WK-Automata

Theorem 1. *RWKC_m^k is more powerful than dRWKC_m^k .*

Proof. Clearly all deterministic machines with m counters and k reversals are a special case of nondeterministic machines with the same number of counters and reversals.

For $n \geq 0$ we define the language

$$L_n = \{a^i b^{j_1} a b^{j_2} \dots a b^{j_n} \mid i, j_k \geq 0, k = 1, 2, \dots, n, \\ \text{and } i = j_\ell \text{ where } \ell = 1, \text{ or } \ell = 2, \text{ or } \dots \ell = n\}.$$

Then $L' = \bigcup_{n=1}^\infty L_n$ cannot be accepted by a deterministic stateless realtime WK-automata with any number of reversals and any number of counters. The machine does not know for which $\ell \in \{1, 2, \dots, n\}$ the equality stands, so it

must be tested for all possible values. It is clear that for any numbers $k, m \in \mathbb{N}$: $w \in L_{2(k+1)(m+1)+1}$ cannot be accepted by a machine in WKC_m^k .

Furthermore, in the nondeterministic case, only one counter is enough to accept the language. If $i = j_k$ for a k in $\{1, 2, \dots, n\}$, then the right head can read the end of the input till the end of the subword b^{j_k} without changing the counter, then at the beginning of that block (i.e., at the previous symbol a or $\$$ in case of $k = n$, i.e., at the last block) the counter is changed to 1. If the counter is positive, both heads move while reading symbols. The left head reads the prefix a^i while the right head reads b^{j_k} . Thus they are reading the same number of symbols. At the end of the blocks the counter set to be zero again only if both blocks ended at the same step. Then the remainder of the input can be read by the first head to finish the input. \square

The following theorem is a known result regarding to deterministic WK-automata.

Proposition 4. [2] *For fixed $i \geq 3$, languages in the form $L = \{a^{in} \mid n \geq 0\}$, cannot be accepted by any stateless deterministic realtime reversal bounded WK-automaton.*

These languages are clearly accepted by nondeterministic stateless WK-automata with k -reversals and m -counters, where k, m depend on the value of i . If the machine is 1-reversal, $m = \lceil \frac{i}{2} \rceil - 1$ counters are needed to accept L (for $i = 1, 2, \dots$).

A linear grammar $G = (N, T, P, S)$ is in normal form if every rule has one of the following forms $A \rightarrow aBb, A \rightarrow aB, A \rightarrow Ba, A \rightarrow a$ with $A, B \in N$ and $a, b \in T$. This can be achieved by basic transformations of the rules.

Theorem 2. *Let $G = (N, T, P, S)$ be a linear grammar in normal form. Then $L(G)$ can be accepted by a nondeterministic stateless realtime WK-automaton with unbounded number of reversals and $m = |N| - 1$ counters so that $\sum_{i=1}^m c_i \leq 1$ at any time of the computation (where c_i is the value of the i -th counter).*

Proof. We give a sketch of the construction of the machine which accepts $L(G)$. Each counter represents a nonterminal which is not S . The symbol S is represented by the sign vector $(0, 0, \dots, 0)$. Initially both heads make a step from the boundary markers without changing the values of the counters. For a rule $X \rightarrow aYb \in P$ the move

$$\begin{aligned} &((a, b; 0, 0, \dots, 0, s_X = 1, 0, \dots, 0), \infty) \rightarrow \\ &(1, 1; 0, 0, \dots, 0, e_X = -, 0, e_Y = +, 0, \dots, 0) \end{aligned}$$

is added. The order of C_X and C_Y in the sequence may be different depending on how the counters were assigned to the nonterminals. If $X = Y$, then the counter is not changed. For a rule $X \rightarrow aY \in P$ similar moves are added (for $\varphi(M) > 1$). This time H_2 does not move: b can be any terminal symbol, so the moves should be constructed for all $b \in T$, and only the left head makes a step. The case of $X \rightarrow Ya \in P$ is very similar and $X \rightarrow a \in P$ can be handled with $\varphi(M) = 1$ by reading an a with the left input head and decreasing the counter C_X . \square

Remark 1. It is known that counter machines with a finite control and two counters accept all recursively enumerable languages [8]. In this way we can add two additional counters to the automaton in Theorem 2 and get machine equivalent to the Turing-machine. Thus, reversal boundedness is a key factor at hierarchy of stateless multicounter WK-automata by limiting their power.

Let us define the following languages over $\Sigma = \{a, b, c\}$ for $j \geq 1$:

$$L_j = a^{n_1} c a^{n_2} c \dots a^{n_j} c b^{n_1} c^{n_j} b^{n_2} c^{n_{j-1}} \dots b^{n_{j-1}} c^{n_2} b^{n_j} c^{n_1},$$

where $n_i \geq 0$ for all $i = 1, \dots, j$.

Theorem 3. *For any $j \geq 1$, L_j is in $dRWKC_j^1$, but not in $RWKC_{j-1}^k$ for any $k \geq 1$.*

Proof. First we remark that without the delimiters c between the subwords a^{n_i} , only one counter is enough to accept L_j .

It is easy to see that L_j can be accepted by a 1-reversal machine with j counters. The machine first counts the number of a 's in the prefix a^{n_1} subword with counter 1, then, if a c is read, the second counter is increased by 1 to indicate, that the a^{n_2} subword is going to be read. After reading a^{n_2} and increasing counter 2 in each step, another c comes. This time, while reading it, the second counter is decreased by 1 (it contained $n_2 + 1$, because of the first c) and the third counter is increased to indicate, that now a^{n_3} is to be read and counted. In this way, the machine counts all these a^{n_i} subwords and after the j th one, while reading a c with the left head, it also reads the right end marker, $\$$ with the right head, while decreasing the j th counter by 1 (this is also because of the previously read c). At this point, the value of counter i ($i = 1, \dots, j$) is n_i . After these steps, the subword $b^{n_1} c^{n_j} b^{n_2} c^{n_{j-1}} \dots b^{n_{j-1}} c^{n_2} b^{n_j} c^{n_1}$ is processed while decrementing the corresponding counters.

We use induction on j to show, that L_j cannot be accepted by $j - 1$ counters with any number of reversals. For $j = 1$, it is obvious, that the language $L_1 = a^n c b^n c^n$ cannot be accepted without counters. Note, that the expression of L_{k+1} can be constructed from the expression of L_k in the following way:

1. let w and x be the sequences $b^{n_1} c^{n_k} \dots b^{n_k} c^{n_1}$ and $a^{n_1} c a^{n_2} c \dots a^{n_k} c$ respectively
2. $L_{k+1} = a^\ell c x b^\ell h(w) c^\ell$, where h is a morphism such that $h(b) = c$ and $h(c) = b$.

Suppose, that for some $k \geq 1$, L_k cannot be accepted with $k - 1$ counters with any number of reversals. For L_{k+1} , we have an additional counter. In this counter, we should store the number ℓ to test whether the first block of b 's have the same number of symbols. So, the additional counter is needed to store ℓ . Then, after reading a^ℓ , or c^ℓ (or both) we have $k - 1$ counters and the remaining part of the input to be processed (without the b^ℓ subword) is in L_k , except that the corresponding b 's and the c 's in the suffix are exchanged. But that does not help the machine to accept L_k . And we know that L_k cannot be accepted with $k - 1$ counters. Thus there is no way to get to b^ℓ without loss of information, so L_{k+1} cannot be accepted with k counters. □

Corollary 2. $dRWKC_m^1$ and $RWKC_{m-1}^k$ are incomparable (for $k \geq 1, m \geq 2$).

Proof. L_m can be accepted by a 1-reversal m -counter machine, but cannot be accepted by any k -reversal $m - 1$ counter machines for any $k \geq 1$. On the otherside, the language L' (after Theorem 1) is in $RWKC_{m-1}^k$ for $k \geq 1, m \geq 2$, but it is not in $dRWKC_m^1$. \square

5 Conclusions

Figure 1 contains some hierarchy results about WK-automata based on our new results and [2]. Arrows stand for proper inclusion. RE stands for the class of recursively enumerable languages. Some open problems are represented by dotted arrows. Another further direction of research can be to investigate such multicounter WK-automata where the heads do not sense when they meet, therefore both heads read the full input in an accepting computation. Other extension can also be considered: multicounter WK-automata which may read strings and not just letters in a single transition, for example.

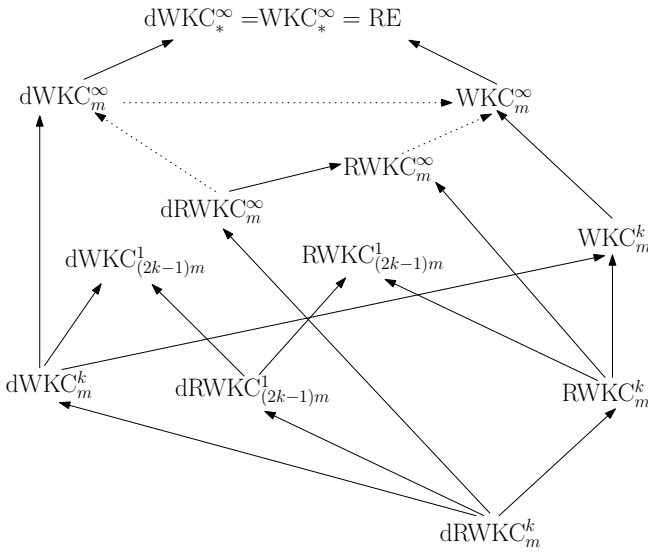


Fig. 1. Hierarchy of stateless multicounter $5' \rightarrow 3'$ WK automaton languages

Acknowledgements

The research is supported by the TÁMOP 4.2.1/B-09/1/KONV-2010-0007 project, which is implemented through the New Hungary Development Plan, co-financed by the European Social Fund and the European Regional Development Fund.

References

1. Eğecioğlu, Ö., Ibarra, O.H.: On stateless multicounter machines. In: Ambos-Spies, K., Löwe, B., Merkle, W. (eds.) CiE 2009. LNCS, vol. 5635, pp. 178–187. Springer, Heidelberg (2009)
2. Eğecioğlu, Ö., Hegedüs, L., Nagy, B.: Stateless multicounter $5' \rightarrow 3'$ Watson-Crick Automata. In: Fifth International Conference on Bio-Inspired Computing: Theories and Applications, pp. 1599–1606. IEEE Press, Los Alamitos (2010)
3. Freund, R.: Păun, Gh., Rozenberg, G., Salomaa, A.: Watson-Crick Finite Automata. In: Third Annual DIMACS Symposium on DNA Based Computers, Philadelphia, pp. 535–546 (1994)
4. Ibarra, O.H., Eğecioğlu, Ö.: Hierarchies and characterizations of stateless multicounter machines. In: Ngo, H.Q. (ed.) COCOON 2009. LNCS, vol. 5609, pp. 408–417. Springer, Heidelberg (2009)
5. Ibarra, O.H., Karhumäki, J., Okhotin, A.: On stateless multihead automata: Hierarchies and the emptiness problem. In: Laber, E.S., Bornstein, C., Nogueira, L.T., Faria, L. (eds.) LATIN 2008. LNCS, vol. 4957, pp. 94–105. Springer, Heidelberg (2008)
6. Leupold, P., Nagy, B.: $5' \rightarrow 3'$ Watson-Crick automata with several runs. In: Workshop on Non-Classical Models of Automata and Applications (NCMA), Wrocław, Poland, pp. 167–180. book@ocg.at, Österreichischen Computer Gesellschaft (2009)
7. Leupold, P., Nagy, B.: $5' \rightarrow 3'$ Watson-Crick automata with several runs. *Fundamenta Informaticae* 104, pp. 71–91, extended version of [6] NCMA2009 (2010)
8. Minsky, M.L.: Computation: finite and infinite machines. Prentice-Hall, Englewood Cliffs (1967)
9. Nagy, B.: On a hierarchy of $5' \rightarrow 3'$ sensing WK finite automata languages. In: CiE 2009: Mathematical Theory and Computational Practice, Abstract Booklet, pp. 266–275. University of Heidelberg (2009)
10. Nagy, B.: On $5' \rightarrow 3'$ Sensing Watson-Crick Finite Automata. In: Garzon, M.H., Yan, H. (eds.) DNA 2007. LNCS, vol. 4848, pp. 256–262. Springer, Heidelberg (2008)
11. Păun, G.: Computing with Membranes. *Journal of Computer and System Sciences* 61/1, 108–143 (2000)
12. Păun, G.: Computing with Membranes: An Introduction. Springer, Berlin (2002)
13. Păun, G., Rozenberg, G., Salomaa, A.: DNA Computing: New Computing Paradigms. Springer, Heidelberg (1998)
14. Yang, L., Dang, Z., Ibarra, O.H.: On stateless automata and P systems. In: Pre-Proceedings of Workshop on Automata for Cellular and Molecular Computing, pp. 144–157 (2007)

Towards a Bio-computational Model of Natural Language Learning

Leonor Becerra-Bonache

Laboratoire Hubert Curien, UMR CNRS 5516
Université de Saint-Etienne, Jean Monnet
Rue du Professeur Benoit Luras, 42000 Saint-Etienne, France
leonor.becerra@univ-st-etienne.fr

Abstract. This paper tries to bring together the theory of Grammatical Inference and the studies of natural language acquisition. We discuss how the studies of natural language acquisition can improve results in the field of Grammatical Inference, and how a computational model inspired by such studies can help to answer several key questions in natural language learning.

1 Introduction

Children, independently of their culture and the language they are exposed to, are able to acquire their native language easily, efficiently and without any specific training. However, the ease with which children acquire their language skills contrasts with the difficulty to explain this process.

The desire to better understand how children acquire their native language has motivated research in formal models of language learning [14,13]. Such models can allow us to address several key questions in natural language learning, such as: what types of input are available to the learner? what is the impact of semantic information on learning the syntax of a language? Moreover, such models can have important implications in the field of human language technologies. Therefore, it is of great interest to study formal models of language learning. However, in order to better simulate the human processing and acquisition of language, it is important that such models are inspired by studies of natural language acquisition.

Based on all these ideas, we present and discuss some of our main contributions within the field of *Grammatical Inference* (GI), and its implications for studies of natural language acquisition. We claim that ideas coming from studies of natural language acquisition can help GI to improve models and techniques used in this field, and even to obtain new challenging results. Moreover, GI models can also help in the understanding of natural language acquisition/processing. The simulation of such human capacity would provide natural interfaces to improve the communication between machines and humans.

2 Grammatical Inference versus Natural Language Acquisition

GI is a subfield of Machine Learning that deals with the learning of formal languages. The basic framework can be considered as a game played between two players: a teacher and a learner. The teacher provides information to the learner, and the learner must identify the underlying language from that information [10]. For example, imagine that the target language is ba^+ (i.e., a language that contains strings starting with one b , followed by at least one a), and the teacher provides to the learner strings that belong to the language (i.e., positive data), such as $ba, baa, baaa \dots$. The learner, from this information, should infer that the target language is ba^+ .

Several formal models of language learning have been proposed in the field of GI. The main ones are: *Identification in the limit* [11], *Query learning model* [1], and *PAC learning model* [15]. The problem of these models is that they do not take into account some relevant aspects of natural language learning. Therefore, they have aspects that make them useful to study the problem of natural language acquisition to a certain extent, but other aspects of the models make them unsuitable for this task.

For example, let us focus on the model proposed by E.M. Gold, *Identification in the limit*. E.M. Gold was really motivated by the question of how children acquire their native language. His goal was to construct a formal model of language learning in order to investigate from a theoretical point of view how to learn a language artificially. *Identification in the limit* views learning as an infinite process. In this model, the learner passively receives more and more examples, and he has to produce a hypothesis of the target language (i.e., the language to be learned). If he receives new examples that are not consistent with the hypothesis, then he has to change it. The hypothesis has to converge to a correct hypothesis.

Note that in this model, the learner can never be sure of having correctly guessed the language, since new examples could appear at any time. The justification of Gold for studying this model was that

a person does not know when he is speaking a language correctly; there is always the possibility that he will find that his grammar contains an error. But we can guarantee that a child will eventually learn a natural language, even if it will not know when it is correct [11]

Two traditional settings are considered: learning from *text* and learning from *informant*. In learning from text, only positive data (i.e., strings that belong to the language) are given to the learner. In learning from informant, positive and negative data are given to the learner (i.e., strings that belong and do not belong to the language).

Although we can find some similarities between learning in Gold's model and natural language acquisition (for example, in both cases there is a process of improvement), this model has several aspects that are controversial from a linguistic point of view.

In Gold's model, there is not limit on how long it can take the learner to guess the correct language.

That is, a language has been correctly identified when the learner no longer changes its guess through the presentation of all of the (possible infinite) strings in the language. If the learner is lucky, the first guess could be correct. Alternatively, it might take several billions of years to come up with the correct guess. [12]

Hence, considerations of efficiency form a somewhat separate line of analysis from Gold's work. However, from natural language acquisition point of view efficiency is also important. Although learning natural language is an infinite process, we are able to learn the language in an efficient way.

Moreover, in *identification in the limit* the learner hypothesizes complete grammars instantaneously. From a linguistic point of view this assumption is unrealistic, since this is not the case in children's language acquisition.

Although natural language learning is mainly based on positive examples, positive data only is less than what a child actually gets in the learning process and, informant is much more than what a learner can expect. Moreover, the distinction that Gold does between positive and negative data is clear within the framework of formal languages, but not within the framework of natural languages, since we can find data that contains positive and negative information at the same time (as we will see in the next section), and hence it is difficult to classify them as positive or negative.

Learning from text or informant is also known as passive learning, as the learner passively received strings of the language. We know that natural language learning is more than that. Children also interact with their environment. They produce sentences that could be grammatically correct or not, and they can also ask questions to the adults, etc. Therefore, there is an interaction between child and adult, that is not gained by Gold's model.

Therefore, the model of *identification in the limit* postulates greatly idealized conditions, as compared to the conditions under which children learn language. For a longer discussion of the main models proposed in GI, see [6].

3 Towards a Bio-computational Model of Language Learning

The problem of language learning in GI presents similarities with the process of language acquisition. For example, in the context of natural language acquisition, instead of a teacher and a learner, we have an adult and a child. Moreover, a child learns a language from the data that he/she receives (a child with an English environment will learn to speak English, and the same child with a Chinese environment will learn to speak Chinese). Therefore, GI provides a good theoretical framework for investigating the process of language learning.

However, as we have seen in the previous section, the formal models proposed so far within the field of GI are not satisfactory. Therefore, it would be of great

interest to develop a new computational model inspired by studies of natural language acquisition. We claim that studies of natural language acquisition can help to improve models and techniques in GI, and even to improve formal results obtained in this field. Moreover, a model with such bio-inspiration can also help to studies of natural language acquisition to better understand how children acquire and process their native language, and to answer several key questions about natural language learning (for example, what kind of data is available to the learner? what is the role of semantics in language learning?) The application of such a model could also be of great interest, for example, to improve the communication between humans and machines.

Next we present some works that we have done in these directions, and discuss the results obtained.

3.1 Learning from Corrections

Computational models of language learning should ideally provide the learner the same kind of information that are available to children. But, what kind of data is available to children during the learning process?

It is generally accepted that positive data are available to children. However, the availability of another kind of data, which is often called negative data, remains a matter of substantial controversy. As we can see, the definition of negative data is oversimplified, and could have different interpretations. Therefore, beliefs about whether or not children receive negative data are going to depend on how we define that concept. Moreover, as we have pointed in the previous section, this distinction between positive and negative seems not to be very adequate within the framework of natural languages, since we can find sentences that are grammatically correct but contain negative information.

For example, let us consider the following conversation extracted from the CHILDES database:

CHILD: milk, milk
 ADULT: you want milk?
 CHILD: uh-huh
 ADULT: Ok. Just a second and I'll get you some

As we can see, the child produces a sentence that is grammatically incorrect. Immediately after, the adult tries to reformulate the sentence by checking on what the child had intended to say. Moreover, after that, the child acknowledges the reformulation.

This kind of conversations occurs very often during the first stages of children's language acquisition. Adults try to correct child's erroneous utterances based on the meaning that the child intend to express (then, the context in which this sentence is produced is very important). Adults correct them just to be sure that they have understood the child's intentions. Therefore, child's utterance and adult's correction have the same meaning, but different form.

What kind of data are these corrections? Positive or negative? As we can see, corrections contain positive and negative information at the same time. A correction is a sentence grammatically correct, then, contains positive information. But, as Chouinard and Clark pointed out

Since, like adults, children attend to contrast in form, any change in form that does not mark a distinct, different, meaning will signal to children that they may have produced something that is not acceptable in the target language. [9]

Therefore, negative information is also available.

Based on linguistic arguments that support the presence of corrections in children's language acquisition [9], we have applied the idea of corrections to GI studies, and showed that GI models can benefit from corrections, for instance, the query learning model proposed by D. Angluin. In this model, the learner is allowed to make queries to the teacher, and the teacher has to answer correctly his queries. Membership and equivalence queries have established themselves as the standard combination to be used. In the case of a membership query, the learner asks to the teacher if a string is in the target language, and the teacher answers "yes" or "not". In the case of an equivalence query, the learner asks if his conjecture is correct, and the teacher answers "yes" or gives to the learner a counterexample (if the conjecture is not correct).

The queries available to the learner in Angluin's model are quite unnatural for real learning environments. Based on the corrections that children receive during the first stages of language acquisition, we have proposed a new type of query called *correction query* (CQ). In a CQ, the learner asks if a string is in the language, and if the string does not belong to the language, the teacher returns a correction.

In [8], we present the first attempt to learn from corrections. Taking into account the simplicity of DFA and their adequacy for some applications of natural language processing, we considered that a good starting point was to apply corrections to learn deterministic finite automata (DFA). We design an algorithm called *Learning from Corrections Algorithm* (LCA), which is able to infer a DFA using CQs and equivalence queries. In this context, a correction of a string consists of the shortest extension of the queried string. We showed that it is possible to learn DFA from corrections, and that the number of queries needed by the learner is reduced considerably.

In [7] we proposed a new CQ based on edit distance. When the learner submits to the teacher a string that does not belong to the target language, the teacher returns a string of the language close to the query with respect to the edit distance (the edit distance is the minimum number of deletion, insertion or substitution operations needed to transform one string into another). We consider non-standard classes of languages defined via edit distance : the balls of strings. We showed that this class is not learnable in Angluin's model, but is with a linear number of CQs. We also conducted several experiments with a teacher simulating a human Expert, and showed that our algorithm is resistant to approximate answers.

Therefore, all these results show that new challenging results can be obtained in the field of GI by using ideas coming from studies of natural language acquisition.

3.2 Learning with Semantics

The kinds of corrections considered in the papers cited above, are mainly syntactic corrections based on proximity between strings. However, as we have pointed out in the previous section, the corrections given to children during the first stages of language acquisition *preserve* the meaning that the child intend to express. Therefore, one of our goals has been to develop a formal model that gives an account of this kind of correction, and in which we can address the following questions: What are the effects of corrections on learning syntax? Can corrections facilitate the language learning process? Can semantic information simplify the problem of learning formal languages?

Inspired by the two-word stage of children's language acquisition, we have developed a formal model that takes into account semantics for language learning. This model accommodates two different tasks: comprehension and production. Such a model tries to reflect several aspects of natural language acquisition. For example, our model does not rely on a complex syntactic mechanism; in that way, we try to represent the fact that, although the child and adult grammars are different, the semantic situation allows communication

The first attempt to incorporate semantics in the field of GI can be found in [3,2,5]. We have presented an algorithm that learns a meaning function and prove that it finitely converges to a correct result under a specific set of assumptions about the transducer and examples used. The learning problem has been formulated as follows: (i) The teacher provides to the learner several example pairs consisting of a situation and an utterance denoting something in the situation (like in the 2-words stage, where in addition to hearing utterances, children have access to the context in which these utterances are generated); (ii) The goal of the learner is to learn the meaning function, allowing the learner to comprehend novel utterances. We have shown that a simple algorithm can learn to comprehend an adults utterance (in the sense of producing the same sequence of predicates), even without mastering the adults grammar. We have presented and analyze the results of empirical tests of our algorithm with natural language samples in an example domain of geometric shapes and their properties and relations.

We have also explored the possibility of applying existing automata-theoretic approaches to machine translation (concretely, subsequential transducers and the OSTIA algorithm) to model language production [4]. For ten natural languages and a limited domain of geometric shapes and their properties and relations we have defined sequential transducers to produce pairs consisting of an utterance in that language and its meaning. Using this data we have empirically explored the properties of OSTIA and DD-OSTIA algorithms for the tasks of learning comprehension and production in this domain, to assess whether they may provide a basis for a model of meaning-preserving corrections. Our results suggest

that OSTIA and DD-OSTIA may be an effective method to learn to translate sequences of predicates into natural language utterances in our domain. However, some of our objectives seem incompatible with the properties of OSTIA (e.g., the intermediate results of the learning process do not seem to have the properties we expect of a learner who is progressing towards mastery of production).

Finally, we have considered a statistical approach to model comprehension and production, which has produced a more powerful version of our initial model and has allowed us to model corrections [5]. In this new approach, the teacher is able to understand a flawed utterance produced by the learner and respond with a correct utterance for that meaning. Moreover, the learner can recognize that the teachers utterance has the same meaning but different form. This approach allows us to compare a learner that only receives positive data, a learner that is corrected sometimes (with different probabilities) and a learner that is corrected whenever this is possible, and therefore, to study the effect of meaning-preserving corrections on language learning. The results obtained so far show that: the access to the semantics facilitates language learning, and the presence of corrections by the teacher has an effect on language learning by the learner (even if the learner does not treat corrections specially). Hence, this new approach points out the relevance of semantics and corrections in language learning, and sheds interesting questions about them.

4 Conclusions

The understanding and simulation of natural language acquisition constitutes one of the biggest challenges of the 21st century. Therefore, it is of great interest to develop formal models of language learning that can help us to better understand how children acquire their native language. Such a models could also have important implications in the field of human language technologies. If we are able to create machines that can recognize, understand and generate natural languages, we will make possible for the user to interact with the computer, without any special skill or training, just as they would do to a person.

In this paper we have discussed how the theory of GI and the studies of natural language acquisition can be brought together. Based on the fact that language learning in GI exhibits similarities with natural language acquisition, the need for an adequate/sophisticated bio-computational model for language learning has been discussed and confirmed.

To employ the results of the field of GI and natural language acquisition in each other, both theories should be developed. On one hand, we have argued why GI models need to involve the concept of corrections, and we have demonstrated how the models get advantage in this way. On the other hand, we have shown how a computational model that incorporates semantics as well, can allowed us to investigate aspects of the roles of semantics and corrections in the process of learning to understand and speak a natural language.

Therefore, we have tried to bring together the theory of GI and studies of natural language acquisition, and shown the benefits that can be obtained by

doing so. Ideas coming from linguistics can be useful in GI in order to obtain new perspectives of the problem and possible new solutions and, of course, the theory of GI can also help to understand the process of language acquisition. Hence, it is of great interest to study natural language acquisition from an interdisciplinary point of view. Ideas and techniques coming from different areas can help us to develop computer systems with human-like capabilities and go deeper in the understanding of children's language acquisition.

References

1. Angluin, D.: Learning Regular Sets from Queries and Counterexamples. *Information and Computation* 75, 87–106 (1987)
2. Angluin, D., Becerra-Bonache, L.: A model of semantics and corrections in language learning. YALEU/DCS/TR-1425 (April, 2010)
3. Angluin, D., Becerra Bonache, L.: Experiments with an algorithm to learn meaning before syntax. In: *ForLing2008*, pp. 1–12 (2008)
4. Angluin, D., Becerra-Bonache, L.: Learning meaning before syntax. In: *ICGI*, pp. 1–14. Springer, Berlin (2008)
5. Angluin, D., Becerra-Bonache, L.: Experiments using OSTIA for a language production task. In: *CLAGI 2009*, pp. 16–23 (2009)
6. Becerra-Bonache, L.: On the Learnability of Mildly Context-Sensitive Languages using Positive Data and Correction Queries. PhD thesis, Rovira i Virgili University (2006)
7. Becerra-Bonache, L., de la Higuera, C., Janodet, J.C., Tantini, F.: Learning balls of strings from edit corrections. *JMLR* 9, 1841–1870 (2008)
8. Becerra-Bonache, L., Dediu, A.-H., Tîrnăucă, C.: Learning DFA from correction and equivalence queries. In: Sakakibara, Y., Kobayashi, S., Sato, K., Nishino, T., Tomita, E. (eds.) *ICGI 2006. LNCS (LNAI)*, vol. 4201, pp. 281–292. Springer, Heidelberg (2006)
9. Chouinard, M.M., Clark, E.V.: Adult reformulations of child errors as negative evidence. *Journal of Child Language* 30, 637–669 (2003)
10. Clark, A.: Grammatical inference and first language acquisition. In: *Psychocomputational Models of Human Language Acquisition*, Geneva, pp. 25–32 (2004)
11. Gold, E.M.: Language identification in the limit. *Information and Control* 10, 447–474 (1967)
12. Gordon, P.: Learnability and feedback. *Developmental Psychology* 26, 217–220 (1990)
13. Parekh, R., Honavar, V.: Grammar inference, automata induction and language acquisition. In: Moisl, Dale, S. (eds.) *Handbook of Natural Language Processing*, pp. 727–774. Marcel Dekker, New York (2000)
14. Pinker, S.: Formal models of language learning. *Cognition* 7, 217–283 (1979)
15. Valiant, L.G.: A theory of the learnable. *Communication of the ACM* 27, 1134–1142 (1984)

Computing Languages with Bio-inspired Devices and Multi-Agent Systems

M. Dolores Jiménez-López

Research Group on Mathematical Linguistics
Universitat Rovira i Virgili
Av. Catalunya, 35, 43002 Tarragona, Spain
mariadolores.jimenez@urv.cat

Abstract. This paper is an overview of agent-based models and/or bio-inspired devices that have been defined in the field of formal language theory. Grammar systems, colonies and eco-grammar systems are presented as syntactical devices for multi-agent architectures. L systems, DNA computing and P systems show the biological inspiration/motivation in formal languages. And finally, Networks of Evolutionary Processors can be defined as bio-inspired multi-agent systems. All these models can be used to compute languages, either natural or formal.

1 Introduction

Languages can be natural or formal/artificial. In both cases, we can define a language as a set of sentences, where a sentence is a finite string of symbols over an alphabet. Therefore, languages, either natural or artificial, are particular cases of symbol systems. The manipulation of symbols is the stem of formal language theory. The theory of formal languages mainly originated from *mathematics* and *linguistics*. From the area of mathematics, A. Thue and E. Post introduced the formal notion of a rewriting system, while A. Turing introduced the general idea of finding models of computing where the power of a model could be described by the complexity of the language it generates/accepts. From linguistics, the study of grammars and the grammatical structure of a language was initiated by N. Chomsky in the 1950s. Chomsky introduced his grammar hierarchy as a tool for modeling and investigating syntax of natural languages. After 1964, formal language theory developed as a separate branch with specific problems, techniques and results and since then it has had an important role in the field of computer science, in fact it is considered as the stem of theoretical computer science.

The first generation of formal languages, fitted into the Chomskian hierarchy, were based on rewriting, and caused the generalization of the tree-like models for computing languages. Turing machines, Chomsky grammars, Post, Thue, or Markov systems are rewriting formalisms. However, the idea of *modeling* certain objects or phenomena has initiated large and significant parts of the theory of

formal languages. In this paper, we review two of the most important motivations in the developing of formal languages models in the last decades: 1) *agent technologies* and 2) *biological ideas*.

According to [14], computing has enjoyed several different metaphors for the notion of computation. Until the mid-1960s, most people thought of computation as *calculation*, or operations undertaken on numbers. From the 1960s, computation was re-conceptualized more generally as *information processing*, or operations on text, audio or video data. With the growth of the Internet, a new metaphor for computation has appeared: computation as *interaction*. In order to exploit this new metaphor of computing, many people deal with agent technologies. In fact, agent-based systems are one of the most important areas of research and development that have emerged in information technology in the 1990s. Roughly speaking, an agent is a computer system that is capable of flexible autonomous action in dynamic, unpredictable, multi-agent domains. The concept of agent can be found in a range of disciplines as, for example, computer networks, software engineering, artificial intelligence, human-computer interaction, distributed and concurrent systems, information retrieval, etc. In general, multi-agent systems offer strong models for representing complex and dynamic real-world environment. The idea of multi-agent systems has been taken into account in formal language theory as well. While in classic formal language theory, grammars and automata modelled classic computing devices where the computation was accomplished by one central agent, new models in formal languages take into account the idea of modern computer science where distributed computation plays a major role. The idea of several devices collaborating for achieving a common goal was formalized in the theory of *colonies* [12] or *grammar systems* [5] that were developed as grammatical models for distributed computation.

Biology has become a pilot science, so that many disciplines have formulated their theories under models taken from biology. The knowledge of the behavior of nature has influenced a number of areas such as artificial intelligence, mathematics or theoretical computer science, giving rise to new perspectives in research. In the last decades, natural computing has become the most extended framework where new models for formal language theory have been developed. One of the most developed lines of research in natural computing is molecular computing, a model based on molecular biology. An active area in molecular computing is *DNA computing* [19], inspired in the way DNA performs operations to generate, replicate or change the configuration of the strings. During the last years, systems biology and cellular biology have achieved an important development. These advances have provided new models for computer science. One of them is cellular computing, a model that emphasizes the concept of microbiological populations as well as the equilibrium of the devices and the relationships between the elements. *P systems* [17] can be considered an example of this emerging paradigm. On the other hand, natural computing has evolved from the first numeric models –like neural networks– to symbolic models –as cellular computing– which are closer to multi-agent systems. *Networks of evolutionary processors* (NEPs) [3] are inspired in both, bio cellular models and basic structures for parallel and

distributed symbolic processing. The main reason for adopting such theoretical perspectives is the need to reach a more realistic human-designed computing, both understanding the processes the nature carries out and taking advantage of the natural mechanisms that science is discovering.

In what follows, we review some of the models in formal language theory that have been motivated by agent theory and/or biological ideas. In section 2, we introduce colonies, grammar systems and eco-grammar systems as multi-agent formal models for computing languages. In section 3, biological inspired theories like Lindenmayer systems, DNA computing and P systems are sketched. In section 4, we refer to Networks of Evolutionary Processors as a model where bio-inspiration and agent architecture have been the basis of the definition of this new formal language framework.

2 Computing Languages with Multi-Agent Systems

2.1 Grammar Systems

Grammar systems theory is a branch of the field of formal languages that provides syntactic models for describing multi-agent systems at a symbolic level using tools from formal grammars and languages. The theory was launched in 1988 [4], when Cooperating Distributed Grammar Systems (CDGS) were proposed as a syntactic model of the blackboard architecture of problem solving. One year later, Parallel Communicating Grammar Systems (PCGS) –very much inspired by the ‘classroom model’ of problem solving– were introduced as a grammatical model of parallelism [20]. Since 1988, the theory has developed in several directions related to several scientific areas. Easy generation of non-context-free structures using context-free rules, modularity, parallelism, interaction, distribution, and cooperation are just some of the advantages that grammar systems have over classical models.

Roughly speaking, a grammar system is a set of grammars working together, according to a specified protocol, to generate a language. Note that while in classical formal language theory *one* grammar (or automaton) works individually to generate (or recognize) *one* language, here we have *several* grammars working together in order to produce *one* language. There are two basic classes of grammar systems:

1. *Cooperating Distributed Grammar Systems* which work sequentially.
2. *Parallel Communicating Grammar Systems* that work in parallel.

A CDGS consists of a finite set of generative grammars that cooperate in the derivation of a common language. Component grammars generate the string in turns (thus, sequentially), under some cooperation protocol. The basic model of CDGS presents sequentiality in its work and homogeneity in the cooperation protocol, however, variants have been introduced that have some parallelism in their function (teams) and that change the initial homogeneity into heterogeneity

of modes of cooperation (hybrid systems). The basic model has been extended, also, by the addition of ‘extra’ control mechanisms.

A PCGS consists of several usual grammars, each with its own sentential form. In each time unit, each component uses a rule, which rewrites the associated sentential form. Cooperation among agents takes place thanks to the so-called query symbols that permit communication among components. In PCGS, we also find variants of the basic model. Modifications in the type of components or in the way strings are communicated produce several new types of PCGS. For more information and formal results about all these variants see [5] and [7].

2.2 Eco-grammar Systems

Eco-grammar systems, introduced in [6] as a subfield of grammar systems, provide a syntactical framework for ecosystems, this is, for communities of evolving agents and their interrelated environment. An eco-grammar system can be defined as a multi-agent system where different components, apart from interacting among themselves, interact with a special component called ‘environment.’ So, within an eco-grammar system we can distinguish two types of components: *environment* and *agents*. Both are represented at any moment by a string that identifies the current state of the component. These strings change according to sets of evolution rules (L systems). Interaction between agents and environment is carried out through agents’ actions performed on the environmental state by the application of some rewriting rules from the set of the action rules of the agents. For formal definitions of eco-grammar systems see [6].

The concept of eco-grammar system is based on six postulates formulated according to properties of Artificial Life:

1. An *ecosystem* consists of an *environment* and a set of *agents*. Both the state of the environment and the states of the agents are described by strings of symbols of given alphabets.
2. In an ecosystem there is a *universal clock* which marks time units, the same for all the agents and for the environment, according to which the evolution of the agents and of the environment is considered.
3. Both the environment and the agents have characteristic *evolution rules*, which are in fact Lindenmayer systems, hence are applied in a parallel manner to all the symbols describing agents and environment; such (rewriting) step is done in each time unit.
4. Evolution rules of the environment are *independent* on agents and on the state of the environment itself. Evolution rules of the agents *depend* on the state of the environment (at a given moment, a subset of applicable rules is chosen from a general set associated to each agent).
5. Agents act on the environment according to *action rules*, which are pure rewriting rules used sequentially. In each time unit, each agent uses one action rule which is chosen from a set depending on the current state of the agent.

6. *Action has priority over evolution* of the environment. In a given time unit exactly the symbols which are not affected by action (in the environment) are rewritten (in parallel manner) by evolution rules.

2.3 Colonies

Colonies as well-formalized language generating devices have been proposed in [12], and developed during the nineties in several directions in many papers.

Colonies can be thought of as grammatical models of multi-agent systems motivated by Brooks' subsumption architectures [2]. They describe language classes in terms of behavior of collections of very simple, purely reactive, situated agents with emergent behavior. Roughly, a colony consists of a finite number of simple modules (regular grammars) which generate finite languages and operate on a shared string of symbols –the *environment* of the colony– without any explicitly predefined strategy of cooperation. Each component has its own reactive behavior which consists in: 1) sensing some aspects of the *context* and 2) performing elementary tasks in it in order to achieve some local changes. The environment is quite passive, its state changes only as result of acts agents perform on it. Because of the lack of any predefined strategy of cooperation, each component participates in the rewriting of current strings whenever it can participate in it. The behavior of a colony –this is, the language– is defined as the set of all the strings which can be generated by the colony from a given starting string. For formal definitions of colonies see [12].

3 Computing Languages with Bio-inspired Devices

3.1 Lindenmayer Systems

Lindenmayer systems (L systems) are the first bio-inspired model in the field of formal language theory. Aristid Lindenmayer introduced L systems in 1968 [13] as a theoretical framework in order to model the development of filamentous organisms, which are composed of cells. These cells receive inputs from their neighbors and change their states and produce outputs based on their states and the input received. L systems model biological growth and because growth happens in multiple areas of an organism, growth is parallel. This parallelism is the main difference from *sequential* rewriting systems of the Chomsky hierarchy. The investigations of L systems are an important and wide area in the theory of formal languages. The modeling of different environmental influences, for example, growth during day versus night, lead to different L systems and thus to different L languages.

The study of L languages has resulted in a language hierarchy, namely the L system hierarchy. Lindenmayer systems are well investigated parallel rewriting systems. For an overview see [11] and for the mathematical theory of L systems see [21].

3.2 DNA Computing

One of the most developed lines of research in natural computing is *molecular computing*, a model based on molecular biology. An active area in molecular computing is *DNA computing* [19] inspired in the way DNA performs operations to generate, replicate or change the configuration of the strings. Splicing systems or H systems –introduced by Tom Head [9]– represent a model for DNA computation that is part of formal language theory. H systems can be viewed as a development in formal language theory that provides new generative devices that allow close simulation of molecular recombination processes by corresponding generative processes acting on strings.

According to [9], in H systems an initial set of molecules is represented by an initial set of strings. The action of a set of endonucleases and a ligase is represented by a set of splicing rules acting on strings. The language of all possible strings that may be generated is a representation of the set of all possible molecules that may be generated by the biochemical recombination processes. The splicing operation consists of splitting up two strings in an arbitrary way an sticking the left side of the first one to the right side of the second one (direct splicing) and the left side of the second one to the right side of the first one (inverse splicing).

Application of molecular computing methods to natural language syntax gives rise to *molecular syntax* [1]. Molecular syntax takes as a model two types of mechanisms used in biology (especially in genetic engineering) in order to modify or generate DNA sequences: *mutations* and *splicing*. Mutations refer to changes performed in a linguistic string, being this a phrase, sentence or text. Splicing is a process carried out involving two or more linguistic sequences. Methods used by molecular syntax are based on basic genetic processes: *cut*, *paste*, *delete* and *move*. Combining these elementary rules most of the complex structures of natural language can be obtained, with a high degree of simplicity.

3.3 P Systems or Membrane Computing

P systems –introduced in [17]– are a model of computation inspired by some basic features of biological membranes. They can be viewed as a new paradigm in the field of natural computing based on the functioning of membranes inside the cell.

P systems, as a computational model based in biology, consist of multisets of objects which are placed in the compartments defined by the membrane structure that delimits the system from its environment. Such structure is represented by a Venn diagram where all the sets, *membranes*, are inside a unique *skin membrane*. A membrane without any membrane inside is called *elementary membrane*. Every membrane delimits a *region*. *Objects*, placed in these regions, are able to evolve traveling to other membranes or being transformed in different objects.

Formal definitions and main issues related to the topic can be found in [17] and [18].

4 Computing Languages with Bio-inspired MAS

4.1 NEPs-Networks of Evolutionary Processors

Networks of Evolutionary Processors (NEPs) are a new computing mechanism directly inspired in the behavior of cell populations. NEPs, introduced in [3] and [15], can be defined as systems consisting of several devices whose communication is regulated by an underlying graph. Such devices, which are an abstract formalization of cells, are described by a set of words (DNA) evolving by mutations, according to some predefined rules. Their outcome travels to the other nodes if they accept it after passing a filtering process. At the end of the process, only the cells with correct strings will survive.

The cellular basis of NEPs relate them with P systems, especially with Tissue P systems [16]. In tissue P systems, cells form a multitude of different associations performing various functions. NEPs could be linked to systems biology as well, because the model aims to develop a holistic theory where the behavior of each agent can influence the environment and the other agents. From the computational point of view, NEPs are related to the Connection Machine [10] and the Logic Flow paradigm [8]. Another important theoretical relationship of NEPs is the theory of grammar and eco-grammar systems [5,6] which share with NEPs the idea of several devices working together and exchanging results.

With all this background and theoretical connections, it is easy to understand how NEPs can be described as agential bio-inspired context-sensitive systems. Many disciplines are needed of these types of models that are able to support a biological framework in a collaborative environment. The conjunction of these features allows applying the system to a number of areas, beyond generation and recognition in formal language theory.

5 Concluding Remarks

Taking into account the achievements of the theory of formal languages, its initial linguistic motivation and its important role in computation, it is not possible to do without this research area in the study of computing languages, either formal or natural. Being part of what is called theoretical computer science, nobody doubts about the adequacy of formal language theory in the field of programming languages. What is maybe not so obvious is that all those devices can be very useful in the description, analysis and processing of natural languages. The multi-agent capabilities of some of these models make them a suitable tool for simulating the processes of generation and recognition in natural language. The biological inspiration of some of those devices may be very useful in the field of natural language, since they provide simple, flexible and intuitive tools for describing natural languages, making easier their implementation in natural language processing systems.

References

1. Bel-Enguix, G., Jiménez-López, M.D.: Biosyntax. An Overview. *Fundamenta Informaticae* 64, 1–12 (2005)
2. Brooks, R.A.: Elephants don't Play Chess. *Robotics and Autonomous Systems* 6, 3–15 (1990)
3. Castellanos, J., Martín-Vide, C., Mitrana, V., Sempere, J.M.: Networks of Evolutionary Processors. *Acta Informatica* 39, 517–529 (2003)
4. Csuhaj-Varjú, E., Dassow, J.: On Cooperating/Distributed Grammar Systems. *Journal of Information Processing and Cybernetics (EIK)* 26, 49–63 (1990)
5. Csuhaj-Varjú, E., Dassow, J., Kelemen, J.: *Grammar Systems: A Grammatical Approach to Distribution and Cooperation*. Gordon and Breach, London (1994)
6. Csuhaj-Varjú, E., Kelemen, J., Kelemenová, A., Păun, G.: Eco-Grammar Systems: A Grammatical Framework for Life-Like Interactions. *Artificial Life* 3(1), 1–28 (1996)
7. Dassow, J., Păun, G., Rozenberg, G.: Grammar Systems. In: Rozenberg, G., Salomaa, A. (eds.) *Handbook of Formal Languages*, vol. 2, pp. 155–213. Springer, Berlin (1997)
8. Errico, L., Jesshope, C.: Towards a New Architecture for Symbolic Processing. In: Plander, I. (ed.) *Artificial Intelligence and Information-Control Systems of Robots 1994*, pp. 31–40. World Science, Singapore (1994)
9. Head, T.: Formal Language Theory and DNA: An Analysis of the Generative Capacity of Specific Recombination Behaviors. *Bulletin of Mathematical Biology* 49, 737–759 (1987)
10. Hillis, W.D.: *The Connection Machine*. MIT Press, Cambridge (1985)
11. Kari, L., Rozenberg, G., Salomaa, A.: L Systems. In: Rozenberg, G., Salomaa, A. (eds.) *Handbook of Formal Languages*, vol. 1, Springer, Berlin (1997)
12. Kelemen, J., Kelemenová, A.: A Grammar-Theoretic Treatment of Multiagent Systems. *Cybernetics and Systems* 23, 621–633 (1992)
13. Lindenmayer, A.: Mathematical Models for Cellular Interaction in Development. *Journal of Theoretical Biology* 18, 280–315 (1968)
14. Luck, M., McBurney, P., Shehory, O., Willmott, S. (eds.): *Agent Technology: Computing as Interaction*. University of Southampton (2005)
15. Martín-Vide, C., Mitrana, V., Pérez-Jiménez, M., Sancho-Caparrini, F.: Hybrid Networks of Evolutionary Processors. In: Canté-Paz, E., et al. (eds.) *Genetic and Evolutionary Computation*, pp. 401–412. Springer, Berlin (2003)
16. Martín-Vide, C., Pazos, J., Păun, G., Rodríguez-Patón, A.: A New Class of Symbolic Abstract Neural Nets: Tissue P Systems. In: Ibarra, O.H., Zhang, L. (eds.) *COCOON 2002*. LNCS, vol. 2387, pp. 290–299. Springer, Heidelberg (2002)
17. Păun, G.: Computing with Membranes. *Journal of Computer and System Sciences* 61, 108–143 (2000)
18. Păun, G.: *Membrane Computing. An Introduction*. Springer, Berlin (2002)
19. Păun, G., Rozenberg, G., Salomaa, A.: *DNA Computing. New Computing Paradigms*. Springer, Berlin (1998)
20. Păun, G., Sântean, L.: Parallel Communicating Grammar Systems: The Regular Case. *Annals of the University of Bucharest* 38, 55–63 (1989)
21. Rozenberg, G., Salomaa, A.: *The Mathematical Theory of L-Systems*. Academic Press, New York (1980)
22. Rozenberg, G., Salomaa, A.: *Handbook of Formal Languages*. Springer, Berlin (1997)

A Novel Strategy for Improving the Quality of Embedded Zerotree Wavelet Images Transmitted over Alamouti Coding Systems

Josmary Labrador, Paula M. Castro,
Héctor J. Pérez-Iglesias, and Adriana Dapena

Department of Electronics and Systems, University of A Coruña,
Campus de Elviña s/n, 15.071. A Coruña. Spain
{jlabrador, pcastro, hperez, adriana}@udc.es

Abstract. This work deals with the transmission of images, previously coded using the *Embedded Zerotree Wavelet* (EZW) transform, over wireless systems in which *Space-Time Coding* (STC) is used. It is shown how the system performance, measured in terms of *Peak Signal to Noise Ratio* (PSNR), can be improved using bit allocation strategies that take into account the special structure of the EZW bitstream, where the bits firstly allocated are associated to the lowest frequency subbands, and therefore, an error-free transmission of such bits will be crucial to appropriately recover the transmitted image.

Keywords: Artificial neural networks, learning rules, EZW transform, Alamouti coding, PSNR metric, image processing, bit allocation, channel estimation.

1 Introduction

The *Embedded Zerotree Wavelet* (EZW) transform is a quite simple image compression algorithm based on a tree-ordering of the wavelet coefficients [1,2]. By taking into account that, for wavelet transforms, the highest energy coefficients reside in the lowest frequency subbands placed at the root node, and that a parent-child relationship is defined between wavelet coefficients from frequency subbands spatially related, so that the children correspond to higher frequencies than their respective parents, one or more subtrees will entirely have coefficients whose value is zero or almost zero with high probability. Those subtrees are called *zerotrees*. Therefore, the bitstream is organized according to the wavelet coefficients ordered from lowest to highest frequency subbands, which allows the decoder to stop the decoding process at any point of the bitstream and still recover the transmitted images but with lower quality. This property is termed as *progressive* or *incremental* decoding.

This paper focuses on the transmission of EZW-coded images over wireless systems making use of the popular *Orthogonal Space-Time Block Code* (OS-TBC) scheme proposed by Alamouti [3], which has been incorporated to the

IEEE 802.11 and IEEE 802.16 standards, for example. Coherent detection using Alamouti-coded systems demands a unitary channel matrix, which is commonly acquired from transmitted training symbols (pilots) by means of supervised algorithms [4,5]. In order to increase the system throughput, there exists a great interest about the development of algorithms to directly estimate the channel from the observations without using pilots. This type of algorithms are termed as *unsupervised* or *blind* [6].

Principal Component Analysis (PCA) was developed in 1901 by Karl Pearson and it is one of the most important paradigms in *Artificial Neural Networks* (ANN) since it is connected with Hebbian-type learning rules [7]. Nowadays, PCA is mostly used as a tool for data analysis and for predictive modeling. Shahbazpanahi et al. in [8] have shown that the channel matrix of Alamouti coding systems can be estimated using PCA although it requires different energies for signal transmission. This unbalanced energy implies a degradation in terms of *Bit Error Rate* (BER) for the signal transmitted with lower energy. In fact, for EZW-coded images it will be shown how *Peak Signal to Noise Ratio* (PSNR) can be considerably improved when the bits associated to the lowest frequency subbands are sent using higher energy than that assigned for the bits corresponding to higher subbands. However, this bit allocation strategy penalizes the EZW property for incremental decoding and, for this reason, it is also proposed in this work a simple decision criterion to decide when that strategy must be used or not.

The work is structured as follows. The channel model is shown in Section 2, where the utilization of PCA for channel estimation is also described. Section 3 shows three different strategies to convert the EZW bitstream to the signals transmitted through the antennas, and Section 4 compares those strategies by performing several computer simulations. Finally, Section 5 presents the main work conclusions.

2 Alamouti Coding Systems

We define the sources s_1 and s_2 as independent equiprobable discrete random variables with values from a finite set of symbols belonging to a real or complex modulation (PAM, PSK, QAM...). In a 2×1 *Multiple-Input/Single-Output* (MISO) case, the vector $\mathbf{x} = [x_1 \ x_2]^T$ of received signals (so-called observations) can be written as $\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{v}$, where $\mathbf{s} = [s_1 \ s_2]^T$ is the source vector, $\mathbf{v} = [v_1 \ v_2]^T$ is the *Additive White Gaussian Noise* (AWGN) vector, and the 2×2 channel matrix has the form

$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix}. \quad (1)$$

Note that this matrix is orthogonal, i.e. $\mathbf{H}^H\mathbf{H} = \mathbf{H}\mathbf{H}^H = \mathbf{I}_2$ with $\|\mathbf{h}\|_2^2 = |h_1|^2 + |h_2|^2$ and thus, the sources can be recovered applying $\hat{\mathbf{s}} = \mathbf{H}^H\mathbf{x}$.

In recent years, several unsupervised methods to estimate the channel matrix (and to recover the sources) have been developed assuming that both sources

and channel matrices are completely unknown at the receiver side (see, for instance, [9] and references therein). PCA constitutes an interesting unsupervised method to estimate the channel matrix since it implies to perform a reduced number of operations. By considering the covariance matrix \mathbf{C}_x obtained from the observations, i.e. $\mathbf{C}_x = E[\mathbf{x}\mathbf{x}^H]$, PCA computes an orthogonal basis \mathbf{U} , so that a new set of orthogonal signals, given by $\mathbf{z} = \mathbf{U}^H \mathbf{x}$, can be obtained [7].

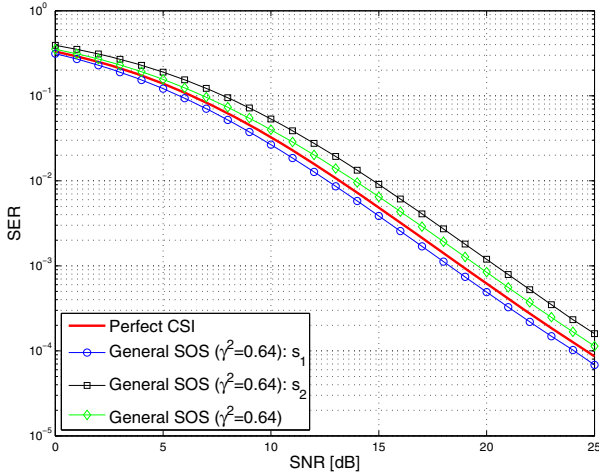


Fig. 1. SER performance of general SOS algorithm for randomly generated symbols

For the scenario of Alamouti coding systems, since \mathbf{H} is unitary, the matrix \mathbf{U} is simply a normalized version of the channel matrix \mathbf{H} . It has been proved in [8] that the identification of the channel matrix is possible only when the sources have different energies, i.e. when the source s_1 is transmitted with an energy given by $E[|s_1(n)|^2] = 2/(1 + \gamma^2)$, while s_2 is sent with the energy $E[|s_2(n)|^2] = 2\gamma^2/(1 + \gamma^2)$, where γ is the parameter of energy unbalance.

Many methods can be applied to compute the PCA decomposition. For instance, Via et al. in [10] have proposed an adaptive learning procedure, while Pérez et al. in [9] have presented a block algorithm for that purpose. In both cases, the matrix \mathbf{C}_x is estimated by sampling averaging of the N_B symbols received per frame.

In order to illustrate the degradation associated to the source energy unbalance, we consider a scenario where blocks of $N_B = 1000$ symbols are generated from an equiprobable distribution. These symbols are modulated using 4-QAM and transmitted through block fading Rayleigh channels. We use the PCA implementation presented in [9] referred to as general *Sum-Of-Square* (SOS) reconstruction, where the aforementioned unbalancing parameter, γ , is used and whose value is set up to $\gamma^2 = 0.64$, i.e. $\gamma = 0.8$. Figure 1 plots the *Symbol Error Rate* (SER) in terms of *Signal Noise Ratio* (SNR) for each source, s_1 and s_2 , and the corresponding mean SER value. As a reference, it is also depicted SER under

Perfect Channel Side Information (Perfect CSI) assumptions. Note that Perfect CSI corresponds to the mean SER obtained for the two sources (s_1 and s_2) when CSI is perfectly known at the receiver side. By comparing the perfect CSI curve to those obtained using PCA, it can be seen from the figure that the source with highest energy (s_1) exhibits lower SER, while the source with smaller one (s_2) suffers from a loss in terms of SER with respect to perfect CSI scenario.

3 Bit Allocation Strategies

The results above reported show that the unbalanced energy is an important drawback for the use of PCA because it produces a SER degradation for one of the sources. In this section, we show that this degradation can be compensated with an adequate bit allocation taking into account the bit organization into the stream. The common strategy to convert a bitstream to a modulated signal (like, for example, an M-QAM signal), consists of sequentially processing that bitstream by taking groups of $b = \log_2(M)$ bits, which are mapped to their corresponding modulated symbol. The modulated symbols are subsequently divided into two sources (remember that a 2×1 Alamouti coded system is implemented): odd symbols, which are sent by the source s_1 , and even symbols, which are sent by s_2 . For EZW, such bit allocation implies that bits associated to any frequency subband are transmitted by both sources and, as a consequence, the degradation caused by unbalanced energy affects to all the subbands. This bit allocation is termed as *Approach 1* in the following.

On the other hand, as a result of the structure observed for the EZW bitstream, we propose a novel bit allocation strategy (termed as *Approach 2* in this work), where the bitstream is divided into two parts. The first part (corresponding to the lower subbands) is transmitted by s_1 , while the second part (corresponding to higher subbands and to the so-called *refinement* bits) by s_2 . This method permits that the energy degradation does not affect to the lower subbands but note, however, that it keeps from the incremental decoding of EZW. For this reason, it is desirable to use this approach only when the resulting quality of the recovered image is considerably better than that obtained applying Approach 1.

The immense majority of current wireless communication standards make use of feedback channels (usually limited in terms of throughput) between both sides of the link to periodically send channel state information from the receiver to the transmitter. For example, *Worldwide Interoperability for Microwave Access* (WiMAX) standard uses this channel to send an index for selecting the most adequate code according to channel conditions. In this work, we propose to use this feedback channel to indicate to the transmitter which bit allocation strategy must be used in order to obtain an adequate PSNR.

Since it is not possible to compute the PSNR at the receiver-end for a given image, it is also proposed to use a set of training images to get the estimated PSNR according to visual image quality as a function of SNR. Then, this information is stored at the receiver. Before the transmission of each new image, the

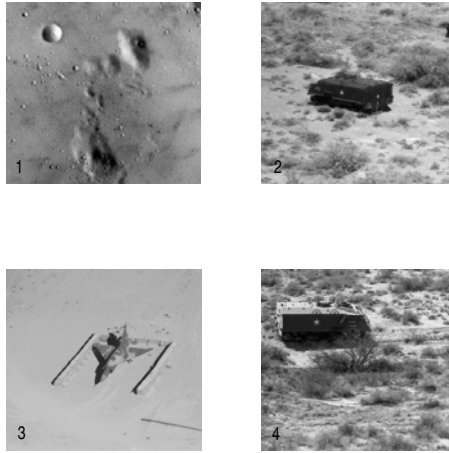


Fig. 2. Computer simulations: Training images

receiver estimates the SNR and decides the bit allocation approach with higher estimated PSNR for a given SNR. An alternative way of interpreting this rule consists of defining the SNR threshold, denoted by SNR_t , which marks out the working regions for each approach. In other words, this approach, termed as *Hybrid Approach*, can be described by the following decision rule

$$\begin{cases} \text{estimated SNR} \geq \text{SNR}_t \rightarrow \text{Use Approach 1} \\ \text{estimated SNR} < \text{SNR}_t \rightarrow \text{Use Approach 2,} \end{cases} \quad (2)$$

whose result is sent to the transmitter through the feedback channel. The open issue is how to find that SNR threshold, SNR_t , which defines the border between the two working regions. In the next section, we will show a method based on the visual quality of the recovered images.

4 Computer Simulations

In order to compare the proposed bit allocation strategies, we consider a computer scenario where the bitstream of EZW-coded images is modulated using 4-QAM. The symbols are transmitted in blocks of size $N_B = 1000$ using Alamouti coding. In order to guarantee that the channel matrix can be estimated using PCA, the transmitter unbalances the source energy by means of a parameter $\gamma^2 = 0.64$. Thus, PCA is used to acquire the channel matrix estimate per received frame. Note that the covariance matrix obtained from the observations is computed using all the frame symbols, i.e. N_B symbols.

For the training step, the four images plotted in Figure 2 have been coded using EZW [1]. Figure 3 plots the PSNR in terms of SNR obtained by averaging

¹ The original images are constituted by 256×256 pixels with 256 gray levels.

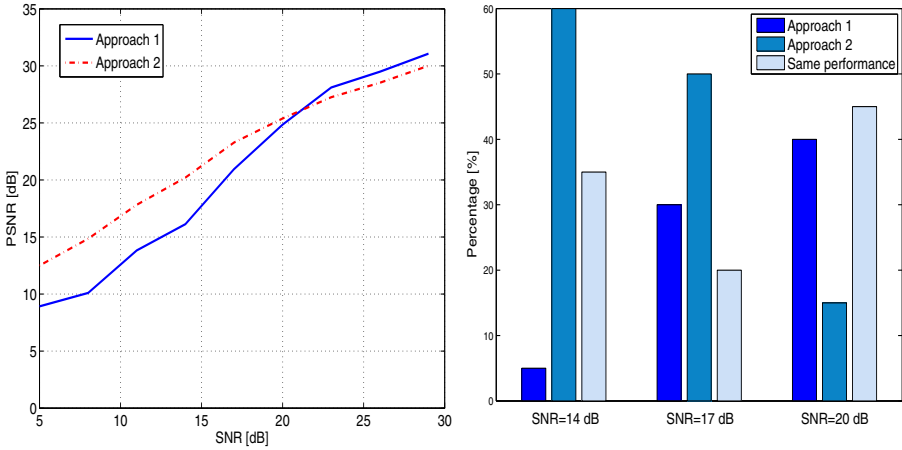


Fig. 3. Training step: Establishment of SNR_t threshold using visual test

the results for 10 Rayleigh channels randomly generated. It can be seen from this figure that the Approach 2 provides the best PSNR for low and medium SNR values since bits associated to lower frequency subbands are sent by means of the source having highest energy. With the goal of establishing the threshold parameter SNR_t for the decision rule described in Equation (2), the recovered images corresponding to different SNR values, specifically 14, 17, and 20 dB for 10 channel realizations, have been empirically observed. A test oriented to decide which approach provides the best quality percentage or ratio taking into account all the set of training images is applied. Thus, Figure 3 also shows the results obtained from this visual test, which allows us to conclude that

- Firstly, for an SNR value of 14 dB, the improvement achieved with Approach 2 compared to Approach 1 is substantial. Therefore, it is apparent that Approach 2 is the best choice in such a case leading to better quality reconstruction.
- Secondly, for an SNR value of 17 dB, the improvement achieved with Approach 2 compared to Approach 1 is not as significant. By considering the results obtained with Approach 1 and that both approaches exhibit exactly the same performance for about a ratio of 20 to 100 (this scenario is labeled in the figure as *Same performance*), Approach 1 can be decided as a more adequate choice than Approach 2 if progressive decoding is desired.
- Finally, for an SNR value of 20 dB, Approach 1 is clearly the best choice, since it is only outperformed by Approach 2 less than 20% of cases.

As a result, it can be established a threshold parameter of SNR equal to 17 dB.

Figure 4 shows a comparison in terms of PSNR versus SNR between the three bit allocation strategies studied in this work only considering one of the test images depicted in Figure 2 and 10 channel realizations. Figure 5 illustrates

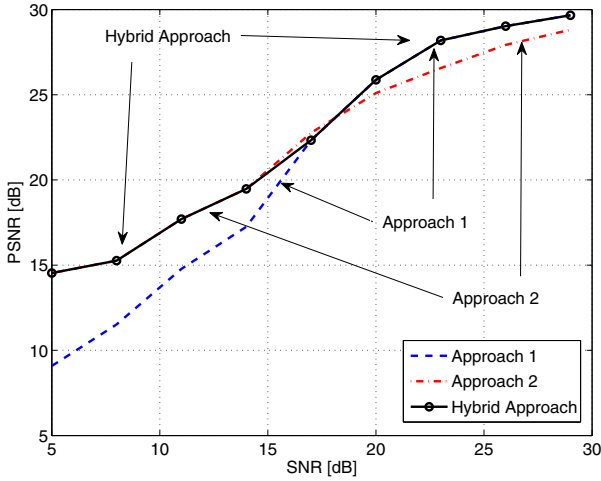


Fig. 4. PSNR results obtained from computer simulations for image 1

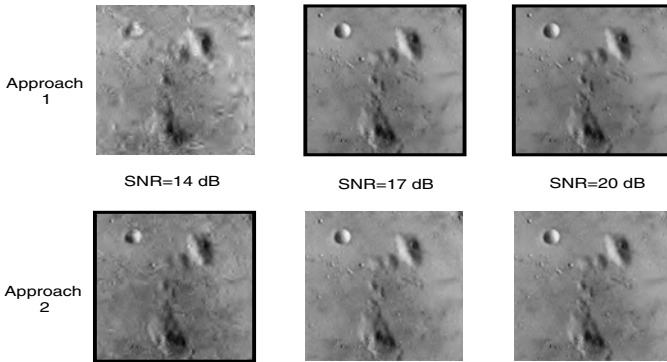


Fig. 5. Black rectangles indicate the reconstructed images for each SNR

the visual quality of the recovered images corresponding to SNR values of 14, 17, and 20 dB for a given channel realization, where the black rectangle marks the recovered ones in the event of implementing the above proposed hybrid approach.

5 Conclusions and Future Work

A lot of algorithms based on wavelets for image compressing, like EZW, produce bitstreams by ordering the subbands from lowest to highest, which allows to achieve different image qualities after the progressive decoder. This means that the first bits of the EZW sequence are “more important” than the following bits. In this work, we have used this property to improve the performance of the EZW images transmitted over wireless systems where Alamouti space-time block coding is used. We have focused on the utilization of PCA to acquire the channel

matrix estimate avoiding the use of pilot symbols. As a result, three proposed bit allocation strategies combined with a decision rule based on an empirical SNR threshold parameter allow us to improve the quality of the reconstructed images after their transmission affected by fluctuating wireless channel conditions.

Future work to be developed by the authors is focused on studying different bit allocation strategies like, for instance, the sending of the refinement bits corresponding to LL subband through the channel with largest energy. Since PSNR criterion above explained is empirically obtained, other assessment metrics must be analyzed, specially those based on Human Visual System, as for example the Visual Information Fidelity measure. Additionally, this work could be extended to the standard of image processing JPEG2000, thus avoiding some of the drawbacks inherent to EZW.

Acknowledgments. This work was supported by Xunta de Galicia (grant numbers 10TIC105003PR and 09TIC008105PR) and Ministerio de Ciencia e Innovación of Spain (grant numbers TEC2010-19545-C04-01 and CSD2008-00010).

References

1. Taubman, D.S., Marcellin, M.W.: JPEG2000 Image Compression Fundamentals, Standards and Practice. Kluwer Academic Publishers, Dordrecht (2002)
2. Shapiro, J.M.: Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing* 41(12), 3445–3462 (1993)
3. Alamouti, S.M.: A Simple Transmit Diversity Technique for Wireless Communications. *IEEE Journal Select. Areas Communications* 16, 1451–1458 (1998)
4. Budianu, C., Tong, L.: Channel Estimation for Space-Time Orthogonal Block Codes. In: Proc. of International Conference on communications, pp. 1127–1131 (2001)
5. Naguib, A.F., Tarokh, V., Seshadri, N., Calderbank, A.R.: A Space-Time Coding Modem for High-Data-Rate Wireless. *IEEE Journal on Selected Areas in Communications* 16(8), 1459–1478 (1998)
6. Comon, P., Jutten, C.: Handbook of Blind Source Separation, Independent Component Analysis and Applications. Academic Press, London (2010)
7. Haykin, S.: Neural Networks: A Comprehensive Foundation. Macmillan, Basingstoke (1994)
8. Shahbazpanahi, S., Gershman, A.B., Manton, J.: Closed-form Blind MIMO Channel Estimation for Orthogonal Space-Time Block Codes. *IEEE Trans. on Signal Processing* 53(12), 4506–4516 (2005)
9. Pérez-Iglesias, H.J., García-Naya, J.A., Dapena, A., Castedo, L., Zarzoso, V.: Blind Channel Identification in Alamouti Coded Systems: A Comparative Study of Eigendecomposition Methods in Indoor Transmissions at 2.4 GHz. *European Transactions on Telecommunications (Special Issue)* 19(7), 751–759 (2008)
10. Vía, J., Santamaría, I., Pérez, J., Ramírez, D.: Blind Decoding of MISO-OSTBC Systems based on Principal Component Analysis. In: Proc. of International Conference on Acoustic, Speech and Signal Processing, vol. IV, pp. 545–549 (2006)

Applying Data Mining Techniques in a Wyner-Ziv to H.264 Video Transcoder*

José Luis Martínez¹, Alberto Corrales-García², Pedro Cuenca²,
and Francisco José Quiles²

¹ Architecture and Technology of Computing Systems Group, Complutense University
Ciudad Universitaria s/n, 28040 Madrid, Spain

jose Luis.martinez@fdi.ucm.es

² Instituto de Investigación en Informática de Albacete (I3A)

University of Castilla-La Mancha. Campus Universitario, 02071 Albacete, Spain
{albertocorrales, pcuenca, paco}@dsi.uclm.es

Abstract. This paper proposes an improved Wyner-Ziv to H.264 transcoder for supporting mobile-to-mobile video communications. In this framework, both transmitter and receptor should employ video encoders and decoders of low complexity. Taking advantage of both paradigms, in terms of low complexity algorithms, a suitable solution consists in transcoding from Wyner-Ziv to H.264. In order to reduce this process this paper proposes an algorithm which is based on the hypothesis that macroblock coding mode decisions in H.264 video have a high correlation with the distribution of the side information residual in Wyner-Ziv video. The proposed algorithm, which is based on data mining techniques, selects one sub-set of the several coding modes in H.264. Simulation results show that the proposed transcoder reduces the inter prediction complexity in H.264 by up to 53%, while maintaining coding efficiency.

Keywords: Distributed Video Coding, Wyner-Ziv, H.264, Data Mining, Machine Learning, Transcoding.

1 Introduction

Mobile-to-mobile video teleconferencing is one of the most requested services that newest networks for mobile devices (such as 4G) can support. However, in this framework, both transmitter and receptor should employ video encoders and decoders of low complexity. On the one hand, traditional video codecs, such as H.264 Advanced Video Coding (AVC) [1] are based on architectures which have encoders of higher complexity than decoders. On the other hand, Wyner-Ziv (WZ) [2] video coding is an innovative paradigm which reduces the processing complexity of the

* This work was supported by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under Grants CSD2006-00046 and TIN2009-14475-C04. It was also partly supported by JCCM funds under grant “PEII09-0037-2328” and “PII2I09-0045-9916”. The present approach has been implemented by using the VISNET2-WZ-IST software developed in the framework of the VISNET II project.

encoder, leading to a low-cost implementation, while the majority of the computations are taken over by the decoder. Taking advantage of both paradigms, in terms of low complexity algorithms, a suitable solution could consist in transcoding from WZ to H.264 as Fig. 1 depicts. In the scenario the end-user devices will employ the lowest complexity algorithm of traditional and WZ video coding while the majority of computations will be taken over by the transcoder.

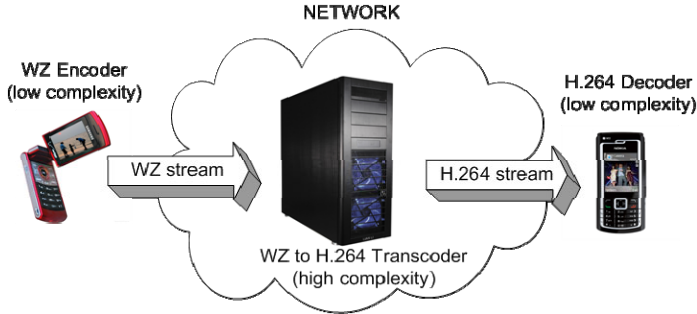


Fig. 1. Scheme of DVC to H.264 transcoding framework

Nevertheless, the transcoding process should be carried out in an efficient way so as to avoid major delays in communication. The fact is that between WZ and H.264 there are many differences, such as type of frames (I, P frames as opposed to Key and Wyner-Ziv frames), GOP patterns (IPPP as opposed to K-WZ-K), GOP sizes (2, 4, 8 as opposed to 12), that need to be resolved in the transcoder. This paper proposes an improved transcoder, with respect to the reference cascade one, which not only efficiently converts the bitstream, but also reduces the time to perform this task. The proposed algorithm reuses part of the information collected in the Side Information (SI) generation process (this algorithm can be seen as the motion estimation performed in the WZ decoding algorithm) to reduce the overall MB partition checked into a sub-set of them; the algorithm is based on a Data Mining (DM) process which generates a decision tree from this statistical information. The results show a transcoder that performs this process efficiently, with a time saving of up to 53% with a negligible Rate – Distortion (RD) penalty drop.

Many different transcoding approaches, based on traditional standards, have been proposed in the literature, yet there are only a few approaches based on Wyner-Ziv which have been proposed recently for transcoding WZ to H.263 [3] or WZ to H.264 [4]. Although the idea of applying the DVC paradigm in a transcoder framework was introduced in [2], it was not until 2008 that Peixoto et al. [3] proposed the first architecture to support mobile-to-mobile communications by a WZ to H.263 transcoder. In this approach a reutilization of the MVs to accelerate the time spent on the H.263 ME was also made.

The rest of the paper is organized as follows: Section 2 briefly reviews the principles of operation of the WZ video coding as well as the H.264 video coding standard; Section 3 introduces our approaches for inter-frame prediction, based on

machine learning techniques; in Section 4, we carry out a performance evaluation of the proposed algorithm; finally, Section 5 draws our conclusions.

2 Technical Background

2.1 Wyner-Ziv Video Codec Architecture

The WZ video coding architecture adopted in this paper was established by the VISNET-II project [5] which is depicted in Fig. 2.

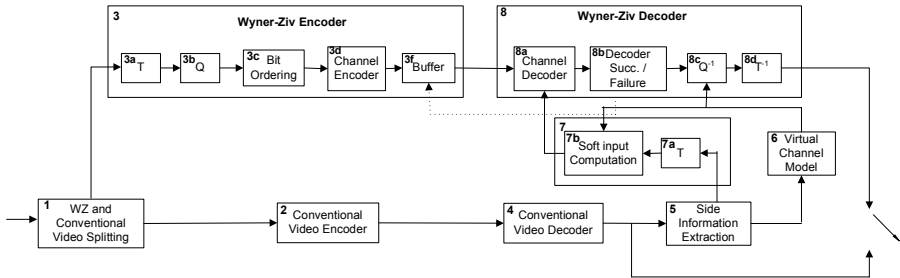


Fig. 2. Block diagram of the reference WZ video coding architecture

In a nutshell, the video sequence is divided into Key (K) frames and WZ frames in the splitting module (1). At the encoder, the K frames are encoded using Intra H.264 video codec [1] (2). WZ frames will follow the WZ encoding algorithm which, firstly, transforms pixel values into coefficients by means of the DCT (3a). In addition, coefficients of WZ frames (already in DCT domain) are organized into bands, and then quantized (3b). Over the resulting quantized symbol stream, bitplane extraction is performed per band (3c). The parity bits produced by the channel encoder are stored in the buffer and transmitted in small numbers upon decoder request via the feedback channel; the systematic bits are discarded (3f).

On the decoder side, the K frames are firstly decoded using Intra H.264 video codec [1] (4) and these frames are used in the SI generation process (5). The frame interpolation module is used to generate the SI frame, an estimate of the WZ frame, based on previously decoded frames [5]. These SI pixel values are also transformed into coefficients and then are used as soft values for the information bits, taking into account the statistical modeling of the virtual noise (7b). A Laplacian model is used to obtain an approximation of the residual distribution (6). The SI is used by an iterative decoding algorithm to obtain the decoded quantized symbol. The success of the channel decoding algorithm is determined by module 8b; if the decoding algorithm does not converge, more parity bits are requested using the feedback channel. After that, the quantized reconstructed coefficients are obtained using the correlation noise model estimated in (6) and the quantized SI coefficients (8c). Finally, module (8d) inverts the transform coefficients to recover pixel values.

2.2 H.264 Inter Prediction

H.264 contains a large amount of compression techniques and innovations compared with the previous standards; it allows more compressed video sequences to be obtained. Although H.264 can achieve higher coding efficiency than any other previous coding standard, the computation complexity also increases significantly. Inter prediction is the most time consuming task in the H.264 encoder. Inter prediction in H.264 allows not only the use of the MBs in which the images are decomposed for the motion estimation and compensation processes, but also the use of partitions resulting from dividing the MB in different ways. Greater flexibility gives greater reliability to the H.264 encoding process. This feature is known as variable block size for the ME.

3 Proposed Transcoder

To provide a mobile video communication framework of low complexity at both ends, this paper proposes a WZ to H.264 video transcoder which collects all the computations while keeping the complexity of the end-user devices as low as possible. The reference transcoder is composed of the full WZ decoding and full H.264 encoding algorithms, but both processes working sequentially implies higher time consumption and thus higher delays. As mentioned in Section 2.2, the inter prediction part of the WZ/H.264 transcoder takes a long time to search all inter modes and intra modes for inter-frame coding exhaustively, and it is the most suitable part to be accelerated. Therefore, the improved transcoder presented in this paper reuses some of the operations performed in the first half (WZ decoding) to prevent unnecessary operations from being performed in the second half (H.264 encoding).

The homologous procedure to the inter prediction carried out on the WZ side is the SI creation process. This process tries to estimate the movement of the called WZ frames by means of motion compensated interpolation between adjacent K frames. It thus seems that most of the operations and data generated in the SI generation process can be useful in the inter prediction algorithm, or at least this information can be used to reduce the time taken.

The starting point for this idea was the “look-and-feel” of the mathematical statistics applied to the WZ SI motion residual information. For example, the Flower Garden motion estimation behaviour against the mean and the variance is shown in Fig. 3. Fig. 3.a shows the original YUV second frame, Fig. 3.b the MB mode selection made by the H.264 reference software for coding each MB within the frame (for legend: gray are skip MB and black are Intra MB). Fig. 3.c and 3.d shows the mean and the variance statistic applied to this WZ motion residual information in 4x4 blocks respectively. As the coding mode changes from low complexity modes (skipped, 16x16, 16x8, etc.) to high complexity modes (8x8 and Intra modes), the mean and the variance value increases for the corresponding MB, as can be clearly seen. Comparing the images shown in Fig. 3, we can see there are some relationships between the processed WZ motion residual information (generated in the SI generation process) and the H.264 mode partition selection.

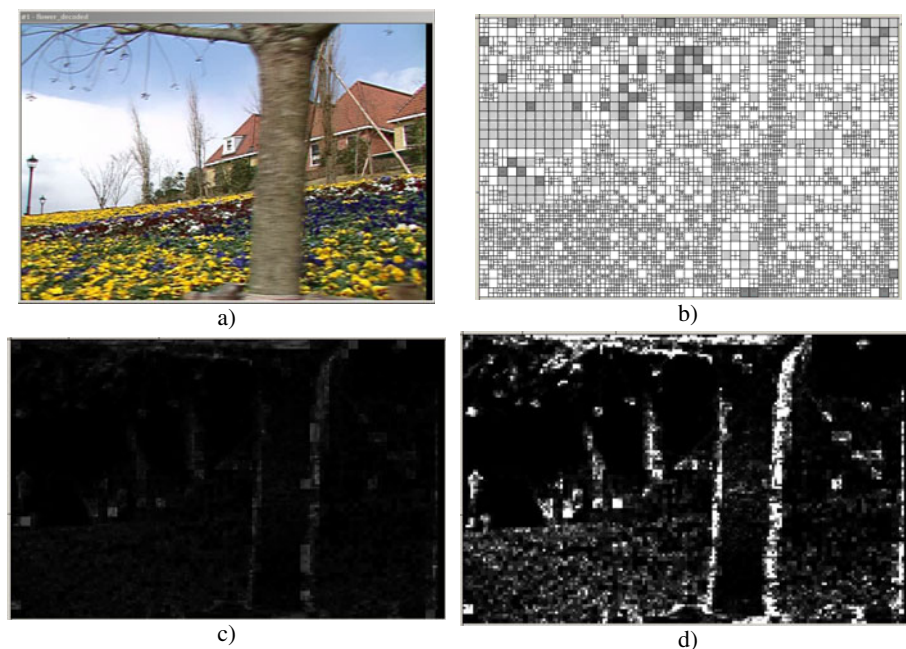


Fig. 3. Block diagram of the reference WZ video coding architecture a) Original Frame; b) MB selection; c) Mean residual and d) variance residual

DM is the process of finding correlations or patterns among dozens of fields in large relational databases. The information gathered in this way can be converted into knowledge, for use in applications that rely on knowledge. Machine Learning (ML) is concerned with the design and development of algorithms and techniques that allow computers to "learn". [6]. At this point, the present approach consists of trying to understand (using an ML algorithm) all this statistical information and create a decision tree to be implemented as part of the H.264 encoder algorithm in order to replace the more complex reference one.

For the training stage, the software used in this data mining process was WEKA v3.6 [7]. WEKA is a collection of ML algorithms for data mining tasks. The training files used by WEKA are formed using some statistical information extracted from the SI generation process and the corresponding MB codec decision. The minimum unit to apply these concepts is the MB; this procedure extracts the statistical information from every MB.

The JRip rule learner [6], a fast algorithm for learning "IF-THEN" rules, was used for creating the rules of the different nodes in the decision tree. The idea consist thus on once the WZ video is decoded, try to gather as much information as we can in order to give WEKA attributes to work with. From the SI generation process the proposed algorithm stores: 1) the amount of the whole MB residual (16x16 residual); 2) the means of the variances; it computes the sixteen 4x4 variances of the sixteen sub-partitions that an MB can be divided into, and then returns the mean of all of them; 3) the variance of the means; in this case, the sixteen 4x4 means are calculated,

and then their variance is returned; 4) the kelastic variable, defined as: $\text{kelastic} = (\text{residual}/\text{count}) * (1.0 + (\sqrt{(\text{dx} + \text{dy})}/2))$; where residual is the amount of the residual of the whole MB; count is the number of pixels per MB ($16 * 16$, 256 pixels); and dx and dy are the corresponding MVs for this MB; and finally, 5) the MV length defined as $\sqrt{(\text{dx} + \text{dy})}$. The class variable used for classifying the samples will be the decisions made by the H.264 reference software encoder (JM version 17.0 [8]), in the training sequence.

In this process, we applied supervised learning because we found that, after some experimentation, there are MB partitions that are more correlated to each other than others. Therefore, in order to define mode sets when we need them, the H.264 reference software decisions will be modified to create binary decisions, i.e., instead of trying to determine the final MB mode decision (such as 16×16 , 16×8 , Intra or whatever), the proposed algorithm only determines whether the final MB codec decision belongs to LOW COMPLEXITY or HIGH COMPLEXITY sets, which are defined as follows: The first one is made up of {SKIP, 16×16 , 16×8 , 8×16 } and the second one of { 8×8 , 8×4 , 4×8 , and 4×4 }. Intra mode will always be checked, although at the beginning the algorithm also tries to determine whether an MB can be coded as Intra. After some experimentation, we found that the RD results are better if both classifications (LOW COMPLEXITY and HIGH COMPLEXITY) can check for Intra coding. In addition, we emphasize that this procedure could be applied again for each leave in order to further divide both LOW COMPLEXITY and HIGH COMPLEXITY modes. In this work we have only implemented the first level decision; include more level of the proposed tree deal with more time reduction as well as an increase of RD penalty.

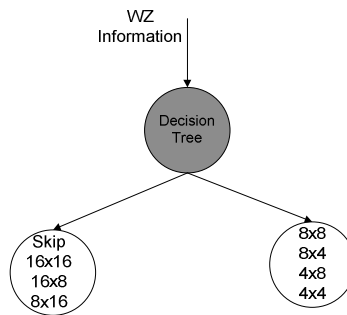


Fig. 4. The Decision Tree

After trying with different kinds of sequences, and for each sequence a different number of frames, we found that sequences which contain varying regions from homogenous to high-detail serve as good training sets. Good sample sequences could be Flower and Soccer sequences. Finally, the best training set was made using 150 P-frames (the whole sequence) in the Soccer sequence (QCIF format, frame rate 15 Hz), encoded at matrix for quantization of the WZ frames of 7 [3], and key frames were encoded with the H.264 [1] Quantization Parameter (QP) 31.

The decision trees proposed in this paper as a solution to replace the MB coding mode decisions in WZ to H.264 video transcoders consist of leaves and branches, as is shown in Fig. 4. The leaves are the classifications and the branches are the features that lead to a specific classification. The gray circle in Fig. 4 represents the decision tree and the white circles represent a set of MB partitions where the reference standard can choose. The output of the tree is a set of H.264 MB modes. Fig. 5 shows the different rules defined in the decision tree algorithm where, basically, these thresholds are found by the ML process.

```

if((meansofvariances4x4 >= 280.86) && (residual16x16 >= 2109)) return HIGH_COMPLEXITY;

if((_vectorlength >= 2.83) && (meansofvariances4x4 >= 595.22)) return HIGH_COMPLEXITY;

if( (meansofvariances4x4 >= 439.07) && (kelastic >= 6.82)) return HIGH_COMPLEXITY;

if((_vectorlength >= 2) && (meansofvariances4x4 >= 384.40) && (kelastic >= 5.81) return HIGH_COMPLEXITY;

if((meansofvariances4x4 >= 280.86) && (_vectorlength >= 2) && (residual16x16 <= 1554) && (residual16x16 >= 1109) && (varianceofmeans4x4 <= 3351.37))return HIGH_COMPLEXITY;

Otherwise: return LOW_COMPLEXITY

```

Fig. 5. The Decision Tree Algorithm

4 Performance Evaluation

The proposed WZ to H.264 Transcoder for the Baseline Profile has been implemented in the H.264/AVC JM 17.0 reference software [1]. First, the transcoder fully decodes the WZ sequence, and the information required by the decision trees is gathered in this stage. The MB coding mode decision determined by the decision trees is used in the low complexity H.264 encoding stage, and the H.264 MB mode decision is replaced by a simple mode assignment, or a range of them.

This performance evaluation included an extensive set of experiments with videos representing a wide range of motion, texture, and colour. Four sequences were fully transcoded, at commonly used frame rates: 15. All of them were in QCIF (176x144) format, which is the most suitable resolution for mobile to mobile video conferencing. The parameters used in the H.264 encoder configuration file for testing the mechanism proposed against a cascade WZ to H.264 transcoder are those that are included in the encoder_baseline.cfg file. The metrics of interest are Δ Time, Δ PSNR and Δ Bitrate where Δ Time (%) = $((T_{JM}-T_{FI})/ T_{JM}) * 100$ and Δ PSNR and Δ Bitrate are calculated according to [8].

Table 1 shows the results; compared with the cascade WZ to H.264 reference transcoder, and for the average of all the sequences, the proposed transcoder has a PSNR drop of at most 0.14 dB on average for a given bitrate, and a bitrate increase of at most 3.3 % for a given quality. This negligible drop in RD performance is more than offset by the decrease in computational complexity, which is reduced by around 53% for the average of all the sequences.

Table 1. Δ Time, Δ PSNR, and Δ Bitrate for the compared sequences

Sequence	RD performance of the WZ/H.264 video transcoder					
	GOP 2			GOP 4		
	Δ Bitrate (%)	Δ PSNR (dB)	TR (%)	Δ Bitrate (%)	Δ PSNR (dB)	TR (%)
Foreman	4,34%	-0,171	50,99%	3,21%	-0,124	53,09%
Soccer	6,56%	-0,272	46,01%	5,91%	-0,239	47,43%
Hall	0,54%	-0,026	58,76%	0,73%	-0,034	59,46%
Mobile	2,89%	-0,134	46,98%	1,63%	-0,078	53,93%
Coastguard	2,41%	-0,102	50,32%	1,48%	-0,060	54,92%
<i>mean</i>	3,35%	-0,141	50,61%	2,59%	-0,107	53,77%

5 Conclusions

This work presents a low complexity WZ to H.264 transcoder which is able to support mobile to mobile video communications. The complex MB mode coded partition process is replaced by a decision tree algorithm which can determine a sub-set of the MB partitions and sub-partitions in which the H.264 encoder can search. This decision tree is based on a DM process which reuses most of the operations carried out in the WZ decoding algorithm, or to be more precise, in the SI generation process. In this way, a complexity reduction of 53% is achieved without negligible RD loss.

References

1. ISO/IEC International Standard 14496-10:2003, Information Technology – Coding of Audio – Visual Objects – Part 10: Advanced Video Coding
2. Girod, B., Aaron, A., Rane, S., Rebollo-Monedero, D.: Distributed video coding. Proceedings of the IEEE 93, 71–83 (2005)
3. Peixoto, E., Queiroz, R.L., Mukherjee, D.: A Wyner-Ziv Video Transcoder. IEEE Transaction on Circuits and Systems for Video Technology 20, 189–200 (2010)
4. Martínez, J.L., Fernández-Escribano, G., Kalva, H., Fernando, W.A.C., Cuenca, P.: Wyner-Ziv to H.264 Video Transcoder for Low Cost Video Communications. IEEE Transactions on Consumer Electronics 55, 1453–1461 (2009)
5. Ascenso, J., Brites, C., Dufaux, F., Fernando, A., Ebrahimi, T., Pereira, F., Tubaro, S.: The VISNET II DVC Codec: Architecture, Tools and Performance. In: Proceedings of EURASIP European Signal Processing Conference (EUSIPCO), Denmark (2010)
6. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco (2005)
7. WEKA. The University of Waikato, <http://www.cs.waikato.ac.nz/ml/weka/>
8. Bjøntegaard, G.: Calculation of Average PSNR Differences between RD-Curves. In: 13th VCEG-M33 Meeting, USA (2001)

On the Use of Genetic Algorithms to Improve Wavelet Sign Coding Performance

Ricardo García^{1,*}, Otoniel López¹, Antonio Martí², and Manuel P. Malumbres¹

¹ Miguel Hernández University,
Avda. Universidad s/n, 03202, Elche, Spain
{r.garcia, otoniel, mels}@umh.es

² Universidad Politécnica de Valencia,
Camino de Vera s/n, 46222, Valencia, Spain
amarti@disca.upv.es

Abstract. Compression of wavelet coefficient sign has been assumed to be inefficient for a long time. However, in the last years several proposals have been developed and, in fact several image encoders like JPEG 2000 include sign coding capabilities. In this paper, we present a new sign coding approximation using a genetic algorithm in order to efficiently predict the sign of wavelet coefficients. We have included that prediction in a fast non-embedded image encoder. Preliminary results show that, by including sign coding capabilities to a non-embedded encoder, the compression gain is up to 17.35%, being the Rate-Distortion (R/D) performance improvement up to 0.25 dB.

Keywords: sign coding, wavelets, image coding, genetic algorithms.

1 Introduction

Wavelet transforms have proved to be very powerful tools for image compression. Many state-of-the-art image codecs, including the JPEG2000 standard [1], employ a wavelet transform in their algorithms. One advantage is the provision of both frequency and spatial localization of image energy. The image energy is compacted into a small fraction of the transform coefficients and compression can be achieved by coding these coefficients. The energy of a wavelet transform coefficient is restricted to non-negative real numbers, but the coefficients themselves are not, and they are defined by both a magnitude and a sign. Shapiro stated in [2] that a transform coefficient is equally likely to be positive or negative and thus one bit should be used to encode the sign. In recent years, several authors have begun to use context modeling for sign coding [3][4][5].

In [5], A. Deever and S. Hemami examines sign coding in detail in the context of an embedded wavelet image coder. The paper shows that a Peak Signal to Noise Ratio (PSNR) improvement up to 0.7 dB is possible when sign entropy

* Thanks to Spanish Ministry of education and Science under grant DPI2007-66796-C03-03 for funding.

coding and a new extrapolation technique based on the mutual information that biorthogonal basis vectors provide to improve the estimation of insignificant coefficients are combined. However, the contribution of sign coding by itself to the PSNR improvement is only up to 0.4 dB.

In [4] the Embedded Block Coding with Optimized Truncation of the embedded bit-streams (EBCOT), core coding tool of the JPEG 2000 standard, encodes the sign of wavelet coefficients using context information from the sign of horizontal and vertical neighbor coefficients (North, South, East, West directions). Five context are used to model the sign coding stage.

In [3], X. Wu presents a high order context modeling encoder. In this coder, the sign and the textures share the same context modeling. This model is based on a different neighborhood for the HL, LH and HH wavelet subbands. For the HL subband, the information of North, North-West, North-East, North-North and South sign is used to predict the current coefficient sign. The neighbors sign information used for the LH subband are North, North-West, North-East, West-West and East. Finally, for the HH subband, an inter-band prediction is used besides the intra-band prediction used by the HL and LH subbands.

Genetic algorithms were first introduced by Holland in [6] and they are nowadays well known techniques for finding nearly optimal solutions of very large problems and also, they have been used in image processing [7][8].

In a genetic algorithm, the evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated by means of a cost function that determines the optimal degree we are looking for (i.e compression rate). Multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

In this paper, we will explore the convenience of employing genetic algorithms to efficiently predict the wavelet coefficient signs. If we find a genetic algorithm that help us to define a good wavelet sign prediction, then, instead of coding the sign, we will encode the result of the prediction (i.e success or failure). If the sign prediction is really good, a binary entropy encoder will be able to get significant compression rates. So, our goal is to define a genetic algorithm that finds out the parameters of our sign predictor that achieve the best prediction performance. As studied in the literature, the parameters to be found by our genetic algorithm will be a) the neighbor set that defines the prediction context, and b) the sign values (sign patterns) of wavelet coefficient neighbor set with the correspondent sign prediction for current wavelet coefficient.

After running the genetic algorithm and configured the sign predictor, we will evaluate the impact of the sign coding module in the overall performance of an image wavelet encoder. In particular, we will use the LTW wavelet encoder [9] to determine the bit-rate savings for several test images.

The remainder of the paper is organized as follows: Section 2 describes our sign coding approximation. In Section 3, we show the results of the global encoder system (with sign coding stage) and compare it with SPIHT and JPEG 2000. Finally, in Section 4 some conclusions are drawn.

2 Wavelet Sign Prediction

Most wavelet image codecs do not consider the use of sign coding tools since the wavelet coefficients located at the high frequency subbands form a zero-mean process, and therefore equally likely positive as negative.

Schwartz, Zandi and Boliek were the first authors to consider sign coding, using one neighboring pixel in their context modeling algorithm [10]. The main idea behind this approach is to find correlations along and across edges.

The HL subbands of a multi-scale 2-D wavelet decomposition are formed from low-pass vertical filtering and high-pass horizontal filtering. The high-pass filtering detects vertical edges, thus the HL subbands contain mainly vertical edge information. Oppositely defined are the LH subbands that contain primarily horizontal edge information.

As Deever explained in [5], given a vertical edge in an HL subband, it is reasonable to expect that neighboring coefficients along the edge have the same sign as the coefficient being coded. This is because vertical correlation often remains very high along vertical edges in images. When a low-pass filter is applied along the image columns, it results in a series of similar rows, as elements in a row tend to be very similar to elements directly above or below due to the high vertical correlation. Subsequent high-pass filtering along similar rows is expected to yield vertically correlated transform coefficients.

It is also important to consider correlation across edges, being the nature of the correlation directly affected by the structure of the high pass filter. For Daubechies' 9/7 filters, wavelet coefficient signs are strongly negatively correlated across edges because this filter is very similar to a second derivative of a Gaussian, so, it is expected that wavelet coefficients will change sign as the edge is crossed. Although the discrete wavelet transform involves sub sampling, the sub sampled coefficients remain strongly negatively correlated across edges. In this manner, when a wavelet coefficient is optimally predicted as a function of its across-edge neighbors (e.g. left and right neighbors in HL subbands), the optimal prediction coefficients are negative, indicating an expected sign change. This conclusion is general for any wavelet with a shape similar to a second derivative of a Gaussian.

To estimate sign correlation in a practical way, we have applied a 6-level Dyadic Wavelet Transform decomposition of the source image and then a low quantization level to the resulting wavelet coefficients. As a first approach and taking into account that the sign neighborhood correlation depends on the subband type (HL,LH,HH) as Deever assesses in [5], we have used three different neighbors depending on the subband type. So, for HL subband, the neighbors used are N, NN and W. Taking into account symmetry, for the LH subband,

those neighbors are W, WW, and N. For the HH subband they are N, W, and NW, exploiting the correlation along and across the diagonal edges. This lead us to a maximum of 3^3 Neighbor Sign Patterns (NSP) for each subband type.

Table 1. Probability distribution of neighbor sign patterns (NSPs) of HL_6 subband (8×8 coefficients) in Lena image

C	N	NN	W	Occurrences	%Probability
+	+	+	+	13	20.31
+	+	+	-	8	12.50
-	-	-	+	8	12.50
-	+	+	+	6	9.38
-	-	+	+	6	9.38
Others				23	35.93

In Table 1 we show the NSP probability distribution for HL_6 subband (from the sixth decomposition level) of Lena test image. As shown, the probability that the current coefficient (C) is positive when its N, NN and W neighbors are also positive is around 20%. Besides, if the N and NN neighbors have the same sign and the W neighbor has the opposite sign, the current coefficient (C) has the opposite sign of its W neighbor with a probability of 25% as shown in rows two and three in Table 1. The visible sign neighborhood correlation suggest that the sign bits of wavelet coefficients are compressible. Using the previously mentioned neighborhood for each subband type, we have developed a genetic algorithm (GA) in order to find an accurate sign estimation.

2.1 Genetic Algorithm for Wavelet Sign Prediction

The goal of the desired genetic algorithm would be to find a table where for each Sign Neighborhood Pattern (V_k) we have a sign prediction ($S_{i,j}$) for coefficient $C_{i,j}$. There is no an univocal relationship between a neighbor sign combination, i.e not always for a same V_k pattern, $S_{i,j}$ is always positive or negative. However, it is possible that for a V_k pattern, $S_{i,j}$ is more probably to be positive or negative. But, the problem is still more complex, because a sign prediction for a neighbor sign pattern could fit well for an image and not for others. Therefore, the idea is to find suboptimal neighbor sign pattern predictions that better fit for a representative set of images.

The use of genetic algorithms to compress the sign of wavelet coefficients is twofold. First, when the number of neighbors used to analyze the sign correlation grows or when there is a great number of images to be used in the analysis, the search space is excessively wide. Second, it is not intuitive to find a way of combining the predictions obtained for several images.

In Fig. 2 we show the genetic algorithm pseudocode for sign prediction. First of all we define each individual, containing a sign prediction for each 3^3 NSP, then

each NSP sign prediction of each individual of the universe is randomly initialized as a positive or negative sign. Then, during evolution, sequences mate and mutate to generate new sequences in the population and best sequences are selected for survival on the basis of their fitness function. The mating of sequences is performed through crossover operator, where parents are randomly selected and its gens (NSPs) are mixed. The best two individuals, the ones that exhibit best prediction performance, are selected for survival. Individuals can also undergo mutation, where a sequence prediction is randomly modified. Finally, after performing the maximum iterations, the algorithm finishes, obtaining an optimal/suboptimal sign prediction for each NSP. We have performed the fitness evaluation over Lena and Barbara test images, because these images are representative for both low and high textured images respectively. Several parameters should be taken into account when training a genetic algorithm: The population size, the individuals initialization, the number of iterations performed, the mutation probability, the crossover point, the crossover method, the selection criteria of the best sequences to be selected for survival, etc. We have performed lots of tests varying these parameters to tune the genetic algorithm. The parameters used to obtain the sign prediction are: population size (100), individuals initialization (randomly), number of iterations (1000), mutation probability (0.001), crossover point (randomly) and crossover method (best two fitness individuals over four randomly selected parents).

```

Individual Structure{
    sign[NSP]; //Prediction array for each neighbor sign pattern combination
    fitness; //indicates the goodness of the individual
}Individual universe[NUM-POPULATION]; //Individual array

function SignPrediction (SubbandType, ImageFiles, mutation Probability)
//Initialization phase: sign[NSPs]= random(POSITIVE/NEGATIVE)
Initialize(universe, NUM-POPULATION, NSP);
//we evaluate each individual of the universe. For each image in ImageFiles
EvaluateFitness(SubbandType, ImageFiles, universe);
for i=0 to NUM-ITERATIONS
//Select the best two individuals from universe for survival.
    best = SelectBestIndividuals(2);
    //Crossover
    crossPoint=random(NSP);
    //randomly selects a father and a mother to mix its gens
    SelectFatherAndMother(random(NUM-POPULATION));
    universe = MergeFatherAndMother(crossPoint);
    Mutation(universe, mutation Probability);
    universe = universe + best;
    EvaluateFitness(SubbandType, ImageFiles, universe);
end
//Finally get the best individual.
best = SelectBestIndividuals(1);
end of function

```

Fig. 1. Genetic algorithm for sign prediction

After running the genetic algorithm for each subband type, we obtain an individual containing the prediction of the current coefficient sign ($\hat{S}C_{i,j}[k]$), for each NSP (k) of each subband type. So, what we are going to encode is the correctness of this prediction, i.e., a binary valued symbol from $\hat{S}C_{i,j}[k] \cdot SC_{i,j}$ (see Table 2). In order to compress this binary valued symbol, we use two contexts in the arithmetic encoder for each subband type, distributing all sign coding predictions from NSPs between them so as to minimize the zero order entropy of both contexts. The selection criterion is to isolate in one context those NSPs with the highest correctness prediction probability and highest number of occurrences derived from the probability distribution found in the previous analysis. The rest of them are grouped into the other context. However, there are certain NSPs with low correctness probability but with a great amount of occurrences, so we have to heuristically determine the convenience of including them in the first context or not.

Table 2. Sign prediction for HL subband in Lena image for some NSPs

NSP(k)	N	NN	W	Prediction ($\hat{S}C_{i,j}[k]$)
0	*	*	*	-
...				
13	+	+	+	+
14	+	+	-	+
...				
26	-	-	-	+

3 Performance Evaluation

In this section we analyze the behavior of the sign coding when implemented on LTW image encoder [9]. This new encoder implementation is called S-LTW. We will also compare the S-LTW encoder versus JPEG2000 (Jasper 1.701.0) and SPIHT (Spiht 8.01) in terms of R/D and coding delay. All encoders have been tested on an Intel PentiumM Dual Core 3.0 GHz with 2 Gbyte RAM memory.

In Table 3 we show the relative compression gain with respect to the original LTW due only to the sign coding capability for several test images. As we can see, the maximum sign compression gain is 17.35%. Furthermore, we show an estimation of the bit savings for SPIHT encoder.

In Figure 2 we show the R/D improvement when comparing original LTW versus JPEG2000/SPIHT and S-LTW versus JPEG2000/SPIHT. As shown, there is an increase in the PSNR difference between SPIHT and the new S-LTW encoder, and regarding JPEG2000, we can see that now S-LTW has a minor loss in PSNR than original LTW. Regarding coding delay, the use of a higher context modeling in the arithmetic encoder implies a higher computational cost. In order to compensate the coding speed loss, we have changed the arithmetic encoder stage by a fast arithmetic encoder [11]. As it can be seen in Table 4, S-LTW

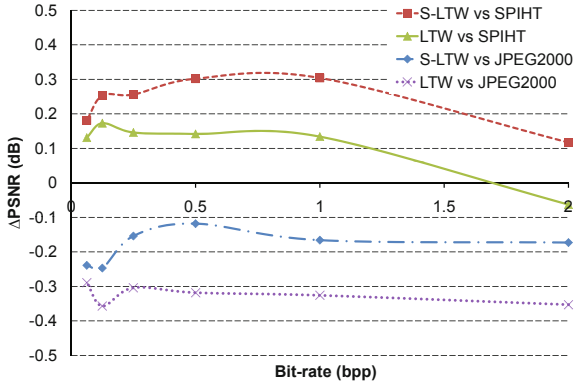


Fig. 2. PSNR-Gain for Bike image

Table 3. Sign compression performance at different bit-rates

Bit-rate (bpp)	S-LTW		SPIHT		%Gain
	#Significant Coefficients	#Bits Saved	#Significant Coefficients	#Bits Saved	
Barbara (512x512)					
1	45740	7936	54657	9482	17.35
0.5	22331	3648	27535	4499	16.34
0.25	10484	1520	13460	1951	14.50
0.125	4343	304	6016	421	7.00
Bike (2048x2560)					
1	855266	115200	1371280	184711	13.47
0.5	412212	64424	798202	124758	15.63
0.25	198943	30472	366927	56213	15.32
0.125	91767	11992	162990	21302	13.07

Table 4. Coding delay (seconds)

Bit-rate (bpp)	JPEG 2000	SPIHT	LTW Orig.	S-LTW
CODING Barbara (512x512)				
1	0.080	0.042	0.037	0.023
0.5	0.076	0.026	0.022	0.014
0.25	0.074	0.018	0.013	0.009
0.125	0.073	0.014	0.010	0.006
CODING Bike (2048x2560)				
1	2.623	0.920	0.647	0.430
0.5	2.543	0.521	0.381	0.259
0.25	2.507	0.323	0.224	0.162
0.125	2.518	0.221	0.158	0.117

encoder is 49% faster on average in the coding process than SPIHT encoder and 86% faster on average than JPEG2000. Furthermore, S-LTW encoder is even faster than the original LTW version which does not include the sign coding stage (1.5 times faster on average in the coding process).

4 Conclusions

We have presented a genetic algorithm that is able to find a good sign predictor of wavelet coefficient sign. So, by encoding the sign prediction result (success or failure) with an arithmetic encoder, the sign information will be highly compacted in the final bitstream. To prove our proposal we have implemented it over the LTW encoder. The new S-LTW proposed encoder has slightly better R/D performance (up to 0.25 dB), or in terms of bitstream, it is able to reduce it up to 17% for the same quality level. Regarding coding delay, the new image encoder is on average 2 times as fast as SPIHT in the coding process and 1.5 times as fast as original LTW.

References

1. ISO/IEC 15444-1: JPEG2000 image coding system (2000)
2. Shapiro, J.M.: A fast technique for identifying zerotrees in the EZW algorithm. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, vol. 3, pp. 1455–1458 (1996)
3. Wu, X.: High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression. In: Proc. of 31st Asilomar Conf. on Signals, Systems, and Computers, pp. 1378–1382 (1997)
4. Taubman, D.: High performance scalable image compression with EBCOT. IEEE Transactions on Image Processing 9(7), 1158–1170 (2000)
5. Deever, A., Hemami, S.S.: What's your sign?: Efficient sign coding for embedded wavelet image coding. In: Proc. IEEE Data Compression Conf., Snowbird, UT, pp. 273–282 (2000)
6. Holland, J.: Adaption in Natural and Artificial Systems. University of Michigan Press (1975)
7. Chabrier, S., Rosenberger, C., Emile, B., Laurent, a.H.: Optimization-based image segmentation by genetic algorithms. EURASIP Journal on Image and Video Processing 2008, 1–10 (2008)
8. Anam, S., Islam, M. S., Kashem, M., Islam, M., Islam, M., Islam, M.: Face recognition using genetic algorithm and back propagation neural network. In: International MultiConference of Engineers and Computer Scientists, Hong Kong (2009)
9. Oliver, J., Malumbres, M.P.: Low-complexity multiresolution image compression using wavelet lower trees. IEEE Transactions on Circuits and Systems for Video Technology 16(11), 1437–1444 (2006)
10. Schwartz, E.L., Z, A., Boliek, M.: CREW: Compression with reversible embedded wavelets. In: In Proc. SPIE, pp. 212–221 (1995)
11. Said, A.: Comparative analysis of arithmetic coding computational complexity. Technical report, Hewlett-Packard Laboratories HPL-2004-75 (2004)

Kernel-Based Object Tracking Using a Simple Fuzzy Color Histogram

Juan Villalba Espinosa, José María González Linares, Julián Ramos Cózar,
and Nicolás Guil Mata

University of Málaga, Dept. of Computer Architecture, Málaga, Spain
{juanvillalba,jgl,julian,nguill}@uma.es
<http://www.ac.uma.es/~vip>

Abstract. In this paper, we present an approach for kernel-based object tracking using the *HSV* color space as the feature space and fuzzy color histograms as feature vectors. These histograms are more robust to illumination changes and quantization errors than common histograms. To avoid a significant increase in the computational complexity, a simple fuzzy membership function is used. The efficiency of this approach is demonstrated using videos from the PETS database and comparing the results using the fuzzy color histogram and the common color histogram.

1 Introduction

Objects tracking is an important area of research in computer vision. This technique is useful in several situations where the object movement is important, as in surveillance applications, human-computer interaction, video communication and compression, etc.

Sports games analysis is another application scenario. By tracking the players, statistics can be automatically gathered, plays can be virtually recreated and game highlights can be easily found, thus allowing to enhance the game broadcast. Nevertheless, tracking player position is an especially difficult task, because during the game the player body shape, size and color change. Also the difficulty in tracking increases due to players irregular movement and changes in velocity. For instance, the player may rotate and change its shape when approaching or moving away from the camera, the lighting and background may change too, and even some players may be hidden by other objects or players.

A visual tracker is usually composed of a target representation and localization process and a filtering and data association procedure. In this paper we have developed a version of the kernel tracking algorithm by Comaniciu et al. [3] that is typically used in the target representation and localization process. In this algorithm an object feature vector is selected to represent the object, and the mean shift method is used to locate this feature vector in the next frame.

One of the main contributions presented in this paper is the use of the *HSV* color space as the feature space, and a fuzzy color histogram as the feature vector to increase robustness to changes in lighting.

Fuzzy color histograms (*FCH*) have been successfully used in image retrieval [5] and objects tracking applications [7,6]. This type of histograms are more robust to illumination changes and quantization errors, but are computationally intensive. A *FCH* is built using a fuzzy-set membership function that determines the degree of association of the color of a pixel to every bin in the *FCH*. In the work presented in this paper we use a simple triangular-shaped fuzzy membership function. Thus, small variations due to lighting and shading artifacts are compensated with a low computational complexity increase.

The rest of the paper is organized as follows: in section 2 we describe the method for kernel-based object tracking and in section 3 we present a modification that consists in using fuzzy color histograms. In section 4, we present some experimental results and finally conclusions are drawn in section 5 of the paper.

2 Kernel-Based Object Tracking

In the kernel-based object tracking approach proposed by Comanicu et al. [3], a color histogram is calculated to represent the target model and the target candidate in the next frame. An isotropic kernel, with a convex and monotonic decreasing kernel profile, is used to regularize the target representation. In particular, it is used a kernel with an Epanechnikov profile that assigns smaller weights to the pixels farther from the centre of the region that is being tracked. With this kind of kernel the histogram robustness against occlusion and background interference increases.

The target is localized in the next frame using the mean-shift algorithm. In this approach, with an *m*-interval histogram, the target model is represented as:

$$\hat{q} = \{\hat{q}_u\}_{u=1\dots m}, \quad \hat{q}_u = C \sum_{i=1}^n b_i k(\|\mathbf{X}_i^*\|^2) \delta[b(\mathbf{X}_i^*) - u], \quad (1)$$

where $\{\mathbf{X}_i^*\}_{i=1\dots n}$ are the normalized pixel locations in the region defined by the target model centered at the spatial location *O*. Also, δ is the Kronecker delta function, b_i is a function that associates to a pixel the index of its bin in the quantized feature space, k is the Epanechnikov profile and C is a normalization constant derived from the condition $\sum_{u=1}^m \hat{q}_u = 1$:

$$C = \frac{1}{\sum_{i=1}^n k(\|\mathbf{X}_i^*\|^2)} \quad (2)$$

The target candidate is represented as follows:

$$\hat{p}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1\dots m}, \quad \hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} b_i k\left(\left\|\frac{\mathbf{y} - \mathbf{X}_i}{h}\right\|^2\right) \delta[b(\mathbf{X}_i) - u], \quad (3)$$

where $\{\mathbf{X}_i\}_{i=1\dots n_h}$ are the normalized pixel locations of the target candidate, centered at *y* in the current frame, with $C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - \mathbf{X}_i}{h}\right\|^2\right)}$.

The normalization is inherited from the frame containing the target model using the same kernel profile, but with bandwidth h . This bandwidth h is used to make the algorithm invariant to changes in the object scale. To adapt the target to this changes, the localization is computed three times using the same bandwidth and with bandwidth $h \pm 0.1h$. The best localization result is selected and the bandwidth adapted to reflect the change in scale.

In order to measure how similar is one histogram to the other, the Bhattacharyya coefficient can be used:

$$\hat{\rho}(\mathbf{y}) \equiv \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{y})\hat{q}_u} \tag{4}$$

and a metric distance can be defined as

$$d(\mathbf{y}) = \sqrt{1 - \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}]} \tag{5}$$

This distance gets closer to 0 as the target model and candidate histograms are more similar. The goal is to minimize the distance as a function of y . The localization process starts in the target model position in the previous frame and looks for the position where the distance is minimum in the neighborhood. As the distance function is smoothed thank to the use of the kernel, the process uses gradient information provided by the mean-shift vector.

Minimizing the distance is equivalent to maximizing the Bhattacharyya coefficient $\hat{\rho}(y)$. Firstly, it is computed the target candidate histogram $\{\hat{p}_u(\hat{y}_0)\}_{u=1\dots m}$ at \hat{y}_0 in the current frame. Using Taylor expansion around $\hat{p}_u(\hat{y}_0)$ the following approximation to the Bhattacharyya coefficient is obtained:

$$\rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{\mathbf{y}}_0)\hat{q}_u} + \frac{C_h}{2} \sum_{u=1}^{n_h} w_i k \left(\left\| \frac{\mathbf{y} - \mathbf{X}_i}{h} \right\|^2 \right) \tag{6}$$

with

$$w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{y}_0)}} \delta[b(\mathbf{X}_i) - u] \tag{7}$$

To minimize the distance $d(\mathbf{y}) = \sqrt{1 - \rho[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}]}$, the second term of (6) must be maximized. This second term represents the density estimation calculated with the kernel profile $k(x)$ at y , in the current frame and with the data weighted by w_i . The mode of this density in the local neighborhood is the maximum searched, which may be found using the mean-shift procedure.

According to the following equation, the kernel moves from the current position \hat{y}_0 to the new position \hat{y}_1 until converging at the accurate target position.

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} \mathbf{X}_i w_i g \left(\left\| \frac{\hat{\mathbf{y}}_0 - \mathbf{X}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n_h} w_i g \left(\left\| \frac{\hat{\mathbf{y}}_0 - \mathbf{X}_i}{h} \right\|^2 \right)}. \tag{8}$$

3 Tracking Using Fuzzy Color Histograms

The color histogram used in [3] is computed in the *RGB* color space. This space is very sensitive to changes in scene lighting, thus we have selected the *HSV* space that is less susceptible to variations in shades of similar colors. The hue component of the *HSV* space has been successfully used for face tracking [1], but in this work we use a fuzzy *HSV* color histogram representation using 16 bins for each channel (hue, saturation and value).

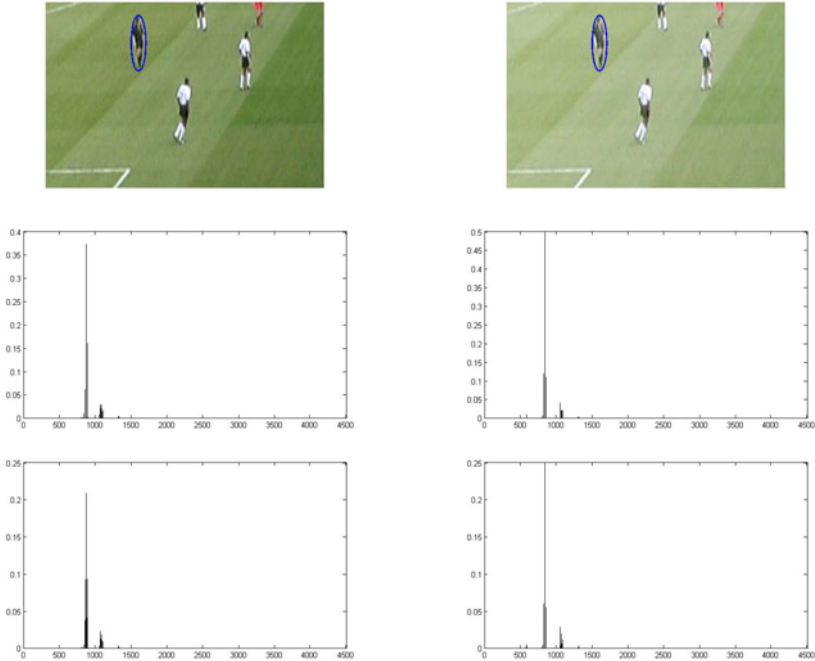


Fig. 1. Comparison of common and fuzzy color histograms. First row: image on the right is a brighter version of image on the left. Second row: common color histograms of patches inside blue bounding boxes in previous images. Third row: *FCH* versions.

Other authors have previously used *FCH* with kernel tracking. In [6] the most representative fuzzy clusters are extracted from a training data set that are used to build the *FCH* of the target using just a few bins. This *FCH* represents the target model better than the common *RGB* histogram, but computation of the set of fuzzy clusters is expensive and results show an improvement only in the number of mean-shift iterations. In [7] a training image is transformed to the *CIELAB* color space and a fuzzy c-means algorithm is used to compute a membership matrix that can transform a common color histogram in a *FCH* with a simple matrix multiplication. The cost of computing this membership

matrix is very high, thus this method is not suitable for tracking objects in different scenarios or with changing background.

In this work we use a simple triangular-shaped membership function, thus the *FCH* can be efficiently computed without the need of a training data set:

$$b_i = \begin{cases} \frac{b_r - |x - b_c|}{b_r} & \text{if } |x - b_c| < b_r \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where x is the value of the pixel being added to the histogram, b_c and b_r are the centre point and range respectively of bin b , and b_i is the resulting bin increment (0→1) for bin b . In order to extend the fuzzy increment to multiple dimensions, we have to combine the bin increment for each bin: $b_i = \prod_{n=1}^N b_i(n)$.

In Figure 1 it is compared the common *HSV* color histogram with the *FCH* proposed in this work. First row shows two almost identical images: the image on the right is a gamma transformed version of the image on the left using a value of 0.5. The bounding box over the referee shows the area used to compute the weighted histogram in (10). The weighted common color histograms for both images are displayed in the second row while their fuzzy counterparts are shown in the third row.

Although there is no significant differences between both images, both histograms are dissimilar because the color distribution has changed. Nevertheless, if the distance metric in (5) is computed between the histograms in the second row of Figure 1, a value of 0.1187 is obtained while the distance between the *FCH* in the third row is 0.073. That is, the *FCH* is more robust to lighting changes than the common color histogram

4 Experimental Results

A number of experiments have been done to test the performance of the kernel tracking algorithm proposed in this work. To that end we have run the algorithm on a video of PETS'2003 dataset that consists of football players moving around a pitch. This dataset provides a total of 59 objects and a list of ground truth tracks.

In a real tracking application the kernel tracking algorithm is a part of the target representation and localization procedure. Problems like objects detection, total occlusions or very fast movement cannot be solved with this algorithm and other methods like background segmentation, Kalman filters, multiple hypothesis testing, etc. are combined with kernel tracking to obtain a robust track estimation. To evaluate the performance of our kernel tracking algorithm, object detection is done manually using the first instance of every ground truth track. The object bounding box and frame number of its first appearance is passed to the algorithm to follow the track and to obtain a so called system track.

The tracking results were compared with the given ground truth by three different metrics. The first one is the intersection area between the system bounding box and the ground truth bounding box for each frame. For that metric we have

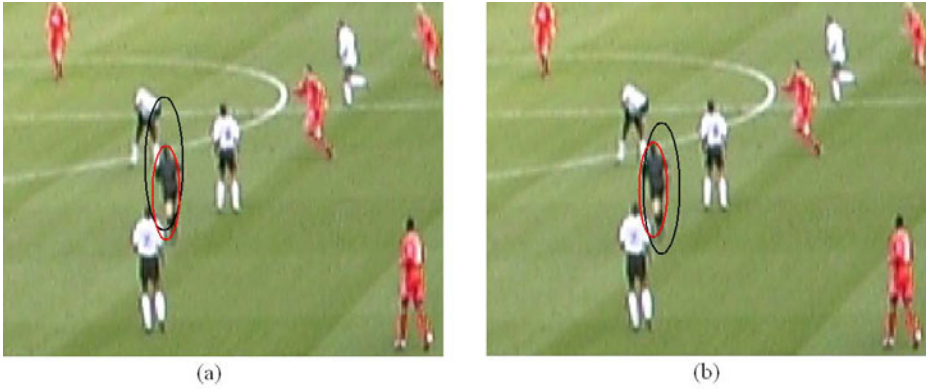


Fig. 2. Comparison of the tracking result (big black ellipse) of both versions with the ground truth bounding box (small red ellipse). Area intersection and centroid distance are worse in the normal version (a) than the fuzzy version (b).

used the Video Performance Evaluation Resource (ViPER) [4] that provides a set of tools for ground truth generation, metrics for evaluation, and visualization of video analysis results. We have used that evaluation tool to obtain the bounding box area precision for each object. The final result is given by the average bounding box area precision among every object.

The second metric used is based on proximity. We have measured the average euclidean distance between the system bounding box centroid and the ground truth bounding box centroid for each object [2] as follows:

$$TDuration \sum_i Dist(p_i^{GT}, p_i^{Sys}) \quad (10)$$

where p_i^{GT} is the centroid of the ground truth track at the i^{th} ground truth frame, p_i^{Sys} is the centroid of the system track, and $TDuration$ is the duration, in frames, of the object track. Finally, we have taken the average distance among every object.

The last metric used is related with the internal performance of the mean-shift algorithm by measuring the average number of iterations for each tracked object.

The final results are summarized in Table 1. It must be noticed that 29 of the 59 objects suffer a large occlusion at some point that should be detected using another algorithm, thus we have collected two results for each evaluation, one considering the full ground truth and another discarding the values after an occlusion (between parenthesis in the table). The system has been evaluated using the normal color version and the fuzzy color version. As it can be observed the fuzzy version is better than the normal one in every test. In Figure 2, where the referee is being tracked, it can be seen how the target candidate found by the

Table 1. Summary of results for different metrics

Metric	Normal version	Fuzzy version	Improvement
Box Area Precision	0.43 (0.49)	0.47 (0.52)	9.30% (6.12%)
Centroid Distance	48 (38)	39 (30)	23.07% (26.67%)
Number of Iterations	3.4	3.1	9.67%

fuzzy version is closer to the ground truth. It is worth mentioning that the fuzzy version needs less number of iterations than the normal one to obtain better tracking results.

5 Conclusions

In this work it has been proposed the use of the *HSV* color space as the feature space for the kernel-based object tracking algorithm, and a simple triangular shaped fuzzy membership function to obtain a fuzzy color histogram.

The *FCH* is more robust than the common color histogram to illumination changes and quantization errors. Several experiments have been conducted to demonstrate that this modification produces better tracking results while keeping the computational complexity low.

References

1. Bradski, G.: Computer Vision Face Tracking For Use in a Perceptual User Interface. Intel (1998)
2. Brown, L., Senior, A., Tian, Y., Connell, J., Hampapur, A., Shu, C., Merkl, H., Lu, M.: Performance evaluation of surveillance systems under varying conditions. In: Proceedings of IEEE PETS Workshop, pp. 1–8 (2005)
3. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25(5), 564–577 (2003)
4. Doermann, D., Mihalcik, D.: Tools and Techniques for Video Performance Evaluation. In: Proceedings of the International Conference on Pattern Recognition (ICPR 2000), Barcelona, pp. 4167–4170 (September 2000)
5. Ju Han, J., Ma, K.-K.: Fuzzy Color Histogram and Its Use in Color Image Retrieval. *IEEE Trans. on Image Processing* 11(8), 944–952 (2002)
6. Ju, M.-Y., Ouyang, C.-S., Chang, H.-S.: Mean shift tracking using fuzzy color histogram. In: International Conference on Machine Learning and Cybernetics (ICMLC 2010), vol. 6, pp. 2904–2908 (2010)
7. Wang, Y., Pan, Q., Zhao, C., Cheng, Y.: Fuzzy Color Histogram Based Kernel Tracking under Varying Illumination. In: Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), vol. 4, pp. 235–239 (2007)

Computational Intelligence in Multimedia Processing

Nicolás Guil¹, Julio C. Bregáins², and Adriana Dapena²

¹ Departamento de Arquitectura de
Computadores. Universidad de Málaga.
Boulevard Louis Pasteur.
Campus de Teatinos, 29071, Málaga, Spain
Tel.: +34-952-133327
`nguil@uma.es`

² Departamento de Electrónica e Sistemas.
Universidade da Coruña
Campus de Elviña s/n
15071, A Coruña, Spain
Tel.: ++34-981-167000
`{jbregains,adriana}@udc.es`

Abstract. Computational intelligence (CI) is a well-established paradigm that incorporates characteristics of biological computers (brains) to perform a variety of tasks that are difficult or impossible to do with conventional computers. This paper reviews some of the applications of CI in multimedia processing, including shot detection in video, logotype detection, video copy detection and retrieval, and faces coding in video sequences.

1 Introduction

Multimedia processing is a very important research domain with a broad range of applications that cover techniques like object-based representation and coding, segmentation and tracking, patten detection and recognition, multi-modal signals fusion, content-based indexing, subject-based retrieval, etc. [1]. The utilization of Computational Intelligence (CI) in multimedia processing has allowed the inclusion of more sophisticated methods to solve these applications than those used in traditional computation. These methods are based on characteristics of biological computers (i.e. brains) and include techniques such as rough sets, neural networks, fuzzy logic, swarm intelligence, reinforcement learning and evolutionary computation. This paper reviews some applications of CI in multimedia processing. Section 2 focuses on shot detection, Section 3 presents logotype detection and its applications, Section 4 presents the problem of video copy detection and retrieval and, finally, Section 5 focuses on face recognition and its utilization in face coding in video sequences.

2 Shot Detection in Video

As video demand continues growing, fast and reliable methods for indexing purposes are needed. One of the most useful techniques in video indexing is shot detection, where a shot is defined as a sequence of frames captured from a single camera operation. Shot detection is performed by means of shot transition detection algorithms. Two different types of transitions are used to split a video into shots:

- Abrupt transitions, also referred as cuts or straight cuts, occur when a sudden change from one shot to the next one is performed in just one frame.
- Gradual transitions use several frames to link two shots together, referred by other authors as optical cuts [2].

Depending on how the shots are mixed up in the transition, there are many different types of gradual transitions. Dissolves are the most common. Fades and wipes are also frequently used. What is most important about gradual transitions is that they are often used to establish some kind of semantic information in the video. For example, dissolves have been widely used to perform scene change in video edition, where a scene is a set of shots closely related in terms of place and time. While abrupt transitions detection is a relatively easy task [3], gradual transitions detection is still an open issue, as the amount of false positives reported by the algorithms is very high for certain sequences. The main problem in gradual transitions detection is that camera operation (pan, tilt, swing, zoom, etc.) originates similar patterns to those generated by gradual transitions [4]. Thus, a method to estimate global motion in video is needed in order to discard false positives induced by camera operations [5].

The first stage of shot transition detection algorithms is the extraction of characteristics from the video streams. One or more metrics are then used to compute several parameters from the characteristics. These metrics can be based on pixel luminance, contour information, block tracking, etc. Although most of the proposed methods make use of only one metric, using several of them is recommended, as drawbacks from one metric could be compensated by the others [6], as long as the used metrics rely on different video characteristics. The computed parameters are then used to determine the occurrence of a transition. Here, data driven methods address the problem from the data analysis point of view.

On the other hand, model driven methods, based on mathematical models of video data, allow a systematic analysis of the problem and the use of domain specific constraints, which helps to improve the efficiency [7], [8]. Other authors use non-deterministic classifiers to study the computed parameters in order to perform pattern recognition, as transitions generally result in a characteristic pattern in the parameters. Using a non-deterministic classifier makes unnecessary a specifically designed pattern recognition method, which usually needs several parameters to be tuned. Also, by using a supervised classification scheme, the system is able to learn the patterns generated by different types of gradual transitions. Some shot transition detection algorithms using neural

networks [9][10], Hidden Markov Models [11] or Support Vector Machines [12] have been proposed.

Neural networks have been proposed to detect abrupt shot transitions using four parameters extracted from video [13]. The parameters were computed using an evolution of the classical sliding window thresholding algorithm. Jun [10] proposed a neural network based algorithm for abrupt transitions detection using only I frames from the MPEG streams. Histograms and pixel-by-pixel comparison between frames are performed. Nevertheless, this last metric is very sensitive to motion. Moreover, only news videos were used to test the proposed method. A proposal which is highly dependent on the MPEG codifier was stated by Mallikarjuna [9], reporting results difficult to compare with other proposals.

Another work, [14], presents a reliable real time approach to temporal video segmentation in MPEG compressed video. A scheme based on luminance and contour information is proposed.

3 Logotype Detection and Learning

Logotypes convey information that can be crucial to infer semantics implicit in broadcasted videos. It is a common practice for broadcasters and specific TV programs, e.g., news, talk shows, advertisements, etc., to superimpose a specific logotype on the broadcasted material. Usually such logotypes refer to the actual video content or the content creator and, as such, they can be used to support automatic semantic-based video annotation. Logotypes extracted from broadcasted videos can be annotated in a database by indicating their shape and shots or scenes where they appear. Then, video retrieval tools can be used to search for a particular TV program or to group different pieces of video material with related contents. As a consequence, accurate logotype detection can be efficiently exploited for semantic-based video classification, video retrieval, aggregation and summarization.

Many works related to logo detection in document analysis have been reported in the literature [15]. However, few approaches consider logo detection in conventional video sequences. Logo detection techniques have been used to differentiate advertisements from TV programs in [16]. This approach assumes that a logo exists if an area with stable contours can be found in the image. The authors claim that their approach does not require any supervised training and can be easily used for any type of logos without human interaction. In [17], a neural network is trained using two sets of logo and non-logo examples to detect a transparent logo. It obtains a good detection rate at the expense of a rather large training set. In [18], color outliers are used to detect pixels different from the background. No temporal information is used, thus many false detections can arise.

The work presented in [19] shows an application for logo removing. The logotype is detected by exploiting frame differences in video sequence. This procedure fails in video with low motion activity. In this case, however, the authors propose to use a logo database and to search for them using a Bayesian approach. The

detection accuracy is improved by assuming that the probability of the logos appearing in the four corners of the video frames is higher than in the center. This prior knowledge is combined with a neural network-based classifier. Other works argue correctly that logos can provide a helpful visual cue for finding related news stories. Usually, a logo is defined as the small graphic or picture that appears behind the anchor person on the screen. In [20], it is assumed that each broadcasted TV channel contains some representative semantic objects, including the channel logotype, that is displayed only during news programs. Channel logotype detection and tracking is performed to automatically classify news events in conventional broadcasting material. The approach relies on the use of logotype models stored in a database. Information about logo position and scale helps to identify the channel and the type of news. In [21], the detection of logos is used to mark news stories, as an alternative approach for tracking them. Here, from each logo, three sets of 2D Haar coefficients are computed (one for each of the RGB channels). The logos feature vector is formed by selecting the coefficients representing the overall averages and the low frequency coefficients of the three color channels. However, in all these works, logos must be known a priori.

There are related works that try to identify known brand logotypes in video data. In [22], certain variability in the logotype appearance must be allowed. In practice, due to the high computational cost of this method, only a few logotypes can be identified simultaneously. Similarly, in [23], a search for specific instances of brand logos is performed. Logo detection is achieved by exploiting homogeneously colored regions surrounding large intensity frame differences.

In [24], a framework to support semantic based video classification and annotation is described. The backbone of the proposed framework is a technique for logotype extraction and recognition which is able to continuously update a metadata base in an iterative process where new logos are learned as they are detected and classified.

4 Video Copy Detection/Retrieval

Video copy detection carries out the comparison of a query video sequence with a target video sequence in order to establish if a copy of the query video is present in the target one. This application is a useful tool for both digital content management and protection of intellectual copyrights (IPR). Thus, on the one hand, the localization of video copies, e.g. an advertisement, can be used to catalog broadcasted material in a multimedia database, to check whether the material was broadcasted at the suitable time or if its duration was correct. On the other hand, the video copy can be compared with original video to check if any violation of IPR has happened. In this sense, video copy detection is an alternative to watermarking techniques when no marks can be inserted in the video copies.

Signature selection is a key point to develop a specific approach to video comparison. Spatial and/or temporal information can be used to generate the

video signature based on global or local descriptors [25]. Global descriptors are, in general, efficient to compute, compact in storage, but insufficiently accurate in terms of retrieval quality. Local descriptors present more invariance to aggressive transformations, such as PiP (Picture in Picture), cropping, insertions of patterns, change of gamma, etc.

In addition, signature content can be determined for every video frame or it can be given for a summary in the form of a set of key frames which is calculated through a temporal clustering process. These methods require less computational resources during the comparison process and generate shorter signatures. Kim and Park present in [26] an approach to video sequence matching by calculating the similarity between sets of key frames. Key frames are extracted using the cumulative directed divergence. Then, the similarity between videos is calculated by employing the modified Hausdorff distance between sets of key frames. Cheung and Zakhor [27] propose a measure for video similarity calculation where the percentage of clusters of similar frames shared between two video sequences is obtained. Previously, each video is summarized by selecting frames that are similar to a set of predefined feature vectors common to all video sequences. Guil et al. [28] divide the query video in clusters and extract a representative key frame for each cluster. Features from these key frames constitute the signature of the video. Then, they perform a dense comparison between the signature of the query video and every frame of the target video using relaxed distance constraints to speed-up the search process.

Comparison between query and target video sequences is carried out using some kind of similarity search. If the database of target videos is large, some efficient indexing process needs to be performed. Some research has been carried out on databases of more than ten thousand hours of videos, where the indexes have to be stored in hard disks. In this context, a trade-off between computing time and robustness must be considered. In this research line, Joly et al. have showed that trading quality for time during the search is highly profitable when statistical filtering is implemented, even when the size of the DB becomes very large [29]. When databases are not so large other authors [30] [31] address the problem of searching for repeated subsequences by hash tables. However, if both strong temporal and spatial transformation are applied to copied videos, more robust indexing techniques are needed. Thus, Douze et al., [32], use two refinements in order to carry out the query. Firstly, the feature descriptor is used as a quantization index which is augmented with a binary structure. Then, partial geometry consistency is calculated between the matching frames. This method obtains excellent results for the TRECVID 2008 copy detection task [33].

5 Coding Faces in Video Sequences

A facial recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. Moghaddam and Pentland introduced the use of Principal Component Analysis (PCA) for face recognition from still images [34] and further explores

in [35] [36]. The main idea of using PCA for face recognition is to express the large 1-D vector of pixels constructed from 2-D facial image into the compact principal components of the feature space. This can be called eigenspace projection. Eigenspace is calculated by identifying the eigenvectors of the covariance matrix derived from a set of facial images (vectors) [37].

In applications like video-mobile the information to transmit is a sequence of the same face (maybe in different poses). To encode this kind of video sequences, Piqué and Torres [38] represent the frames using PCA and adapt the eigenspace taking into account the different poses, expressions and lighting conditions of the faces. The idea is to predict the actual frame by calculating their projections onto the eigenspace calculated from previous frames. The coefficients are therefore coded and transmitted. Full frames are only coded when a poor representation is obtained and, in this case, the eigenspace is updated. The quality of the recovered image is obtained by using a metric based on the peak signal to noise rate but it is also possible to include visual information. In [39], it has been introduced the idea of predicting the faces in two directions which allows us to define three kinds of frames (I, P, and B) similar to those used in MPEG.

Acknowledgements

This work was supported by Xunta de Galicia (grants numbers 10TIC105003PR and 09TIC008105PR), Junta de Andalucía (grants number P07-TIC-02800) and Ministerio de Ciencia e Innovación of Spain (grants numbers TEC2010-19545-C04-01 and CSD2008-00010).

References

1. Hassanien, A.-E., Abrahan, A., Kacprzyk, J., Peters, J.: Computational Intelligence in Multimedia Processing: Recent Advances. In: Computational Intelligence in Multimedia Processing: Foundation and Trends, pp. 3–49. Springer, Heidelberg (2008)
2. Patel, N.V., Sethi, I.K.: Video shot detection and characterization for video databases. *Pattern Recognition* 30(4), 583–592 (1997)
3. Lupatini, G., Saraceno, C., Leonardi, R.: Scene break detection: a comparison. In: Eighth International Workshop on Research Issues in Data Engineering, 1998. Continuous-Media Databases and Applications, pp. 34–41 (February 1998)
4. Lu, H., Zhang, Y., Yao, Y.: Robust gradual scene change detection. In: Proc. of International Conference on Image Processing (ICIP), vol. 3, pp. 304–308 (1999)
5. Zhang, H., Kankanhalli, A., Smoliar, S.W.: Automatic partitioning of full-motion video. *Multimedia Systems* 1, 10–28 (1993)
6. Yusoff, Y., Kittler, J., Christmas, W., Surrey, G.: Combining multiple experts for classifying shot changes in video sequences. In: IEEE International Conference on Multimedia Computing and Systems, pp. 700–704. IEEE, Los Alamitos (1998)
7. Aigrain, P., Joly, P.: The automatic real-time analysis of film editing and transition effects and its applications. *Computers and Graphics* 18(1), 93–103 (1994)
8. Hampapur, A., Jain, R., Weymouth, T.E.: Production model based digital video segmentation. *Multimedia Tools and Applications* 1, 9–46 (1995)

9. Mallikarjuna, R., Ramakrishnan, K., Balakrishnan, N., Srinivasan, S.: Neural net based scene change detection for video classification. In: IEEE 3rd Workshop on Multimedia Signal Processing, pp. 247–252 (1999)
10. Jung, S., Park, S.: An automatic cut detection algorithm using median filter and neural networks. In: International Technical Conference on Circuits/Systems, pp. 1049–1052 (2000)
11. Boreczky, J., Wilcox, L.: A hidden markov model framework for video segmentation using audio and image features. In: Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 6, pp. 3741–3744 (May 1998)
12. Qi, Y., Hauptmann, A., Liu, T.: Supervised classification for video shot segmentation. In: IEEE International Conference on Multimedia and Expo., vol. 2, pp. 689–692 (2003)
13. Zhang, Y.J., Lu, H.B.: A hierarchical organization scheme for video data. *Pattern Recognition* 35(11), 2381–2387 (2002)
14. Saez, E., Benavides, J., Guil, N.: Combining luminance and edge based metrics for robust temporal video segmentation. In: International Conference on Image Processing (ICIP), vol. 4, pp. 2231–2234 (October 2004)
15. Doermann, D.: The indexing and retrieval of document images: A survey. *Computer Vision and Image Understanding* 70(3), 287–298 (1998)
16. Albiol, A., Ch, M., Albiol, F., Torres, L.: Detection of TV commercials. In: Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 541–544 (May 2004)
17. Duffner, S., Garcia, C.: A neural scheme for robust detection of transparent logos in TV programs. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4132, pp. 14–23. Springer, Heidelberg (2006)
18. Ekin, A., Braspenning, R.: Spatial detection of TV channel logos as outliers from the content. In: Apostolopoulos, J.G., Said, A. (eds.) Visual Communications and Image Processing 2006, vol. 6077(1), SPIE, CA (2006)
19. Yan, W.-Q., Wang, J., Kankanhalli, M.: Automatic video logo detection and removal. *Multimedia Systems* 10, 379–391 (2005)
20. Günsel, B., Ferman, A.M., Tekalp, A.M.: Temporal video segmentation using unsupervised clustering and semantic object tracking. *Journal of Electronic Imaging* 7(3), 592–604 (1998)
21. Duygulu, P., Pan, J.-Y., Forsyth, D.A.: Towards auto-documentary: tracking the evolution of news stories. In: Proc. of the 12th Annual ACM International Conference on Multimedia, pp. 820–827. ACM, New York (2004)
22. Hall, D., Pelisson, F., Riff, O., Crowley, L.: Brand identification using gaussian derivative histograms. *Mach. Vision Appl.* 16, 41–46 (2004)
23. den Hollander, R.J.M., Hanjalic, A.: Logo recognition in video stills by string matching. In: Proc. of International Conference on Image Processing (ICIP), pp. 517–520 (2003)
24. Cózar, J., Guil, N., González-Linares, J., Zapata, E., Izquierdo, E.: Logotype detection to support semantic-based video annotation. *Signal Processing: Image Communication* 22(7-8), 669–679 (2007)
25. Law-To, J., Chen, L., Joly, A., Laptev, I., Buisson, O., Gouet-Brunet, V., Bouje-maa, N., Stentiford, F.: Video copy detection: a comparative study. In: Proc. of the 6th ACM international conference on Image and video retrieval ser. CIVR 2007, pp. 371–378. ACM, New York (2007)
26. Kim, S.H., Park, R.-H.: An efficient algorithm for video sequence matching using the modified hausdorff distance and the directed divergence. *IEEE Transactions on Circuits and Systems for Video Technology* 12(7), 592–596 (2002)

27. Cheung, S.-S., Zakhor, A.: Fast similarity search and clustering of video sequences on the world-wide-web. *IEEE Transactions on Multimedia* 7(3), 524–537 (2005)
28. Guil, N., González-Linares, J., Cózar, J., Zapata, E.: A clustering technique for video copy detection. In: Martí, J., Benedí, J.M., Mendonça, A.M., Serrat, J. (eds.) *IbPRIA 2007*. LNCS, vol. 4477, pp. 451–458. Springer, Heidelberg (2007)
29. Joly, A., Frelicot, C., Buisson, O.: Content-based video copy detection in large databases: a local fingerprints statistical similarity search approach. In: *IEEE International Conference on Image Processing, ICIP 2005*, vol. 1, pp. 505–508 (September 2005)
30. Naturel, X., Gros, P.: Detecting repeats for video structuring. *Multimedia Tools Appl.* 38, 233–252 (2008)
31. Döhring, I., Lienhart, R.: Mining tv broadcasts for recurring video sequences. In: *Proc. of the ACM International Conference on Image and Video Retrieval*, vol. 8, pp.28:1–28:8. ACM, New York (2009)
32. Douze, M., Jegou, H., Schmid, C.: An image-based approach to video copy detection with spatio-temporal post-filtering. *IEEE Transactions on Multimedia* 12(4), 257–266 (2010)
33. Smeaton, A.F., Over, P., Kraaij, W.: High-Level Feature Detection from Video in TRECVID: a 5-Year Retrospective of Achievements. In: Divakaran, A. (ed.) *Multimedia Content Analysis, Theory and Applications*, pp. 151–174. Springer, Heidelberg (2009)
34. Moghaddam, B., Pentland, A.: Probabilistic visual learning for object representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19(7), 696–710 (1997)
35. Chandrasekaran, S., Manjunath, B., Wan, Y., Winkler, J., Zhang, H.: An eigenspace update algorithm for image analysis. *Graphical Models and Image Processing* 59(5), 321–332 (1997)
36. Liu, X., Chen, T., Thornton, S.M.: Eigenspace updating for non-stationary process and its application to face recognition. *Pattern Recognition, Special issue on kernel and subspace methods for computed vision, 1945–1959* (2002)
37. Kyungnam, K.: Face recognition using principle component. In: *International Conference on Computer Vision and Pattern Recognition*, pp. 586–591 (1996)
38. Piqué, R., Torres, L.: Efficient face coding in video sequences combining adaptive principal component analysis and a hybrid codec approach. In: *Proc. International Conf. on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 629–632 (2003)
39. Pérez-Iglesias, H., Dapena, A., Castedo, L.: A novel video coding scheme based on principal component analysis. In: *Proc. Machine Learning for Signal Processing* (2005)

Isometric Coding of Spiking Haptic Signals by Peripheral Somatosensory Neurons

Romain Brasselet¹, Roland S. Johansson², and Angelo Arleo^{1,*}

¹ CNRS - UPMC Univ Paris 6, UMR 7102, F75005, Paris, France
Tel.: +33 1 44 27 27 80
angelo.arleo@upmc.fr

² UMEA Univ, Dept Integr Medical Biology SE-901 87 Umeå, Sweden

Abstract. We study how primary tactile afferents encode relevant contact features to mediate early processing of haptic information. In this paper, we apply metrical information theory to perform temporal decoding of human microneurography data. First, we enrich the theory by deriving a novel spike train metrics inspired by neuronal computation. This spike train metrics can be interpreted biologically and its behaviour is not influenced by spontaneous activity, which decreases the ability of other spike metrics to separate input patterns. Second, we employ our metrical information tools to demonstrate that primary spiking signals allow a putative neural decoder to go beyond stimulus discrimination. They transmit information about geometrical properties of the input space. We show that first-spike latencies are enough to guarantee maximum information transmission of tactile stimuli. However, entire primary spike trains are necessary to encode isometric representations of the stimulus space, a likely basis for generalisation in haptic perception.

Keywords: Temporal coding, spike-train metrics, information theory, isometric mapping, microneurography recordings, mechanoreceptors.

1 Introduction

We study neurotransmission at the early stages of the somatosensory pathway and focus on encoding/decoding of primary afferent tactile signals. We previously extended Shannon information theory to account for the metrical properties of spike time patterns when assessing neurotransmission reliability [2,3]. Here we stress the importance for metrical information analysis to reflect the properties of an actual neural decoder —as opposed to an ideal observer— when studying the statistical dependence between stimulus and neural response. To do so, we derive a novel spike train metrics based on a parametric non-linear distance inspired by the probabilistic behaviour of the Spike-Response Model [5]. We apply the resultant metrical information theory to a data set of human microneurography recordings [6]. We perform a temporal decoding analysis of the responses of fingertip mechanoreceptors to a set of tactile stimulations. The main rationale beneath this work is to go beyond input discriminability and to capture possible

* Corresponding author.

geometrical regularities of the input space encoded by primary afferent spiking signals. We suggest that the ability to extract information about the geometrical organisation of the input depends on the characteristics of the neural decoder. We show that a properly tuned decoder can detect isometric mapping suitable for generalisation in haptic perception.

2 Methods

2.1 Metrical Information Theory

We define the metrical entropy $H^*(R)$ on the response space R as:

$$H^*(R) = - \sum_{r \in R} p(r) \log_2 \left(\sum_{r' \in R} p(r') \phi(r, r') \right) \tag{1}$$

where the similarity measure $\phi(r, r')$ can be any real function with values in $[0, 1]$. The conditional entropy given a set of stimuli S can then be taken as:

$$H^*(R|S) = \sum_{s \in S} p(s) H(R|s) = - \sum_{s \in S} \sum_{r \in R} p(r, s) \log_2 \left(\sum_{r' \in R} p(r'|s) \phi(r, r') \right) \tag{2}$$

Finally, similar to Shannon information theory [114], the metrical mutual information $I^*(R; S)$ is the difference between marginal and conditional entropies:

$$I^*(R; S) = H^*(R) - H^*(R|S) = \sum_{s \in S} \sum_{r \in R} p(r, s) \log_2 \left(\frac{\sum_{r' \in R} p(r'|s) \phi(r, r')}{\sum_{r' \in R} p(r') \phi(r, r')} \right) \tag{3}$$

If the event space consists of spiking signals we can take the similarity measure $\phi(r, r')$ as a decreasing function of the distance $D(r, r')$ between two spike trains $r, r' \in R$. Henceforth, we define $\phi(r, r')$ as a Heaviside function of the distance:

$$\phi(r, r') = \mathcal{H}(D_c - D(r, r')) \tag{4}$$

where the *critical distance* D_c is the cutoff parameter: for $D(r, r') < D_c$, responses r, r' are considered as identical, otherwise they are classified as different. If $D_c = 0$ we recover the Shannon entropy from Eq. 1.

Optimality condition for neurotransmission. Akin to the principle of redundancy reduction in biological sensory processing [1], the metrical conditional entropy $H^*(R|S)$ constitutes a fundamental quantity for neurotransmission [9]. Optimal information transfer must both maximise $I^*(R; S)$ and (at the same time) minimise $H^*(R|S)$ [3]. In order to set the optimal parameter D_c for $\phi(r, r')$ we consider two sets of $D(r, r')$ distances: (i) the distances between the responses elicited by the same stimulus —referred to as *intra-stimulus distances*; (ii) the distances between the responses elicited by different stimuli —named *inter-stimulus distances*.

As the input spike waves across multiple afferents flow in the readout system, the relationship between intra- and inter-stimulus distance distributions evolves over time. Optimal discrimination occurs when the distributions of intra- and inter-stimulus distances stop overlapping, which implies that $H^*(R|S)$ is nil and $I^*(R;S)$ is maximum [3]. The cut-off parameter D_c corresponding to optimal neurotransmission can then be set as the distance at which the maximum intra-stimulus distance becomes smaller than the minimum inter-stimulus distance. The time at which the critical distance D_c can be determined indicates when perfect input discrimination occurs [3].

Definition of a novel spike train metrics. Several parametric approaches have quantified the distance between spike trains (e.g. [13],[2],[10],[8]). We previously employed the Victor-Purpura (D_{VP}) distance [13] in order to define the similarity function $\phi(r, r')$, i.e. $D(r, r') = D_{VP}(r, r')$ in Eq. 4, and to embed spike metrics in our information theoretical analysis [2],[3]. In this paper we presents some results based on the Victor-Purpura distance (Sec. 3.2). In addition, we derive a novel spike train metrics to take into account the properties of a putative neural decoder (e.g. a population of downstream neurons along a processing pathway). We do that by defining a parametric non-linear spike train distance mimicking the Spike-Response Model (SRM) [5]. When a SRM neuron receives an afferent spike train s_i , its membrane potential $V(t)$ is computed as:

$$V(t) = \frac{1}{\tau} \sum_i \mathcal{H}(t - \hat{t}_i) \sqrt{t} e^{-(t - \hat{t}_i)/\tau} \quad (5)$$

where \mathcal{H} is the Heaviside function, \hat{t}_i is the time of an input spike emitted by the presynaptic neuron i , and τ is a free parameter determining the decay time constant of the EPSP (excitatory post-synaptic potential) of the neuron. The discharge probability $p(t)$ is a sigmoid function of the membrane potential:

$$p(t) = \frac{1}{1 + e^{(V_0 - V(t))/\Delta V}} \quad (6)$$

with V_0 denoting the probabilistic threshold potential. We can think of this probability of firing as a fundamental output of the SRM, because it does not depend on a particular trial but represents the way the signal is processed (on average). We thus define the distance between two spike trains r_1, r_2 in response to two stimuli s_1, s_2 as the difference between the corresponding firing probabilities:

$$D_{SRM}(r_1, r_2) = \left(\int_t (p_1(t) - p_2(t))^z dt \right)^{1/z} \quad (7)$$

with $z = 2$ in this study. The definition of D_{SRM} is reminiscent of that of Van Rossum distance [12] due to the filtering of the spike train with an exponential kernel function. However, we take into account a non-linearity, which is highly relevant because it captures the probabilistic threshold behaviour of a neuron — in the present case, this is modelled with escape noise. An advantage of the SRM

distance is that all the spikes do not contribute equally to it. Those that yield only subthreshold potentials have no significant contribution to the probability of spiking and thus do not contribute to the distance. Spikes evoking superthreshold potentials, induce large probabilities that may, in turn, contribute largely to the distance. Thus, the SRM distance can be employed to implement synchrony detection: two different spike trains whose interspike intervals are large may have a nil distance if the threshold is high. However, periods of intense discharge make the potential overcome the threshold and thus potentially increase the distance. This may be relevant in the case of spontaneous input activity that should not be taken into account. Distances such as Victor-Pupura and Van Rossum do not account for this property, i.e. they are always non-zero for spontaneous activities.

3 Results

3.1 Metrical Analysis of Poisson Spike Trains

We first compared the Victor-Purpura (VP), Van Rossum (VR) and SRM metrics in terms of their efficiency in assessing spike train separability. We simulated stochastic input spike trains generated according to a Poisson process with refractory period. At each time t , the effective probability of spiking was:

$$p_{eff}(t) = \frac{p(t)}{1 + e^{-t + \hat{t} + \tau_{ref}}} \quad (8)$$

with \hat{t} denoting the latest spike time, and $\tau_{ref} = 5$ ms the refractory period.

In a first protocol, we set the baseline firing probability to $p_0 = 0.01$. We simulated two stimuli: the first increased its probability of spiking from p_0 to 0.2 within the time window 300–600 ms, whereas the second went from p_0 to 0.4 within 300–600 ms. We then computed $n = 400$ VP, VR and SRM distances between pairs of spike trains from identical stimuli (i.e. intra-stimulus distances), as well as $n = 400$ between pairs of spike trains from different stimuli (i.e. inter-stimulus distances). Figure 1A compares the distributions of inter- and intra-stimulus distances assessed according to VP, VR, and SRM measures after 600 ms. The plots suggest, qualitatively, that the SRM distance captured better than VP and VR the separation between the two stimuli. To quantitatively assess the efficiency of stimulus separation, we computed the integral of the absolute difference between the histograms of intra- and inter-stimulus distances (i.e. if two histograms were identical, the efficiency would be nil; if they were non-overlapping, the efficiency would be maximum). The SRM efficiency was significantly larger than the two others (Fig. 1C), suggesting a better discriminability due to the smaller overlap between intra- and inter-stimulus distances.

In a second protocol, we set again the baseline probability to $p_0 = 0.01$ and we simulated two stimuli: the first increased its probability of spiking to 0.2 within the interval 300–500 ms, whereas the second increased it to 0.2 within 350–550 ms. Then, we computed $n = 400$ VP, VR and SRM distances between pairs of spike trains from identical stimuli, and $n = 400$ between pairs of spike trains from different stimuli. Figures 1B, C show that again the SRM distance provided

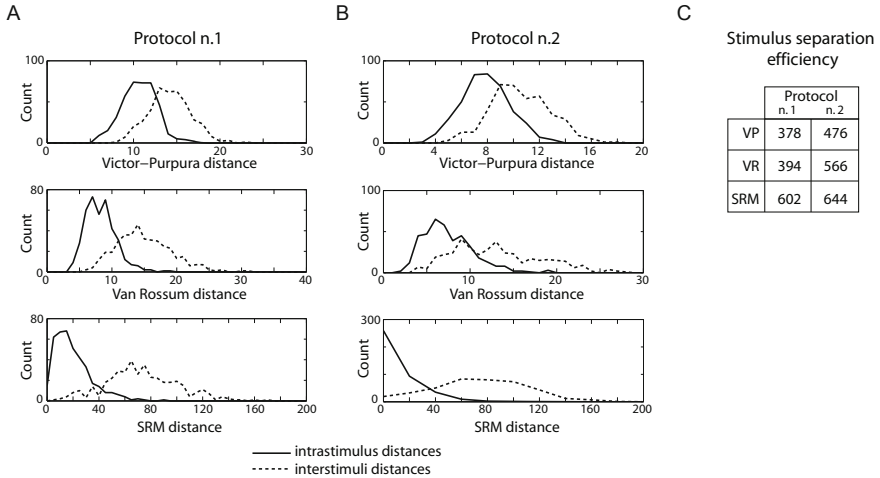


Fig. 1. Results. Comparative analysis between Victor-Purpura (VP), Van Rossum (VR), and SRM distances between two Poisson spike processes.

a better separation between the two stimuli compared to VP and VR distances. Note that, in this case, the efficiency of the SRM distance was not much larger than that of the VR distance (Fig. 1C). However, the SRM intra-stimulus distances were low compared to the inter-stimuli distances (Fig. 1B), which can be interpreted as a high signal-to-noise ratio (SNR) —since the level of intra- and inter-stimulus distances can be understood as *noise* and *signal*, respectively. Thus, although in this example the efficiency of spiking pattern separation was only slightly improved by the SRM distance, the SNR significantly increased compared to VP and VR measures. If we defined the SNR as the mean inter-stimuli distance over the mean intra-stimulus distance, then the SNR would be ≈ 2 for the VR distance and ≈ 4.5 for the SRM distance.

3.2 Isometric Coding of Human Microneurography Haptic Signals

We applied our metrical information theory (Sec. 2.1) to a data set of human microneurography recordings of fingertip mechanoreceptor responses [6]. Mechanoreceptors innervate the epidermis and discharge according to mechanical indentations of the skin. The spike latencies of mechanoreceptor responses convey information about contact parameters faster than the fastest rate code, and fast enough to account for the use of tactile signals in natural manipulation [6, 7]. We focused on temporal coding of fast adapting (FA-I) mechanoreceptor activity [6]. The data set consisted of the responses of 42 FA-I units to 81 distinct tactile stimuli obtained by varying four contact parameters (Fig. 2A). The discharge of FA-I units (Figs. 2B,C) occurs rapidly with respect to stimulus onset and it is phasic (as opposed to other mechanoreceptor types which exhibit slow and tonic responses to skin indentation, [7]).

We first focused on the first spike latencies of FA-I responses. Each tactile stimulus corresponded to a single volley of spikes forming a spatiotemporal pattern

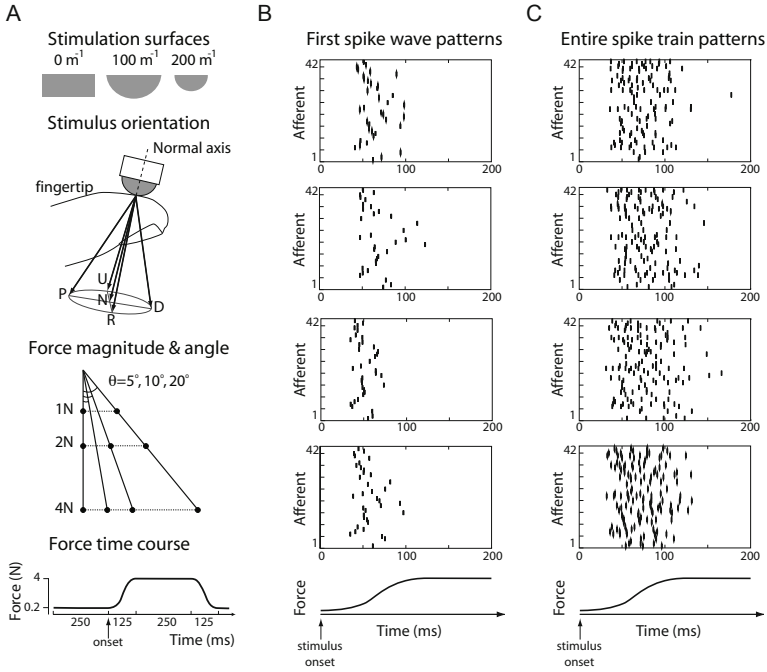


Fig. 2. Human microneurography recordings. (A) We considered 42 FA-I responses to 81 stimuli obtained by varying four contact parameters: probe curvature, force magnitude, force direction, and force angle relative to the normal direction. Black circles indicate available data for given force-angle combinations. (B,C) Samples of FA-I spikegrams when considering first spike waves only or entire spike train patterns, respectively.

(Fig. 2B). We subsequently accounted for the entire spike train patterns emitted by the population of 42 mechanoreceptors (Fig. 2C), to investigate the possible contribution of second and following spike waves to information transmission. We studied temporal encoding/decoding of mechanoreceptor discharges in terms of rapid and reliable discriminability —i.e. in terms of optimal information transfer as defined in Sec. 2.1. The results of Figure 3A compare the time course of metrical information $I^*(R; S)$ (solid lines) and conditional entropy $H^*(R|S)$ (dashed lines) when considering first-spike waves (black lines) or full spike trains (grey curves). Perfect discrimination of 81 tactile stimuli based on the relative spike timing of mechanoreceptor responses occurred within 40 ms of the first spike emitted, i.e. very early in the protraction phase of stimulation (Fig. 3A, bottom). These results also suggested that taking into account entire spike trains did not provide any significant benefit in terms of discrimination capacity and velocity, corroborating the hypothesis on the relevance of the variability in the first-spike latency domain. To study the potential role of subsequent spikes we employed the Victor-Purpura metrics D_{VP} to measure the distances between FA-I responses to a specific stimulus set. We computed the D_{VP} between the responses to stimuli with force amplitudes 1 and 2 N, i.e. $D_{VP}(1N, 2N)$, then to

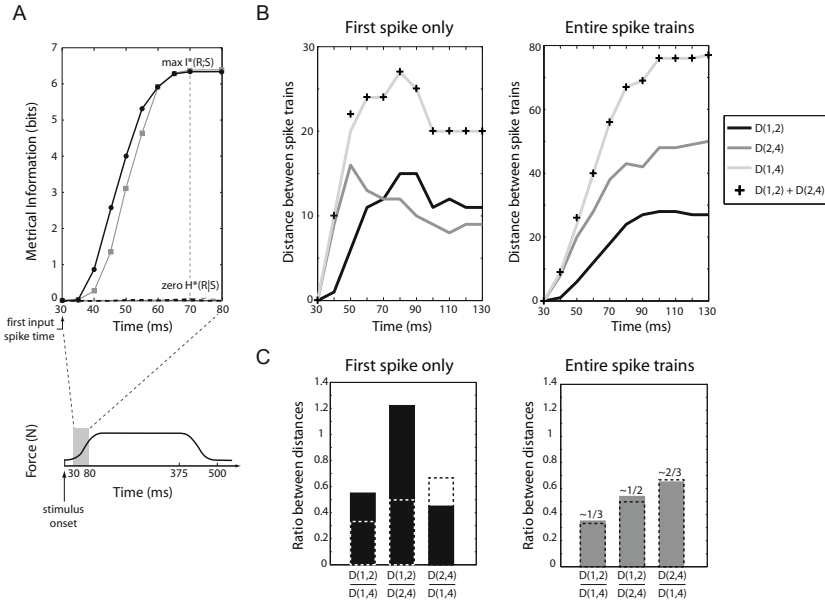


Fig. 3. Results. Metrical information analysis of 42 FA-I responses to 81 tactile stimuli. Each stimulus was presented 100 times. (A) Metrical information over time when considering either first spike waves only (black curve) or entire spike trains (grey line). (B) Time course of the distances between FA-I responses to stimuli with force amplitudes 1 and 2 N, $D(1,2)$, then to stimuli with force 2 and 4 N, $D(2,4)$, and finally to stimuli with force 1 and 4 N, $D(1,4)$. For each combination, all the other contact parameters were varied. (C) Only entire spike trains allowed an isometric input-output mapping to be captured (dashed rectangles indicate the perfect isometric mapping).

stimuli with force 2 and 4 N, i.e. $D_{VP}(2N, 4N)$, and finally to stimuli with force 1 and 4 N, i.e. $D_{VP}(1N, 4N)$. Figure 3B shows that the following relation was verified: $D_{VP}(1N, 2N) + D_{VP}(2N, 4N) = D_{VP}(1N, 4N)$. This result held when both only first spike waves (Fig. 3B, left) and entire spike trains were considered (Fig. 3B, right). Thus, in both cases the one-dimensional stimulus space was mapped onto a non-curved one-dimensional response space. This “alignment” property could be suitable to dissociate the problem of decoding the force of the stimulus from that of determining other features of the stimulus (i.e. pruning of the search state space). Importantly, Figure 3C shows that only when considering entire spike trains the distances between the responses tended to reflect the distances between the stimuli, suggesting an isometry-like property of the decoding measure. We further investigated this isometric coding property by analysing entire spike train responses to the stimulus set of Figure 4A, top. Decoding based on Victor-Purpura metrics —as for data in Figure 3B— proved to be unsuitable to capture the geometrical regularities (i.e. both “alignment” and isometric properties) encoded by the mechanoreceptor response space (not shown). We then applied a decoding scheme based on the SRM metrics (Eqs. 5-7), which allowed

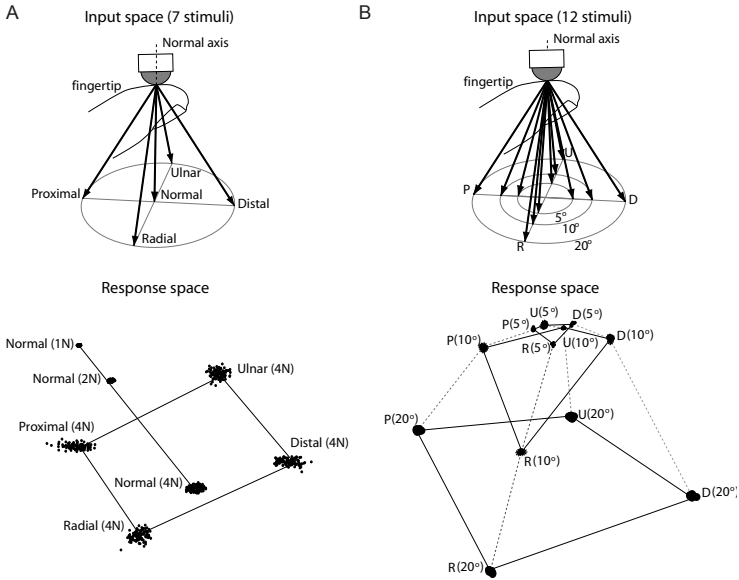


Fig. 4. Results on isometric coding. (A) Multi-dimensional scaled representation (bottom) of the metrical organisation of FA-I responses to 7 stimuli (top; i.e. a probe of fixed curvature applied along the Normal direction with force 1, 2, 4 N, and along the Ulnar, Radial, Distal and Proximal directions with force 4 N). (B) Multi-dimensional scaled representation (bottom) of FA-I responses to 12 stimuli (top, i.e. a probe a fixed probe applied along the 4 directions Ulnar, Radial, Distal, and Proximal with different angles 5°, 10°, 20°; the force was fixed and equal to 4 N).

us to mimic a population of 20 putative 2^{nd} -order neural decoders stochastically connected to the 42 recorded mechanoreceptors. We computed the distances between responses elicited by the set of stimuli shown in Figure 4A (top) as the sum of the SRM distances for each of the 20 hypothetical 2^{nd} -order neurons. For visualisation purposes, we applied a multi-dimensional scaling to this set of distances to determine the three dimensions that best preserved all measured distances. The projections of the multidimensional responses onto the identified 3-dimensional space are shown in Figure 4A, bottom. We obtained both isometrically-organised and “aligned” clusters in the response space. Stimulations in the Normal direction with forces 1, 2 and 4 N formed a line that passed orthogonally in the middle of a square formed by the responses to 4 orthogonal stimulations. Finally, Figure 4B shows that isometric coding was also obtained when considering a more complex data set of 12 tactile stimuli.

4 Discussion

This paper highlights the importance of accounting for the metrical relations between spike trains when studying neurotransmission. We contribute to spike metrics theory by deriving a novel distance that embeds the characteristics

(e.g. membrane time constant, threshold and slope of the non-linear firing function) of a probabilistic neural decoder. This distance helps us to determine the decoder's parameters leading to effective information transfer—in terms of both input discrimination and detection of geometrical input organisation. When decoding the activity of human FA-I mechanoreceptors we provided evidence that the temporal structure of their activity contains enough information to discriminate large number of tactile stimuli within few milliseconds and to infer geometrical relations between them. Our results suggest that first-spike latencies across multiple mechanoreceptors are relevant to rapid and non-ambiguous reconstruction of the input, whereas only by decoding entire spike train responses we can go beyond discrimination and capture higher-order properties such as isometric stimulus-response transformations. This observation may be the basis for the ability to extrapolate and/or generalise tactile perception and recognition in the presence of never experienced stimuli.

Acknowledgements. Granted by the EC project SENSOPAC (no. IST-027819-IP) and by the French Medical Research Foundation (no. FDT20090916556).

References

1. Barlow, H.B.: The coding of sensory messages. In: Thorpe, W.H., Zangwill, O.L. (eds.) *Current Problems in Animal Behaviour*, pp. 331–360. Cambridge University Press, Cambridge (1961)
2. Brasselet, R., Johansson, R.S., Arleo, A.: Optimal context separation of spiking haptic signals by second-order somatosensory neurons. In: Bengio, Y., et al. (eds.) *Adv. Neural Inf. Process Syst.*, vol. 22, pp. 180–188 (2009)
3. Brasselet, R., Johansson, R.S., Arleo, A.: Quantifying neurotransmission reliability through metrics based information analysis. *Neural Comput.* 23(4), 852–881 (2011)
4. Cover, T., Thomas, J.: *Elements of information theory*. Wiley, Chichester (1991)
5. Gerstner, W., Kistler, W.: *Spiking Neuron Models*. Cambridge Univ. Press, Cambridge (2002)
6. Johansson, R.S., Birznieks, I.: First spikes in ensembles of human tactile afferents code complex spatial fingertip events. *Nat. Neurosci.* 7, 170–177 (2004)
7. Johansson, R.S., Flanagan, J.R.: Coding and use of tactile signals from the fingertips in object manipulation tasks. *Nat. Rev. Neurosci.* 10, 345–359 (2009)
8. Kreuz, T., Haas, J.S., Morelli, A., Abarbanel, H.D.I., Politi, A.: Measuring spike train synchrony. *J. Neurosci. Meth.* 165, 151–161 (2007)
9. MacKay, D., McCulloch, W.: The limiting information capacity of a neuronal link. *Bull. Math. Biol.* 14(2), 127–135 (1952)
10. Schreiber, S., Fellous, J.M., Whitmer, S., Tiesinga, P., Sejnowski, T.: A new correlation-based measure of spike timing reliability. *Neurocomput.* 52-54, 925–931 (2003)
11. Shannon, C.: A mathematical theory of communication. *Bell Syst. Tech. J.* 27, 379–423 (1948)
12. Van Rossum, M.: A novel spike distance. *Neural Comput.* 13, 751–763 (2001)
13. Victor, J., Purpura, K.: Nature and precision of temporal coding in visual cortex: a metric-space analysis. *J. Neurophysiol.* 76, 1310–1326 (1996)

Context Separability Mediated by the Granular Layer in a Spiking Cerebellum Model for Robot Control

Niceto R. Luque¹, Jesús A. Garrido¹, Richard R. Carrillo², and Eduardo Ros¹

¹ Department of Computer Architecture and Technology,
University of Granada, Periodista Daniel Saucedo s/n, Granada, Spain

² Department of Computer Architecture and Electronics,
University of Almeria Almeria, Spain

{nluque, jgarrido, rcarrillo, eros}@atc.ugr.es

Abstract. In this paper, we study how a biologically-plausible cerebellum architecture can store and retrieve different robotic-arm internal models (in synaptic connections between granular layer and Purkinje cells) at the granule layer (dynamic modifications of a base robot-arm-plant model), and how the model microstructure and input signal representations can efficiently infer models in a robot control scenario during object manipulation. More specifically, we have evaluated the contribution of the granular layer to the ability of the cerebellum to generate corrective actions. To achieve this we have embedded a spiking cerebellar model into an analog control loop whose output commands a simulated robot arm. The performance results obtained by using a cerebellum which includes granular layer are compared to those using a cerebellum without this layer. The results show that this layer effectively contributes to the generation of accurate cerebellar corrections. This work represents a well defined case of study in the field of neurobotics, in which biologically plausible neural systems and robots are used to study the functionality of biological systems.

Keywords: Granular Layer, Biological Robot Control, Trajectory Following, Plasticity, Context Switching, STDP, Cerebellum, Spiking Neural Network.

1 Introduction

The granular layer contains approximately half of number of neurons in the entire human brain [1]. The granular layer represents a highly divergent structure (there are approximately 10^3 granule cells per mossy fiber) whose functionality seems to be essentially unsupervised [2]. Granular cells (GR) seem to be responsible for building a sparse representation of the mossy fibers inputs. In this layer an efficient processing of mossy fibers (MF) activity takes place and a sparse representation is supposed to be delivered through the parallel fibers (PFs) which are the axons of the GRs. This processing may facilitate the learning capability at the parallel fibers-to-Purkinje cells (PF-PC) connections in order to minimize destructive interferences among consolidated pre-learned contexts. The granular layer includes the contextual information provided by MFs [3] [4]. In these works, the authors study how the granular layer produces sparse activity in the parallel fibers (through a comparison between cerebellar models with or without this layer) in a way that maximizes the

mutual information transfer between MFs and PCs (reducing statistical dependencies between information transmitted through different parallel fibers). The activity in the granule cells is mainly determined by the activity of the incoming connections from the input mossy fibers and the synaptic weight of these connections. Although the cerebellum seems to possess very diverse adaptation mechanisms at different sites, the approach presented in this work uses fixed MF-GR connections with constant strength and LTD and LTP as adaptation mechanisms at the PF-PC connections as described in Section 2.4. This seems to be the main supervised adaptation mechanism at these PF-PC synapses [5], which in our approach remains central for the corrective functionality of the cerebellum in an object manipulation task.

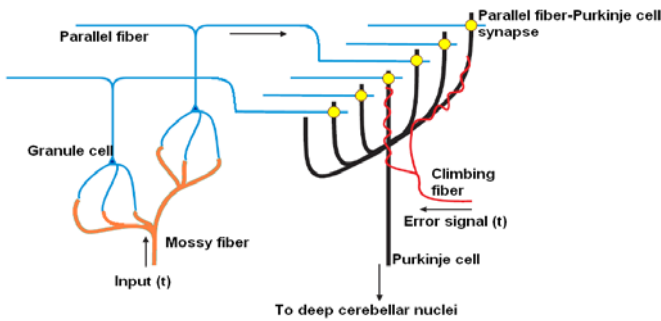


Fig. 1. The simplified microcircuitry of the cerebellum. Human cerebellum is formed by three layers: granular, molecular and Purkinje layer. The human cerebellum involves about 10 million Purkinje cells receiving excitatory inputs from parallel fibers. Each parallel fiber synapses on about 200 Purkinje cells; these parallel fibers are GR axons. Some of these GRs are excited by the mossy fibers (with afferent connections from the spinal cord, with sensory and motor estimates). Each Purkinje cell receives further excitatory synaptic input from one single climbing fiber. The output of a Purkinje cell is an inhibitory input signal to the deep cerebellar nuclei which is the only output pathway of the cerebellum.

These adaptive synaptic weights define a direction in the MF input space along which the GR layer (synaptic PF-PC) has the highest sensitivity. According to [3], the desired situation would be the one that drives GRs to the sparsest code representation making all neighboring cells as independent as possible. This leads to a representation with the most interesting and useful properties for subsequent computations in the cerebellum. The sparser the coding is, the more storage capacity information the network has [4] [6] [7]. Therefore, to adequately activate a PC a sparsely-activated group of GRs is required [8]. In order to specifically evaluate the ability of the granular layer to conveniently pre-process the MF information, we have carried out simulations including and excluding this layer in the cerebellum architecture. These simulations assess whether this pre-processed signals properly activate PCs improving subsequent torque computations in the next layers.

The working hypothesis and methodology of this work can be briefly described as follows:

1. Achieving a biologically relevant task which consists in accurate manipulation of objects which affect a base dynamic model of the base plant. Different objects represent different contexts for the network in the manipulation task.
2. Definition and implementation of a spiking-neuron based cerebellum model in order to evaluate how the granular layer of the cerebellar model affects the functionality of the system in this multi-context manipulation scenario.

2 Methods

We have used an efficient event-driven simulator for extensive spiking network simulations (EDLUT) [9] [10]. To achieve this, we have embedded a cerebellar spiking network in an analog control loop which controls a three-degrees-of-freedom robot. The robot performance results are evaluated with regard to inclusion of granule cells in the cerebellar model.

2.1 Arm Simulator and Training Trajectory

We have integrated the robot plant simulator of the LWR (Light-Weight-Robot developed at DLR) [11] [12] with the control loop defined in Fig. 2. The simulated-robot-plant physical characteristics can be dynamically modified to emulate different contexts. The LWR (Fig.2.a) robot is a 7-DOF arm composed of revolute joints. In our experiments we used the first (so-called q_1), second (q_2) and fourth joint (q_3) keeping the others fixed. The task for the experiments with LWR robot is to follow an eight-like trajectory. This is a rather standard benchmarking trajectory. The inverse kinematics is computed to obtain a smooth trajectory in joint angle space [13] [14]. The trajectories of individual joints have enough variation ensuring a sufficiently rich movement that allows dynamic robot arm features to be revealed [14] [15].

$$q_1 = A \sin(\pi t) + C_1. \quad (1.A)$$

$$q_2 = A \sin(\pi t + \theta) + C_2. \quad (1.B)$$

$$q_3 = A \sin(\pi t + 2 \cdot \theta) + C_3. \quad (1.C)$$

2.2 Control Architecture

In this section, the Feedback-Error Learning (FEL) architecture [16], shown in Fig. 2.b, is presented. It consists of both the crude inverse dynamic robot model which generates the ideal joint torque values (the needed torque when neither disturbances or context switching scenarios appear) and the cerebellum module which supplies the feed-forward joint corrective torque [16] [17]. When the crude inverse dynamic model is faced with a non-ideal scenario (i.e. when the robot is manipulating an object), it is not able to generate a correct global torque on its own to drive the robot plant properly; therefore, an error between the desired and the actual robot's trajectory comes up. This is the so-called feedback error. The cerebellum receives this signal

which will modulates the learning process (FEL) [18]. The cerebellum model overcomes the lack of a precise robot arm dynamic model ensuring the stability of the system and allowing a better performance in terms of accuracy.

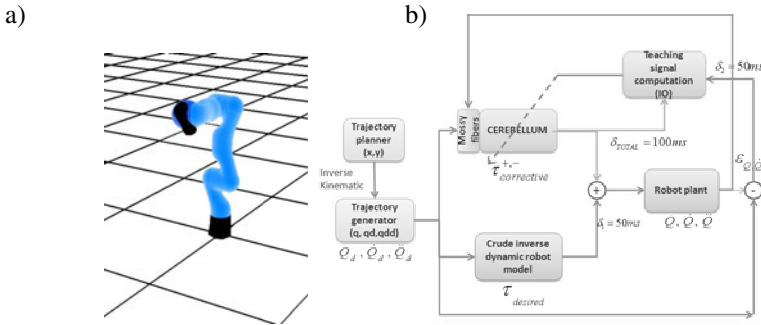


Fig. 2. a) LWR robot b) FEL control loop, using the feedback error, the cerebellar model tries to infer a model to produce effective corrective commands. These delivered add-on corrective actions to compensate deviations in the base dynamic plant model in a context switching scenario. The desired arm states (x,y) are generated by the trajectory planner to follow the desired trajectory. This trajectory in Cartesian coordinates is translated into joint coordinates (positions (), velocities () and accelerations ()) by the trajectory generator (a crude inverse kinematic model representing the output of motor pre-cortex and other motor areas). These desired arm states in joint coordinates are used at each time step to compute crude torque commands (crude inverse dynamic robot model). They are also used as input to cerebellum which produces the predictive corrective commands ($\tau_{corrective}$) which are added to these crude torque commands ($\tau_{desired}$). Total torque is delayed (δ_1 , on account of the biological motor pathways) and supplied to the robot plant. The difference (ϵ) between the actual robot trajectory and the desired one is also delayed (δ_2) and used by the teaching-signal computation module to calculate the inferior olive (IO) activity which is supplied to the cerebellum as a teaching input signal (for the computation of adapting the cerebellar synaptic weighs.). Thus, this signal is used by the cerebellum to adapt its output.

2.3 Cerebellum Model

An advanced Event-Driven simulator based on LookUp Tables (EDLUT) [9] has been used in order to generate the output of the spiking cerebellar architecture. EDLUT is an open-source tool [10] which accelerates the simulation of spiking neural networks by compiling the response of pre-defined cell models into lookup tables before the actual network simulation. Then, the simulation can be performed efficiently without requiring intense numerical analysis.

The neuron models used in the cerebellum are an evolved version of the leaky integrate-and-fire model in which each synapse is implemented as an input-driven conductance [10].

In order to study the potential functional role of the granular layer, we compare results using the two proposed cerebellar architectures illustrated in Fig. 3. The specific connectivity pattern among different cell types is similar to the one used in [19].

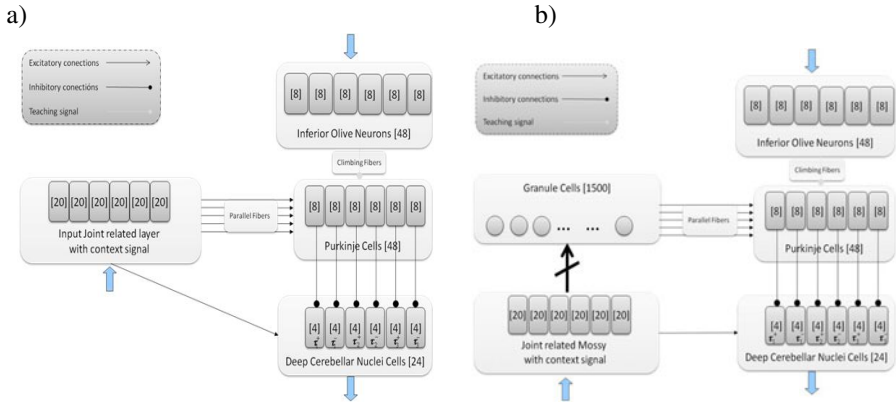


Fig. 3. Cerebellum configuration diagram with and without granular layer (the number of neurons per layer is represented in brackets). MFs are fed by an analog current signal generated by means of overlapping radial basis functions (RBF) as receptive fields in the value space of the input signals. These inputs (translated into spikes) encoding the movement are sent (upward arrow) through the parallel fibers (PFs). Error-related inputs are sent (upper downward arrow) through the climbing fibers (CFs). Outputs are provided by the cell of the deep cerebellar nuclei (DCN) (lower downward arrow). The DCN collects activity from the input layer (excitatory inputs) and the Purkinje cells (PC) (inhibitory inputs). The outputs of the DCNs are added as corrective activity in the control loop (τ_i where $i=1$ to number of joints) of Fig.2.b. [19] a) Cerebellum model diagram without granular layer. The inputs reach directly the PFs. b) Cerebellum model diagram with granular layer. Inputs are combined in the granular layer.

2.4 Learning Process

Although there seems to be adaptation mechanisms at many sites within the cerebellar structure [20] [21] the main synaptic adaptation driven by teaching or temporal signals (from the inferior olive (IO) by means of climbing fibers) take place at the PF-PC synapses [22] [23]. We have adopted a STDP (spike-timing dependent plasticity) mechanism in the spiking cerebellar neural network. The adaptation mechanism drives the efficacy of the PF-PC synapses in the cerebellar model, based on the concept of “eligibility trace” [13], which is a measure of the past activity of the afferent PF. This trace aims at relating spikes from IO (error-related activity) and the previous activity of the PF (sensory-motor-related activity) which is supposed to have generated this error signal.

This plasticity has been implemented including LTD and LTP mechanisms in the following way:

- LTD produces a synaptic efficacy decrease when a spike from the IO reaches a PC as indicated in Eq. (2.A). The amount of synaptic efficacy, which is reduced, depends on the previous activity arrived through the PF. This previous activity is convolved with an integral kernel as defined by Eq. (3). This kernel mainly takes into account all the PF spikes which arrived about 100ms before the IO spike to overcome the effect of transmission delays of this range on sensory and motor signals.

- b) LTP produces a fixed increase in synaptic efficacy each time a spike arrives through a PF to the corresponding PC as defined by Eq. (2.B).

$$LTD: \forall i, \Delta w_i = - \int_{-\infty}^{I_{Ospike}time} k(t - t_{IOspike}) \delta_{GRspike-i}(t) dt. \quad (2.A)$$

$$LTP: \Delta w_i = \alpha. \quad (2.B)$$

$$k(t) = e^{-(t-t_{postsynapticspike})} \sin(t - t_{postsynapticspike})^{20}. \quad (3)$$

STDP Learning Mechanism

Purkinje cells receive the spatiotemporal activity (coding in time the desired entries; positions, velocities and accelerations) from the GRs and a teaching signal through climbing fibers (according to the error estimate obtained in the control loop which is handled properly by the corresponding IO). STDP learning mechanism modifies synaptic weight of active GRs at the arrival time of climbing fiber spikes. Finally Purkinje cells inhibit deep cerebellar nuclei. This deep cerebellar nuclei is in charge of generating the corrective torque. In conclusion, the final generated corrective torque is related not only with the incoming error through the climbing fiber but also the desired trajectory entries coded in time by GRs.

2.5 Experimental Methods

We study the capability of the granular layer to aid in the generation of corrective models for dynamics changes to a base plant model. For each training process we use four objects attached at the end of the last segment of the robot arm (effector): 0.5kg, 1kg, 1.5kg and 2kg. For this purpose, we have used a benchmark trajectory Eq. (1) that is repeated in each iteration and evaluated how learning adapts the GR-PC weights [24], to tune accurate corrective actions in the control loop (Fig. 2.b). If learning takes place appropriately, during the process, the obtained error in each trajectory execution decreases until it reaches a final stable value.

Besides the learning curve itself, the global learning performance is assessed using three estimates which are calculated from the evolution of the mean absolute error (MAE) of the three robot joint coordinates along the executed trajectory (trial) during the learning process:

- Accuracy gain: Difference between the initial MAE error and the one reached after the learning process (average error of the last 30 trials).
- Final-error average: Average error of the last 30 trials.
- Final-error standard deviation: Standard deviation of the error of the last 30 trials. Larger standard deviation indicates less stable behavior.

3 Results

The cerebellum configurations in Fig.3 facilitate the evaluation of how the granular layer contributes to inferring different contexts (results shown in Fig. 4 and Fig. 5).

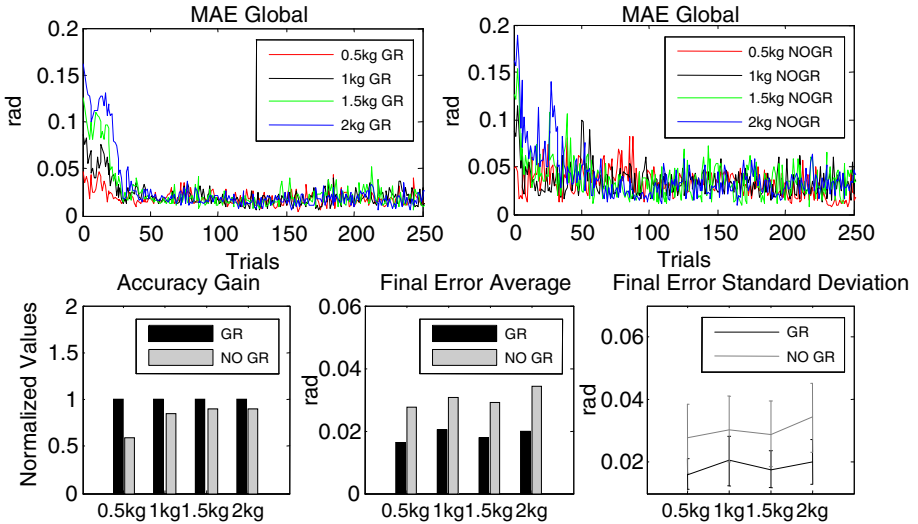


Fig. 4. Upper plots show learning curves with (left panel) and without (right panel) granular layer. Different masses are been attached to the robot-arm end effector (0.5kg, 1kg, 1.5kg and finally 2kg). Lower plots show the values obtained applying the three estimates described in method section. The model that includes granular layer reaches lower errors and reveals a more stable behavior after learning in each different context. Final-error average values: 0.5kgNoGR = 0.027 ± 0.011 rad 0.5kgGR = 0.016 ± 0.048 rad 1.0kgNoGR = 0.030 ± 0.010 rad 1.0kgGR = 0.020 ± 0.008 rad 1.5kgNoGR = 0.029 ± 0.010 rad 1.5kgGR = 0.017 ± 0.006 rad 2.0kgNoGR = 0.034 ± 0.011 rad 2.0kgGR = 0.002 ± 0.007 rad

As shown in Fig. 4, the learning curves present a better behavior using granular layer, since a sparse population of active cells changing in time is effectively obtained (4.7 spikes per second of average activity per granule cell in contrast to 21.6 spikes per second average activity per mossy fiber). The ideal situation would be represented by an activity of 1spike per trajectory execution and cell (PF), which means, a sole cell that unambiguously represents a joint position/velocity along the trajectory, the more concurrent activity we have, the less precise representation we obtain. This time-varying sparse population is able to represent exclusively a specific trajectory-execution time interval. That means, that our cerebellum has a proper sparse-population codification per each time-varying input value we may have (positions, velocities, and acceleration), thus, improving the input-pattern classification carried out by PCs through the STDP learning mechanism (previously described in section 2.4).

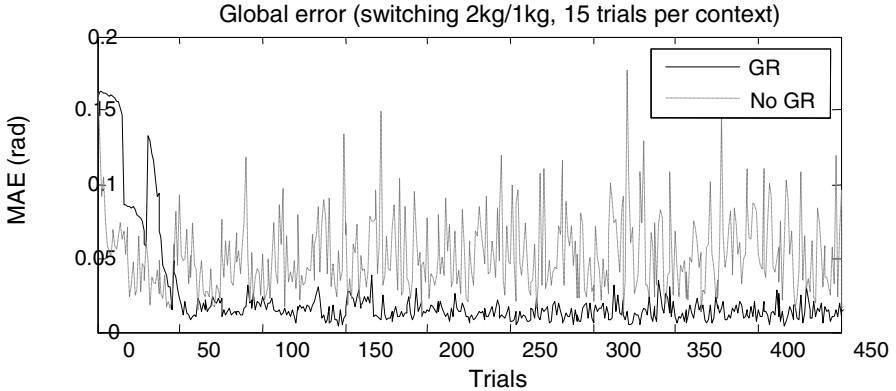


Fig. 5. The role of the granular layer in a multi-context scenario. Mean Absolute Error during the learning process switching between 2kg and 1kg object. Each trial represents a complete movement (eight-like trajectory). Each context-learning process is composed of 15 trials. We compare a cerebellar model with GRs and without them.

Fig. 5 shows performance results of experiments using granular layer in comparison with others without granular layer. We have conducted these experiments with 1kg and 2kg in a context switching scenario (the two objects attached at the end of the last segment of the robot arm are interchanged each 15 trials). The simulations of cerebellar topologies without granular layer obtain mean final errors of 0.0542 ± 0.0267 radians (where the first value represents the mean of final-error averages obtained in these experiments and the second value represents the mean of final-error standard deviations obtained in their corresponding experiments) which are larger than the ones using a granular layer (0.0151 ± 0.0057 radians). These more precise and stable outputs suggest that GRs are essential in the model inference process.

4 Conclusions and Future Work

In the framework of multiple-context management we have studied the role of the granular layer when abstracting different models using the presented spiking cerebellar model. Although a simple cerebellar structure has been used for the control tasks, it has become clear that the separability capability at the granular layer is of high interest in this task. Furthermore, the granular layer also performs spatio-temporal filtering of signals and spatio-temporal redistribution of activity coming from mossy fibers. These features are also used for computational operations involving coincidence detection and pattern separation [25] and also for reducing interferences between different co-existing learned models. According to Fig. 5, the granular layer is of critical importance to facilitate the generation of accurate specific corrective torques for different contexts in a multi-context manipulation scenario. Thus, the translation of input signals into sparse codes seems to be important, since no other specific discrimination module is included in this kind of network topology.

Inter-context discrimination and corrective torque generation is conjointly performed in a distributed manner in the cerebellar structure.

The results of this paper aim to advance in the comprehension of the functional role of cerebellar components and how this role is supported by specific network topologies. As future work, a more extensive set of experiments will be conducted to further assess the relevance of the granular layer and to statistically evaluate the results. We also will study the scalability of this cerebellar configuration, their robustness against noise in MF and IO entries and their response in the framework of different control loops.

Acknowledgements. We would like to acknowledge the constructive discussions of Patrick van der Smagt and Holger Urbanek related to the robot plant simulator and the whole SENSOPAC consortium. This work has been supported by the EU grant SENSOPAC (IST 028056) and national projects DINAM-VISION (DPI2007-61683), MULTIVISION (TIC-3873), ITREBA (TIC-5060) and the Spanish Funding program Juan de la Cierva 2009 (MICINN).

References

1. Kandel, E.R., Schwartz, J.H., Jessell, T.M.: Principles of Neural Science. McGraw-Hill Professional Publishing, New York (2000)
2. Schweighofer, N., Doya, K., Lay, F.: Unsupervised learning of granule cell sparse codes enhances cerebellar adaptive control. *Neuroscience* 103(1), 35–40 (2001)
3. Coenen, O.J.-M.D., Arnold, M., Courchesne, E., Jabri, M., Sejnowski, T.: A hypothesis for parallel fiber coding in the cerebellum. *Society for Neuroscience Abstracts* 25 (1999)
4. Coenen, O.J.-M.D., Arnold, M., Sejnowski, T., Jabri, M.: Parallel fiber coding in the cerebellum for life-long learning. *Autonomous Robots* 11(3), 291–297 (2001)
5. Jorntell, H., Hansel, C.: Synaptic memories upside down: bidirectional plasticity at cerebellar parallel fiber-Purkinje cell synapses. *Neuron*, 52, 227–238 (2006)
6. Meunier, C., Nadal, J.-P.: The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge (1995)
7. Brunel, N., Hakim, V., Isope, P., Nadal, J.P., Barbour, B.: Optimal information storage and the distribution of synaptic weights: perceptron versus Purkinje cells. *Neuron*, 43, 745–757 (2004)
8. Barbour, B.: Synaptic currents evoked in Purkinje cells by stimulating individual granule cells. *Neuron*, 11(4), 759–769 (1993)
9. Ros, E., Carrillo, R.R., Ortigosa, E.M., Barbour, E., Agís, B.: Event-Driven Simulation Scheme for Spiking Neural Networks Using Lookup Tables to Characterize Neuronal Dynamics. *Neural Computation* 18, 2959–2993 (2006)
10. Carrillo, R.R., Ros, E., Boucheny, E., Coenen, C.: A real-time spiking cerebellum model for learning robot control. *Biosystems* 94(1-2), 18–27 (2008), <http://edlut.googlecode.com>
11. Butterfaß, J., Grebenstein, M., Liu, H., Hirzinger, G.: DLR Hand II: next generation of a dextrous robot hand. In: IEEE International Conference on Robotics and Automation, pp. 109–114 (2001)

12. Hirzinger, G., Butterfab, J., Fischer, M., Grebenstein, M., Hähle, M., Liu, H., Schäfer, N., Sporer, I.: A mechatronics approach to the design of light-weight arms and multifingered hands. In: ICRA, pp. 46–54 (2000)
13. Kettner, R., Mahamud, S., Leung, H., Sittko, N., Houk, J., Peterson, B., Barto, A.: Prediction of complex two-dimensional trajectories by a cerebellar model of smooth pursuit eye movement. *Journal of Neurophysiology* 77(4), 2115–2130 (1997)
14. Haith, A., Vijayakumar, S.: Robustness of VOR and OKR adaptation under kinematics and dynamics transformations. In: Proceedings of 6th IEEE International Conference on Development and Learning (ICDL 2007), London (2007)
15. Hoffmann, H., Petckos, G., Bitzer, S., Vijayakumar, S.: Sensor-assisted adaptive motor control under continuously varying context. In: International Conference on Informatics in Control, ICINCO (2007)
16. Kawato, M., Gomi, H.: A computational model of four regions of the cerebellum based on feedback-error learning. *Biological Cybernetics* 68(2), 95–103 (1992)
17. Miller, L., Holdefer, R., Houk, J.C.: The role of the cerebellum in modulating voluntary limb movement commands. *Archives Italiennes de Biologie* 140(3), 175–183 (2002)
18. Ito, M.: Control of mental activities by internal models in the cerebellum. *Brain Res.* 886(1-2), 237–245 (2008)
19. Luque, N., Garrido, J., Carrillo, R., Ros, E.: Cerebellar Spiking Engine: Towards Object Model Abstraction in Manipulation. In: IJCNN (July 2010)
20. Bliss, T., Lomo, T.: Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *Journal of Physiology* 232, 331–356 (1973)
21. Hansel, C., Linden, D., D’Angelo, E.: Beyond Parallel Fiber LTD: The Diversity of Synaptic and Non-Synaptic Plasticity in the Cerebellum. *Nature Neuroscience* 4, 467–475 (2001)
22. Ito, M., Kano, M.: Long-lasting depression of parallel fiber-Purkinje cell transmission induced by conjunctive stimulation of parallel fibers and climbing fibers in the cerebellar cortex. *Neuroscience Letter* 33, 253–258 (1982)
23. Ito, M.: Long-term depression. *Annu. Rev. Neurosci.* 12, 85–102 (1989)
24. Kawato, M., Wolpert, D.: Internal models for motor control. *Novartis Foundation Symposium* 218, 291–307 (1998)
25. Mapelli, J., Gandolfi, D., D’Angelo, E.: Combinatorial responses controlled by synaptic inhibition in the cerebellum granular layer. *Journal of Neurophysiology* 103, 250–261 (2009)

Realistic Modeling of Large-Scale Networks: Spatio-temporal Dynamics and Long-Term Synaptic Plasticity in the Cerebellum

Egidio D'Angelo^{1,2} and Sergio Solinas^{2,3}

¹ Department of Physiology, University of Pavia, Via Forlanini 6, I-27100, Pavia, Italy

² Brain Connectivity Center, Istituto Neurologico IRCCS Fondazione C. Mondino,
Via Mondino 2, I-27100 Pavia, Italy

³ Consorzio Interuniversitario per le Scienze Fisiche della Materia (CNISM),
Via Bassi 6, I-27100 Pavia, Italy
{dangelo, solinas}@unipv.it

Abstract. A large-scale computational model of the cerebellum granular layer has been adapted to generate long-term synaptic plasticity in response to afferent mossy fiber bursts. A simple learning rule was elaborated in order to link the average granule cell depolarization to LTP and LTD. Briefly, LTP was generated for membrane potentials >-40 mV and LTD for membrane potentials <-40 mV. The result was to generate LTP and stronger excitation in the core of active clusters, which were surrounded by LTD. These changes were accompanied by a faster and stronger spike generation compared to the surround. These results reproduce the experimental observations and provide a valuable and efficient tool for implementing autonomous learning algorithms in the cerebellar neuronal network.

Keywords: NEURON, cerebellum, LTP, LTD, granule cells, modeling.

1 Introduction

Realistic large-scale representations of central neuronal networks can be obtained using the NEURON simulator[1]. These networks implement a bottom-up approach, which can provide important validations and predictions about network activity. Realistic models are tightly bound to experiments, with which usually co-evolve. We present here the case of the granular layer of the cerebellum (Fig. 1), a basic version of which has recently been published[2].

The cerebellar network is composed of a little number of neuronal types connected through a well defined architecture[3,4,5]. This has simplified the development of network models, which have been elaborated in several steps. Initially, realistic detailed representations of single granule cells and Golgi cells have been generated revealing that the whole set of complex properties of intrinsic excitability and synaptic transmission can be reproduced by appropriate mechanisms derived from experimental observations. Then, the single cell models have been used to generate the network model, which proved able to reproducing all known granular layer spatio-temporal

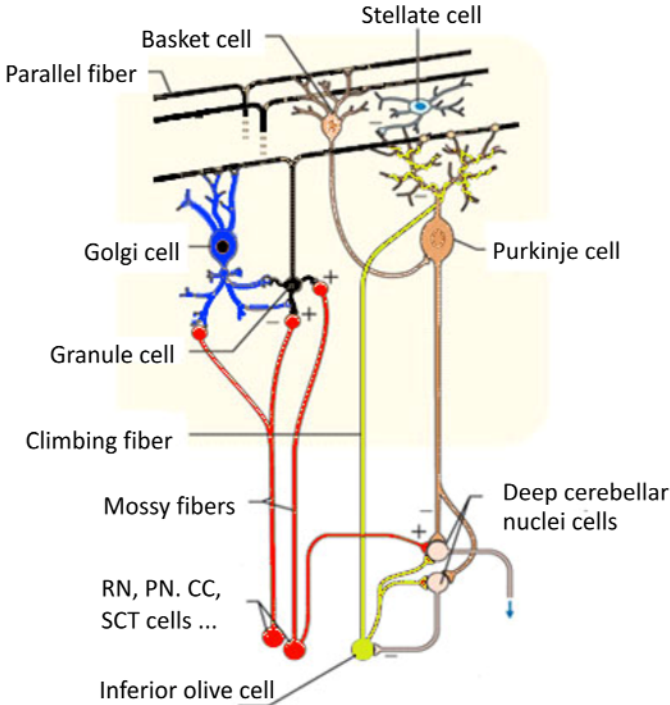


Fig. 1. Schematic representation of the cerebellar circuit. The granule cell represents the gate to the cerebellar cortex. It receives excitatory connection from mossy fibers and sends its axon to the molecular layer forming the parallel fibers, which activate the Golgi cells, the Purkinje cells and the molecular layer inhibitory interneurons (stellate and basket cells). Note the double feed-back and feed-forward inhibitory loop formed by the Golgi cells. Other elements of the cerebellar cortex are also indicated.

dynamics [2] and reconnecting molecular and cellular properties of the granule cell to global network computations[6,7,8,9,10].

The granule cells conductance-based models have been based on a large amount of experimental information (e.g. see [11,12,13,14,15,16,17,18,19]). These models allowed explaining properties like resonance [20,21] and synaptic plasticity[21], Na channel localization and spike generation [22], stochastic release and mutual information (MI) transfer [23].

The Golgi cell conductance-based models have also been based on a experimental information[24,25] although more limited than for the granule cells. These models allowed explaining pace-making and resonance, adaptation, phase-reset and rebound excitation[26,27,28,29]. Synaptic transmission has also been reproduced (Cesana, Dieudonne, D'Angelo and Forti, in preparation).

The current version of the large-scale model contains as many as 105 granule cells and several tens of Golgi cells with all the synapses in between. This model is currently under extension with an algorithm capable of generating long-term synaptic plasticity and reconfiguring network activity.

1.1 LTP and LTD Rules

Long-term synaptic plasticity at the mossy fiber – granule cell synapse is induced by NMDA receptor activation and by the consequent calcium influx in a voltage-dependent manner [19,30,31,32]. It has been shown that Golgi cell inhibition, by preventing granule cell depolarization, can effectively regulate the balance between LTP and LTD in response to high-frequency mossy fiber trains [33]. A robust NMDA receptor-dependent calcium influx occurs above -40 mV and can drive LTP. Between -40 mV and -50 mV, the contribution of the NMDA channels is modest. Moreover, mGlu receptors can generate a voltage-independent calcium influx, probably through release from intracellular stores enhancing LTP and inducing LTD (this latter mechanism occurred at low frequency but may also be extended for high-frequencies at low voltages) [30,31,34]. Therefore we have used the following simple plasticity rule for LTP and LTD generated by a short high-frequency train:

LTP for average $V_m > -40$ mV
 LTD for average $V_m < -40$ mV

Experimentally, LTP and LTD have been reported to reflect changes in release probability [19,31]. This parameter in our models is reported explicitly [2,21] and can therefore be modified by activity.

2 Methods

The large scale model used for these simulations is the same as that published previously [2], except for the fact that the number of synapses between mossy fibers and Golgi cell has been increased from 50 to 150. This allowed accelerating the rate of Golgi cell synaptic depolarization, improving control over the timing of inhibition. With a bundle of 23 active mossy fibers, the granule cell cluster included 625 granule cells and inhibition in granule cells peaked in about 4.3 ± 0.9 ms. Therefore, this cluster reproduces properties compatible with those observed experimentally [7,29,33,35,36]. In these simulations, all the mossy fiber granule cell synapses were initially set at the release probability, $p=0.42$. Then, the mossy fibers bundle was stimulated with a 3-spikes at 300 Hz train. The average membrane depolarization of the activated granule cells was then computed and used to modify p according to the plasticity rule illustrated above. After p modification, the net was stimulated again and the results compared.

3 Results

The response of the granular layer was organized in center - surround according to previous reports [33,35,36] (Fig. 2A). This occurred because the core provides both the strongest excitation of granule cells and the strongest lateral inhibition through Golgi cells. While the percentage of active granule cells was 11% in control ($p=0.42$), the percentage decreased with LTD ($p=0.2$) and increased with LTP ($p=0.8$). Interestingly, the number of discharging granule cells decreased from center (where it was as high as 50%) to periphery of the active area, in agreement with the “center-surround” mechanism [13]. The increase in p was accompanied by an anticipation of

the first spike and by an increase in the number of spikes per cell (Fig. 2B), in agreement with the “time-window” mechanism [7].

The voltage-dependent plasticity rule reported above influenced the center-surround, in that the center became broader with a sharp transition between discharging and non discharging granule cells. The overall percentage of discharging cells increased to 21%. This result was compared to the case of a uniform change in release probability over the whole cluster. In this case, the size of the discharging core changed remarkably, with a contraction at $p=0.2$ (5% discharging cells) and an expansion at $p=0.8$ (22% discharging cells). The profile of the discharging area obtained at $p=0.8$ was very similar to that observed applying the voltage-dependent plasticity rule (Fig. 2C).

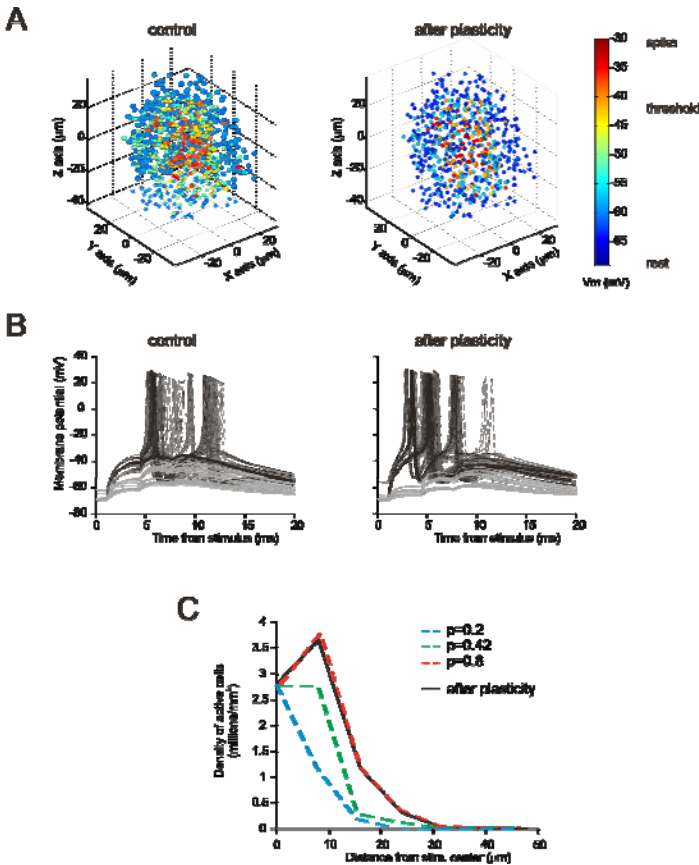


Fig. 2. Network response modifications induced by LTP and LTD. (A) Granule cells are represented by dots with a color corresponding to their membrane potential. Note that a few discharging cells (red) are added in the core. After induction, an area of LTP is manifest in the core and an area of LTF in the surround.(B) Granule cell making spikes become more numerous after LTP. Moreover, spikes occur earlier. (C) The density of active granule cells (i.e. those making spikes) is distributed from center to periphery of the cluster. The density changes remarkably with a uniform change in release probability. The change caused by the voltage-dependent learning rule is almost indistinguishable from that caused by $p=0.8$.

4 Discussion

This work reports a simple and efficient plasticity rule for implementing use-dependent synaptic changes in response to incoming input trains. The rule is based on the well-known voltage dependence of NMDA channel opening, which brings about a proportional regulation of intracellular calcium concentration [31,32]. It remains to be demonstrated whether a fine representation of internal calcium dynamics, which are influenced by mGlu receptors as well as by voltage-dependent calcium channels and calcium release from intracellular stores, could modify the results. The fact that the changes in cluster organization caused by the voltage-dependent plasticity rule and by a homogeneous change in release probability were similar, indicates that granule cell discharge is strongly influenced by synaptic inhibition. This latter prevented granule cell firing outside the core independent from release probability. However, clearly, non discharging granule cells with low release probability would be even more disadvantaged while responding to incoming inputs, generating a sharp edge between core and periphery.

The changes in network response obtained in these simulations strongly resemble those observed experimentally. Indeed, multi-electrode array recordings and voltage-sensitive dye imaging have shown that, following induction of long-term synaptic plasticity, LTP is condensed in the center and LTD in the surround [33,35,36]. This result directly addresses the mechanism through which the cerebellar granular layer is supposed to operate. Incoming inputs need to be separated and selectively amplified and filtered [37,38]. Since LTP and LTD regulate the transmission properties in terms of spike delay and frequency, the generation of sharp center-surround structures would eventually generate effective reconfigurable spatio-temporal filters [39,40].

The present method remapping plasticity over average depolarization could be automated, causing release probability to change in accordance to specific granule cell response patterns. This could allow generating autonomous machine learning exploiting the computational and plastic properties of cerebellar neuron and synapses.

Acknowledgments. This work was supported by the European Union (SENSOPAC, FP6-IST028056; CEREBNET FP7-ITN238686, REALNET FP7-ICT270434) to Prof. Egidio D'Angelo.

References

1. Hines, M.L., Carnevale, N.T.: The NEURON simulation environment. *Neural Comput.* 9, 1179–1209 (1997)
2. Solinas, S., Nieuwenhuis, T., D'Angelo, E.: A realistic large-scale model of the cerebellum granular layer predicts circuit spatio-temporal filtering properties. *Front Cell Neurosci.* 4, 12 (2010)
3. Eccles, J.C., Ito, M., Szentagothai, J.: *The cerebellum as a neural machine*. Springer, Heidelberg (1967)
4. Eccles, J.C.: The cerebellum as a computer: patterns in space and time. *J. Physiol.* 229, 1–32 (1973)
5. Ito, M.: Cerebellar circuitry as a neuronal machine. *Prog. Neurobiol.* 78, 272–303 (2006)

6. D'Angelo, E.: The critical role of Golgi cells in regulating spatio-temporal integration and plasticity at the cerebellum input stage. *Front Neurosci.* 2, 35–46 (2008)
7. D'Angelo, E., De Zeeuw, C.I.: Timing and plasticity in the cerebellum: focus on the granular layer. *Trends Neurosci.* 32, 30–40 (2009)
8. D'Angelo, E., Koekkoek, S.K., Lombardo, P., Solinas, S., Ros, E., et al.: Timing in the cerebellum: oscillations and resonance in the granular layer. *Neuroscience* 162, 805–815 (2009)
9. D'Angelo, E.: Rebuilding cerebellar network computations from cellular neurophysiology. *Front Cell Neurosci.* 4, 131 (2010)
10. D'Angelo, E., Mazzarello, P., Prestori, F., Mapelli, J., Solinas, S., et al.: The cerebellar network: From structure to function and dynamics. *Brain Res Rev* (2010)
11. D'Angelo, E., De Filippi, G., Rossi, P., Taglietti, V.: Synaptic excitation of individual rat cerebellar granule cells in situ: evidence for the role of NMDA receptors. *J. Physiol.* 484(Pt 2), 397–413 (1995)
12. D'Angelo, E., De Filippi, G., Rossi, P., Taglietti, V.: Ionic mechanism of electroresponsiveness in cerebellar granule cells implicates the action of a persistent sodium current. *J. Neurophysiol.* 80, 493–503 (1998)
13. Chadderton, P., Margrie, T.W., Hausser, M.: Integration of quanta in cerebellar granule cells during sensory processing. *Nature* 428, 856–860 (2004)
14. Rancz, E.A., Ishikawa, T., Duguid, I., Chadderton, P., Mahon, S., et al.: High-fidelity transmission of sensory information by single cerebellar mossy fibre boutons. *Nature* 450, 1245–1248 (2007)
15. Arenz, A., Silver, R.A., Schaefer, A.T., Margrie, T.W.: The contribution of single synapses to sensory representation in vivo. *Science* 321, 977–980 (2008)
16. Mitchell, S.J., Silver, R.A.: Shunting inhibition modulates neuronal gain during synaptic excitation. *Neuron.* 38, 433–445 (2003)
17. Saviane, C., Silver, R.A.: Fast vesicle reloading and a large pool sustain high bandwidth transmission at a central synapse. *Nature.* England, pp. 983–987 (2006)
18. Jorntell, H., Ekerot, C.F.: Properties of somatosensory synaptic integration in cerebellar granule cells in vivo. *J. Neurosci.* 26, 11786–11797 (2006)
19. Sola, E., Prestori, F., Rossi, P., Taglietti, V., D'Angelo, E.: Increased neurotransmitter release during long-term potentiation at mossy fibre-granule cell synapses in rat cerebellum. *J. Physiol.* 557, 843–861 (2004)
20. D'Angelo, E., Nieuws, T., Maffei, A., Armano, S., Rossi, P., et al.: Theta-frequency bursting and resonance in cerebellar granule cells: experimental evidence and modeling of a slow k^+ -dependent mechanism. *J. Neurosci.* 21, 759–770 (2001)
21. Nieuws, T., Sola, E., Mapelli, J., Saftenu, E., Rossi, P., et al.: LTP regulates burst initiation and frequency at mossy fiber-granule cell synapses of rat cerebellum: experimental observations and theoretical predictions. *J. Neurophysiol.* 95, 686–699 (2006)
22. Diwakar, S., Magistretti, J., Goldfarb, M., Naldi, G., D'Angelo, E.: Axonal Na^+ channels ensure fast spike activation and back-propagation in cerebellar granule cells. *J. Neurophysiol.*, 519–532 (2009)
23. Arleo, A., Nieuws, T., Bezzi, M., D'Errico, A., D'Angelo, E., et al.: How synaptic release probability shapes neuronal transmission: information-theoretic analysis in a cerebellar granule cell. *Neural Comput.* 22, 2031–2058 (2010)
24. Dieudonne, S.: Submillisecond kinetics and low efficacy of parallel fibre-Golgi cell synaptic currents in the rat cerebellum. *J. Physiol.* 510(Pt 3), 845–866 (1998)
25. Forti, L., Pietrobon, D.: Functional Diversity Of L-Type Calcium Channels In Rat Cerebellar Neurons. *Neuron.* 10, 437–450 (1993)

26. Aizenman, C.D., Manis, P.B., Linden, D.J.: Polarity of long-term synaptic gain change is related to postsynaptic spike firing at a cerebellar inhibitory synapse. *Neuron*. 21, 827–835 (1998)
27. Solinas, S., Forti, L., Cesana, E., Mapelli, J., De Schutter, E., et al.: Computational reconstruction of pacemaking and intrinsic electroresponsiveness in cerebellar Golgi cells. *Front Cell Neurosci.* 1, 2 (2007)
28. Solinas, S., Forti, L., Cesana, E., Mapelli, J., De Schutter, E., et al.: Fast-reset of pacemaking and theta-frequency resonance patterns in cerebellar golgi cells: simulations of their impact in vivo. *Front Cell Neurosci.* 1, 4 (2007)
29. Kanichay, R.T., Silver, R.A.: Synaptic and cellular properties of the feedforward inhibitory circuit within the input layer of the cerebellar cortex. *J. Neurosci.*, 8955–8967 (2008)
30. D’Angelo, E., Rossi, P., Armano, S., Taglietti, V.: Evidence for NMDA and mGlu receptor-dependent long-term potentiation of mossy fiber-granule cell transmission in rat cerebellum. *J. Neurophysiol.* 81, 277–287 (1999)
31. D’Errico, A., Prestori, F., D’Angelo, E.: Differential induction of bidirectional long-term changes in neurotransmitter release by frequency-coded patterns at the cerebellar input. *J. Physiol.* 587, 5843–5857 (2009)
32. Gall, D., Prestori, F., Sola, E., D’Errico, A., Roussel, C., et al.: Intracellular calcium regulation by burst discharge determines bidirectional long-term synaptic plasticity at the cerebellum input stage. *Journal of Neuroscience* 25, 4813–4822 (2005)
33. Mapelli, J., D’Angelo, E.: The spatial organization of long-term synaptic plasticity at the input stage of cerebellum. *J. Neurosci.* 27, 1285–1296 (2007)
34. Maffei, A., Prestori, F., Rossi, P., Taglietti, V., D’Angelo, E.: Presynaptic current changes at the mossy fiber-granule cell synapse of cerebellum during LTP. *J. Neurophysiol.* 88, 627–638 (2002)
35. Mapelli, J., Gandolfi, D., D’Angelo, E.: Combinatorial responses controlled by synaptic inhibition in the cerebellum granular layer. *J. Neurophysiol.* 103, 250–261 (2010)
36. Mapelli, J., Gandolfi, D., D’Angelo, E.: High-Pass Filtering and Dynamic Gain Regulation Enhance Vertical Bursts Transmission along the Mossy Fiber Pathway of Cerebellum. *Front Cell Neurosci.* 4, 14 (2010)
37. Albus, J.: The theory of cerebellar function. *Math. Biosci.* 10, 25–61 (1971)
38. Marr, D.: A theory of cerebellar cortex. *J. Physiol.* 202, 437–470 (1969)
39. Dean, P., Porrill, J., Ekerot, C.F., Jorntell, H.: The cerebellar microcircuit as an adaptive filter: experimental and computational evidence. *Nat. Rev. Neurosci.* 11, 30–43 (2010)
40. Schweighofer, N., Doya, K., Lay, F.: Unsupervised learning of granule cell sparse codes enhances cerebellar adaptive control. *Neuroscience* 103, 35–50 (2001)

Event and Time Driven Hybrid Simulation of Spiking Neural Networks

Jesus A. Garrido¹, Richard R. Carrillo², Niceto R. Luque¹, and Eduardo Ros¹

¹ Dept. of Computer Architecture and Technology, CITIC,
ETSI Informática y Telecomunicaciones, University of Granada, Spain
{jgarrido,nluque,eros}@atc.ugr.es

² Dept. of Computer Architecture and Electronics, University of Almería, Spain
rcarrillo@atc.ugr.es

Abstract. Emerging research areas in neuroscience are requiring simulation of large and detailed spiking neural networks. Although event-driven methods have been recently proposed to simulate these networks, they still present some drawbacks. To obtain the advantages of an event-driven simulation method and a traditional time-driven method, we present a hybrid method. This method efficiently simulates neural networks composed of several neural models: highly active neurons or neurons defined by very-complex model are simulated using a time-driven method whereas other neurons are simulated using an event-driven method based in lookup tables. To perform a comparative study of this hybrid method in terms of speed and accuracy, a model of the cerebellar granular layer has been simulated. The performance results showed that a hybrid simulation can provide considerable advantages when the network is composed of neurons with different characteristics.

Keywords: spiking neural networks, simulation, time-driven, event-driven.

1 Introduction

Many research projects are studying and modeling nervous circuits of specific brain areas. To accomplish these tasks, well-known simulators such as GENESIS [2] and NEURON [8] have been traditionally used since they provide a good accuracy when simulating detailed biophysical models of neurons. Their main drawback is the low simulation speed, which is caused by the simulation method that they mainly employ: time-driven simulation. This simulation method divides the simulated time into short time steps and in each step the neural state variables are approximated and updated through a numerical analysis method [11]. This iterative processing involves an intensive computation which hinders the simulation of large scale neural networks.

The demand for fast simulations of neural networks has given rise to the application of another simulation method: event-driven simulation. This simulation method only computes and updates the neural state variables when a new event

modifies the normal evolution of a neuron, that is, when an input is received or an output is produced. At the same time, it is known that most information transmission in biological neural circuits is carried out by the so-called spikes. These events are relatively infrequent and localized in time: less than 1% of neurons are simultaneously active [9] and the activity is extremely sparse in many nervous areas such as the granular layer in the cerebellum [4]. This makes event-driven simulation schemes particularly efficient.

Most common event-driven simulators [5,14] use relatively-simple neural models described by equations which can be evaluated repeatedly at arbitrary times (e.g. the Spike-Response model). However, even the limited complexity of these models makes it difficult to predict the future behavior of a neuron, especially to detect the nearest threshold-crossing point that corresponds to the next firing time [10,6]. To mitigate these two limitations (i.e. model-complexity restriction and firing-time prediction which allows a straightforward event-driven simulation) EDLUT (Event-Driven neural simulator based on LookUp tables) was implemented [13]. This application software is an open source project [1] for efficient simulation of biological neural networks. It is of particular interest in the field of neurobotics and embedded neural computing in which real-time processing is required, for example, for experiments which include perception-action loops.

EDLUT uses lookup tables which store all the possible values (with certain precision) of the neural-model state variables [3] in addition to the future states (firing times) [12]. Therefore, a whole neural model is encoded in each set of model-characterization tables. These tables are calculated using time-consuming numerical analysis (e.g. Runge-Kutta method). However, once they are generated, the network simulation can be run efficiently through the event-driven method, just by accessing tables when the neural state must be updated or predicted.

The initial EDLUT's processing scheme allowed the fast simulation of more-complex neural models. Nonetheless, this scheme presented a downside: if the number of state variables of a neural model was relatively large, the size of the corresponding lookup table could be excessive.

In this work, we describe how EDLUT has been upgraded to provide a hybrid time-and-event driven simulation method. It has been proven that this method allows the concurrent simulation of some neuronal model using the event-driven method (the models which can be translated into lookup tables) and other models using the time-driven method in the same network. Finally, we have performed a comparative study of this event-and-time-driven computation method in terms of speed and accuracy.

2 Description of the Time-Driven Simulation Method

In order to achieve a simulation including both event-driven and time-driven methods, the previously-developed Event-Driven LookUp-Table simulator (EDLUT [13]) has been upgraded. The previous architecture was mainly composed of a simulation engine capable of processing spikes. Now, it handles a spike as a

concrete type of event, so that, different classes of events can be processed homogeneously. Thus, the simulation loop could be seen as a simple event handler as shown in algorithm 1.

Algorithm 1. EDLUT Simulation Engine. The simulation algorithm mainly follows an event-driven method. However, the homogeneous event treatment allows the implementation of a time-driven method without modifying this loop. Many new features can be added by means of adding a new class of event which implements the ProcessEvent method.

```

program NeuralSimulator ()
  InitializeSimulation(...);
  repeat
    Event := GetTheNextSimulationEvent();
    SimulationTime := Event.Time;
    Event.ProcessEvent();
    Event.RemoveFromHeap();
  until Heap.IsEmpty();
end.

```

The time-driven events interface with the time-driven simulation loop. Each time that one of this kind of event is caught, all the time-driven neurons are processed and their state is updated. The processing algorithm is mainly composed of 3 different parts:

1. The neural state of each time-driven neuron is updated. This evolution is computed according to the neural model associated to that neuron and the elapsed time from the last state update.
2. If one of those updated neurons fires an spike, this is inserted on the top of the event heap as an internal spike and it will be processed according to its time stamp.
3. Finally, a new time-driven update event is inserted in the event heap. This upcoming event will be handled after the shortest time step of all neurons has been processed (in case of simulating with several time steps).

Also the input spike type has been added to the original EDLUT data structures. Following the same methodology proposed in EDLUT, external inputs reach the network through *stub* neurons. Thus, an incoming spike does not modify the *stub* neuron state, it only generates the corresponding input spikes (propagated spike events [13]) for every neuron which is connected to the output of this neuron. In this way, input spike processing is really simple, it only inserts the first (the earliest) output spikes produced by the input neuron.

Another kind of implemented event is the internal spike. These are directly fired by the dynamics inherent to the neurons (as a response to input conductance or as a consequence of the autonomous activity of the neurons). In these

events, a different processing is carried out depending on whether the neuron model is time-driven or event-driven. The last one updates the neural state and after that, checks if other spike is scheduled for later, and in that case it inserts it in the simulation heap. However, this state-update stage is not necessary when simulating event-driven neural models due to the periodical evolution of the state variables. Finally, the first propagated spike is inserted in the simulation heap.

The last implemented stage of the spike processing is the propagated-spike event which corresponds to one spike being transmitted between neurons. Thus, the input spike will be processed and only if the neural model is event-driven the state will be updated. After that, the next propagated spike (generated by the same source neuron) will be inserted in the simulation heap.

Finally, some other different types of events have been implemented in order to handle different situations which could happen during the simulation (such as communication with external systems following an simulation loop, saving the synaptic weights periodically to explore learning processes...).

3 The Granular Layer Model

Two different alternatives of the same neuron model (Leaky Integrate and Fire) has been implemented and subsequently tested: time-driven and event-driven based in look-up tables. Using these models, an abstraction of the granular layer of the cerebellum has been built. This network is composed of partially overlapped neurons including as a whole 4877 neurons of two different types (see Figure 1).

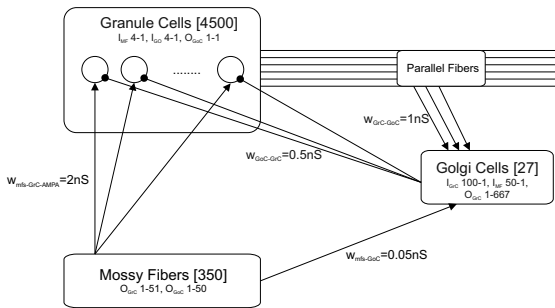


Fig. 1. Simulated architecture of the granular layer. The whole network is composed of 350 mossy fibers (MFs), 4500 granule cells (GrCs) and 27 Golgi cells (GoCs). The convergence/divergence ratios are represented into each layer (e.g. each GrC receives activity from 4 MFs and 4 GoCs, and each GoC receives activity from 100 GrCs and 50 MFs).

3.1 Leaky Integrate-and-Fire Model (LIF)

Both neuronal types (GrCs and GoCs) have been implemented using a Leaky Integrate-and-Fire model [7]. However, different constants have been defined in order to get realistic dynamics according to the cell type. Firstly, this model was strictly implemented following a time-driven scheme. Afterwards, an adaptation process was carried out in order to develop a more efficient event-driven model based in look-up tables.

The state of the neuron is characterised by membrane potential (V_{m-c}) which is expressed by equation (II).

$$C_m \frac{dV_{m-c}}{dt} = g_{AMPA}(t)(E_{AMPA} - V_{m-c}) + g_{GABA}(t)(E_{GABA} - V_{m-c}) + G_{rest}(E_{rest} - V_{m-c}) \quad (1)$$

where C_m denotes the membrane capacitance, E_{AMPA} and E_{GABA} represent the reversal potential of each synaptic conductance and E_{rest} is the resting potential. g_{AMPA} and g_{GABA} conductances integrate all the contributions received through individual synapses and are defined as decaying exponential functions. The parameters of the neuronal model and a more detailed description can be found in [7].

These neuron models account for synaptic conductance changes rather than simply for fixed current flows, providing an improved description over common I&F models. The version of these neural models implemented for EDLUT simulator can be found and downloaded at the EDLUT project official site (see [1]).

LIF models can be directly implemented following event-driven schemes based on lookup tables with only regarding the correctly adjusting the table dimensions. However, once both time-driven and event-driven schemes were implemented, we used the first of them to validate and test the performance of the second one.

4 Hybrid Event-Time Driven Simulation Experiments

A hybrid network including a time-driven model (the GoC model) and an event-driven model (the GrC model) has been built. This hybrid simulation method includes the advantages of the well-controlled accuracy of a time-driven model in GoCs and a fast simulation of GrCs with the event-driven model. Moreover, these advantages are more acute if we take into account the number of GrCs (4500 in our model and more than 50 billions in the human cerebellum) and the small number of GoCs (only 27 in our experiment and also a much smaller number than GrCs in the human brain) which diverge into a wider area in GrCs. These characteristics make the cerebellum more suitable to be simulated using this method.

In order to test the accuracy of the whole system, a complete-time-driven granular layer has been simulated using different time steps and has been compared with a hybrid simulation scheme (event-driven model in GrCs and time-driven

model in GoCs). The mean-absolute error (MAE) of the GrC activity histogram (taking 1-ms bins) has been used as accuracy measure, following a population code strategy. The reference model was a complete time-driven model implementing 4th order Runge-Kutta methods with 0.1 ms time step. The network was stimulated using random inputs at different frequencies, and the MAE has been calculated after simulating the network with each input frequency.

Figure 2a shows the accuracy of the tested schemes. As expected, the error in relation to the 0.1ms-step time-driven model increased as we simulated higher input frequencies. However, this enhancement seems to be more marked in a hybrid scheme (simple line), due to the progressive accumulation of inputs arriving at very near times. However, a hybrid system presented similar precision to a time-driven scheme with 1 ms time step (triangle line). Thus, the time-driven methods with the shortest time steps obtained the better accuracy, while the hybrid model presented reasonably good results (similar to 1ms-step time-driven method, but better than 10ms-step time-driven method -cross line-).

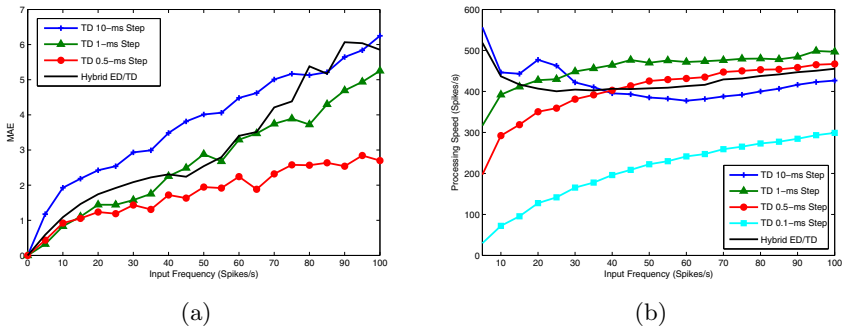


Fig. 2. (a) Obtained error (MAE) after simulating 0.5 s of real activity. Different simulation methods are compared with the reference model (a whole time-driven network using 4th-order Runge-Kutta methods with 0.1 ms step) (b) Simulation speed measured as the rate of processed spikes per second. While hybrid scheme achieves a nearly-constant processing rate due to the intrinsic characteristics of event-driven schemes, time-driven schemes increased their performances as the input frequency increased.

However, an event-driven simulation method based on look-up tables supposed an improvement on the simulation speed specially when processing sparse activity, as it is believed to occur in some layers of the cerebellum (e.g. the parallel fibers) [4]. The performance of the system was measured as the rate of processed spikes per seconds. Using this estimator, we obtained a fair comparison between methods with different levels of activity (due to the accumulation of the error along the simulation).

Figure 2b shows the performance of all the studied systems. The hybrid scheme (simple line), using both time-driven (0.1ms time step) and event-driven

models, kept a nearly constant rate of processed spikes per second due to the intrinsic characteristics of the simulation method. On the contrary, full time-driven networks (square, circle and triangle line) improved their performances as the input frequency increased (more activity). However, this improvement was not completely linear (as expected in a time-driven model) due to the implementation of the method (using an event-driven method as the base of time-driven models). This aspect became more evident when we simulated the time-driven network with 10ms time step (cross line). In this case, the performance is extremely close to the hybrid network because the computational load of the simulation is mainly caused by the event processing (using long time steps remarkably reduces the number of evaluations of the neural-model differential equations).

The enhancement of the performance when using hybrid networks can be noticed if we compare the rate of processed spikes between the hybrid system (simple line) and the 0.1ms-time-step time-driven model (square line). In both models, GoCs are implemented using a 0.1ms time-driven model. However, the difference lies in the model of GrCs. In this case, using an event-driven model (hybrid system) the number of spikes per second approximately doubled the pure event-driven model. Thus, using event-driven models in layer with a huge quantity of neurons markedly accelerates the network simulation.

5 Discussion

In this work, an event-driven simulation environment has been widely improved to make it capable of natively simulating time-driven methods (alone or in conjunction with event-driven methods). This allowed the simulation of hybrid networks where some layers were using time-driven-based models while others were running event-driven-based models.

A correct decision on what simulation method should implement each neuron model could lead to an impressive enhancement of the simulation performance. This decision should take into account some of above commented factors such as the activity rates in which the neural model will be working, the number of neurons to simulate, the influence of each neuron on the rest of the network activity, the possibility of implementing an event-driven alternative, the complexity of the neuronal dynamics, etc...

Event-driven methods (or more generally hybrid networks) showed good performance and accuracy when working with low rates of activity. In the opposite, time-driven model would be preferred when the characteristics of the experiment produces high neural activity. However, as long as different layers in a biological system could present different levels of activity, a hybrid system including both time-driven and event-driven methods can be convenient.

The availability of a hybrid simulation environment also allows the validation of new developed neural models for the event-driven method including this model in the network and comparing its behaviour with the previously implemented time-driven model. In this way, faster and more precise table-based neural models could be designed and subsequently tested in a realistic simulation.

Finally, as future work, the time-driven implementation will include a variable and independent time step for each neuron, speeding up and enhancing the precision of this method.

Acknowledgements. This work was supported in part by the Spanish Subprogram FPU 2009 (MICINN), the Spanish Subprogram Juan de la Cierva 2009 (MICINN), the national projects ITREBA (TIC-5060) and ARC-VISION (TEC2010-15396), and the EU project REALNET (IST-270434).

References

1. Edlut official web site, <http://edlut.googlecode.com>
2. Bower, J.M., Beeman, D.: *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral Simulation System*, 2nd edn. Springer, Heidelberg (1998)
3. Brette, R.: Exact simulation of integrate-and-fire models with synaptic conductances. *Neural Computation* 18, 2004–2027 (2006)
4. Coenen, O.J.-M., Arnold, M.P., Sejnowski, T.J., Jabri, M.A.: Parallel fiber coding in the cerebellum for life-long learning. *Autonomous Robots* 11, 291–297 (2001), <http://dx.doi.org/10.1023/A:1012403510221>, doi:10.1023/A:1012403510221
5. Delorme, A., Gautrais, J., VanRullen, R., Thorpe, S.J.: Spikenet: A simulator for modeling large networks of integrate and fire neurons. *Neurocomputing* 26-27, 989–996 (1999)
6. D’Haene, M., Schrauwen, B., Campenhout, J.V., Stroobandt, D.: Accelerating event-driven simulation of spiking neurons with multiple synaptic time constants. *Neural Computation* 21, 1068–1099 (2009)
7. Gerstner, W., Kistler, W.M.: *Spiking Neuron Models*. Cambridge University Press, Cambridge (2002)
8. Hines, M.L., Carnevale, N.T.: The neuron simulation environment. *Neural Computation* 9, 1179–1209 (1997)
9. Kandel, E.R., Schwartz, J., Jessell, T. (eds.): *Principles of Neuroscience*. Elsevier, Amsterdam (1981)
10. Makino, T.: A discrete-event neural network simulator for general neural models. *Neural Computing and Applications* 11, 210–223 (2003)
11. O’Reilly, R.C., Munakata, Y.: *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*. MIT Press, Cambridge (2000)
12. Reutimann, J., Giugliano, M., Fusi, S.: Event-driven simulation of spiking neurons with stochastic dynamics. *Neural Computation* 15(4), 811–830 (2003), doi:10.1162/08997660360581912
13. Ros, E., Carrillo, R., Ortigosa, E.M., Barbour, B., Agis, R.: Event-driven simulation scheme for spiking neural networks using lookup tables to characterize neuronal dynamics. *Neural Computation* 18, 2959–2993 (2006)
14. Watts, L.: Event-driven simulation of networks of spiking neurons. *Advances in Neural Information Processing Systems* 6, 927–934 (1994)

Author Index

- Ababneh, Jehad II-484
Abengózar Sánchez, J.G. II-308
Aguiar-Pulido, Vanessa II-252
Aguirre, Carlos I-49
Aizenberg, Igor I-33
Alejo, R. I-199
Alhazov, Artiom I-441
Allison, Brendan I-362
Alonso, Izaskun I-256
Alonso, Luis M. II-356
Alonso, Ricardo S. II-661
Alonso-Jordá, Pedro I-409
Álvarez, Daniel I-345
Anagnostou, Miltiades I-113
Andrejková, Gabriela II-145
Angelopoulou, Anastassia II-42, II-58,
II-98, II-236, II-244
Angulo, Cecilio II-581, II-605
Aranda-Corral, Gonzalo A. II-461
Arcay, Bernardino I-273
Arenas, M.G. I-393, I-433, II-316,
II-333, II-341
Arleo, Angelo I-528
Arsene, Corneliu T.C. II-210
Atencia, Miguel II-516
Azzouz, Marouen I-265
- Badillo, Ana Reyes II-284
Bajo, Javier II-661
Baldassarri, Paola II-121
Banos, Oresti II-185
Baños, R. I-73, II-300
Barreira, N. II-66
Barreto, Guilherme A. I-97
Barrios, Jorge II-524, II-540
Barrios, Luis I-370
Becerra, Roberto I-323
Becerra-Alonso, David II-161
Becerra-Bonache, Leonor I-473
Bel-Enguix, Gemma I-441
Benítez-Rochel, Rafaela II-105
Biganzoli, Elia II-210
Black, Norman II-677
Blanca-Mena, María José I-337
- Bojanić, Slobodan I-183
Borrego Díaz, Joaquín II-461
Boufaras, K. I-401
Brasselet, Romain I-528
Brawanski, A. I-299
Bregáins, Julio C. I-520
Briceño, J. II-129
Bueno, G. II-268
- Cabestany, Joan II-557
Cabrera, I.P. II-437
Camargo-Olivares, José Luis II-477
Campos, Doris I-49
Carbonero-Ruz, Mariano II-161
Carrillo, Richard R. I-537, I-554
Cascado, D. I-124
Castellanos, Juan I-307
Castillo, P.A. I-433, II-316, II-333,
II-341
Castrillón, Modesto I-191
Castro, Alfonso I-273
Castro, Paula M. I-489
Català, Andreu II-597
Cazorla, Diego II-380
Cazorla, Miguel II-9, II-50
Cerezuela-Escudero, E. II-548
Charte, F. I-41
Cheung, Willy I-362
Chung, Mike I-362
Civit, A. I-157
Cléménçon, Stéphan II-276
Colla, Valentina I-57, I-256
Comas, Joan II-605
Corchado, Juan M. II-629
Cordero, P. II-412, II-437
Cornelis, Chris I-174
Coronado, J.L. I-124
Corralejo, Rebeca I-345
Corrales-García, Alberto I-497
Cotta, Carlos II-284, II-308, II-348
Couce, Yasel II-202
Crassidis, John II-621
Cruz Echeandía, M. I-457
Cruz-Ramírez, M. II-129

- Cuartero, Fernando II-380
 Cuenca, Pedro I-497
- Dahl, Veronica I-449
 Damas, Miguel II-185
 Danciu, Daniela II-565, II-573
 D'Angelo, Egidio I-547
 d'Anjou, Alicia II-83
 Dapena, Adriana I-489, I-520, II-500
 de Arazoza, Héctor II-276, II-524,
 II-540
 de Armas, Jessica II-292
 de la Encina, Alberto II-388
 de la Mata, M. II-129
 de la Prieta, Fernando II-669
 del Campo-Ávila, José II-137
 del Castillo, M. Dolores I-370
 del Jesus, María Jose I-41
 del Saz-Orozco, Pablo I-315
 Del Ser, Javier I-17
 Déniz, O. II-268
 De Paz, Juan F. II-629, II-669
 Derderian, Karnig II-396
 Derrac, Joaquín I-174
 de Toro, Francisco I-105
 Díaz, Antonio F. I-232
 Diaz-del-Rio, F. I-133
 Díaz Martínez, Miguel A. I-329
 Diaz-Rubio, Eduardo II-260
 Diez Dolinski, L. I-457
 Domínguez, Enrique II-1, II-17, II-98
 Domínguez-Morales, M. I-124, II-548
 Doquire, Gauthier I-9, I-248
 Dragoni, Aldo Franco II-121
- Edlinger, G. I-386
 Eduardo, Ros II-90
 Egecioğlu, Ömer I-465
 Enciso, M. II-412
 Escalera, Sergio II-581
 Escuín, Alejandro I-291
 Eugenia Cornejo, M^a II-453
- Faltermeier, R. I-299
 Faundez-Zanuy, Marcos II-220
 Fernandes, C.M. II-325
 Fernández, A. II-300
 Fernández, José M. II-637
 Fernández, M. II-268
 Fernández-Ares, A. II-325
- Fernández-Caramés, Tiago M. II-500
 Fernandez de Canete, Javier I-315
 Fernández de Vega, F. II-308
 Fernández-Leiva, Antonio J. II-284,
 II-348
 Fernández López, Pablo II-169
 Florentín-Núñez, María Nieves II-34
 Florido, J.P. II-194
 Franco, Leonardo II-202
 Friedrich, Elisabeth C.V. I-362
 Fuentes-Fernández, Rubén II-637
- Galán Páez, Juan II-461
 Galindo, Pedro L. I-291
 Gallardo-Estrella, L. I-240
 Gallego, Juan Álvaro I-370
 Galuszka, Adam II-613
 García, Elena II-629
 García, Jesús II-621, II-653
 Garcia, Jose II-9, II-50
 García, Ricardo I-505
 García, Rodolfo V. I-323
 García, Salvador I-174
 García, V. I-199
 García Arroyo, José Luis II-74
 García Báez, Patricio II-169
 García Zapirain, Begoña I-265, II-74
 García-Chamizo, Juan Manuel II-58,
 II-98
 García-Córdova, Francisco I-166
 García Rodríguez, José II-236, II-244
 Garcia-Moral, Inmaculada I-315
 García-Naya, José A. II-500
 García-Rodríguez, José II-58, II-98
 García-Rojo, M. II-268
 García-Sánchez, P. II-316, II-325
 Garrido, Jesús A. I-537, I-554
 Gascón-Moreno, J. I-25, II-113, II-153
 Gautero, Fabien I-65
 Ghaziasgar, Mehran I-215
 Gil, C. I-73, II-300
 Gil-Lopez, Sergio I-17
 Gómez, J. I-73
 Gómez, Sandra I-307
 Gómez-Pulido, Juan A. II-364, II-372
 Gómez-Rodríguez, Francisco I-133,
 I-157
 González, Jesús I-323
 González-Álvarez, David L. II-372
 González Linares, José María I-513

- González-López, Miguel II-500
 Gracia, Jesús II-645
 Graña, Manuel II-83
 Gräser, Axel I-353
 Grassi, Marco II-220
 Grzejszczak, Tomasz II-613
 Guerrero, Elisa I-291
 Guerrero-González, Antonio I-166
 Guger, C. I-386
 Guil Mata, Nicolás I-513
 Guil, Nicolás I-520
 Guillén, A. I-393
 Gutiérrez, P.A. II-129, II-177
 Guzmán, I. Pérez de II-412
- Haddadi G., Ataollah I-207
 Hegedüs, László I-465
 Hernández, Daniel I-191
 Herrera, Francisco I-174
 Herrera, L.J. I-393
 Herrero-Carrón, Fernando II-532
 Hervás-Martínez, C. II-129, II-177
 Hidalgo-Herrero, Mercedes II-388
 Hierons, Robert M. II-396, II-404
 Hintermüller, C. I-386
 Hornero, Roberto I-345
 Hornillo-Mellado, Susana II-477
 Hsieh, Ying-Hen II-524
 Hwang, Chih-Lyang I-223, II-25
- Ibáñez, Jaime I-370
 Igual, Carmen II-484
 Igual, Jorge II-484
- Javier, Díaz II-90
 Jerez, José M. II-202
 Jimenez, G. I-124, I-133, I-157, II-548
 Jimenez-Fernandez, Angel I-124, I-141, II-548
 Jiménez-López, M. Dolores I-481
 Jimenez-Moreno, Gabriel I-149
 Jin, Lizuo II-228
 Johansson, Roland S. I-528
 Joya, Gonzalo II-516, II-540
 Joya Caparrós, Gonzalo I-329
 Julián-Iranzo, Pascual II-421, II-429
- Kaviani, Nima I-449
 Krassovitskiy, Alexander I-441
- Labrador, Josmary I-489
 Lamp, Torsten I-256
 Landa-Torres, Itziar I-17
 Lang, E.W. I-299
 Laredo, J.L.J. II-316, II-333
 León, Coromoto II-292
 Linares-Barranco, A. II-548
 Linares-Barranco, Alejandro I-124, I-141, I-149, I-157
 Linares-Barranco, Bernabe I-141
 Lisboa, Paulo J. II-210
 Litovski, Vančo I-183
 Linares, Raul II-484
 Llinas, James II-621
 López, Otoniel I-505
 López-Alomso, Victoria II-260
 López-Campos, Guillermo H. II-260
 Lopez-Gordo, M.A. I-378
 López-Rubio, Ezequiel II-17, II-34
 López-Rubio, Francisco Javier II-34
 Lorenzo, Javier I-191
 Lounes, Rachid II-524
 Lu, Kai-Di II-25
 Luong, T.-V. I-401
 Luque, Niceto R. I-537, I-554
 Luque, R.M. II-1, II-17
- Madani, Kurosh I-65, I-81
 Malumbres, Manuel P. I-505
 Manjarres, Diana I-17
 Márquez, A.L. I-73, II-300
 Marrero, Aymée II-540
 Martí, Antonio I-505
 Martí, Enrique II-653
 Martín, Beatriz II-669
 Martín-Clemente, Rubén II-477
 Martínez, José Luis I-497
 Martínez-Estudillo, Alfonso Carlos II-161
 Martínez-Estudillo, Francisco José II-161
 Martín-Merino, Manuel I-89
 Martín-Sánchez, Fernando II-260
 Matarese, Nicola I-256
 McCullagh, Paul II-677
 Medina, Jesús II-429, II-453
 Mekyska, Jiri II-220
 Melab, N. I-401
 Méndez, Juan I-191
 Méndez Zorrilla, Amaia I-265, II-74

- Mentzelopoulos, Markos II-42
 Merayo, Mercedes G. II-396
 Merelo, J.J. II-316, II-325, II-333, II-341
 Mikulka, Jan II-220
 Milojković, Jelena I-183
 Miranda, Gara II-292
 Miró-Amarante, L. I-133
 Molina, José Manuel II-621, II-645, II-653
 Molinero, Carlos II-404
 Montero-Gonzalez, Rafael J. I-141, I-149
 Montoya, F.G. I-73
 Montoya, M.G. I-73, II-300
 Moor, Anton I-353
 Mora, A.M. I-433, II-316, II-325, II-333, II-341, II-412
 Mora-Gimeno, Francisco José II-98
 Morales-Bueno, Rafael II-137
 Morcillo, Pedro J. II-429, II-445
 Morell, Vicente II-58
 Moreno, Ginés II-429, II-445
 Moreno, Juan Manuel II-557
 Moreno, Ramón II-83
 Moreno Arostegui, Juan Manuel II-589
 Moreno, David I-307
 Morgado, A. I-157
 Morgado-Estevez, Arturo I-141, I-149
 Morgado-León, Agustín I-291
 Morillas, Christian I-417
 Mosquera, A. II-66
 Muñoz, J.L. I-124, II-1
 Muñoz-Pérez, José II-105
 Munteanu, Cristian R. II-252

 Nagy, Benedek I-465
 Neuper, Christa I-362
 Nieto-Taladriz, Octavio I-183
 Novo, J. I-282
 Nugent, Chris II-677
 Núñez, Manuel II-396, II-404
 Núñez Hervás, R. I-457

 Ojeda-Aciego, M. II-429, II-437
 Olivier, Paul II-589
 Oravec, Jozef II-145
 Ortega, A. I-457
 Ortega, J. II-300
 Ortega, Julio I-232
 Ortiz, Andres I-232

 Ortiz-de-Lazcano-Lobato, J.M. II-17
 Ortiz-García, E.G. II-113, II-153
 Orts, Sergio II-58, II-98
 Ortuño, F. II-194

 Pablo, Guzmán II-90
 Palacios, Juan I-256
 Palomo, E.J. II-1, II-17
 Pan, Hong II-228
 Pani, Tommaso I-105
 Paniagua-Tineo, A. I-25, II-113, II-153
 Papaioannou, Ioannis I-113
 Pardo, Diego II-605
 Pascual, Pedro I-49
 Patricio, Miguel A. II-645
 Pavón, Juan II-637
 Paz, R. I-124
 Pazos, Alejandro II-252
 Paz-Vicente, R. II-548
 Peinado-Pinilla, Jesús I-409
 Pelayo, Fernando L. II-380
 Pelayo, Francisco J. I-417
 Pelayo Valle, Francisco I-378
 Penabad, Jaime II-445
 Penas, M. II-66
 Penedo, M.G. I-282
 Pérez, Carlos II-597
 Pérez-Arjona, Isabel I-409
 Perez-Carrasco, Jose Antonio I-141
 Pérez-García, Jesús II-137
 Pérez-Godoy, M.D. I-41
 Pérez-Iglesias, Héctor J. I-489
 Perez-Peña, Fernando I-141, I-149
 Perez-Sala, Xavier II-581
 Pérez-Villamil, Beatriz II-260
 Pianezzola, Marco I-256
 Piétrus, Alain II-540
 Pilar, Ortigosa II-90
 Pomares, Hector I-393, II-185, II-194
 Poncela, A. I-240
 Portilla-Figueras, Jose A. I-17, I-25, II-113, II-153
 Pozo, Alberto II-645
 Prieto, Alberto I-232
 Pröll, Markus I-362
 Psarrou, Alexandra II-42, II-58, II-98, II-236, II-244
 Puntonet, C. I-299

 Quiles, Francisco José I-497

- Rabanal, Pablo II-356, II-388
 Ramík, Dominik M. I-81
 Ramírez, Eloisa II-453
 Ramos, L. II-66
 Ramos Cózar, Julián I-513
 Ramos-Jiménez, Gonzalo II-137
 Rao, Rajesh P.N. I-362
 Räsvan, Vladimir II-565
 Remeseiro, B. II-66
 Rey, Alberto I-273
 Reyneri, Leonardo I-57
 Rivas, M. I-133
 Rivas-Perez, Manuel I-157
 Rivera, A.J. I-41
 Rocha Neto, Ajalmar R. I-97
 Rocon, Eduardo I-370
 Rodrigo, Agis II-90
 Rodríguez, Francisco B. I-1, II-532
 Rodríguez, Ismael II-356
 Rodríguez, Roberto I-323
 Rodríguez, Sara II-629, II-661
 Rodríguez-Corral, Jose Maria I-149
 Rodríguez-Jiménez, J.M. II-412
 Rogozhin, Yurii I-441
 Rojas, Fernando I-323
 Rojas, Ignacio I-393, II-185, II-194
 Romera-López, Alejandro II-260
 Romero, G. I-433, II-316, II-333, II-341
 Romero, Samuel I-417
 Ron-Angevin, Ricardo I-337, I-378
 Ros, Eduardo I-537, I-554
 Rosenhahn, Bodo I-425
 Rossi, Fabrice II-276
 Roussaki, Ioanna I-113
 Rozado, David I-1
 Rubio, Fernando II-388
 Rubio-Largo, Álvaro II-364
 Rubio-Manzano, Clemente II-421
 Ruiz, Francisco J. II-597
 Ruiz, Ibon I-265
 Ruiz-Sepúlveda, Amparo II-105

 Saavedra-Moreno, B. I-25, II-113, II-153
 Sabourin, Christophe I-65, I-81
 Safont, Gonzalo II-469, II-508
 Sahebi, Mahmoodreza I-207
 Salazar, Addisson II-469, II-508
 Salcedo-Sanz, Sancho I-17, I-25, II-113,
 II-153
 Samà, Albert II-597
 Sancha-Ros, Salvador I-337
 Sánchez, Andres II-524
 Sánchez-Morcillo, Victor J. I-409
 Sánchez-Pérez, Juan M. II-364, II-372
 Sanchis, Lorenzo I-291
 Sandoval, Francisco II-516
 Santos, J. I-282
 Scherer, Reinhold I-362
 Schiewe, Siegfried I-256
 Seoane, José A. II-252
 Serrano, Eduardo I-49
 Serrano, J. Ignacio I-370
 Silva-Sauer, Leandro da I-337
 Singh, Tarunraj II-621
 Sistachs Vega, Vivian I-329
 Skvortsov, Evgeny I-449
 Smirg, Ondrej II-220
 Solinas, Sergio I-547
 Soto, Javier II-557
 Sotoca, J.M. I-199
 Sovilj, D. I-393
 Stephens, Gerard I-256
 Suárez Araujo, Carmen Paz II-169
 Subirats, José L. II-202

 Talbi, E.-G. I-401
 Tapia, Dante I. II-661
 Tavakoli Naeini, Armin I-215
 Tomé, A.M. I-299
 Tran, Viet-Chi II-276

 Urda, Daniel II-202
 Ureña, Raquel I-417
 Urquiza, J.M. II-194

 Valdovinos, R.M. I-199
 Valenzuela, Olga I-323
 Vallesi, Germano II-121
 van Heeswijk, M. I-393
 Vannucci, Marco I-57, I-256
 Varona, Pablo I-1, II-532
 Vasconcelos, Cristina Nader I-425
 Vázquez, Carlos II-445
 Vega-Rodríguez, Miguel A. II-364,
 II-372
 Velasco-Álvarez, Francisco I-337
 Velázquez, Luis I-323
 Vergara, Luis II-469, II-508

Verleysen, Michel I-9, I-248

Viejo, Diego II-9, II-50

Villalba Espinosa, Juan I-513

Volosyak, Ivan I-353

Wang, Ting I-65

Xia, Liangzheng II-228

Xia, Siyu II-228

Yáñez, Andrés I-291

Yang, Chen-Han I-223

Yebra-Pimentel, E. II-66

Zato, Carolina II-669

Zdunek, Rafal II-492

Zeiler, A. I-299

Zhang, Shumei II-677

Zheng, Huiru II-677