Ambient detection for distributed speech recognition

Jose A. Gonzalez, Angel M. Gómez, Antonio M. Peinado, Jose L. Carmona
Dpto. de Teoría de la Señal, Telemática y Comunicaciones, University of Granada, Spain
{joseangl,amgg,amp,maqueda}@ugr.es

ABSTRACT
This paper describes a framework for distributed speech recognition (DSR) that exploits the information about the acoustic environment in order to improve the system performance. At the client side, the ETSI front-end is used. At the server, we propose an acoustic ambient detector that identifies the acoustic environment so that the recognition engine can employ a set of acoustic models which match the identified environment. The experimental results show that the proposed framework outperforms a system using the ETSI advanced front-end (AFE) with multicondition-trained acoustic models both in well-matched and mismatched conditions. Furthermore, our proposal maintains a more light-weighted front-end, which enables DSR in resource-limited clients (cellular phones or PDAs).

Categories and Subject Descriptors
H.5 [**INFORMATION INTERFACES AND PRESENTATION**]: User Interfaces—*Voice I/O*

General Terms
Performance

Keywords
Distributed speech recognition, robust speech recognition, noise classification, context-awareness.

1. INTRODUCTION
In the last years, the proliferation of mobile devices with Internet access has caused a revolution in our lives. Currently, people want to be connected anywhere, anytime in order to access information. However, the access to information services using the traditional limited and small interfaces (displays, keyboards, stylus, ...) is cumbersome for many people. A possible solution for this problem is the use of speech interfaces [1, 2], which improve the usability by reducing the complexity of the interaction and are more user-friendly.

Unfortunately, a speech recognition system embedded in a typical mobile device (phone, PDA,...) can be severely limited. The paradigm known as distributed speech recognition (DSR) [3] alleviates this problem by distributing the recognition tasks amongst the user terminal and a remote server. Under this approach, the user device extracts and encodes a parametrized representation of speech which is suitable for recognition. Then, the speech features are transmitted over the network to a remote back-end where recognition is performed. This client-server architecture is very suitable for packet networks and allows speech-enabled services for PDAs, portable thin clients and other handheld devices connected to Internet. Moreover, powerful centralized recognizers could be shared between multiple users and easily upgraded with new technologies and services.

The European Telecommunications Standard Institute (ETSI) has proposed two standards (and their corresponding extended versions) for the implementation of DSR front-ends. The first one, usually referred to as ETSI front-end (FE, ETSI 201 108 [4]), carries out a usual cepstral analysis scheme where 13 Mel frequency cepstral coefficients (MFCCs) are calculated along with the frame energy as fourteenth parameter. The second one, named ETSI advanced front-end (AFE, ETSI 202 050 [5]), extends the ETSI FE by including additional processing blocks for reducing the influence of acoustic noise. However, the complexity of the AFE front-end can be excessive for many of the available mobile devices in terms of computational complexity and power consumption [6].

In addition to the restricted performance of the mobile terminals, one of the major problems of DSR systems is the context where they work. As opposed to the traditional automatic speech recognition (ASR) systems, the context in DSR systems generally changes rapidly in terms of acoustic noise, network quality, location, etc. This means that systems that automatically adapt their operation to the characteristics of the context become necessary [7].

With the deployment of sensor networks and pervasive devices, more and more environmental information can be regularly monitored. The use of this information in *context-aware* systems [7]-[11], make them more robust since they can react to the environmental changes. In particular, information like user gender, network and device capacities, time, location, acoustic noise, etc., can improve the robustness of the ASR systems.

In the ASR literature, much of the work to adapt the system to the environmental changes has focused on the ambient noise. It is well known that the performance of the ASR systems degrades when the training and testing acoustic conditions are different. This mismatch has been traditionally reduced by means of different signal processing and compensation techniques. An alternative introduced along the last years is that of employing environmental information in the recognition engine (back-end) [12, 13].

In this paper, we consider the problem of the acoustic ambient mismatch under this new approach. Our goal is to improve the recognition accuracy in noisy environments of current DSR systems by increasing only the complexity of the back-end. Thus, simple front-ends can be used in low-end mobile devices with restrictions in power consumption or CPU performance. In particular, we propose an acoustic ambient detector which identifies the acoustic ambient and its corresponding signal-to-noise ratio (SNR) value using only the received speech features. According to this information, a suitable set of acoustic models is selected for recognition.

This paper is organized as follows. In section 2 we describe the overall architecture of the proposed framework, as well as the design of the acoustic ambient detector and the speech recognizer. The experimental framework is described in section 3 while the recognition results are presented in section 4. Finally, conclusions are presented in section 5.

2. SYSTEM ARCHITECTURE

The architecture of the proposed system is shown in figure 1. On the client side, the ETSI FE front-end segments the speech signal into overlapped frames of 25 ms every 10 ms. Each speech frame is represented by a 14-dimensional feature vector containing 13 MFCCs (including the 0th order one) plus the log-Energy (logE). These features are grouped into pairs and quantized by means of seven Split Vector Quantizers (SVQ). All codebooks have a 64-center size (6 bits), except the one for MFCC-0 and logE, which has 256 centers (8 bits). In the server, the feature vectors are decoded and extended with their first and second derivatives.

The server consists of an acoustic ambient detector and a speech recognizer. The purpose of the ambient detector is to categorize the acoustic ambient which is contaminating the speech signal in terms of the ambient type (e.g. *babble*, *car*, *street*, ...) and the corresponding SNR value. Finally, the speech recognizer uses this information to select a set of acoustic models which match (approximately) the current ambient and performs speech recognition.

2.1 Ambient detector

As in [11, 12], we are interested in identifying ambients of the daily life such as car, babble, street, etc. Acoustic ambient recognition is very different in nature to speech recognition. Speech signals are produced from a single point source and are reasonably well modelled. On the contrary, environmental noise is a complex signal originated by different noise sources. In addition, each of them can include from fully random events, e.g. thunders in a storm, to quasi-predictable ones, e.g. computer fan. These and other reasons justify the difficulty of modelling the ambient noise.

In general, ambient detection can be considered an optimization problem where we try to maximize a criterion function $\phi(\cdot)$ as follows,

$$\widehat{\alpha} = \underset{1 \le l \le \mathcal{A}}{\operatorname{argmax}} \, \phi\left(\alpha_l; X, I\right) \tag{1}$$

where $\{\alpha_1, ..., \alpha_{\mathcal{A}}\}$ is the set of possible ambients, X is the sequence of feature vectors, and I is any available a priori information such as the a priori probability for each ambient type, position, date, time, etc. In the literature, the ambient detection has been traditionally carried out by modelling each ambient α_l with an hidden Markov model (HMM) [14] Θ_l. Using these models, the ambient detection can be seen as to find the ambient model that maximizes the posterior probability,

$$\widehat{\alpha} = \underset{1 \le l \le \mathcal{A}}{\operatorname{argmax}} \, p(\Theta_l|X) = \underset{1 \le l \le \mathcal{A}}{\operatorname{argmax}} \, \frac{p(X|\Theta_l) \cdot p(\Theta_l)}{p(X)} \tag{2}$$

where $p(X|\Theta_l)$ is the probability of the sequence of feature vectors X given the ambient model Θ_l, $p(\Theta_l)$ is the a priori probability of the ambient α_l, and $p(X)$ is the probability of the sequence of feature vectors (this probability is considered a constant).

As mentioned before, in this paper we consider that every ambient α consists of an acoustic ambient type a and a SNR value. In order to categorize the ambient of the current utterance, the received feature vectors are processed by several blocks included in the ambient detector, namely: a voice activity detector (VAD), an SNR estimator, an acoustic ambient recognizer, and a decision block. The proposed ambient detector is shown in figure 1.

First, the received sequence of feature vectors is classified and segmented by the VAD. The resulting speech and non-speech segments will be later used by the other blocks of the ambient detector. The VAD has been implemented by means of a support vector machine (SVM) [15]. A Radial Basis Function (RBF) kernel is used in the SVM training process. No additional parameters are required, the SVM-VAD training and classification is performed using only the feature vectors extracted by the ETSI FE.

Using the information provided by the VAD, the SNR estimator carries out an estimation of the SNR of the speech. Under the assumption that the acoustic noise is additive, the SNR estimation can be computed as,

$$\widehat{SNR} = 20 \log_{10} \frac{P_x}{P_n} \simeq 20 \log_{10} \frac{P_y - P_n}{P_n} \tag{3}$$

where P_x, P_n, and P_y are the powers of the clean speech, the noise, and the contaminated speech, respectively. Since the feature vectors received by the server correspond to contaminated speech, the SNR estimator must use the information returned by the VAD and the log-Energy (logE) parameters to compute the previous powers as,

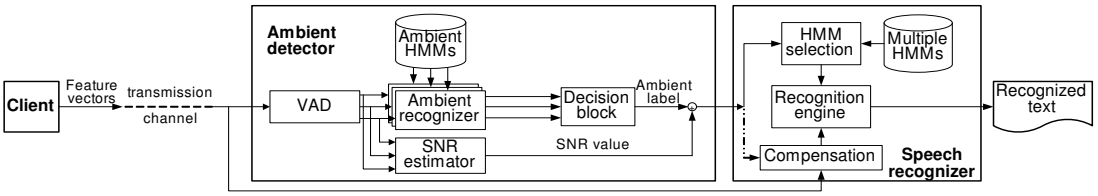

Figure 1: Overview of the proposed framework.

$$P_y \simeq \frac{1}{\mathcal{S} \cdot \mathcal{F}} \sum_{i \in \Psi} e^{logE_i} \tag{4}$$

$$P_n \simeq \frac{1}{\mathcal{N} \cdot \mathcal{F}} \sum_{i \notin \Psi} e^{logE_i} \tag{5}$$

where Ψ is the set of speech frames detected by the VAD in the utterance, $\mathcal{S}$ is the number of speech frames, $\mathcal{N}$ is the number of non-speech frames, and $\mathcal{F}$ is the number of samples in each frame (200 samples at 8 KHz in the ETSI FE).

In order to obtain an estimated ambient type $\widehat{a}$, an ambient recognizer based on HMMs is used. The ambient recognizer considers a total number $\mathcal{A}$ of possible ambient types, thus, for each acoustic ambient $1 \leq j \leq \mathcal{A}$, an HMM $\widehat{\lambda}_j$ has been trained. An ergodic topology (i.e. fully connected) is selected to model the random evolutionary nature of the ambients.

Using this set of HMMs, the non-speech feature vector segments $\boldsymbol{\sigma} = (s_1, s_2, ..., s_{\mathcal{T}})$ extracted by the VAD (with lengths $(f_1, f_2, ..., f_{\mathcal{T}})$ in number of frames) are classified. For each segment, s_t, the recognizer returns a vector of log-probabilities (scores),

$$\mathbf{log}\, \boldsymbol{P_t} = \left(\log p_t^1, \log p_t^2, ..., \log p_t^{\mathcal{A}}\right) \tag{6}$$

where $\log p_t^j$ is the score of the non-speech segment s_t for the ambient model $\widehat{\lambda}_j$, i.e. $\log p_t^j = \log\left(p(s_t|\widehat{\lambda}_j)\right)$. We will also define the most likely ambient type (MAP estimate) for the segment s_t as,

$$a_t^* = \underset{1 \leq j \leq \mathcal{A}}{\operatorname{argmax}}\, p\left(\widehat{\lambda}_j|s_t\right) = \underset{1 \leq j \leq \mathcal{A}}{\operatorname{argmax}} \log p_t^j \tag{7}$$

where we have assumed equiprobable acoustic ambient types.

As can be seen, the most likely ambient a_t^* can be different for each non-speech segment s_t. Therefore another block in the ambient detector should process all the information available to finally decide which ambient noise is contaminating the input signal (decision block in figure 1). We have tested three alternative decision rules for this decision block:

- *Mode*: the label corresponding to the most frequent recognized ambient is returned, where only the most likely ambient labels $(a_t^*, t = 1...\mathcal{T})$ for each segment are considered,

$$\widehat{a} = mode(\sigma) = \underset{1 \leq j \leq \mathcal{A}}{\operatorname{argmax}}\, count(j) \tag{8}$$

  where $count(j)$ returns how many times the ambient j has been recognized as the most likely ambient type, that is,

$$count(j) = \sum_{t=1}^{\mathcal{T}} \delta(j, a_t^*) \tag{9}$$

  where $\delta(x, y)$ is the Kronecker delta function.

- *Maximum length* (ml): considering only the most likely ambient labels $(a_t^*, t = 1...\mathcal{T})$ for each non-speech segment, this function returns the label of the ambient type that maximizes (over all possible ambients) the sum of the segment lengths recognized as a given ambient,

$$\widehat{a} = ml(\sigma) = \underset{1 \leq j \leq \mathcal{A}}{\operatorname{argmax}} \sum_{t=1}^{\mathcal{T}} f_t \cdot \delta(j, a_t^*) \tag{10}$$

- *Maximum score per frame* (msf): this function maximizes a combination of the segment lengths and recognition scores as follows,

$$\widehat{a} = msf(\sigma) = \underset{1 \leq j \leq \mathcal{A}}{\operatorname{argmax}} \sum_{t=1}^{\mathcal{T}} \left(e^{\overline{\log p_t^j}}\right)^{\frac{1}{f_t}} \tag{11}$$

  where $\overline{\log p_t^j}$ is a normalized score (explained below). The idea is to give more weight to the longest segments since they are expected to be more reliably identified.

  The reason why a normalized score is used is twofold. First, we equalize the values of the recognition scores $\log p_t^j$ since they strongly depend on the segment length. Also, we avoid underflows when these scores are very small. Here we propose the use of the following normalized score,

$$\overline{\log p_t^j} = \log p\left(\widehat{\lambda}_j|s_t\right) = \log p_t^j - \log\left(\sum_{i=1}^{\mathcal{A}} e^{\log p_t^i}\right) \tag{12}$$

2.2 Speech recognizer

In our framework, the acoustic models used for speech recognition are obtained using a Multiple-model training paradigm [13]. Under this paradigm, different HMM sets are trained for a number of ambients. Every ambient is defined by an ambient type and a SNR value. We note the corresponding HMM model set as $\Lambda_j^{SNR_i}$ $(i = 1...\mathcal{M}, j = 1...\mathcal{A})$, in which we have considered $\mathcal{M}$ different SNR values and $\mathcal{A}$ different acoustic ambient types.

The main idea in the Multiple-model paradigm is that the most suitable HMM set is selected to perform recognition. In our proposal, this set contains the HMMs trained for the recognized ambient type and the $SNR_i (i = 1...\mathcal{M})$ value nearest to the SNR obtained from the log-Energies (eq. (3)). Finally, as appears in figure 1, the speech features can be also compensated before recognition using different techniques.

3. EXPERIMENTAL FRAMEWORK

The experimental setup is based on that proposed by ETSI STQ-Aurora working group using the Aurora-2 database [16]. This database consists of utterances of connected digits. The vocabulary is made up of 11 digits between 0 and 9 (zero has two sound descriptions: 'zero' and 'o'). For our purposes, we have extracted the clean training subset and the clean utterances from the test set A of this database.

A set of 9 ambient noises is chosen, namely: airport, highway, babble, bar, car race, stadium, slight swell, restaurant, and train station. Each noise in this set is split into two parts: two-thirds are employed to train the multiple-model HMMs while the other third is reserved for testing. The training part of the 9 ambient noises has been added to the *Clean* training set of Aurora-2 at 7 different SNRs (clean, 20, 15, 10, 5, 0, and -5 dB), resulting in 63 training conditions. Each condition have been used to train a set of HMMs.

In each training condition, every digit is modeled by one HMM model with 16 states and 3 Gaussian mixtures per state. Furthermore, two pause (silence) models are defined. The first one consists of 3 states with a mixture of 6 Gaussian per state. This HMM models the pauses before and after the utterance. The second one models pauses between words. It consists of a single state which is tied with the middle state of the first pause model. During training and testing, the speech features provided by the ETSI FE are extended with their first and second derivatives.

The SVM-based VAD is trained using the LIBSVM software tool [17]. This SVM-VAD operates on ETSI FE feature vectors. Previously, every feature vector from every utterance in the clean training set of Aurora-2 has been classified with a simple energy-based VAD such as the one included in the ETSI AFE. Then, the SVM is trained using a set of 5 randomly selected utterances for each condition (ambient type, SNR). Since our SVM-VAD achieved better performance, we chose it instead of the ETSI AFE VAD.

The ambient recognizer employs 3 states and 9 Gaussians per HMM state, and has been trained using non-speech feature vectors in which the energy parameters have been excluded (log energy, 0th order MFCC and their corresponding first and second derivatives). This exclusion tries to prevent a possible mismatch between training and testing conditions due to differences in SNR levels.

	Well-matched test set	**Mismatched test set**
FE	90.13	83.50
AFE	**92.86**	**89.63**

Table 1: Recognition accuracy (WAcc (%)) for the ETSI FE and AFE front-ends for the well-matched and mismatched test sets using multicondition training.

	SNR thresholds						
	-5	*0*	*5*	*10*	*15*	*20*	*clean*
FE	94.17	94.17	**94.49**	93.10	90.14	86.02	69.52
AFE	94.26	94.26	**94.68**	94.40	93.48	92.38	87.56

Table 2: Recognition accuracy (WAcc (%)) for the ETSI AFE and FE front-ends using multiple-model training (with different SNR thresholds) for well-matched test set.

Two different test sets are defined. In the first one, called *well-matched test set*, the testing part of the 9 ambient noises defined above is added to the clean utterances of the Aurora-2 test set A, at the same 7 SNR values considered in training. The second test set, called *mismatched test set*, is created in exactly the same way, but using four different ambient noises, namely pedestrian street, bus station, heavy sea, and slip road to a highway, and 5 different SNRs (17.5, 12.5, 7.5, 2.5, and -2.5 dB).

Finally, for comparing purposes, multicondition acoustic models [16] has been also evaluated. Multicondition training is the usual way of reducing the mismatch between training and testing, and provides much better results than employing clean speech for training. In order to obtain the multicondition HMMs, the files of the Clean training set of Aurora-2 are split in 45 subsets. According to [16], each subset is contaminated with one of 9 training acoustic ambients at a chosen SNR from five available (clean, 20, 15, 10 and 5 dB). These HMMs are modelled with the same number of states and Gaussian mixtures as in the Multiple-model training described above.

4. EXPERIMENTAL RESULTS

As a reference, table 1 shows the Word Accuracy (*WAcc* (%)) results achieved by two multicondition-based systems for the aforementioned testing sets, *well-matched* and *mismatched*. The first system, named as FE, uses the features extracted by the ETSI FE. The second one, named as AFE, uses the ETSI AFE features. In both systems, the speech parameters are compensated applying cepstral mean normalization (CMN) [3]. As expected, AFE is more robust than FE. This behaviour is specially noticeable for the *mismatched test set*.

In a perfect matching situation, each utterance should be recognized using an HMM set trained under the same acoustic ambient and SNR. The results obtained with this ideal system can be considered an upper limit for our proposal.

Train \ *Test*	airport	highway	babble	bar	car race	stadium	sligth swell	restaurant	train
airport	**63.34**	0.00	0.00	1.63	0.00	0.00	0.00	35.03	0.00
highway	0.02	5.93	0.00	4.40	1.23	**65.15**	22.86	0.23	0.18
babble	1.25	0.00	1.50	0.37	0.00	0.00	0.00	**96.89**	0.00
bar	0.00	0.00	0.00	**97.95**	0.07	1.93	0.05	0.00	0.00
car race	1.78	2.20	0.03	10.59	**83.38**	1.58	0.18	0.20	0.05
stadium	0.02	0.00	0.00	17.98	0.00	**80.99**	0.32	0.68	0.02
sligth swell	0.00	0.00	0.00	0.00	0.00	0.00	**100.00**	0.00	0.00
restaurant	0.17	0.00	0.00	0.20	0.00	0.00	0.00	**99.63**	0.00
train	43.26	0.00	0.02	12.44	0.00	0.02	0.00	**44.17**	0.10

Table 3: Noise classification results and confusion matrix using the *msf* decision function (eq. (10)) for the well-matched test set.

Table 2 shows this limit considering both ETSI front-ends (FE and AFE) with CMN for the *well-matched test set*. For every case, the recognition is carried out using the Multiple-model-trained HMMs, with 6 different SNR thresholds defined between -5 dB and the clean condition. Each SNR threshold restricts the HMM models that can be used during recognition. Thus, if the SNR level is equal or higher than the threshold, the testing utterance is recognized using an HMM set trained under the same acoustic condition (current ambient type and SNR). Otherwise, the utterance is recognized with the HMM set trained with the same ambient type but at the SNR threshold.

As table 2 shows, the Multiple-model framework does not achieve the best recognition results when the testing utterances are recognized in perfect SNR matching (same ambient type and SNR). This is the case of employing -5 dB as threshold. As has been observed by other authors [13], it is better to apply an HMM set trained with speech at higher SNRs for utterances with very low SNRs. Thus, it can be observed that the best recognition results, 94.49 % and 94,68 % (for FE and AFE, respectively), are obtained using the Multiple-model framework with an SNR threshold of 5 dB. These values define the upper limit for our framework.

While in the multicondition-based system the AFE clearly outperforms the FE front-end (see table 1), table 2 shows that this difference is almost negligible in the multiple-model-based system, where the relative improvement achieved by the AFE is only 0.20 % in comparison with FE (SNR threshold of 5 dB). These results justify the use of the FE in our framework since, as can be observed, the potential benefits in terms of WAcc (for both front-ends) are the same while AFE is much more complex.

In order to reach the upper limit in the proposed framework, a perfect performance of the ambient detector is required. As expected, this is not usually the case. Table 3 shows the results obtained by the ambient recognizer using the msf decision function (eq. (11)). As can be seen, a number of acoustic ambients are misclassified. In some cases, it is clear that the ambient detector is unable to classify the noise correctly. However, we can observe that, in many cases, the misrecognized ambient type is quite similar to the actual one (e.g., babble and restaurant). This fact will allow a good behaviour of the proposed ambient-aware framework both for well-matched and mismatched conditions, as it is observed in the recognition results of table 4.

Table 4 compares the best results (with respect to the best SNR threshold) obtained for our framework using the different decision functions defined in subsection 2.1 for the ambient detector. In addition, this table compares the results obtained by applying or not CMN to the feature vectors before performing speech recognition in the proposed framework. As can be seen, all the decision rules provide comparable results, being the msf decision function which achieves the maximum accuracy. Furthermore, we can observe that the best results are obtained when CMN is applied. This fact suggests that this compensation technique helps to reduce the mismatch between training and testing due to the misclassification errors introduced by the ambient detector.

With respect to ideal system, we can see that the results achieved by our proposed framework are close to those of the ideal system (94.49 % for FE and CMN). The small performance reduction with respect to the ideal system is due to the detection errors introduced by the ambient detector. Nevertheless, the best value achieved in *well-matched test set*, 94.20 % for decision function *msf* and CMN, only supposes a 0.31 % of relative degradation with respect to the upper limit. Furthermore, the proposed framework outperforms the results obtained by the multicondition system using AFE parameters (see table 1) in both test sets: a relative 1.42 % improvement for the *well-matched test set* and 1.48 % for the *mismatched test set*.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have described a framework for robust DSR using information about the acoustic ambient noise. We have proposed an acoustic ambient detector which is used at the server side to identify the ambient type and the SNR of the current input utterance. This information is used in the speech recognizer to select a set of acoustic models trained with speech acquired under the same acoustic ambient. The results show that the proposed system improves the recognition accuracy obtained using the AFE front-end and mul-

	Well-matched test set		Mismatched test set	
	No CMN	*CMN*	*No CMN*	*CMN*
mode	91.47	94.13	82.27	90.60
ml	91.48	94.13	82.29	90.61
msf	92.45	**94.20**	84.51	**90.98**

Table 4: Recognition accuracy (WAcc (%)) for the proposed framework using different noise decision functions with and without applying cepstral mean normalization to the feature vectors.

ticondition training in both well-matched and mismatched conditions. Furthermore, our system achieves a performance only slightly worse than an ideal system, where each utterance would be recognized with an HMM set trained in the same acoustic condition.

Future work includes improving the framework to be more flexible to the ambient changes and including more context information during the recognition. Another task is to incorporate an adaptation technique to make the system more robust to new ambients not considered (mismatch cases). Finally, the effect of the transmission errors in the performance should be considered.

6. ACKNOWLEDGEMENTS

This work was supported by the Spanish MEC in the project FEDER TEC2007-66600.

7. REFERENCES

[1] M. Turunen, T. Hurtig, and J. Hakulinen. *Mobile speech-based and multimodal public transport information services*. Speech in Mobile and Pervasive Environments (SiMPE), Espoo, Finland, September 2006.

[2] T. Brøndsted, L.B. Larsen, B. Lindberg, M. Rasmussen, Z.H. Tan, and H. Xu. *Distributed speech recognition for information retrieval on mobile devices*. Speech in Mobile and Pervasive Environments (SiMPE), Espoo, Finland, September 2006.

[3] A.M. Peinado and J.C. Segura. *Speech recognition over digital channels. Robustness and standards*. Wiley, 2006.

[4] ETSI ES 201 108. *Speech Processing, Transmission and Quality aspects (STQ), Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms*. February 2000.

[5] ETSI ES 202 050. *Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms*. November 2005.

[6] J.Y. Li, B. Liu, R.H. Wang, and L.R. Dai. *A complexity reduction of ETSI advanced front-end for DSR*. In Proc. of ICASSP'04, Montreal, Canada, May 2004.

[7] Z. Tan, P. Dalsgaard, B. Lindberg, and H. Xu. *Robust speech recognition in ubiquitous networking and context-aware computing*. In Proc. of Interspeech'05, Lisboa, Portugal, September 2005.

[8] B. Schilit, D. Hilbert, and J. Trevor. *Context-aware communication*. IEEE Wireless Communications, vol. 9:5, pp. 46-54, October 2002.

[9] G. Chen and D. Kotz. *A survey of context-aware mobile computing research*. Dartmouth computer science technical report TR2000-381.

[10] M. Coen, L. Weisman, K. Thomas, and M. Groh. *A context sensitive natural language modality for the intelligent room*. In Proc. of MANSE'99, Dublin, Ireland, 1999.

[11] L. Ma, D.J. Smith, and B.P. Milner. *Context awareness using environmental noise classification*. In Proc. of Eurospeech'03, Geneva, Switzerland, 2003.

[12] M. Akbacak and J.H.L Hansen. *Environmental sniffing: Noise knowledge estimation for robust speech recognition*. IEEE Trans. on audio, speech, and language processing, vol. 15, no. 2, February 2007.

[13] H. Xu, Z. Tan, P. Dalsgaard, and B. Lindberg. *Robust speech recognition based on noise and SNR classification – a Multiple-Model Framework*. In Proc. of Interspeech'05, Lisboa, Portugal, September 2005.

[14] L.R. Rabiner. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. In Proc. of the IEEE, vol. 77, pp. 257-285, February 1989.

[15] V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York 1982.

[16] H.G. Hirsh and D. Pearce. *The Aurora experimental framework for the performance evaluations of speech recognitions systems under noise conditions*. ISCA ITRW ASR 2000, November 2000.

[17] C. Chang and C.J. Lin. *LIBSVM: a library for support vector machines*. Technical report, Dept. of Computer Science and Information Engineering, National Taiwan University (2001).