Contour motion estimation for address-event data

Francisco Barranco, Cornelia Fermüller, and Yiannis Aloimonos
University of Maryland
CFAR-UMIACS
{barranco, fer, yiannis}@umiacs.umd.edu

1. Introduction

Current motion estimation methods use very sophisticated techniques that require high computational complexity, with low time performance, and large resources. The current framework is considered exhausted. DVSs (Dynamic Visual Sensors) efficiently encode and transmit motion signals (Fig. 1). They are valuable to study the early visual motion providing motion boundaries that are salient. Furthermore, new sensors that provide both, frame-based and event-based information, allow us even more accurate estimates. Current motion estimation methods are based on the OFC (Optical Flow Constraint) that assumes the constancy of the intensity. The problem is ill-posed and the solution usually consists in adding more assumptions about the flow that increase the error. As DVS provides motion contours, we compute the contour or normal flow: the projection of the motion onto the spatial gradient. It does not require additional constraints and allows us to compute the 3D pose estimation.

Since the absolute intensity is not available the gradient reconstruction is not possible. Thus, the DVS estimation is based on the thickness of the motion contours: thicker boundaries intuitively mean higher velocities. The first step consists in locating the boundaries, estimate their width, and finally set the motion direction. Locally, we define a group of connected pixels that belong to the same boundary $\mathcal{C}(p)$ (horizontally and vertically), with p the point for which we are estimating the speed (usually the position in the boundary for which the first event was fired). The velocity is estimated as the ratio of the total number of events of connected pixels and the median number of events $\mathcal{E}(q)$ for every connected pixel q (see Eq. (1)).

$$\mathbf{u}(p) = \sum_i \frac{\Delta t_i}{T} sign(\mathbf{u}(p)) \frac{\sum\limits_{q \in \mathcal{C}(p)} \mathcal{E}(p)}{\underset{q \in \mathcal{C}(p)}{mean}(\mathcal{E}(q))} \quad (1)$$

The speed estimation is initially performed by accumulating all the events during a fixed interval of time t_i. Since we are constantly receiving new events, it is not obvious how to find the beginning and end points of the intensity variations. We are using a bio-inspired technique collecting for different amounts of time $[0 - t_1], [0 - t_2], ..., [0 - t_n]$ and combining the different estimates into the final one.

Figure 1. Dynamic Visual Sensor and contour motion estimation.

	DVS EPE rel	EPE rel	Density
Translation	0.003	7.5	9.72%
Diverging	15.4	19.4	5.4%
Yosemite	12.8	11.7	1.37%
Rubberwhale	25.2	40.1	0.53%
Dimetrodon	7.2	9.1	0.78%
Satellite	9.6	26.1	4.17%

Table 1. Relative EPE results for Middlebury sequences.

2. Solving frame-based method problems

Until now, large motions have been solved with computationally expensive multi-resolution strategies and due to the scale-to-scale error propagation, not very accurate. Additionally, occlusions have to be handled separately and iteratively. DVS sensor provide data with a super temporal resolution. In this case occlusions are not a problem anymore, and large motions are handled as temporal integration of smaller ones. Moreover, due to its dynamic range, motion blurring or light artifacts disappear.

3. Results

We estimate EPE (end-point error) for different sequences from Middlebury obtaining in general, better results than the classic+Nl-fast algorithm from Sun et al., 2014 for the Table 1. In this case we are integrating intensity and DVS data, and comparing only contour motion.