REAL-TIME MOTION ESTIMATION BASED ON EVENT-BASED VISION SENSOR

Jun Haeng Lee, Kyoobin Lee, Hyunsurk Ryu, Paul K. J. Park,
Chang-Woo Shin, Jooyeon Woo, and Jun-Seok Kim

SAIT, SEC, San 14, Nongseo-dong, Giheung-gu, Yongin-si, Gyeonggi-do, Korea

ABSTRACT

Fast and efficient motion estimation is essential for a number of applications including the gesture-based user interface (UI) for portable devices like smart phones. In this paper, we propose a highly efficient method that can estimate four degree of freedom (DOF) motional components of a moving object based on an event-based vision sensor, the dynamic vision sensor (DVS). The proposed method finds informative events occurred at edges and estimates their velocities for global motion analysis. We will also describe a novel method to correct the aperture problem in the motion estimation.

Index Terms— dynamic vision sensor, motion estimation

1. INTRODUCTION

Accurate motion estimation of an object in multi degree of freedom (**DOF**) can provide a useful basis for a number of vision-based applications including gesture-based user interface (**UI**). Estimation of motion or optical flow based on the conventional frame-based cameras has been extensively studied over the last few decades [1]. And there have been many efforts to apply these techniques to the gesture-based UI not only for TV and PC but also for mobile phones [2-4]. However, the response time and the function of the gesture-based UI were significantly limited, especially in mobile devices powered by batteries, due to the high computational cost required by these frame-based techniques. The computational cost of motion estimation can be significantly reduced by using the event-based sensors like dynamic vision sensor (**DVS**) [5-12]. Delbruck [10] developed event-based temporal correlation filters for the DVS to detect the orientations and velocities of edges. Benosman et al. [11] showed that the event-based optical flow calculation required much less computation time than the frame-based one. Lee et al. [12] proposed a method to calculate the pseudo optical flow and showed the possibility of rich gesture UIs based on four DOF motion. Although these previous works clearly showed benefits of using the event-based sensors in the motion estimation, there is still much room for improvement in terms of accuracy and computational cost. For example, the stochastic nature of event occurrence in the DVS was not properly considered in the previous works. In particular, in those works, the aperture problem has not been addressed although it could significantly degrade the accuracy of motion estimation. In this paper, we propose a series of algorithms to estimate edge velocities and global motion of a moving object from the output events of the DVS. The proposed technique can find edge orientations as well as the edge velocities with much less computation cost than the previously proposed methods. It can also decompose the global motion into four DOF components and correct the aperture problem.

2. EDGE PATTERN CLASSIFICATION

The DVS is a kind of edge detector since it tends to produce output events mostly on moving edges. Previously, it has been proposed to estimate edge orientation and velocity from the output of the DVS using temporal correlation filters [10]. However, this method could be inefficient since they usually evaluate correlation for every possible orientation and select the best one among them for velocity estimation. In this paper, we propose a simple but highly efficient method which does not require this kind of exhaustive search to find the best-matching orientation.

Each event from the DVS contains the x and y position, the timestamp of its occurrence, and the polarity information denoting the cause of event occurrence (either intensity increase or decrease). A couple of two dimensional (**2D**) timestamp maps are used to keep the timestamp of the latest event from each pixel of the DVS [10, 12]. When an event arrives, one of the maps is selected based on the polarity of the event and updated with its timestamp. Subsequently, the edge pattern of the 3x3 patch on the event location is classified by observing the timing patterns of surrounding eight neighbor pixels. Based on the timestamp difference, each neighbor is tagged as either E or S-type like:

$$t_{ev} - t_n \begin{cases} \geq T_E \rightarrow \text{E-type} \\ \leq T_S \rightarrow \text{S-type} \end{cases}, \quad (1)$$

where t_{ev} is the timestamp of the current event, t_n is the latest timestamp of the n-th neighbor pixel (n = 1 ~ 8), and T_E and T_S are the threshold values to determine E or S-type, respectively. These types are used to classify edge patterns into one of twenty four classes defined in Fig. 1(a) or a garbage to be abandoned. Those patterns are named as directional edge patterns (**DEP**s) since they have the information of movement direction as well as its orientation (i.e., angular position). When T_E = T_S, the classification of

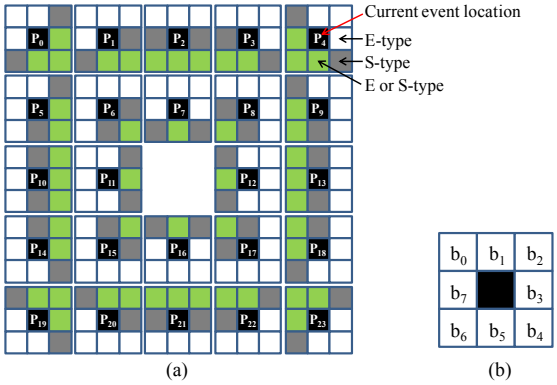
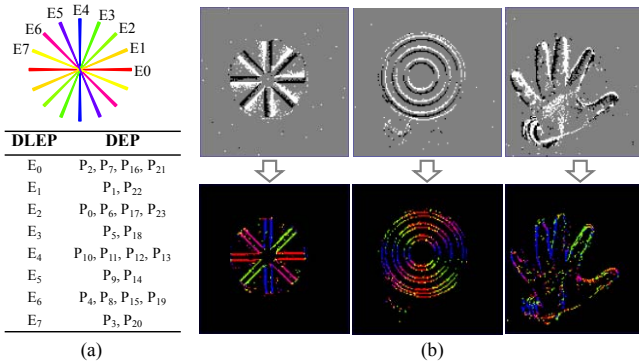


Fig. 1. (a) Definition of directional edge patterns (DEPs). The black colored center is the location of the current event. White boxes: E-type pixels. Gray boxes: S-type pixels. Green boxes: either E or S-type pixels. (b) Position of b_n.



DLEP	DEP
E_0	P_2, P_7, P_{16}, P_{21}
E_1	P_1, P_{22}
E_2	P_0, P_6, P_{17}, P_{23}
E_3	P_5, P_{18}
E_4	$P_{10}, P_{11}, P_{12}, P_{13}$
E_5	P_9, P_{14}
E_6	P_4, P_8, P_{15}, P_{19}
E_7	P_3, P_{20}

(a) (b)

Fig. 2. (a) Definition of directionless edge patterns (DLEPs). (b) Edge extraction results.

these patterns can be significantly simplified. By checking E-type neighbors in the order shown in Fig. 1(b), we can evaluate a value B defined as $B = b_0b_1b_2b_3b_4b_5b_6b_7$, where b_n (n = 0, 1, …, 7) is a bit value that is 1 (or 0) if the *n*-th neighbor is an E-type (or an S-type). Bit-wise **AND** operation of B with its circular shifts gives a value defined as the P-value like:

P-value = (B<<1) **AND** B **AND** (B>>1), (2)

where (B<<1) and (B>>1) represent 1-bit left and right circular shifts, respectively. Each pattern in Fig. 1(a) has a unique P-value. Thus, pattern classification can be done by using a lookup table of P-values. Edge orientation can be inferred from these DEPs as shown in Fig. 2(a). Each DEP can be mapped into one of eight orientations defined as directionless edge pattern (**DLEP**). DLEPs can be utilized as basic visual features for estimating the shape of an object or constructing a depth map in stereo vision configuration. A few examples of edge classification results are shown in Fig. 2(b) when $T_E = T_S = 30$ ms. The black (or white) dots in the upper side figures represent the events with off-polarity (or on-polarity). In the lower side figures, the orientation of edges is color coded as defined in Fig. 2(a). Orientations of edges are correctly detected. In an evaluation using a vertical line, 75% of events were classified into E_4 while

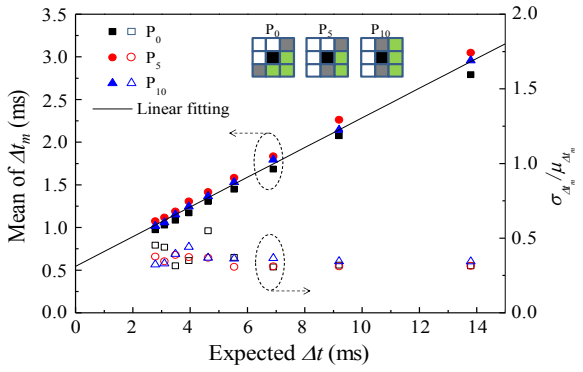


Fig. 3. Mean and standard deviation of Δt_m.

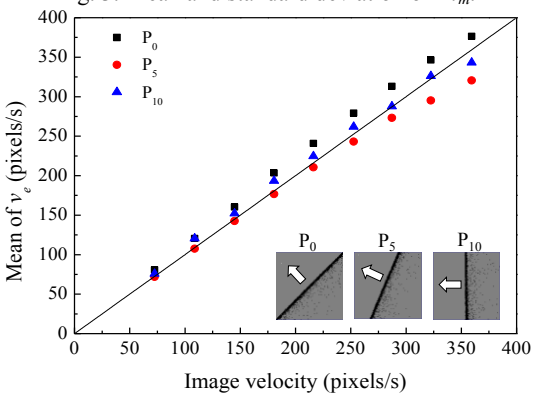


Fig. 4. Estimated velocity of edges from (3).

20% were detected as E3 or E5 due to randomness of event generation.

3. EDGE VELOCITY ESTIMATION

The temporal information recorded in the timestamp maps provides a useful clue for motion analysis [10, 12]. Although the occurrence of events is a stochastic process [5], ensemble average taken over all events tends to give a good estimation of global motion. However, if the velocity is simply calculated from the temporal differences, it is not guaranteed to have a linear relation between the global motion and the true image velocity. To address this problem, in this paper, we propose an estimator for edge velocities. For this purpose, Δt_m is defined as $\Delta t_m \equiv \frac{1}{N}\sum_{i\in A}^{N} dt_i$, where dt_i is the difference between the timestamp of the current event and the latest timestamp of *i*-th neighbor pixel in a set *A*. The set *A* consists of pixels on the side of S-type neighbors. Fig. 3 shows the mean value of Δt_m for three representatives of DEPs (P_0, P_5, and P_{10}). Even though Δt_m is a random variable due to the random nature of event occurrence in the DVS, the mean of Δt_m has linear relationship with the expected timestamp difference (defined as Δt here after) obtained from the image velocities. The ratio of standard deviation to mean of Δt_m also can be regarded as a constant as shown in the right-bottom axes. Thus, Δt_m can be approximated as a linear function with a random noise source like $\Delta t_m = p\Delta t + c + n$, where p and c are constants

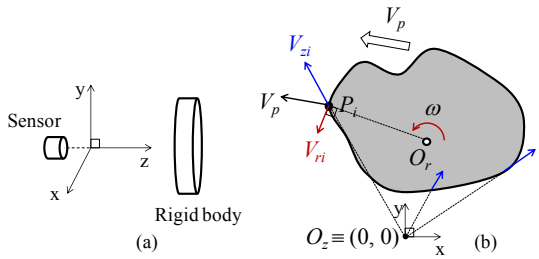


Fig. 5. Four DOF motional model of a rigid body (a) in 3D space, and (b) in the x-y plane.

and n is a random variable. By assuming n as Gaussian random variable and expanding $1/\Delta t_m$ ($\equiv v_m$) into Taylor series, its expectation can be approximated as $E[v_m] \approx \beta v/(1+v/v_0)$, where $v = 1/\Delta t$, β and v_0 are constant. We experimentally found that $E[v_m]$ was not linearly proportional to the true image velocities. Thus, we propose a simple velocity estimator to obtain global motion linearly related to the image velocity as:

$$v_e = \frac{v_m}{\beta - v_m / v_0} \quad (\text{for events with } v_m > 0.8\beta v_0) \cdot \tag{3}$$

Fig. 4 shows that the average values of edge velocities estimated by (3) have good linear correlation with the real image velocities regardless of edge patterns. The result was measured by swiping line edges (P_0, P_5, or P_{10}) with various velocities as shown in the insets of Fig. 4. The direction of v_e is assumed to be that of the normal vector of the edge. This assumption gives rise to the well-known aperture problem. We will consider this issue in detail in Section 5 and propose a novel method to address it.

4. GLOBAL MOTION ANALYSIS

Global motion of an object in four DOF (i.e., 2D translation, zoom, and rotation) can be estimated by integrating local motions obtained by (3). For this purpose, we assume a rigid body capable of 3D translational and 1D rotational motion about an axis parallel to the z-axis as shown in Fig. 5. Rotation about the x or y axes is not considered in this model. If the position of the DVS is fixed, 2D translation of the output events is the result of the planar movement of the object on a plane parallel to the x-y plane. Scaling component (i.e. zoom-in or -out) is caused by the translational movement of the object along the z-axis. Neglecting any distortion or offsets caused by imperfection of optical systems, the center point of zoom can be assumed to be (0, 0) on the x-y plane. With those assumptions, the velocity (V_i) of an edge at a pixel P_i can be modeled like:

$$V_{zi} + V_{ri} + V_p = V_i, \tag{4}$$

where V_{zi}, V_{ri}, and V_p is the motional components caused by zoom, rotation, and 2D translation, respectively. If we consider an object with negligible height along the z direction (i.e. all P_i is virtually on the same plane and any z-directional dependency is neglected), we can simplify (4) like:

$$tP_i + \omega A(P_i - O_r) + V_p = V_i, \tag{5}$$

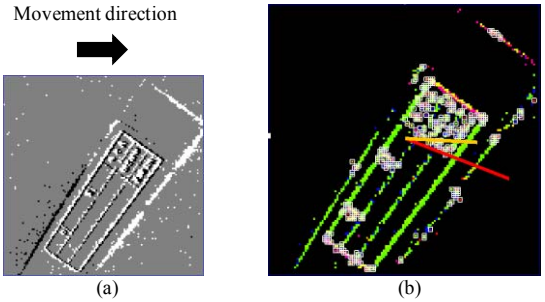


Fig. 6. True direction of movement can be estimated by finding events with low *LOH*. (a) DVS output of a tilted name tag. (b) Result of global motion estimation. White rectangles: events in patches with low *LOH*. Red line: direction estimated from all events. Yellow line: direction estimated from low *LOH* events only.

where $P_i = (x_i, y_i)$, $V_i = (V_{xi}, V_{yi})$, O_r is the axis of rotation, $A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$, and t and ω are zooming and rotating rates, respectively. Thus, we can obtain t and ω from (5) using Moore–Penrose pseudo inverse. O_r and V_p can be estimated by using another constraint like: $(V_i - tP_i - V_p) \bullet (P_i - O_r) = 0$. However, when there is no rotational component (i.e., $V_i \approx tP_i + V_p$ for all i), this constraint does not give correct answers. Thus, in this case, O_r is assumed to be the mean position ($\overline{P}$), and subsequently V_p is estimated by using the mean value of (5) and O_r.

5. APERTURE PROBLEM

The aperture problem (**AP**) refers to ambiguity of the edge velocity observed through a small aperture [14]. Since we assumed the direction of the edge velocity to be that of the normal vector of the edge in Section 3, the global motion estimation described in the previous section could suffer from the AP. The AP could affect on both the direction and the magnitude of the estimated global motion. For example, the direction of the global velocity can be biased to the direction of the normal vector of majority edge types. The magnitude of the global motion also can be significantly underestimated depending on the shape of an object. The more parallel to the movement direction the orientations of edges are, the smaller the global motion is estimated. One of the solutions for the AP is to find the intersection of all lines perpendicular with local velocities [13]. However, this method can solve the AP only for the 2D translation. In this section, we propose a method robust to the stochastic process of the DVS and capable of negating the harmful effect of the AP by using a generative motion model.

The direction of the global motion can be estimated more accurately if there are enough features like corners, branches, or cross points. We find these features by searching areas consisting of heterogeneous orientation types. For this purpose, we accumulate and average event orientation over a certain period of time. Subsequently, the level of homogeneity (***LOH***) is evaluated for the patch around each pixel which produced events during the period as follows:

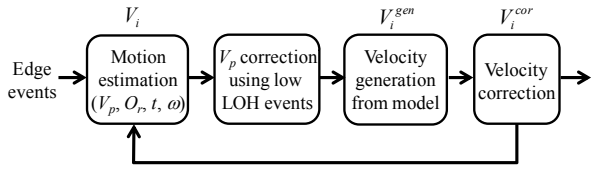


Fig. 7. Principle of the proposed AP correction method.

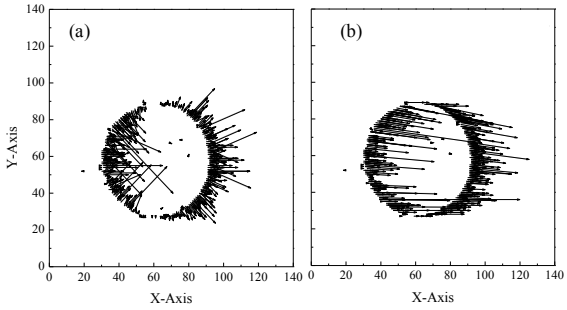


Fig. 8. Correction of aperture problem. (a) Edge velocities w/o correction. (b) Edge velocities after the second correction.

$$Level\ Of\ Homogeneity(LOH) \equiv \sum_{i \in Patch}^{N} \cos^2(\theta_{av} - \theta_i), \qquad (6)$$

where θ_{av} is the average orientation of the edges in the patch and θ_i is the edge orientation at i-th pixel of the patch. More accurate direction of the global motion could be estimated from the events occurred at low *LOH* patches only as shown in Fig. 6. It is clearly shown that the movement direction of a tilted rectangle moving rightward can be detected with events from low *LOH* patches (yellow line) while the direction estimated from entire events (red line) is right downward due to the AP. As expected, we could obtain low *LOH* patches (there locations are marked with white rectangles in Fig. 6) around corners, branches, cross points, and complex areas like texts. Patch size was 7×7.

To correct the shape dependency of global motion caused by the AP, we propose to use a generative motion model as follows. Since we can extract the global motion parameters (i.e. V_p, O_r, t, and ω), we can generate the theoretical velocities at every position. By comparing the estimated velocities (i.e. v_e in Section 3) with the generated counter parts, we can negate the effect of the AP as illustrated in Fig. 7. Firstly, motion parameters of the object are estimated by using the global motion estimation method described in Section 4. Subsequently, the direction of V_p is corrected by using the events from low *LOH* patches described above. The obtained motion parameters are used to generate theoretical velocities at the position of every event as:

$$V_i^{gen} = tP_i + \omega A(P_i - O_r) + V_p. \qquad (7)$$

Assuming that the original edge velocities (V_i) are distorted by the AP and the generated velocities (V_i^{gen}) are their projections to the currently estimated global motion, the effect of the AP can be negated as follows:

$$V_i^{cor} = \begin{pmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{pmatrix} V_i, \qquad (8)$$

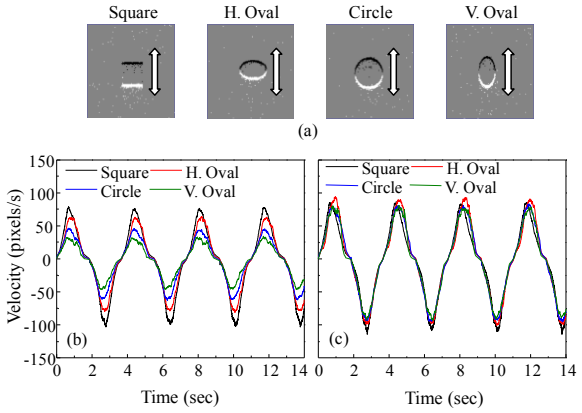


Fig. 9: Correction of AP. (a) DVS outputs of four objects used for AP correction experiment. Each object was oscillated vertically. Velocities of the objects (b) w/o correction and (c) w/ correction.

where θ is the angle between V_i and V_i^{gen}. With these corrected velocities (V_i^{cor}), the motion parameters are updated. For stable operation, the maximum correction angle must be smaller than $\pi/2$ due to $\tan\theta$. Thus, to correct angles up to $\pi/2$, one more iterative correction is necessary. It is also required to limit the total corrected angle to around $\pi/2$ to avoid overcorrection. We set the maximum correction angle for the first correction close to $\pi/2$ to ensure enough correction of velocity magnitude. The second correction was used to correct the residual angle which was not corrected in the first correction.

Fig. 8 shows the experimental result of the AP correction when a circle is horizontally moving from left to right. As shown in Fig. 8(a), without correction, edge velocities are perpendicular with their orientations. In addition, their magnitudes are seriously underestimated due to the AP when their directions are deviated from the true movement direction. Fig. 8(b) shows that these effects were significantly reduced by using the proposed method. Fig. 9 shows shape dependency of global velocities with and without correcting the AP. We measured velocities of four different objects (i.e. square, horizontal oval, circle, and vertical oval) while moving each of them periodically up and down. Without correction, the global velocity is a strong function of the object shape as shown in Fig. 9(a). On the other hand, shape dependency was significantly reduced when the AP was corrected as shown in Fig. 9(b).

6. SUMMARY

We have proposed a method for estimating edge orientations and motion of a moving object. Edge patterns were efficiently classified by using the proposed algorithm without exhaustive search, and edge velocities were obtained by using a linear estimator considering the random nature of event occurrence in the DVS. The global motion of a rigid body in four DOF was estimated from the edge velocities. The harmful effect of the aperture problem could be significantly reduced by using a generative motion model.

7. REFERENCES

[1] Barron T. et al. Performance of optical flow techniques. *Int'l Jour. of Computer Vision*, **12**(1), 43-77 (1994).

[2] Cutler R. & Turk M. View-based Interpretation of Real-time Optical Flow for Gesture Recognition. *Proc. IEEE Conf. Face and Gesture Recognition*, 416-421 (1998).

[3] Wang J. Camera phone based motion sensing: interaction techniques, applications and performance study. *Proc Annual ACM Symp. on User Interface Software and Technology.* 101–110 (2006).

[4] Kratz S. & Ballagas R. Gesture recognition using motion estimation on mobile phones. *Proc. Int'l Workshop on Pervasive Mobile Interaction Devices* (*PERMID*'07) (2007)

[5] Lichtsteiner P. et al. An 128x128 120dB 15us-latency temporal contrast vision sensor. *IEEE J. Solid State Circuits* **43**(2), 566-576 (2008)

[6] Serrano-Gotarredona T. & Linares-Barranco B. A 128x128 1.5% Contrast Sensitivity 0.9% FPN 3us Latency 4mW Asynchronous Frame-Free Dynamic Vision Sensor Using Transimpedance Amplifiers. *IEEE J. Solid-State Circuits*, 48(3), 827-838 (2013).

[7] Lee J. H. et al. Live demonstration: Gesture-based remote control using stereo pair of dynamic vision sensors. *Proc. Int'l Conf. Circuits and Systems* (*ISCAS*'12), 741-745 (2012).

[8] Kohn B. Event-driven body motion analysis for real-time gesture recognition. *Proc. Int'l Conf. Circuits and Systems* (*ISCAS*'12), 703-706 (2012).

[9] Lee J. H. et al. Touchless hand gesture UI with instantaneous responses. *Proc. Int'l Conf. Image Processing* (*ICIP*'12), pp. 1957-1960 (2012).

[10] Delbruck T. Frame-free dynamic digital vision. *Proc. Intl. Symp. on Secure-Life Electronics*, 21-26 (2008).

[11] Benosman R. et al. Asynchronous frameless event-based optical flow. *Neural Networks* **27**, 32-37 (2012).

[12] Lee K. et al. Four DoF gesture recognition with an event-based image sensor. *Proc. Global Conf. on Consumer Electronics* (*GCCE*'12), 293-294 (2012).

[13] Adelson E.H. & Movshon J.A. Phenomenal coherence of moving visual patterns. *Nature* **300**, 523-525 (1982).