An Event-based Optical Flow Algorithm for Dynamic Vision Sensors

Iffatur Ridwan and Howard Cheng*

Department of Mathematics and Computer Science
University of Lethbridge, Canada
iffatur.ridwan@uleth.ca,howard.cheng@uleth.ca

Abstract. We present an event-based optical flow algorithm for the Davis Dynamic Vision Sensor (DVS). The algorithm is based on the Reichardt motion detector inspired by the fly visual system, and has a very low computational requirement for each event received from the DVS.

1 Introduction

Motion detection is a common task in many areas of video processing and computer vision, and optical flow computation is one method of performing such detections. In applications such as autonomous vehicle or robot navigation [5], these computations must be done in real-time, using devices that may have limitations on power consumption as well as computational power. We propose a fast algorithm to compute optical flow that is useful on such restricted platforms, using a camera that has been inspired by biological retinas.

There are existing works on the computation of optical flow for videos obtained from conventional frame-based cameras (for example, [6, 8, 9]). Since consecutive frames are highly correlated, these cameras often capture frames with redundant data which are later removed in the processing. Computational time and electrical power are wasted in capturing and processing this data.

The Davis Dynamic Vision Sensor (DVS) is a camera that is modelled upon the human retina [7]. The DVS is an asynchronous device that only transmits events indicating changes in brightness in individual pixels. If there is no change, this system does not give any output. The DVS has lower power and computational requirements, as well as faster reaction times. Algorithms for the DVS must be designed so that it works with "sparse" input in order to take advantage of the unique properties of the DVS.

In this paper, we propose an algorithm for the DVS based on the Reichardt detector—a simple correlation-based movement detection model inspired by the visual system of flies [4, 10]. This model cannot be used directly for the DVS but we will show that our algorithm can be considered as a variation of the

* Supported by a Discovery Grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Reichardt detector on conventional frame-based video input. We will show that our algorithm requires relatively little processing for each event received from the DVS and therefore maintains the advantage of using the DVS. While there are some other works on optical flow algorithms for this type of cameras [2, 3], our approach is different in that it is based on the Reichardt detector.

The paper is organized as follows. Section 2 reviews some of the previous works our algorithm is based on. Section 3 describes our algorithm, and experimental results and analysis are given in Section 4.

2 Preliminaries

2.1 Reichardt Detector

The Reichardt detector is a model of how neurons detect motion from photoreceptors, and it is inspired by the visual system of flies [4, 10]. The Reichardt detector consists of two mirror symmetric sub-units (Figure 1). In each subunit, the luminance values as measured in two adjacent image locations (one of them is delayed by a low pass filter) are multiplied together. The product can be viewed as the correlation of the two locations at different times. The resulting output signal is the difference between the multipliers of the two sub-units.

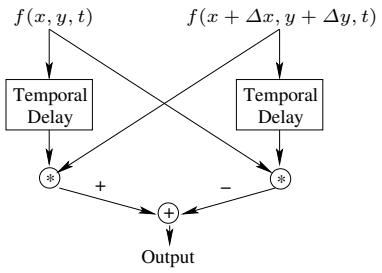


Fig. 1. The Reichardt Detector.

More formally, let $f(x, y, t)$ is the luminance value at location (x, y) at time t. We also let Δx and Δy be the offset between two adjacent sub-units, and Δt be the time delay. Then the output of the Reichardt Detector is

$$RD(f, x, y, t, t') = f(x', y', t) \cdot f(x, y, t') - f(x, y, t) \cdot f(x', y', t'), \qquad (1)$$

where $x' = x + \Delta x$, $y' = y + \Delta y$, and $t' = t + \Delta t$. If $|RD(f, x, y, t, t')|$ exceeds a threshold T_{RD}, motion is detected along the direction $\pm(\Delta x, \Delta y)$, and the sign of $RD(f, x, y, t, t')$ indicates the actual direction. Adjacent pixels in stationary objects will not be detected because the difference is close to zero.

Each Reichardt detector can only detect motion at a particular location with velocities of $\pm(\Delta x/\Delta t, \Delta y/\Delta t)$. Different velocities can be detected by varying

Δx, Δy, and Δt. In practice, this is accomplished by a grid of Reichardt detectors on the pixels of successive frames, so that motion can be detected at all locations in a number of pre-defined directions (e.g. the 8 compass directions).

2.2 The Davis Dynamic Vision Sensor

The Davis Dynamic Vision Sensor (DVS) is a type of "neuromorphic camera" that is inspired by biological retinas [7]. Each pixel is an independent sensor from the others. When the log-luminance level at a particular pixel changes by more than a predefined threshold T_{DVS}, an event is reported indicating the location and its polarity (positive or negative). The events are reported asynchronously as soon as they occur. As a result, it is possible to react to local changes quickly without waiting for a "frame" to be collected. Since only significant changes are reported, redundant data are not reported and do not need to be processed. It may also result in lower power requirement. Of course, the algorithm to process this data must not "convert" these events into a frame-based video or the advantages of the sparseness of the data will be lost.

The events generated by the DVS are communicated using the Address Event Representation (AER) protocol. Conceptually, each of the events we are interested in contains the following information: timestamp, location (x,y), and polarity ($\pm$). The DVS may also generate other types of events but they are ignored by our optical flow algorithm.

2.3 Optical Flow

Optical flow refers to the pattern of motion that are present in a scene. It is generally represented as a vector field at each time step, in which each pixel is associated with a vector indicating the apparent motion for that pixel at that time. These vectors field can then be processed further to detect specific types of motion of interest (e.g. incoming objects). Many optical flow algorithms for conventional frame-based cameras have been proposed and studied (see, for example, [1]).

3 Proposed Algorithm

In this section, our algorithm to compute optical flow for the Dynamic Vision Sensor (DVS) is described. The connection between the algorithm and Reichardt detectors will also be shown.

The input to our algorithm is an event stream in the AER format. Moreover, the output is also an event stream indicating when motion is detected. When motion is detected at a particular pixel, the algorithm generates an event specifying a timestamp, the location (x, y), as well as the direction of the motion. In our algorithm, we only detect one of the eight compass directions, which we denote by the vectors $\boldsymbol{v}_1, = (-1,-1), \boldsymbol{v}_2 = (-1,0), \boldsymbol{v}_3 = (-1,1), \boldsymbol{v}_4 = (0,-1), \boldsymbol{v}_5 = (0,1), \boldsymbol{v}_6 = (1,-1), \boldsymbol{v}_7 = (1,0), \boldsymbol{v}_8 = (1,1)$. Thus, the output of our algorithm

is an event stream indicating the nonzero vectors in the optical flow at specific times. The direction vectors are fixed and only these 8 directions can be reported, but it is possible that multiple directions are reported at the same location and time. If desired, the multiple directions at a location can be combined (e.g. by taking the "average" of the detected directions).

For each pixel location (x, y), we maintain the most recently received event $e_{(x,y)}$. Each event is represented by the timestamp and its polarity (t, p). As each event arrives, we search for a recent event (occurring no more than some threshold T before the current event) that has the same polarity. If a match is found, an event indicating detected motion from the neighbour to the current pixel is reported. This is described in Algorithm 1. The parameter T is used to control how "recent" a neighbouring event is considered a match to the current event received. Note that only the direction of the motion is reported. If desired, the magnitude of the motion can be reported by comparing the timestamps of the two matching events to determine the velocity of the movement.

Algorithm 1: Optical Flow Computation with DVS for a single event received.

Input: an event from the DVS consisting of timestamp t, location (x, y), and polarity $p \in \{+, -\}$; a threshold T.
Output: if motion is detected, event(s) each consisting of timestamp t, location (x, y), and direction $\boldsymbol{v}$.

$e_{(x,y)} \leftarrow (t, p)$;
for $\boldsymbol{v} \in \{\boldsymbol{v}_1, \ldots \boldsymbol{v}_8\}$ **do**
  Let $(x', y') = (x, y) - \boldsymbol{v}$;
  Let $(t', p') = e_{(x',y')}$;
  if $0 < t - t' \leq T$ *and* $p = p'$ **then**
    Output event $(t, x, y, \boldsymbol{v})$;
end

In terms of computational complexity, each input event requires only a small constant number of operations proportional to the number of directions. The number of pixels in the image is irrelevant. This is important because the output of the DVS (the input of our algorithm) is sparse and the complexity of our algorithm is directly proportional to the number of events in the input. Thus, the advantage of the DVS is preserved by our algorithm. Other existing approaches [2, 3] require more complicated calculations for each event and have a higher computational costs, but the output of these algorithms are more general and are not restricted to the 8 directions as in our algorithm.

3.1 Relationship to Reichardt Detectors

Although the main approach in Algorithm 1 can be considered as "event matching," the algorithm is in fact closely related to the Reichardt detector. A single

Reichardt detector along the direction $\boldsymbol{v} = (\Delta x, \Delta y)$ for a conventional frame-based camera is described by (1). Recall that the output of (1) is compared to the threshold T_{RD} to determine if motion is detected along the direction $\boldsymbol{v}$.

We first show that Algorithm 1 can be considered an application of the Reichardt detector on the output of the DVS (instead of the original scene). An event is generated by the DVS at location (x, y) when the change in log-luminance is greater than T_{DVS}. If we denote this change $\Delta f(x, y, t)$, then

$$|\Delta f(x, y, t)| = |\log f(x, y, t') - \log f(x, y, t)| > T_{DVS}, \tag{2}$$

where $t' = t + \Delta t$. When two events are matched in Algorithm 1, each of these events corresponds to a log-luminance change exceeding T_{DVS} at two times t_1 and t_2 with $t_1 < t_2$. In the algorithm, the current event at (x, y) at time t_2 is matched with a previous neighbouring event of the same polarity at time t_1. To simplify notation, we let $t_2 = t_1 + \Delta t_1$, and let $t_3 = t_2 + \Delta t_2$. Applying (1) to the output of the DVS, we have

$$RD(\Delta f, x, y, t_1, t_3) = \Delta f(x', y', t_1) \cdot \Delta f(x, y, t_2) - \Delta f(x, y, t_1) \cdot \Delta f(x', y', t_2). \tag{3}$$

A match in polarity of two events at (x, y, t_2) and (x', y', t_1) means that the product $\Delta f(x, y, t_2) \cdot \Delta f(x', y', t_1)$ is positive and greater than $(T_{DVS})^2$, which can be considered as the threshold T_{RD} that is used in (3) for detecting motion. The second term of (3) may be assumed to be 0 as there are no events at (x', y') at time t_2. Thus, our algorithm can be viewed as applying the Reichardt detector to DVS events.

Furthermore, the application of Reichardt detector to the DVS output can be thought of as a combination of the outputs of different Reichardt detectors on the original scene at closely related times. Simple algebraic manipulation shows that

$$\begin{aligned} RD(\Delta f, x, y, t_1, t_3) = {} & RD(\log f, x, y, t_1, t_2) + RD(\log f, x, y, t_2, t_3) \\ & - RD(\log f, x, y, t_1, t_3). \end{aligned} \tag{4}$$

Thus, we have shown that Algorithm 1 can be considered to be a variation of the Reichardt detector on the original scene.

4 Experimental Results

Our algorithm has been designed and implemented for the DVS. However, to demonstrate the effectiveness of Algorithm 1 in this paper, we use a DVS simulation algorithm to process input videos from conventional frame-based cameras to produce an event stream, which is then processed by Algorithm 1. We did not compare our results to these previous methods [2, 3] as the output of our algorithm is not directly comparable.

Two test videos are used in our experiments on Algorithm 1. The first video consists of a single circular object that moves around in a dark background. The

second video contains a person moving his head and body. The camera is not steady so both the foreground and the background of the scene are moving. The properties of the two videos are shown in Table 1. In the first video, events are generated by the DVS only on the boundary of the circular object, resulting in a significant reduction in the amount of data sent compared to conventional frame-based cameras. This is true even in the second video—the number of events generated is less than a tenth of the total number of pixels among all frames. Two example frames from the second video are shown in Figure 2, and the DVS events generated corresponding to these frames are shown in Figure 3.

Table 1. Properties of the test videos.

	Size	Number of Frames	Number of DVS events
Video 1	1280×720	60	32,940
Video 2	1920×1080	45	8,953,097

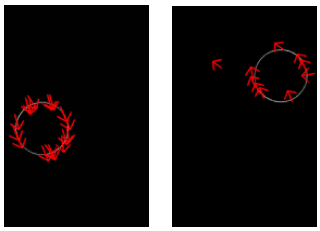
(a) Frame 26 (b) Frame 44

Fig. 2. Two example frames from the second test video.

Figures 4 and 5 show some of the optical flow computed by Algorithm 1 on Video 1 and Video 2, respectively. To visualize the results, motion events generated are collected and those occurring at the same time are displayed as individual images. To make it easier to visualize, not all vectors reported by our algorithm are shown—only one vector from a group of closely located vectors are shown. We can visually observe that the optical flow computed reflect the actual motion present in the videos, though there are very rarely extraneous motion detected due to noise (e.g. the motion vector detected in Frame 30 in Figure 4).

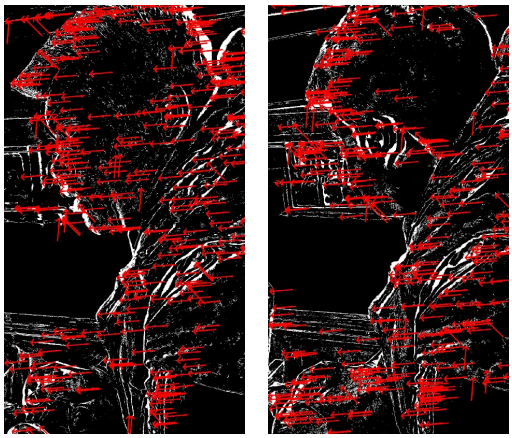
(a) Frame 26 (b) Frame 44

Fig. 3. DVS events corresponding to the example frames in Figure 2. Pixels with DVS events are shown in white.



(a) Frame 3 (b) Frame 30

Fig. 4. A visualization of the output of Algorithm 1 on Video 1.



(a) Frame 11 (b) Frame 43

Fig. 5. A visualization of the output of Algorithm 1 on Video 2.

5 Conclusion

In this paper, we described a new event-based approach to perform optical flow calculations for the Davis Dynamic Vision Sensor based on the Reichardt detector. The proposed algorithm is efficient and requires only a small number of operations for each event received from the DVS. Thus, the advantages of the DVS is maintained.

We are working to incorporate the output of our optical flow algorithm in other applications such as object tracking and looming detection. We believe that the restricted optical flow output will be beneficial for these tasks.

6 Acknowledgement

The authors would like to acknowledge Dr. Matthew Tata for providing access to the Davis Dynamic Vision Sensor for our work, and Cody Barnson for implementing some of our algorithms.

References

1. Beauchemin, S.S., Barron, J.L.: The computation of optical flow. ACM Comput. Surv. 27(3), 433–466 (Sep 1995)
2. Benosman, R., Ieng, S.H., Clercq, C., Bartolozzi, C., Srinivasan, M.: Asynchronous frameless event-based optical flow. Neural Networks 27, 32 – 37 (2012)
3. Brosch, T., Tschechne, S., Neumann, H.: On event-based optical flow detection. Frontiers in Neuroscience 9, 137 (2015)
4. Egelhaaf, M., Reichardt, W.: Dynamic response properties of movement detectors: theoretical analysis and electrophysiological investigation in the visual system of the fly. Biological Cybernetics 56(2-3), 69–87 (1987)
5. Fortun, D., Bouthemy, P., Kervrann, C.: Optical flow modeling and computation. Comput. Vis. Image Underst. 134(C), 1–21 (May 2015), http://dx.doi.org/10.1016/j.cviu.2015.02.008
6. Fülöp, T., Zarandy, A.: Bio-inspired looming object detector algorithm on the eye-ris focal plane-processor system. 2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010) (2010)
7. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128 × 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. IEEE Journal of Solid-State Circuits 43(2), 566–576 (Feb 2008)
8. Pantilie, D., Nedevschi, S.: Real-time obstacle detection in complex scenarios using dense stereo vision and optical flow. 13th International IEEE Conference on Intelligent Transportation Systems (2010)
9. Park, S.S., Sowmya, A.: Autonomous robot navigation by active visual motion analysis and understanding. Proceedings of IAPR Workshop on Machine Vision Applications (1998)
10. Reichardt, W., Egelhaaf, M.: Properties of individual movement detectors as derived from behavioural experiments on the visual system of the fly. Biological Cybernetics 58(5), 287–294 (1988)