Improvement of optic flow estimation from an event-based sensor with recurrently connected processing layers

Vilim Štih

31st March 2014

1 Introduction

The Dynamic Vision Sensor (DVS) is a visual sensor inspired by information processing in the retina. Most output cells of the retina (ganglion cells) do not fire persistently when their receptive field is constantly bright or dark, but in response to changes in brightness. This configuration enables efficient information transmission and high sensitivity within a broad dynamic range. Every pixel reacts only to local changes in brightness and is not influenced by the global level of illumination, making discrimination possible in visual scenes with large differences of illumination. The DVS responds in a similar way, by outputting events on changes of brightness on the pixel level as a a stream of event data blocks. Each block contains the sign of the derivative of a pixel's brightness (either on- or off-events – figure 1), along with the location of the change and a timestamp with microsecond precision.

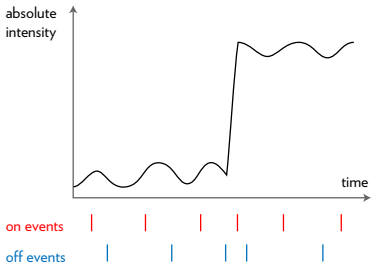

Figure 1: Events emitted for brightness changes on a pixel

Visual information from the retina proceeds via the optic nerve to the lateral geniculate nucleus (LGN) of the thalamus, from where it is relayed to the primary visual cortex (V1) and some higher visual areas (V2 and V3). In the area V1 the cells are sensitive to more complex spatial and temporal patterns than the retinal ganglion cells. For example, some cells are tuned to specific orientations of edges, left-right

eye dominance and velocity. Cells in higher visual areas respond to more complex features and have larger receptive fields. In the cortical area MT (also known as V5), several types of cells were discovered which are tuned to different velocities and directions, on a scale which incorporates cues from many V1 cells, possibly enabling resolution of ambiguities in local motion cues [1]. Another important aspect of visual processing is the big influence of feedback. The majority of connection to the lower visual areas, such as V1 are however not from the LGN, but from areas higher in the processing hierarchy. These connections strongly influence response properties of the V1 neurons [3].

Inspired by these observations Bayerl and Neumann [2] proposed an algorithm with two recurrently connected layers to improve the quality of optic flow estimates. The first layer (named V1) represents motion data as a set of velocity hypotheses for each image pixel, while the second layer (MT) contains velocity estimates in the same form as V1, but on a rougher scale. Initial estimates are set in V1 and they are modulated by the hypotheses from the second layer (MT), which is subsequently updated by subsampling from V1. The initial estimates provided as inputs to this algorithm are computed by comparing two frames, which is inapplicable for an event-based data stream.

The main aim of this project was to apply the recurrent approach to DVS data streams.

2 Time-difference based algorithm for velocity estimation

An initial motion estimate from event data can be derived by considering a simple 1D situation when a spot is moving across an array of detectors. If the time is measured between the activation of two adjacent detectors and the angle in the field of view is known, then the velocity of the spot is simply this angle divided by the time difference of the events. This approach can be extended in two dimensions, with both spatially adjacent events can be taken into account and averaged:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \frac{\Delta x}{t(x,y)-t(x-1,y)} - \frac{\Delta x}{t(x,y)-t(x+1,y)} \\ \frac{\Delta x}{t(x,y)-t(x,y-1)} - \frac{\Delta x}{t(x,y)-t(x,y-1)} \end{bmatrix} \tag{1}$$

where $t(x,y)$ is the time of the last event at the corresponding location and Δx the distance between the detectors.

For this implementation velocities were represented by the number of pixels an event would move within 1 ms, so Δx is set to 10^3 if the time is measured in µs. On and off events are considered separately, but both were used for optic flow estimation. Each new event triggers an update of the velocity estimate at the corresponding pixel, and for display and evaluation the velocity field is sampled at fixed time intervals.

This procedure is referred to further on as the 4-neighbourhood velocity estimate.

3 Recurrently connected layers for motion detection

As in the algorithm described in section 2, each event triggers the processing steps, only here the neighbourhood of the in the processing layers has to be updated as well.

In order to allow for ambiguity in motion data and its subsequent refinement by feedback, velocities for every pixel are represented as a set of wighted hypotheses. For each event, 8 hypotheses are generated initaly (figure 2). 4 velocities are calculated from the time differences of the current pixel with every orthogonal combination of the 4 adjacent pixels, and additional 4 from every orthogonal combination of diagonally adjacent pixels. For example:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \sqrt{2} \begin{bmatrix} \frac{\Delta x}{t(x,y)-t(x-1,y-1)} + \frac{\Delta x}{t(x,y)-t(x-1,y+1)} \\ \frac{\Delta x}{t(x,y)-t(x-1,y-1)} - \frac{\Delta x}{t(x,y)-t(x-1,y+1)} \end{bmatrix}$$

The $\sqrt{2}$ has to be included in diagonal estimates because of the physical distance of pixel centres.

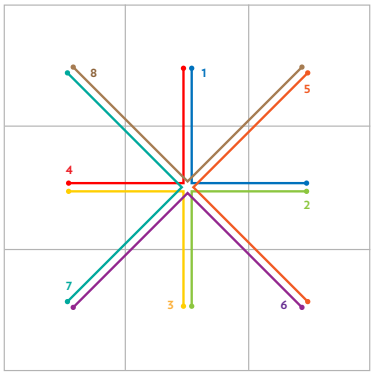

Figure 2: The pairs of time differences from which velocity hypotheses were generated

All hypotheses are initially given the same weight, and without further steps the resulting motion estimate is similar to the one given by the equation (1), but slightly smoother because the diagonally neighbouring pixels are also taken into account.

Two variants of the algorithm were tested, with feedback from the previous values of the same layer (horizontal V1 to V1 projections) and with feedback from the layer representing motion on a coarser scale (MT). In contrast to [2], the feedback enhancement of weights occurs for every velocity from the feedback layer, not just the one identical to it. Also, since the updates are asynchronous, feedback is also weighted by the time difference. The expression for the weight increase of a velocity hypothesis with weight w, velocity v at the time t is:

$$w_{enhanced} = w \left(1 + C \sum_{i}^{n_f} e^{-\frac{\|\mathrm{v}-\mathrm{v}_{f,i}\|^2}{\lambda^2}} e^{-\frac{t-t_{f,i}}{\tau}} w_{f,i} \right)$$

where C is a constant regulating the strength of feedback, n_f the number of feedback hypotheses and $\left(v_{f,i}, t_{f,i}, w_{f;i}\right)$ the parameters of the i-th feedback hypothesis.

If the feedback is sourced from a MT layer, it too has to be updated for each incoming event. Hypotheses are sampled in a neighbourhood of 5x5 pixels, to pool the hypotheses on a coarser spatial scale. Each of the 25 affected locations in MT have to be recalculated. The weights of the pooled velocity hypotheses are then multiplied by the values of a Hamming window function. Of these $25h_{max}$ hypotheses, h_{max} with the highest weight are retained and the rest is deleted. The retained hypothesis weights are squared to increase their relative differences. If the feedback is sampled from the first (V1) layer, the fed back hypotheses are again within a 5x5 neighbourhood with weights multiplied by values of the Hamming window.

To obtain final velocity estimate from a layer, for each pixel a weighted sum of velocities is calculated.

4 Quality evaluation

The performance of the algorithm was evaluated on several simple simulated and real-world datasets.

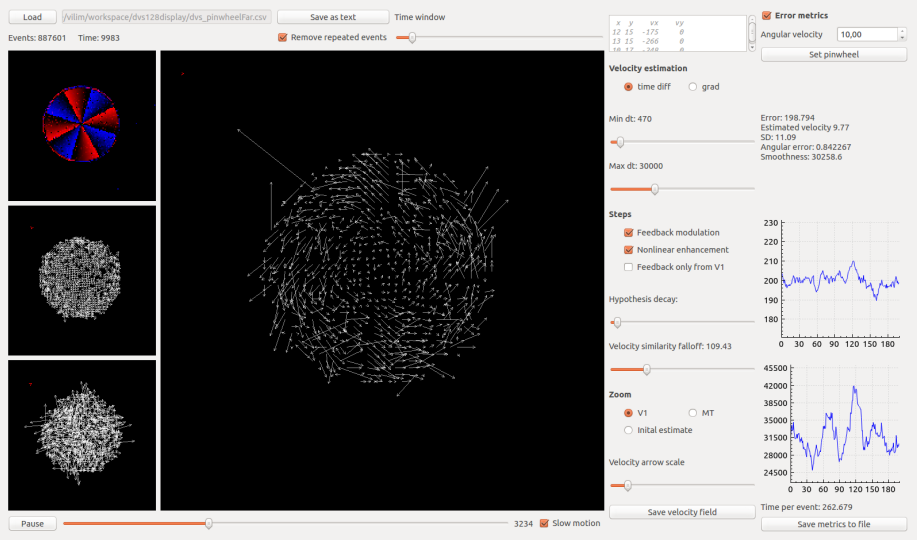

Figure 3: User interface showing the events, estimated velocities and quality metrics

An example where it is easy to establish the ground truth is a rotating pinwheel. The only parameter to estimate is the angular velocity, and it can be determined with independent timing to high accuracy. A previously-built setup with a DC motor and a frame for holding the DVS was used. The pinwheel was rotating with an angular velocity of 10.6 rad/s. The velocity field of a such an image is:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \omega \begin{bmatrix} -y \\ x \end{bmatrix} \tag{2}$$

for each (x, y) within a circle of radius r, with ω being the radial velocity. Two measures of estimation quality were used: the difference of the true and estimated velocity field and the estimate of the pinwheel velocity as obtained from the visual measurements.

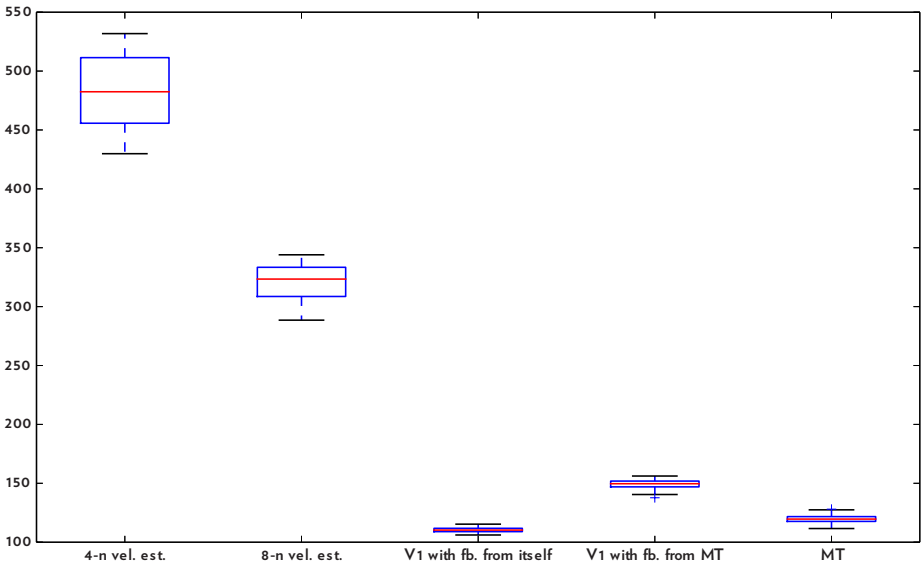

Figure 4: Velocity field errors for the rotating pinwheel

If the velocity field is known, the parameter ω can be determined by solving the equation 2. Estimated values and their distribution are displayed on figure 5.

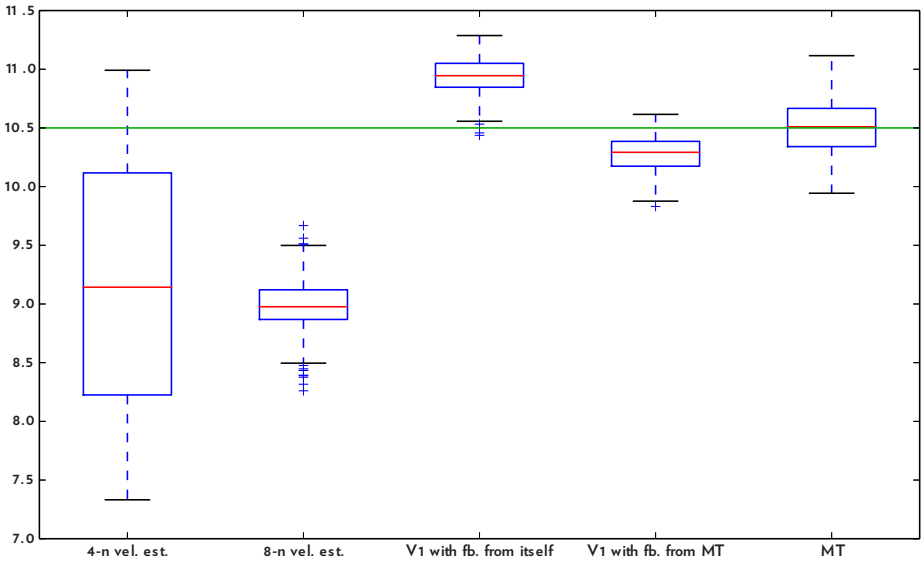

Figure 5: Estimated angular velocity of the pinwheel, the true value is in green

The second metric was the smoothness of the field obtained from panning the DVS. Smoothness was defined using the following expression:

$$S = \sum_{x=1}^{w} \sum_{y=1}^{h} \|\mathbf{v}(x,y) - \mathbf{v}(x-1,y)\|^2 + \|\mathbf{v}(x,y) - \mathbf{v}(x,y-1)\|^2$$

Where x and y are coordinates, w and h width and height and $\mathbf{v}(x,y)$ velocity at (x,y).

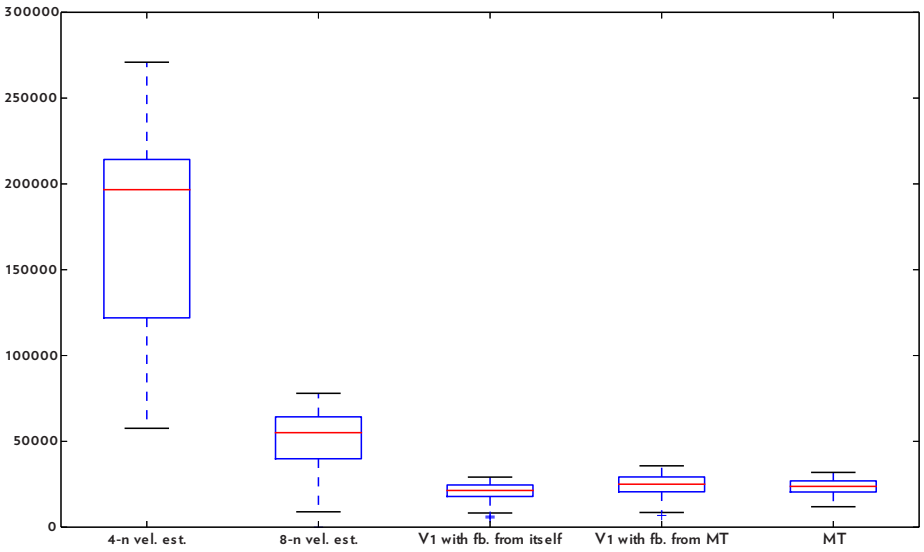

Figure 6: Smoothness for different layers and parameters

Direct comparison with frame-based methods for optic flow estimation is difficult, because there is not enough temporal resolution in publicly available synthetic image sequences to generate appropriate simulated inputs for the DVS.

4.1 Performance considerations

Event streams from the DVS in the test runs contain on the order of magnitude of 10^5 events per second. Time for processing an event was evaluated on two configurations (table 1).

Processor	RAM	OS	Compiler	no fb.	fb. from V1	fb. with MT
AMD Phenom II X4 945	4GB	Ubuntu Linux 13.04	gcc 4.7.3	3µs	70µs	270µs
Intel Core i7-3630QM	16GB	Mac OS X 10.9	CLANG 5.0	1.5µs	25µs	170µs

Table 1: Time per event for different configurations

For real-time performance the maximum time for processing an event should be about 10µs. Parallelisation of the algorithm could be obtained by independently processing events which are not within each other's area of influence.

5 Results and conclusion

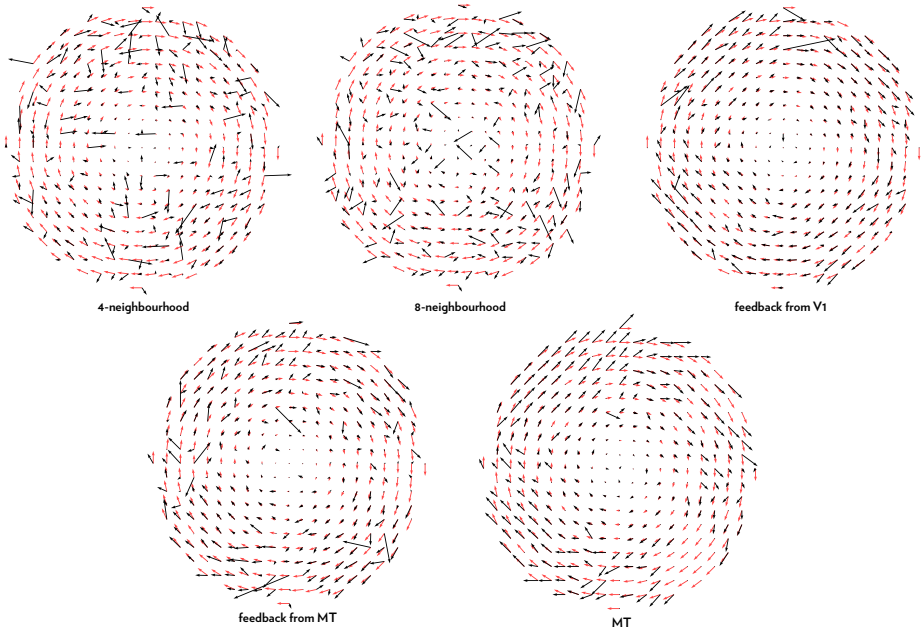

Figure 7: Estimated velocity fields for different cases

Estimating velocity from an 8-instead of 4- neighbourhood improves performance at a negligible computational cost. The feedback architecture with recurrent inputs within one layer offers significant error reduction (around 3 times for the tested case – figure 4), but on a PC runs 2 times slower then necessary for real time processing, and could be much slower for embedded processors. In the evaluated cases, an additional layer (MT) does not improve velocity estimation quality and comes at a significant computational cost, running approximately 20 times slower than real time. Its advantage is in grouping velocities which is not desired in rotating visual fields, but can be useful for object tracking. A parallelised implementation of the 1-layer feedback algorithm, on multi-processor hardware such as the SpiNNaker could yield high-quality optic flow estimation from event based data in real time.

6 Remarks on implementation

Data was captured from the DVS board via a serial interface and recorded using a Matlab script to obtain the test cases.

The artificial test cases were generated by projecting moving shapes with floating point coordinates to a pixel grid and outputting the changes in the grid every 25 µs, with optional noise.

The algorithm was implemented in C++ with the QT framework used for data visualisation and user interface.

Additional data processing was done using Python with the pandas, numpy and matplotlib libraries within an iPython notebook environment.

References

[1] John Allman, Francis Miezin, and EveLynn McGuinness. Direction-and velocity-specific responses from beyond the classical receptive field in the middle temporal visual area (mt). *Perception*, 14(2):105–126, 1985.

[2] Pierre Bayerl and Heiko Neumann. A fast biologically inspired algorithm for recurrent motion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):246–260, 2007.

[3] JM Hupe, AC James, BR Payne, SG Lomber, P Girard, and J Bullier. Cortical feedback improves discrimination between figure and background by v1, v2 and v3 neurons. *Nature*, 394(6695):784–787, 1998.