# Simultaneous Optical Flow and Intensity Estimation from an Event Camera

Patrick Bardow
p.bardow14@imperial.ac.uk

Andrew J. Davison
a.davison@imperial.ac.uk

Stefan Leutenegger
s.leutenegger@imperial.ac.uk

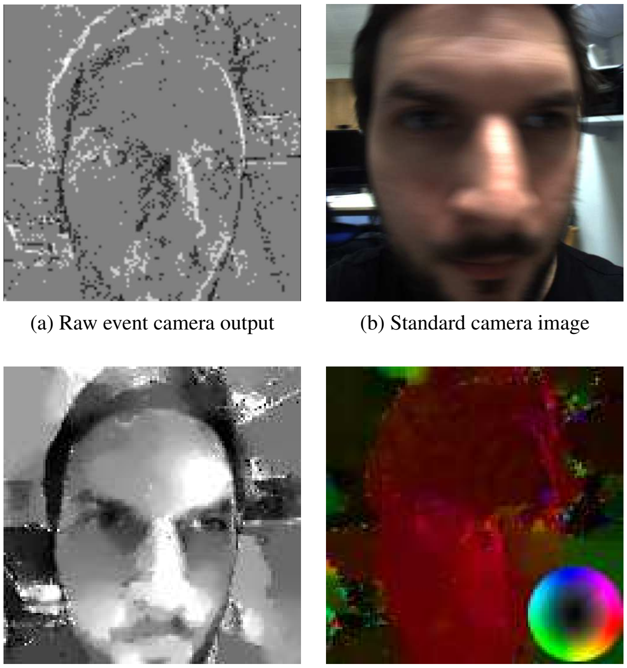
Dyson Robotics Laboratory at Imperial College, Dept. of Computing, Imperial College London, UK

## Abstract

*Event cameras are bio-inspired vision sensors which mimic retinas to measure per-pixel intensity change rather than outputting an actual intensity image. This proposed paradigm shift away from traditional frame cameras offers significant potential advantages: namely avoiding high data rates, dynamic range limitations and motion blur. Unfortunately, however, established computer vision algorithms may not at all be applied directly to event cameras. Methods proposed so far to reconstruct images, estimate optical flow, track a camera and reconstruct a scene come with severe restrictions on the environment or on the motion of the camera, e.g. allowing only rotation. Here, we propose, to the best of our knowledge, the first algorithm to simultaneously recover the motion field and brightness image, while the camera undergoes a generic motion through any scene. Our approach employs minimisation of a cost function that contains the asynchronous event data as well as spatial and temporal regularisation within a sliding window time interval. Our implementation relies on GPU optimisation and runs in near real-time. In a series of examples, we demonstrate the successful operation of our framework, including in situations where conventional cameras suffer from dynamic range limitations and motion blur.*

## 1. Introduction

The ‘silicon retina’ or event camera appears to offer enormous potential for a new level of performance in real-time geometric vision, and in the longer term a drive towards dramatically more efficient algorithms. It discards the frame-based paradigm of standard cameras and instead adopts a bio-inspired approach of independent and asynchronous pixel brightness change measurement. The highly appealing promise is that all of the information contained in a standard video stream of tens or more Megabytes of data per second is present in a natural and much compressed event stream of only tens or hundreds of kilobytes per second — all of the redundancy of sending regular, unchanging values where motion and intensity change is small is removed. As systems like LSD-SLAM [8] very clearly highlight, pixels where edges move are the only ones which give information useful for tracking and reconstruction, and it is precisely these locations which are highlighted in hardware by an event camera. On top of this, the pixels of an event

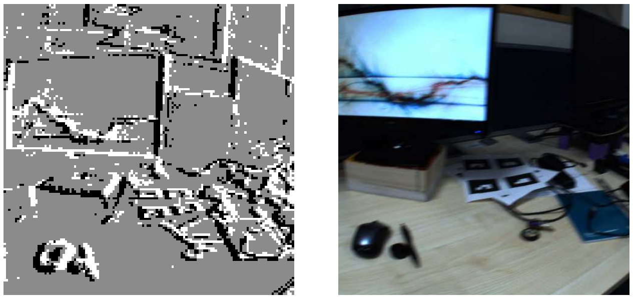

(a) Raw event camera output

(b) Standard camera image

(c) Intensity estimate from events

(d) Optical flow from events

Figure 1: Results from our method: (a) integrated output of a DVS128 event camera; (b) the same scene from a standard camera. (c) and (d) show a snapshots of the intensity and velocity fields we estimate jointly only from event data. A color-wheel is used to show the velocity per pixel.



(a) Event camera output

(b) Standard camera image

Figure 2: Desktop scene captured by an event camera (a) and a standard camera (b). The event camera image is an accumulation of all events during a period of 33 ms, where white and black pixels represents positive and negative events. Areas where no event has fired are grey.

camera with their low latency, microsecond-timestamped response and shutter-free independent intensity measurement offer the prospect of extreme high speed tracking and management of high dynamic range scenes.

However, since the original invention of the silicon retina [14] and the release by iniLabs of the Dynamic Vision Sensor (DVS) [13] as a research device several years ago, adoption of these cameras by the wider computer vision community has been minor. Standard vision algorithms simply do not transfer to an event camera. Traditional cameras capture *images*: synchronous intensity measurements; but events are asynchronous reports of local intensity difference, and there is no global or persistent measurement of intensity which would permit normal approaches to correspondence to be used (feature extraction, patch alignment).

In this paper we take a significant step towards proving the general potential of event cameras for motion and structure estimation by presenting the first algorithm which simultaneously estimates scene intensity and motion with minimal assumptions about the type of scene and motion.

## 2. Related Work

'Optical Flow' is in the title of our paper, because a key goal is to recover a generic motion field from camera data, but it is important to clarify the difference between the common understanding of this term in computer vision and the rather different type of estimation our algorithm achieves. Optical flow is normally understood as the correspondence field between two temporally close intensity images — an estimation of two parameters per pixel, the horizontal and vertical displacement between one frame and the next. In our case, we use the time stream of event data to estimate a continuously varying motion field at any time resolution. Therefore, it is more precise to say that we estimate a continuously time-varying *velocity field* in image coordinates.

Measuring velocity as a camera observes an arbitrary moving scene requires and implies knowledge of the correspondence between entities at different points in time, and this is where we face a particular challenge when the only input is event data which does not directly record image intensity or even oriented edges. We can only interpret an event as a measurement of motion if we know about the intensity gradient in the scene; but on the contrary we can only interpret it as a measurement of intensity gradient if we know the motion. Therefore, we must formulate a joint estimation problem to recover both motion and intensity together. We will show that weak assumptions about regularity in the overall solutions for motion and intensity are enough to allow this.

Previous computer vision work using event cameras, both the DVS and the related Asynchronous Time-based Image Sensor (ATIS) [19], has not aimed at such generic estimation, but instead focused on reduced problems. The first examples presented of the use of the DVS were for the tracking of simple objects to enable reactive robotic control (e.g.ball blocking [7] or pencil balancing [5]), highlighting the new level of performance enabled by the event camera's extremely low latency. Object tracking tasks like these have generally been tackled from a static camera using template fitting, or even more simply by finding the mean pixel coordinates of current event activity. These methods do not extend to tracking the independent motion of multiple, variable and highly textured objects.

Increasingly, authors have attempted to apply event cameras to more sophisticated vision tasks. In [15] an event camera was used to estimate the rapid motion of a quadcopter by using events to track a known target at low latency. Extending this, in [3] the image from a standard CMOS camera was used to provide a target for event tracking to produce a visual odometry system. Neither of these systems performed intensity reconstruction from event data.

The pieces of work which are most closely related to our approach are Benosman *et al*.'s optical flow estimation technique [1] and Kim *et al*.'s work on Simultaneous mosaicing and tracking [12]. In [1] the authors recovered a motion field without explicitly estimating image intensity, by assuming that events which fire spatially and temporally close to each other can be put into correspondence and locally fitting spatiotemporal planes to these. In a scene with sharp edges and monochromatic blocks this works well, but this approach has trouble in more complicated environments.

After some initial work by Cook *et al*. on integrating events into interacting maps [6], Kim *et al*. [12] demonstrated the first true high quality joint estimation of scene intensity and motion from event data, but under the strong assumption that the only movement is due to pure camera rotation in an otherwise static scene. They were able to demonstrate high quality intensity recovery, including super-resolution and HDR aspects, from purely event data. It was these results in particular which inspired us to work towards more generic estimation.

## 3. Method

Our algorithm is formulated as sliding window variational optimisation, and has much in common with well known variational methods for estimating two-view optical flow from standard video [11, 24]. We pre-define an optimisation time window $T$, and within this a fine time discretisation $\delta_t$. We take all of the events in time window $T$ as input and solve jointly for the velocity field $\mathbf{u}$ and log intensity $L$ at all cells in the associated spatio-temporal volume. We then slide the optimisation forward to a highly overlapping position, initialising the values of all cells to either previous estimates or predictions, and solve again.

### 3.1. Event Camera Definitions

Each pixel of an event camera independently measures intensity and reports an event when that intensity changes by a pre-defined threshold amount relative to a saved value. The pixels operate asynchronously but each event is time-stamped relative to a global clock. The electronics of the camera gather events from all pixels and transmit them to a computer as a serial stream.

Each event is therefore defined as a tuple $e_i = (\mathbf{x}_i, t_i, \rho_i)^\top$, where $\mathbf{x}_i \in \Omega$ is the position of the event in the image domain, $t_i$ is its time-stamp to microsecond resolution and $\rho_i \pm 1$ is its polarity (sign of the brightness change). An event is fired when a change in that log intensity exceeds threshold $\theta$:

$$|L(\mathbf{x}, t) - L(\mathbf{x}, t_p(\mathbf{x}, t))| \geq \theta \ , \tag{1}$$

where $L(\mathbf{x}, t)$ is the log intensity at pixel $\mathbf{x}$ at time $t$ and $t_p(\mathbf{x}, t)$ is the time when the previous event occurred.

### 3.2. Estimation Preliminaries

We aim to estimate continously varying image velocity $\mathbf{u}$ and log intensity $L$ at all image pixels over the duration of our input event sequence. The log intensity is related to image intensity as follows: $L := \log(I + b)$, where $b$ is a positive offset constant. Since event cameras do not come with any notion of frames, we note that $\mathbf{u}$ is in velocity units of pixels/second rather than a frame-to-frame displacement. For brevity we will write partial derivatives as $\mathbf{u_x} := \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$.

As in well-known two-view methods for optical flow estimation [11, 24], a key assumption of our algorithm is brightness constancy, which asserts that the brightness value of a moving pixel is unchanged. In differential form this is:

$$I(\mathbf{x} + \delta_t \mathbf{u}, t + \delta_t) = I(\mathbf{x}, t). \tag{2}$$

On a per-pixel basis, this equation is under-determined. As in [11, 24] and many other optical flow methods, we need to introduce regularisation and perform global optimisation in order to achieve a well-defined solution across the whole image domain simultaneously.

But in the case of the input data from an event camera, Equation (2) cannot be directly applied since event measurements do not provide absolute intensity information but only differences. To proceed we must formulate our problem as simultaneous estimation of both intensity and velocity. The details of our approach follow in the next section.

### 3.3. Variational Formulation

As mentioned in the previous section, we add regularisers to (2) to determine the system and to handle the sparse measurements from the event camera. These are, in essence, smoothness priors which have been applied in many image processing application for standard cameras, such as optical flow [23], image denoising [21] and SLAM [16]. Variational methods have been very successful to include smoothness priors such as TV-L$^1$ [18], which approximates natural image statistics [20], while leaving the optimisation problem convex.

With event cameras, smoothness priors allow us to estimate image regions in between events — both spatially and temporally. In other words, we have sensor regions where events are firing and giving information about gradients, and we regions with no data and no events firing, but we can assume smoothness in the absence of events.

Since an event relates an intensity in the past, *i.e. previous event* to an intensity at the *current* event time-stamp, we opt for assimilating the event measurement data to the spatio-temporal smoothness and photometric consistency (optical flow constraint) within a time window. As opposed to traditional optical flow estimation, intensities are unknown, and so is optical flow. Since both quantities are coupled by the optical flow constraint (2), we need to estimate both *jointly*. We assume that the intensity change at a pixel is induced only by optical flow.

We therefore propose the following minimisation:

$$\begin{aligned}\min_{\mathbf{u},\, L} \int_\Omega \int_T \Big(\lambda_1 \|\mathbf{u_x}\|_1 + \lambda_2 \|\mathbf{u}_t\|_1 + \lambda_3 \|L_\mathbf{x}\|_1 + \\ \lambda_4 \|\langle L_\mathbf{x}, \delta_t \mathbf{u}\rangle + L_t\|_1 + \lambda_5\, h_\theta(L - L(t_p))\Big) \mathrm{d}t\, \mathrm{d}\mathbf{x} \\ + \int_\Omega \sum_{i=2}^{|P(\mathbf{x})|} \|L(t_i) - L(t_{i-1}) - \theta\rho_i\|_1 \mathrm{d}\mathbf{x},\end{aligned} \tag{3}$$

where the individual $\lambda$s are positive scalar weights. For brevity we omit the parameters $\mathbf{x}$ and $t$. Readers who are familiar with optical flow and image denoising will recognise the first four terms, which regularise the smoothness of the flow (both spatially and temporally) and the smoothness of the intensities. The fourth term is the first order Taylor approximation of (2), with $\langle \cdot, \cdot \rangle$ being the inner product, which ensures temporal consistency in our intensity estimates.

The last two terms of (3) are the data terms of the event camera: The *event data term* and the *no-event data term*. The event data term is derived from (1), where $P(\mathbf{x})$ is the set of all events fired at $\mathbf{x}$, with $t_i$ and $\rho_i$ being the timestamp and polarity of the $i$-th element in $P(\mathbf{x})$. For this term we assume that events are fired as soon as the threshold $\theta$ in log intensity is reached. While this term models events which have been fired, the no-event data term models the case of no events occurring on a certain pixel: the absence of events gives us the information that the log intensity, after the last event, has not changed more than the given threshold $\theta$. Therefore, we constrain the intensity between two events with an $L^1$-norm cost term containing a dead-zone, which is denoted by $h_\theta$. $h_\theta$ is defined as

$$h_\theta(x) = \begin{cases} |x| - \theta, & \text{if } |x| > \theta \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

and takes as input the difference between $L$ and $L(t_p)$, which is the log intensity at last event relative to $L$. By using the dead zone, the term does not add any cost when the difference is in bound of $[-\theta, \theta]$, but penalises deviation beyond. We use an $\mathrm{L}^1$-norm, in both terms, as we anticipate outliers. Events may be missed by the chip specifically in a short period just after an event has fired – and we anticipate randomly firing events (background noise), which occur in the sensor due to leakage [13]. Next we will describe the discretisation and minimisation of (3).

### 3.4. Discretisation

As described in the previous section, we estimate $L$ and $\mathbf{u}$ over a time period $T$ and over the image domain $\Omega$. For the minimisation we discretise $\Omega$ into a regular pixel grid of size $M \times N$, and $T$ into $K$ cells each of length $\delta_t$ microseconds, which forms a spatio-temporal volume. For each element in the volume created we estimate $L$ and the motion $\mathbf{u}$ by minimising the discretised version of (3). After the minimisation, we then slide the window in time by $\delta_t$ and minimise again. Figure 3 visualises this scheme.

Important here is the choice of $\delta_t$, since a larger choice of $\delta_t$ allows to estimate slower motions, while a smaller value is good for fast motions. We use a constant value for $\delta_t$, but in future work this choice may be automated by adopting it to the rate of incoming events. To adapt our event data term to a lower temporal resolution we linearly interpolate the intensity at the time of each event, as described in Figure 4.

After shifting the sliding window, the oldest estimates and related event data terms are dropping out while we add new incoming events to the window. These new elements in our volume are initialised by assuming a constant motion from our previous estimate. For this we "copy" the previous estimates of $\mathbf{u}$ into the new grid cells and use the newly "copied" motion vectors $\mathbf{u}$ to bilinearly interpolate from the previous log intensity estimate.

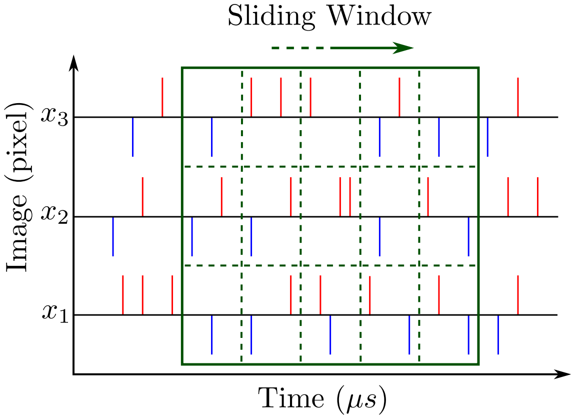


Figure 3: The sliding window (green box) bins the incoming positive events (red bars) and negative events (blue bars) into a regular grid (dashed lines). When the minimisation converges, the window is shifts to the right.

Unfortunately, by deleting estimates from the oldest band of grid cells, we lose information about that period. To compensate this loss, we constrain the oldest estimates in our sliding window with a *prior image* and penalise deviations from those values in the next minimisation. The idea is that consecutive estimates by the sliding window should be discouraged to discard previous estimates, since they are based on measurements, which are not included anymore. To include those priors, we add $\lambda_6 \| L(\mathbf{x}, t_1) - \hat{L}(\mathbf{x}) \|_2^2$ to (3), where $\hat{L} \in \Omega$ is the prior image and $t_1$ is the first event time-stamp at $\mathbf{x}$ in our sliding window. If there is no such $t_1$ at $\mathbf{x}$, then we use instead of $t_1$ the minimum of $T$. Before each minimization, we update $\hat{L}(\mathbf{x})$ by copying the values of $L(\mathbf{x}, t_1)$ for all pixels and if there are no events at $\mathbf{x}$, because they dropped out of the sliding window, we leave the value in $\hat{L}$ unchanged. By using this prior image scheme, we can mitigate the loss of information to a certain extent.

We also want to point out that the prior image scheme would allow to include intensity *measurements* from other sources, *e.g.* with DAVIS240 [2], which provides a standard camera image besides an event stream. However, in this work we focus on exclusively analysing the event stream.

### 3.5. Optimisation

To minimise (3) we use the preconditioned primal-dual algorithm [17]. The advantage of that scheme is its optimal convergence and that it is easily parallelisable. To use the primal dual algorithm we use the duality principle and replace individual L$^1$-norms of (3) by their conjugate using the Legendre-Fenchel transform [10]:

$$\min_{\mathbf{u},\,L} \max_{\substack{|\mathbf{a}|_\infty \le \lambda_1\\ |\mathbf{b}|_\infty \le \lambda_2\\ |\mathbf{c}|_\infty \le \lambda_3\\ |\mathbf{d}|_\infty \le \lambda_4\\ |\mathbf{y}|_\infty \le 1}} \begin{aligned} &\langle \mathbf{D_x u}, \mathbf{a}\rangle - \delta_{\lambda_1 A}(\mathbf{a}) + \langle \mathbf{D}_t \mathbf{u}, \mathbf{b}\rangle - \\ &\delta_{\lambda_2 B}(\mathbf{b}) + \langle \mathbf{D}'_{\mathbf{x}} L, \mathbf{c}\rangle - \delta_{\lambda_3 C}(\mathbf{c}) + \\ &\langle\langle \mathbf{D}'_{\mathbf{x}} L, \mathbf{u}\,\delta_t\rangle + \mathbf{D}'_t L, \mathbf{d}\rangle - \delta_{\lambda_4 D}(\mathbf{d}) + \\ &\lambda_6 h_\theta(L - L(t_p)) + \langle \mathbf{E}L - \mathbf{z}, \mathbf{y}\rangle - \delta_Y(\mathbf{y}), \end{aligned} \tag{5}$$

where $\mathbf{D_x}$, $\mathbf{D}'_{\mathbf{x}}$ represent the finite difference matrices with respect to $\mathbf{x}$ and $\mathbf{D}_t$, $\mathbf{D}'_t$ are the difference matrices with respect to $t$. The individual $\delta(.)$ terms are the indicator functions regards to the dual variables, such that for example $\delta_{\lambda_1 A}(\mathbf{a}) = 0$ if $\|\mathbf{a}\| \le \lambda_1$, otherwise $\infty$. The other indicator functions are defined likewise. Note that we reformulate the event data term to the matrix expression $\mathbf{E}L - \mathbf{z}$, where $\mathbf{z}$ is our measurement vector containing signs of all observed events scaled by $\theta$ and $\mathbf{E}$ is the *event matrix* which transforms the intensity estimate to pairwise differences of linearly interpolated intensities of the observed events.

The optical flow term in (5) is biconvex, due to the inner product of $L_{\mathbf{x}}$ and $\mathbf{u}$. We following here the minimisation strategy for a biconvex function in [9] by minimising this term by alternating between the estimation for $L$ and $\mathbf{u}$. This gives us the following minimisation scheme of (5):

$$\begin{cases} L^{n+1} = (\mathbf{I} + \mathbf{T_1}\lambda_6 \partial h_\theta)^{-1}(\bar{L}^n - \mathbf{T_1}({\mathbf{D}'_{\mathbf{x}}}^\top \mathbf{c}^n \\ \qquad - {\mathbf{D}'_{\mathbf{x}}}^\top(\bar{\mathbf{u}}^n\,\mathbf{d}^n) - {\mathbf{D}'_{\mathbf{t}}}^\top \mathbf{d}^n - \mathbf{E}^\top \mathbf{y}^n)) \\ \bar{L}^{n+1} = 2L^{n+1} - \bar{L}^n \\ \mathbf{u}^{n+1} = (\mathbf{I} + \mathbf{T_2}\lambda_4 \partial G)^{-1}(\bar{\mathbf{u}}^n - \mathbf{T_2}({\mathbf{D_x}}^\top \mathbf{a}^n \\ \qquad - \mathbf{D}_t^\top \mathbf{b}^n))) \\ \bar{\mathbf{u}}^{n+1} = 2\mathbf{u}^{n+1} - \bar{\mathbf{u}}^n \\ \mathbf{a}^{n+1} = (\mathbf{I} + \mathbf{\Sigma_1}\partial F_1^*)^{-1}(\mathbf{a}^n + \mathbf{\Sigma_1 D_x}\bar{\mathbf{u}}^{n+1}) \\ \mathbf{b}^{n+1} = (\mathbf{I} + \mathbf{\Sigma_2}\partial F_2^*)^{-1}(\mathbf{b}^n + \mathbf{\Sigma_2 D}_t\bar{\mathbf{u}}^{n+1}) \\ \mathbf{c}^{n+1} = (\mathbf{I} + \mathbf{\Sigma_3}\partial F_3^*)^{-1}(\mathbf{c}^n + \mathbf{\Sigma_3 D}'_{\mathbf{x}}\bar{L}^{n+1}) \\ \mathbf{d}^{n+1} = (\mathbf{I} + \mathbf{\Sigma_4}\partial F_4^*)^{-1}(\mathbf{d}^n + \mathbf{\Sigma_4}(\langle \mathbf{D}'_{\mathbf{x}}\bar{L}^{n+1}, \bar{\mathbf{u}}^{n+1}\rangle \\ \qquad + \mathbf{D}'_{\mathbf{t}}\bar{L}^{n+1})) \\ \mathbf{y}^{n+1} = (\mathbf{I} + \mathbf{\Sigma_5}\partial F_5^*)(\mathbf{d}^n + \mathbf{\Sigma_5}(\mathbf{E}\bar{L}^{n+1} - \mathbf{z})), \end{cases}$$

where $\mathbf{\Sigma_i}$ and $\mathbf{T_i}$ are diagonal pre-conditioning matrices as described in [17] and $\mathbf{I}$ is the identity matrix. Following the notation in [4], $F_i^*$ represents the indicator functions and $G$ is the optical flow term. Their respective resolvent operators are defined as in [17]. $(\mathbf{I} + \mathbf{T_1}\partial h_\theta)^{-1}$ is the resolvent operator with respect to $h_\theta$, which can be solved by a soft-thresholding scheme for each log intensity estimate $L_i$:

$$(\mathbf{I} + \lambda_6 \tau_i h_\theta)^{-1}(\tilde{L}_i) = \tilde{L}_i + \begin{cases} -\lambda_6\tau_i, & \text{if } (\tilde{L}_i - L(t_p)) > \theta + \lambda_6\tau_i \\ \lambda_6\tau_i, & \text{if } (\tilde{L}_i - L(t_p)) < -\theta - \lambda_6\tau_i \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

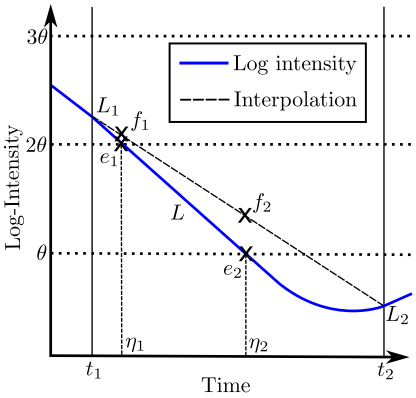


Figure 4: Approximation of the intensity for two given events $e_1$ and $e_2$ between two intensity estimates $L_1$ and $L_2$. For the discrete data term, we use the linear approximations $f_1$ and $f_2$ at the time of each event $\eta_1$ and $\eta_2$.

where we fix $L(t_p)$ during the minimisation step. A fixed $L(t_p)$ can slow down the convergences of the algorithm, but in practise we have not experienced such behavior. Next we discuss the result of this minimisation scheme.

## 4. Experiments

We present the capabilities of our method in a series of experiments with the DVS128 camera [13]. For these experiments we choose a spatial discretisation of $128 \times 128$, which corresponds to the resolution of the actual camera. To highlight these results in this paper, we show log intensity and velocity field estimates at certain time-stamps to highlight how they resemble images and optical flow fields from standard cameras. However, we believe that these results are best viewed in the accompanying video on our project webpage [1], where also show that we can estimate super-resolution log-intensity and velocity, via a simple extension to our formulation based on the work of Unger *et al.* for standard cameras [22]. Although we do not discuss the super-resolution method in detail here, it is important to highlight that event camera data allows us to perform intensity and velocity estimation at sub-pixel resolution.

For the following experiments, we set the sliding window depth $K$ to 128, which we have found to be an appropriate choice for most sequences to capture a large amount of events. If not specified otherwise, $\delta_t$ is set to 15 milliseconds. For all sequences we set $\theta = 0.22$, $\lambda_1 = 0.02$, $\lambda_2 = 0.05$, $\lambda_3 = 0.02$, $\lambda_4 = 0.2$, $\lambda_5 = 0.1$ and $\lambda_6 = 1.0$. All sequences are initialised by assuming no initial motion and only a uniform gray scale intensity distribution, which includes the prior image as well. For comparison, we

[1] http://www.imperial.ac.uk/dyson-robotics-lab/

mounted the DVS128 next to a standard frame-based camera with standard $640 \times 480$ resolution, 30 fps and global shutter settings for all sequences.

### 4.1. The Benefits of Simultaneous Estimation

We begin with an experiment to argue for the simultaneous estimation of intensities and optical flow with event cameras. In this sequence we compare the intensity estimate of our method with and without using of the optical flow term as defined in (3) (Figure 5).

From these results we can see that our method, without the optical flow, can still estimate the intensities well in regions with strong gradient, but between those regions artefacts occur which do not correspond with the real intensities in the scene. Also we see that the term enforces temporal consistency and without it areas become brighter or darker from frame to frame.

### 4.2. Face Sequence

In this sequence we show intensity and velocity reconstruction of a moving face while the camera is in motion as well. The results are shown in Figure 6.

At the beginning of the sequence (left), the intensities have not been properly estimated yet, because only a few events have been captured. We see that more details become visible in the following frames as more events are processed. Velocity visualisation shows clear motion boundaries between the head and the background, proving that our approach can handle motion discontinuities. However, both the intensities and optical flow show noise, presumably caused by outlier events and/or missing data.

### 4.3. High Dynamic Range Scene

In this example, we show the comparison between our method and a traditional camera in a high dynamic range scene. In our experiment, we point the cameras out a window from a dim room, which is a challenging case for a traditional camera, as can be seen in Fig. 7.

We see that our method recovers details both inside the room and outside the window, while the traditional camera, because of its low dynamic range, can only show either the room or the outside at one time.

### 4.4. Rapid Motion

Here we present the capability of our method to estimate fast motion in front of a cluttered background. We throw a ball in front a desktop scene, while the camera is also in motion (Figure 8).

For this sequence, we set $\delta_t$ to a smaller value of 4 milliseconds. We see how the traditional camera is affected by motion blur, while our method is able to recover clear motion boundaries. However, due to the small $\delta_t$ intensity details are not estimated as well as in the previous examples. This gives the impression that the ball is transparent.

### 4.5. Full Body Motion

In our last example we show a person performing jumping jacks. For this sequence we set $\delta_t$ to 7 ms and reduce $\lambda_1$ and $\lambda_3$ to $0.01$, which preserves smaller regions from being smoothed out in the optical flow and intensity estimate. In Fig. 9 we see that our method can estimate arm motion well, even though it occupies a small image region. However, with the decreased smoothness weight, the influence of noisy events is stronger, which becomes visible in the motion field.

## 5. Conclusions

We have shown that event data from a standard DVS128 sensor can be used to reconstruct high dynamic range intensity frames jointly with a dense optical flow field. As demonstrated experimentally, our sliding window optimisation based method does neither impose any restrictions on camera motion nor scene content. In particular we have shown tracking and reconstruction of extreme, rapid motion and high dynamic range scenes which are beyond the capabilities of frame-based cameras. We thus believe that this work is important in supporting the claim that event cameras will play a major role in real-time geometric vision — the information in a low bit-rate event stream really does contain all of the content of continuous video, and more. Having proven a very general capability, we plan to furthermore investigate specific efficient estimation algorithms for particular problems such as model-based 3D tracking, 3D reconstruction and SLAM.

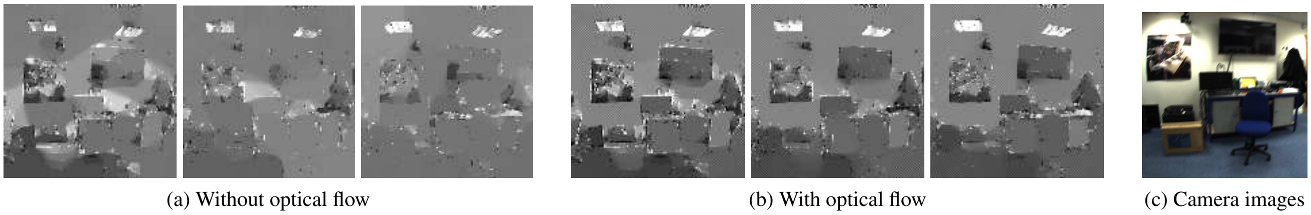
We also believe that this work strengthens the argument that research on embodied real-time vision needs to look at all of the components of a hardware/software vision system together (sensors, algorithms and ultimately also processors) in order to reach for maximum performance. Progress in the specific technology of event cameras must be compared with alternatives, always with an eye on the rapidly improving and strongly market-driven performance of standard frame-based cameras.

## Acknowledgements

Research presented in this paper has been supported by Dyson Technology Ltd. We are grateful to Jan Jachnik and Ankur Handa for very useful discussions.

## References

[1] R. Benosman, C. Clercq, X. Lagorce, S. Ieng, and C. Bartolozzi. Event-Based Visual Flow. *IEEE Transactions on Neural Networks and Learning Systems*, 25:407–417, 2014.


(a) Without optical flow

(b) With optical flow

(c) Camera images

Figure 5: Comparison of intensity estimation without (a) and with (b) the optical flow term. As usual in (c) we show an image from a standard camera for reference.

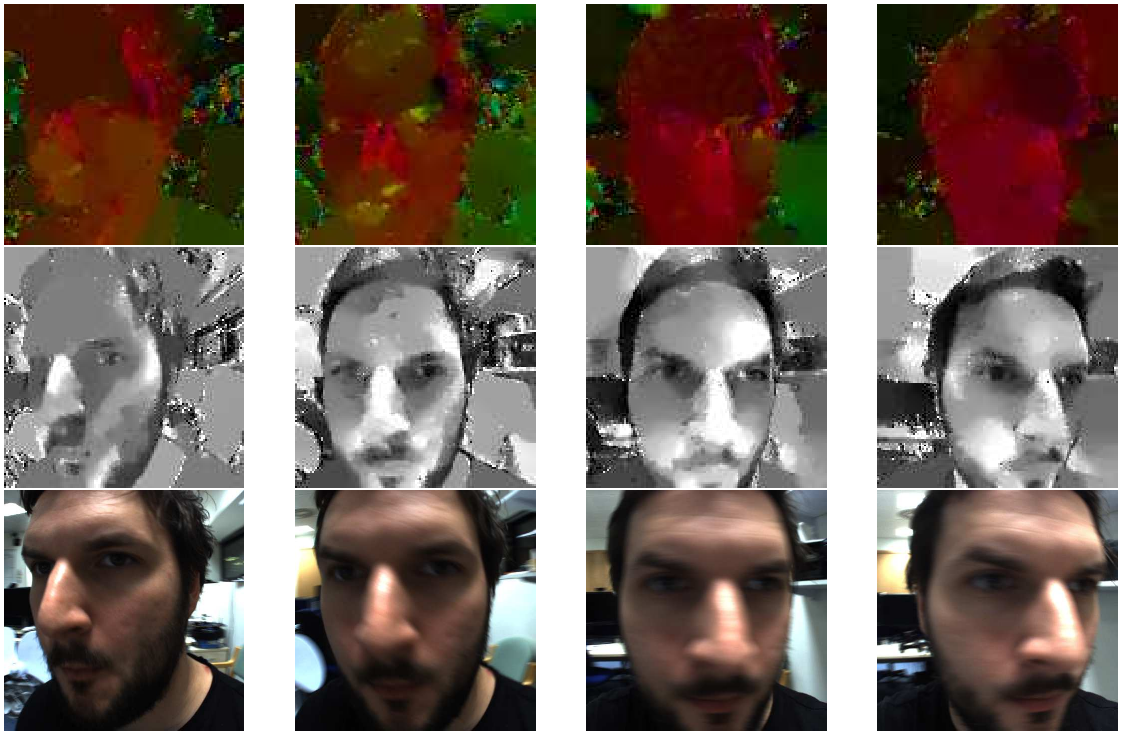


Figure 6: Hand-held event camera moving around a person's head. Here the face is moving from left to right, while the background is moving in the opposite direction, which can be seen in the estimated velocity field (top row). The middle row shows high quality and consistent reconstructed intensity. Bottom row: standard video for comparison.

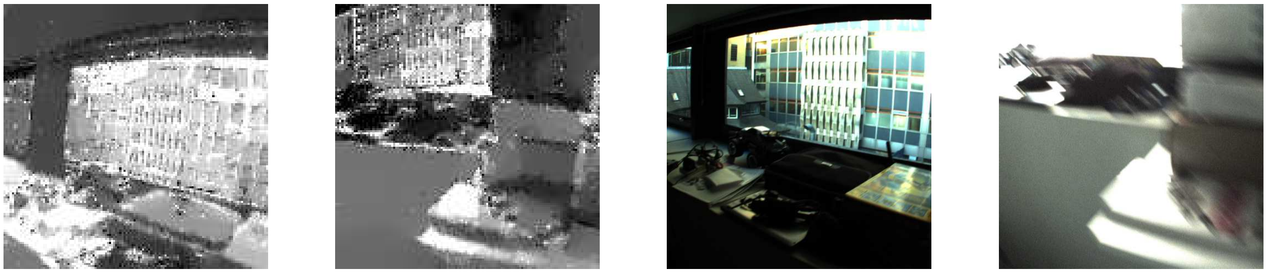


Figure 7: High dynamic range scene. Intensity reconstructions from events (left) contrasted with standard video images (right). The camera moves from observing the bright outside scene to point towards a shelf inside the dim room.

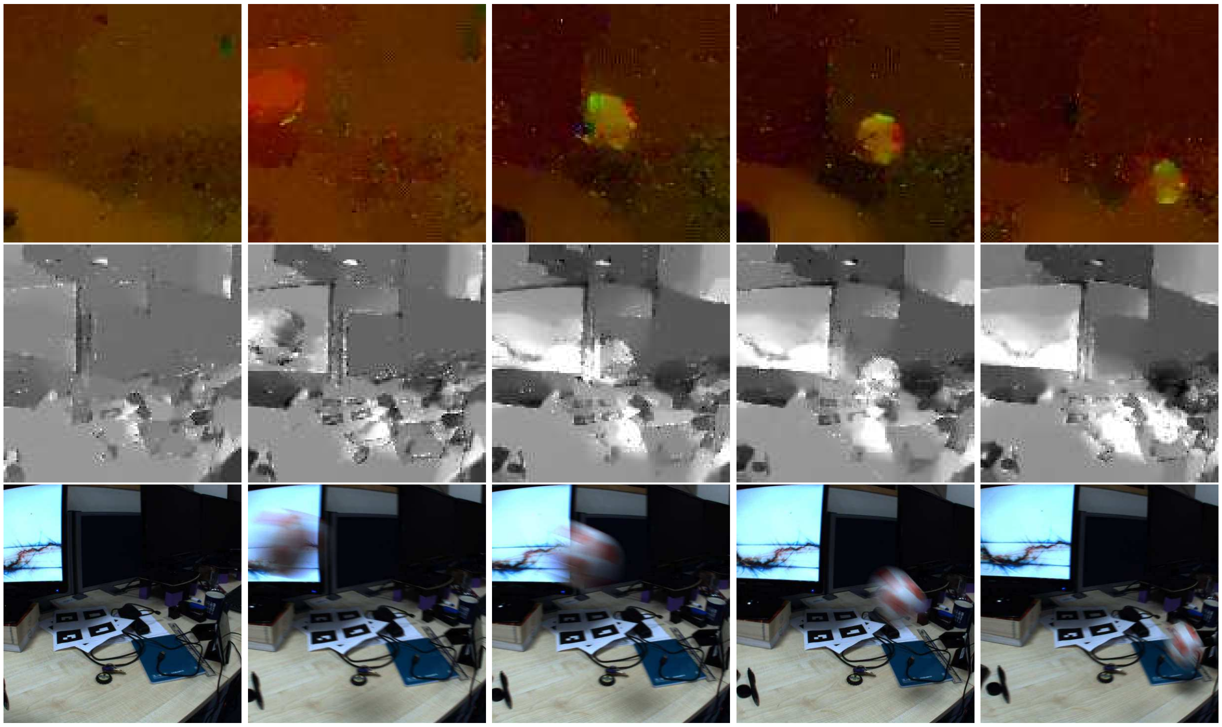


Figure 8: Ball thrown across a desktop scene. In the estimated velocity field (top) we see clear segmentation of the ball, while video from a standard camera (bottom) is heavily blurred. In the intensity reconstruction (middle) we observe good reconstruction of the monitor which was traversed by the ball, causing many events.

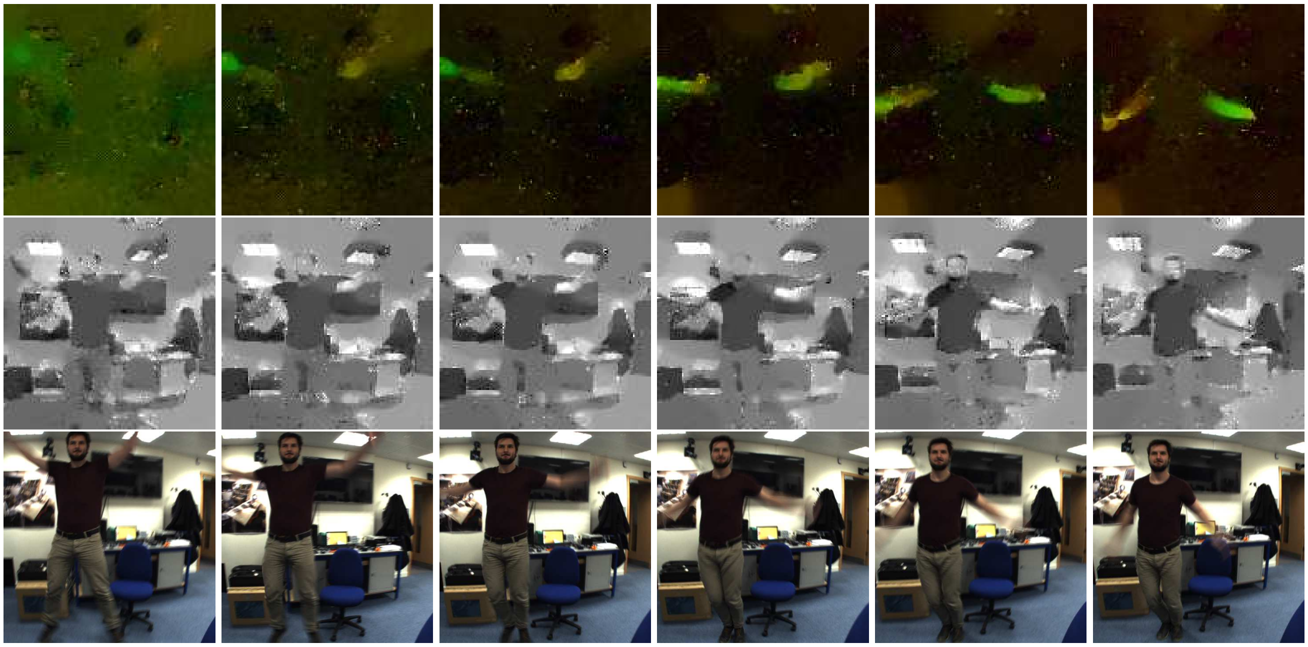


Figure 9: Jumping jacks. In the sequence from a standard camera the arms are blurred, while our reconstruction from events allows clear delineation of the arms in both the intensity and velocity fields.

[2] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck. A 240×180 130 dB 3 $\mu$s Latency Global Shutter Spatiotemporal Vision Sensor. *IEEE Journal of Solid-State Circuits (JSSC)*, 49(10):2333–2341, 2014.

[3] A. Censi and D. Scaramuzza. Low-Latency Event-Based Visual Odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[4] A. Chambolle and T. Pock. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[5] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R. Douglas, and T. Delbruck. A pencil balancing robot using a pair of AER dynamic vision sensors. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2009.

[6] M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger. Interacting maps for fast visual interpretation. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2011.

[7] T. Delbruck and P. Lichtsteiner. Fast sensory motor control based on event-based hybrid neuromorphic-procedural system. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2007.

[8] J. Engel, T. Schoeps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[9] R. Garg, A. Roussos, and L. Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1272–1279, 2013.

[10] A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison. Applications of the Legendre-Fenchel transformation to computer vision problems. Technical Report DTR11-7, Imperial College London, 2011.

[11] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[12] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. J. Davison. Simultaneous Mosaicing and Tracking with an Event Camera. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2014.

[13] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128×128 120 dB 15 $\mu$s Latency Asynchronous Temporal Contrast Vision Sensor. *IEEE Journal of Solid-State Circuits (JSSC)*, 43(2):566–576, 2008.

[14] M. Mahowald and C. Mead. The Silicon Retina. *Scientific American*, 264(5):76–82, 1991.

[15] E. Mueggler, B. Huber, and D. Scaramuzza. Event-based , 6-DOF Pose Tracking for High-Speed Maneuvers. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2014.

[16] R. A. Newcombe, S. Lovegrove, and A. J. Davison. DTAM: Dense Tracking and Mapping in Real-Time. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[17] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[18] T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A convex relaxation approach for computing minimal partitions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[19] C. Posch, D. Matolin, and R. Wohlgenannt. A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS. *IEEE Journal of Solid-State Circuits (JSSC)*, 2011.

[20] E. Reinhard, P. Shirley, and T. Troscianko. Natural image statistics for computer graphics. *Univ. Utah Tech Report UUCS-01-002 (Mar. 2001)*, 2001.

[21] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[22] M. Unger, T. Pock, M. Werlberger, and H. Bischof. A Convex Approach for Variational Super-Resolution. In *Proceedings of the DAGM Symposium on Pattern Recognition*, 2010.

[23] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An Improved Algorithm for TV-L1 Optical Flow. In *Proceedings of the Dagstuhl Seminar on Statistical and Geometrical Approaches to Visual Motion Analysis*, 2009.

[24] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *Proceedings of the DAGM Symposium on Pattern Recognition*, 2007.