Simultaneous Mosaicing and Tracking with an Event Camera

Hanme Kim[1]
hanme.kim@imperial.ac.uk
Ankur Handa[2]
ah781@cam.ac.uk
Ryad Benosman[3]
ryad.benosman@upmc.fr
Sio-Hoi Ieng[3]
sio-hoi.ieng@upmc.fr
Andrew J. Davison[1]
a.davison@imperial.ac.uk

[1] Department of Computing,
Imperial College London,
London, UK

[2] Department of Engineering,
University of Cambridge,
Cambridge, UK

[3] Sorbonne Universités, UPMC Univ Paris 06, UMR_S 968, Institut de la Vision, Paris, F-75012, CNRS, France.

Abstract

An event camera is a silicon retina which outputs not a sequence of video frames like a standard camera, but a stream of asynchronous spikes, each with pixel location, sign and precise timing, indicating when individual pixels record a threshold log intensity change. By encoding only image change, it offers the potential to transmit the information in a standard video but at vastly reduced bitrate, and with huge added advantages of very high dynamic range and temporal resolution. However, event data calls for new algorithms, and in particular we believe that algorithms which incrementally estimate global scene models are best placed to take full advantages of its properties. Here, we show for the first time that an event stream, with no additional sensing, can be used to track accurate camera rotation while building a persistent and high quality mosaic of a scene which is super-resolution accurate and has high dynamic range. Our method involves parallel camera rotation tracking and template reconstruction from estimated gradients, both operating on an event-by-event basis and based on probabilistic filtering.

1 Introduction

Real-time, real-world vision applications such as in robotics and wearable computing require rapid reaction to dynamic motion, and the ability to operate in scenes which contain large intensity differences. Standard video cameras run into problems when trying to supply this, either of huge bandwidth requirements at high frame-rates or diminishing image quality with blur, noise or saturation [11]. To overcome these limitations, researchers in neuromorphics have built new visual sensors aiming to emulate some of the properties of the human retina [5]. An event camera has no shutter and does not capture images in the traditional sense. Instead, each pixel responds independently to discrete changes in log intensity by generating asynchronous events, each with microsecond-precise timing. The bandwidth of an event stream is much lower than for standard video, removing the redundancy in continually repeated image values; but an event stream should in principle contain all of the information of standard video and without the usual bounds on frame-rate and dynamic range.

However, the uses demonstrated of event cameras have been limited. We are interested in scene understanding and SLAM applications where the camera itself moves and there has been little work on building coherent scene models from event data. The clear difficulty is that most methods and abstractions normally used in reconstruction and tracking, such as feature detection and matching or iterative image alignment, are not available.

In this paper we show that the pure event stream from a hand-held event camera undergoing 3D rotations, with no additional sensing, can be used to generate high quality scene mosaics. We use a SLAM-like method of parallel filters to jointly estimate the camera's rotation motion and a gradient map of a scene. This gradient map is then upgraded to a full image-like mosaic with super-resolution and high dynamic range properties.

When an event camera moves, events are triggered at pixel locations where intensity edges cross its field of view. However, an event camera does *not* directly measure image gradients but only the locations, signs and times of brightness changes. The presence, orientations and strengths of edges must be estimated together with the camera's rotation trajectory. As each new event is received, current tracking estimates of the camera's position and velocity relative to the mosaic mean that the event serves as a measurement of the component of gradient parallel to the motion direction of that pixel. We refine estimates of gradients, as well as the motion of the camera, using Bayesian filtering on an event-by-event basis.

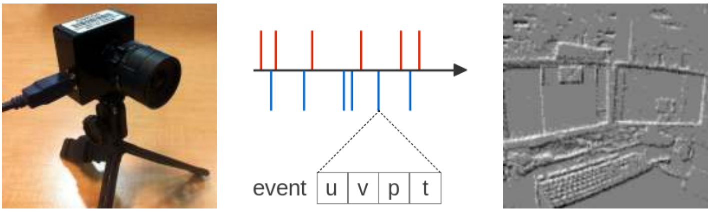
Event cameras can be seen as the logical conclusion of devices such as rolling shutter cameras which have some degree of non-global capture; but are a much more powerful proposition since their output is purely data-driven and they do a lot of the hard things we are used to doing in computer vision to determine which pixels are useful for tracking or reconstruction *in hardware, at no computational cost*. We hope that our work, as well as having a strong engineering interest, also shines some more light on why a biological retina works as it does and suggests that it still has an awful lot to teach us in computer vision.

2 Event cameras

We use the first commercial event camera, the Dynamic Vision Sensor (DVS) [13] shown in Figure 1(a). It has 128×128 resolution, 120 dB dynamic range and 15 microsecond latency, and communicates with a host computer using USB 2.0. It outputs a stream of events, each consisting of a pixel location, a polarity bit for positive or negative change in log intensity and a timestamp in microseconds as depicted in Figure 1(b). We can visualise its output as shown in Figure 1(c) by accumulating events within a time interval; in this figure, white and black pixels represent positive and negative events respectively. We should note that there is a newer version, the Asynchronous Time-based Image Sensor (ATIS) [17], which has higher resolution (304×240), higher dynamic range (143 dB) and lower latency; and ATIS provides an absolute intensity for each event. We can expect much more innovation in neuromorphic cameras in the near future.

2.1 Related Work

Since the emergence of the event cameras, most vision work using them has focused on tracking moving targets from a fixed point of view, where almost all events are generated by



(a) (b) (c)

Figure 1: The first commercial event camera: (a) DVS128; (b) a stream of events (upward and downward spikes: positive and negative events); (c) image-like visualisation of accumulated events within a time interval (white and black: positive and negative events).

the dynamic object motion. For instance, a robot goalkeeper application blocks balls detected by a DVS camera [8], and a pencil balancing application maintains a pencil balanced on its tip by controlling an actuated table underneath using two DVS cameras [7]. This application requires very fast feedback control, successfully proving the remarkable high measurement rate and low latency capabilities of the event camera. More recently, Benosman *et al*. [4] proposed an optical flow estimation algorithm using an event camera which precisely estimates visual flow orientation and amplitude based on a local differential approach on the surface defined by events with microsecond accuracy and at very low computational cost.

Work on reconstructing, understanding and tracking of more general, previously unknown scenes where the event camera itself is moving is at an early stage. The first attempt to use this type of camera for a SLAM problem was made by Weikersdorfer *et al*. [19]. They used a wheeled robot equipped with an upward looking DVS camera to estimate 2D motion and construct a planar ceiling map. Most recently, Mueggler *et al*. [15] presented an onboard 6 DoF localisation quadrotor system using a DVS camera which is able to track high-speed maneuvers, such as flips. Their system starts by integrating events until a known pattern is detected, and then it tracks the borders of the pattern, by updating both the line segments and the pose of the flying robot on an event-by-event basis.

As a DVS camera does not provide absolute brightness values, a few attempts to combine an event camera with an extra full frame camera were made. Weikersdorfer *et al*. [20] developed an event-based 3D sensor combining a DVS with an RGB-D camera which generates a sparse stream of depth-augmented 3D points. In a similar way, Censi and Scaramuzza [6] presented a low-latency event-based visual odometry algorithm combining a DVS with a normal CMOS camera which uses events from the DVS to estimate the relative displacement since the previous frame from the conventional camera. Although these are both certainly possible practical ways to use an event camera for localisation and mapping, in our view this type of approach is sub-optimal; returning to the need for a frame-based camera alongside the event camera reduces many of the advantages of working purely with events.

After the submission of our work, we came across a similar idea to our scene reconstruction from an event stream [3], but in a much more constrained and hardware-dependent setup. They developed a special 360° high dynamic range camera consists of a pair of dynamic vision line sensors, a high-speed rotating mechanical device with encoders and a processing unit, and created a panoramic with greyscale values from event data.

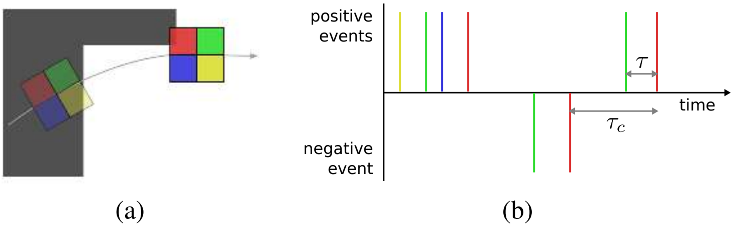


Figure 2: Event time intervals τ and τ_c: (a) A simplified 2×2 event-based camera moving over a scene, with colours to identify the pixels. (b) A stream of events generated by the camera. Upward and downward spikes represent positive and negative events, and their colours indicate the pixel each event came from. τ is the time elapsed since the previous event at any pixel, and τ_c is the time since the previous event at *the same* pixel.

3 Method

Our approach relies on two parallel probabilistic filters to jointly track the global rotational motion of a camera and estimate the gradients of the scene around it; the gradient map is then upgraded to a full image-like mosaic with super-resolution and high dynamic range properties. Each of these components essentially believes that the current estimate from the other is correct, following the approach of most recent successful data-rich SLAM systems such as PTAM [12] and DTAM [16], or in pure rotation mosaicing [14]. Note that we do not currently explicitly address bootstrapping in our method. We have found that simple alternation of the tracking and mapping components, starting from a blank template, will very often lead to rapid convergence of joint tracking and mapping, though there are sometimes currently gross failures and this is an important issue for future research.

We use the notation $e(u,v) = (u,v,p,t)^\top$ to denote an event with pixel location u and v, polarity p and timestamp t. The rotational mosaic or template we aim to reconstruct is denoted $\mathtt{M}(\mathbf{p}_m)$ and has its own fixed 2D coordinate frame with pixel position vector $\mathbf{p}_m$. We define two important time intervals τ and τ_c which are used in our algorithm. For clarity, let us consider a simplified 2×2 event camera moving over a simple scene as shown in Figure 2(a). We receive a stream of positive and negative spikes from the camera; those spike events are depicted along the time axis in Figure 2(b) and can be associated with a specific pixel by its colour. When a new event arrives from a certain pixel, we define τ as the time elapsed since the most recent previous event *from any pixel* and; and τ_c as the time since the most previous event *at the same pixel*. Here, τ is significant as the 'blind time' since any previous visual information was received and is used in the motion prediction component of our tracker; while τ_c is important since its inverse serves as a local measurement of the rate of events at a particular location in image space.

3.1 Event-based camera tracking

We have chosen a particle filter as a straightforward sequential Bayesian way to estimate the rotation motion of our camera over time with the multi-hypothesis capability to cope with the sometimes noisy event stream. In our event-based particle filter, the posterior density function at time t is represented by N particles, $\{\mathbf{p}_1^{(t)}, \mathbf{p}_2^{(t)}, \ldots, \mathbf{p}_N^{(t)}\}$. Each particle $\mathbf{p}_i^{(t)}$ is a set consisting of a hypothesis of the current state $\mathtt{R}_i^{(t)} \in \mathbf{SO}(3)$ and a normalised weight $w_i^{(t)}$.

Initially, all particles are set to the origin with the same weight.

3.1.1 Motion prediction

We first explain the tracking component of our algorithm, whose job is to provide an event by event updated estimate of the rotational location of the camera with respect to the scene mosaic. We use a constant position (random walk) motion model for our particle filter where the predicted mean rotation of a particle at any given time remains constant while the variance of the prediction expands linearly with time. We perturb the current $\mathfrak{so}(3)$ vector on the tangent plane with Gaussian noise independently in all three axes and reproject it onto the $\mathbf{SO}(3)$ unit sphere to obtain the corresponding predicted mean rotation. The noise is the predicted change the current rotation might have undergone since the previous event was generated. This is further simplified by the composition property of rotation matrices and yields the final update at the current time t as:

$$\mathtt{R}_i^{(t)} = \mathtt{R}_i^{(t-\tau)} \exp\left(\sum_{k=1}^{3} \mathbf{n}_k \mathtt{G}_k\right) , \tag{1}$$

where $\mathtt{R}_i$ is the rotation matrix for the i^{th} particle and $\mathtt{G}_k$ are the Lie group generators for $\mathbf{SO}(3)$. The noise vector $\mathbf{n} = (n_1, n_2, n_3)^\top$ is obtained by generating random numbers sampled from Gaussian distributions independently in all three directions i.e. $n_i \sim \mathcal{N}(0, \sigma_i^2 \tau)$.

Note that the high average frequency of events (at least 10kHz typically) relative to the dynamics of a hand-held camera strongly motivates the use of a stronger motion model (e.g. constant velocity or acceleration) [10], and we aim to test such a model soon.

3.1.2 Measurement update

The weights of these perturbed particles are now updated through the measurement update step which applies Bayes rule to each particle (the weights are subsequently normalised):

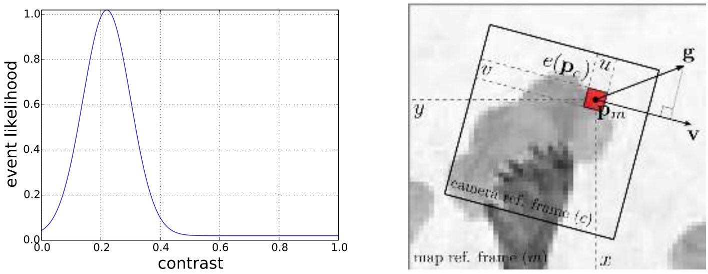
$$w_i^{(t)} = P(z|\mathtt{R}_i^{(t)}) w_i^{(t-\tau)} . \tag{2}$$

We calculate the value of a 'measurement' z given an event $e(u,v)$, the current state $\mathtt{R}_i^{(t)}$ and the previous state $\mathtt{R}_i^{(t-\tau_c)}$ by taking a log intensity difference between the corresponding intensity map positions:

$$z = \log(\mathtt{M}(\mathbf{p}_m^{(t)})) - \log(\mathtt{M}(\mathbf{p}_m^{(t-\tau_c)})) , \tag{3}$$

$$\text{where } \mathbf{p}_m^{(t)} = \pi\left(\mathtt{R}_i^{(t)} \mathtt{K}^{-1} \dot{\mathbf{p}}_c\right) . \tag{4}$$

Here $\dot{\mathbf{p}}_c = (u, v, 1)^\top$ is a camera pixel position in homogeneous coordinates, $\mathtt{K}$ is the camera intrinsics, and $\pi(\mathbf{p}) = \frac{1}{p_2}(p_0, p_1)^\top$ is the homogeneous projection function. The measurement z is now used to calculate the likelihood $P(z|\mathtt{R}_i^{(t)})$ for each particle, essentially asking 'how likely was this event relative to our mosaic given a particular hypothesis of camera pose?'. We first compare the sign of the log intensity difference with the polarity of the event, and we give a particle a fixed low likelihood if the signs do not agree. Otherwise, we look up a likelihood of this absolute log intensity difference (contrast) in the Mexican



(a) (b)

Figure 3: (a) Event likelihood function. (b) Camera and map reference frames.

hat shaped curve shown in Figure 3(a), with mean aligned to a known contrast which highly likely generates an event [13].

For the next measurement update step and the reconstruction block, a particle mean pose is saved for each pixel whenever an event occurs at a specific pixel location. To calculate the mean of particles, we apply the matrix logarithm to all particles' $\mathbf{SO}(3)$ components to map them to the tangent space, calculate the arithmetic mean, and re-map to the $\mathbf{SO}(3)$ group's manifold by applying the matrix exponential. Because of the random walk nature of our motion model which generates noisy motion estimates, a new mean pose is saved in a form of a weighted average with the previous mean pose.

Finally, after giving each particle a likelihood and normalising the distribution we resample the distribution in the standard way, making a new particle set which copies old particles with probability according to their weights. However, due to the very high frequency of events, we do not resample on every step to avoid unnecessarily deleting good particles in cases where all weights are similar. We follow [9] to determine whether the resampling should be carried out depending on the so-called effective number of particles N_{eff}:

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (w_i^{(t)})^2} \, . \tag{5}$$

We resample the set of particles whenever N_{eff} is less than $N/2$.

3.2 Mosaic reconstruction

We now turn to the other main part of our algorithm, which having received an updated camera pose estimate from tracking must incrementally improving our estimate of the intensity mosaic. This takes two steps; pixel-wise incremental Extended Kalman Filter (EKF) estimation of the log gradient at each template pixel, and interleaved Poisson reconstruction to recover absolute log intensity.

3.2.1 Pixel-wise EKF based gradient estimation

We receive an event at a pixel location $\mathbf{p}_c = (u, v)^\top$ in the camera reference frame and, using our tracking algorithm as described in Section 3.1, we find the corresponding location $\mathbf{p}_m = (x, y)^\top$ in the map reference frame as shown in Figure 3(b). Each pixel of the gradient map has an independent gradient estimate $\mathbf{g}^{(t)} = (g_x, g_y)^\top$ and 2×2 covariance matrix $\mathtt{P}_{\mathbf{g}}^{(t)}$. At initialisation, all estimated gradients are initialised to zero vectors with large covariances.

Now, we want to improve an estimate $\mathbf{g}^{(t)}$ based on a new incoming event and a tracking result using the pixel-wise EKF. We know τ_c at $\mathbf{p}_c$ and the velocity of the camera at a pixel $\mathbf{p}_m$ is calculated as below:

$$\mathbf{v}^{(t)} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \frac{\mathbf{p}_m^{(t)} - \mathbf{p}_m^{(t-\tau_c)}}{\tau_c} . \tag{6}$$

Assuming, based on the rapidity of events, that the gradient $\mathbf{g}$ in the template and the camera velocity $\mathbf{v}$ can be considered locally constant, we now say $(\mathbf{g} \cdot \mathbf{v})\tau_c$ is the amount of log grey level change that has happened since the last event. Therefore, if we have an event camera where log intensity change C should trigger an event, the brightness constancy tells us that:

$$\left(\mathbf{g}^{(t)} \cdot \mathbf{v}^{(t)}\right) \tau_c = \pm C , \tag{7}$$

where the sign of C depends on the polarity of an event. We now define z, a measurement of the instantaneous *event rate* at this pixel, and its measurement model h as below:

$$z^{(t)} = \frac{1}{\tau_c} , \tag{8}$$

$$h^{(t)} = \frac{\mathbf{g}^{(t)} \cdot \mathbf{v}^{(t)}}{C} . \tag{9}$$

In the EKF framework, the gradient estimate and the uncertainty covariance matrix are updated using the standard equations at every event:

$$\mathbf{g}^{(t)} = \mathbf{g}^{(t-\tau_c)} + \mathbf{W}\nu , \tag{10}$$

$$\mathtt{P}_{\mathbf{g}}^{(t)} = \mathtt{P}_{\mathbf{g}}^{(t-\tau_c)} - \mathbf{W}\mathtt{S}\mathbf{W}^\top , \tag{11}$$

where the Kalman gain $\mathbf{W}$ is:

$$\mathbf{W} = \mathtt{P}_{\mathbf{g}}^{(t-\tau_c)} \frac{\partial h}{\partial \mathbf{g}}^\top \mathtt{S}^{-1} , \tag{12}$$

the innovation ν is:

$$\nu = z^{(t)} - h^{(t)} , \tag{13}$$

and the innovation covariance $\mathtt{S}$ is:

$$\mathtt{S} = \frac{\partial h}{\partial \mathbf{g}} \mathtt{P}_{\mathbf{g}}^{(t-\tau_c)} \frac{\partial h}{\partial \mathbf{g}}^\top + \mathtt{R} , \tag{14}$$

where $\mathtt{R}$ is the measurement noise, in our case scalar σ_m^2. Finally, Jacobian $\frac{\partial h}{\partial \mathbf{g}}$ is derived as:

$$\begin{aligned} \frac{\partial h}{\partial \mathbf{g}} &= \frac{\partial}{\partial \mathbf{g}} \left(\frac{\mathbf{g} \cdot \mathbf{v}}{C}\right) \\ &= \begin{pmatrix} \frac{\partial}{\partial g_x}\left(\frac{g_x v_x + g_y v_y}{C}\right) & \frac{\partial}{\partial g_y}\left(\frac{g_x v_x + g_y v_y}{C}\right) \end{pmatrix} \\ &= \begin{pmatrix} \frac{v_x}{C} & \frac{v_y}{C} \end{pmatrix} . \end{aligned} \tag{15}$$

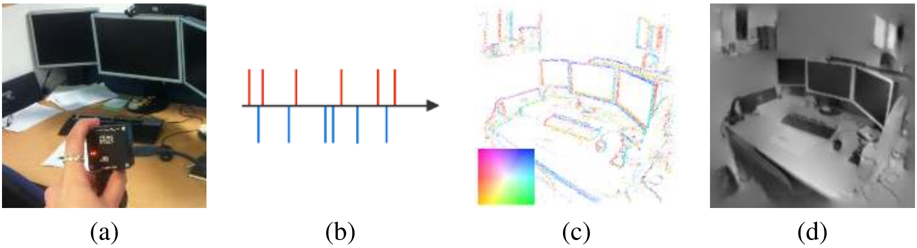


Figure 4: Proposed algorithm: (a) scene and DVS camera; (b) event stream; (c) estimated gradient map (the colours and intensities of this figure represent the orientations and strengths of the gradients of the scene respectively); (d) reconstructed intensity map.

Essentially, each new event which lines up with a particular template pixel improves our estimate of its gradient in the direction parallel to the motion of the camera over the scene at that pixel while we learn nothing about the gradient in the direction perpendicular to camera motion. We visualise an estimated gradient map in Figure 4(c); the colours and intensities of the figure represent the orientations and strengths of the gradients of the scene respectively.

3.2.2 Reconstruction from gradients

Inspired by [18], we reconstruct the log intensity of the image whose gradients M_x, M_y across the whole image domain are close to the estimated gradients g_x, g_y in a least squares sense:

$$J(\mathtt{M}) = \int\int (M_x - g_x)^2 + (M_y - g_y)^2 dxdy \ . \tag{16}$$

The Euler-Lagrange equation to minimise $J(\mathtt{M})$ is:

$$\frac{\partial J}{\partial \mathtt{M}} - \frac{\mathrm{d}}{\mathrm{dx}}\frac{\partial J}{\partial M_x} - \frac{\mathrm{d}}{\mathrm{dy}}\frac{\partial J}{\partial M_y} = 0 \ , \tag{17}$$

which leads to the well known Poisson equation:

$$\nabla^2 \mathtt{M} = \frac{\partial}{\partial x} g_x + \frac{\partial}{\partial y} g_y \ . \tag{18}$$

Here $\nabla^2 \mathtt{M} = \frac{\partial^2 \mathtt{M}}{\partial x^2} + \frac{\partial^2 \mathtt{M}}{\partial y^2}$ is the Laplacian. To solve Equation (18), we use a sine transform based method [1, 2]. We show a reconstructed intensity map in Figure 4(d).

4 Experiments

We recommend readers to view our submitted video[1] which illustrates all of the key results below in a form better than still pictures. We have conducted spherical mosaicing in both indoor and outdoor scenes. Also, we show the potential for reconstructing high resolution and high dynamic range scenes from very small camera motion. Our algorithm runs in real-time on a standard PC for low number of particles and template resolutions; the results we show here were generated at higher resolution and are currently off-line but we believe it is a simple matter of engineering to run at these settings in real-time in the near future.

[1] https://www.youtube.com/watch?v=l6qxeM1DbXU

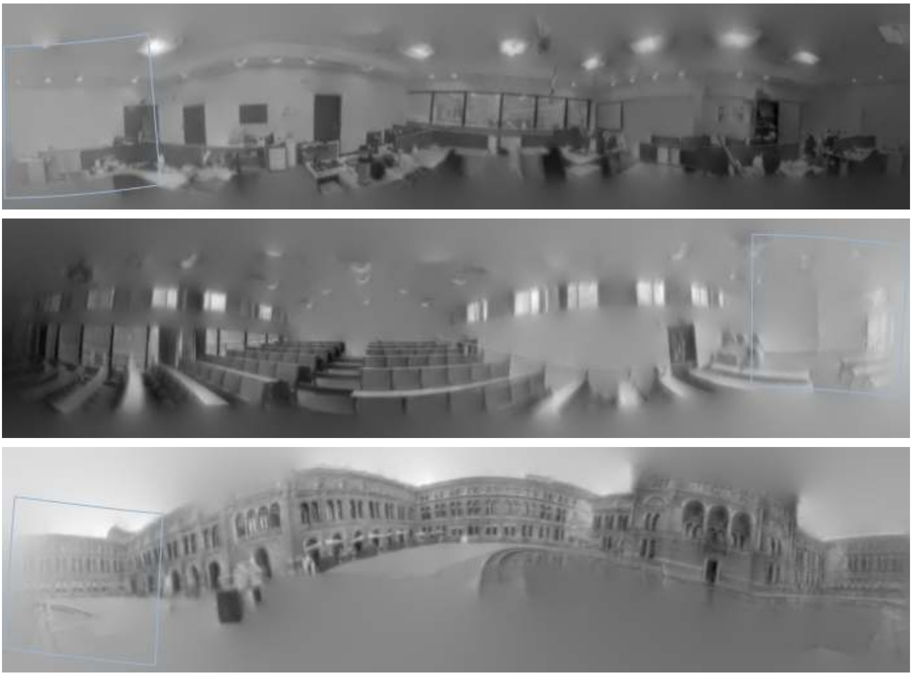


Figure 5: Spherical mosaicing for indoor and outdoor scenes. The overlaid boxes represent the field of view of the event camera.

4.1 Spherical mosaicing

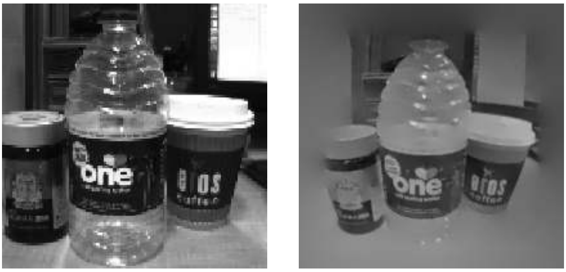
As shown in Figure 5, our algorithm is able to reconstruct indoor and outdoor scenes. In these mosaics, the overlaid box represents the tracked field of view of the event camera.

4.2 High resolution reconstruction

Even though the current event-based cameras have very low resolution (DVS has a 128×128 pixel array), as they provide very fast visual measurements, we can reconstruct high resolution scenes since our algorithm tracks rotation at sub-pixel accuracy. In Figure 6 we compare (a) an image from a standard camera down-sampled to 128×128 resolution with (b) our DVS reconstruction, showing sharper details.

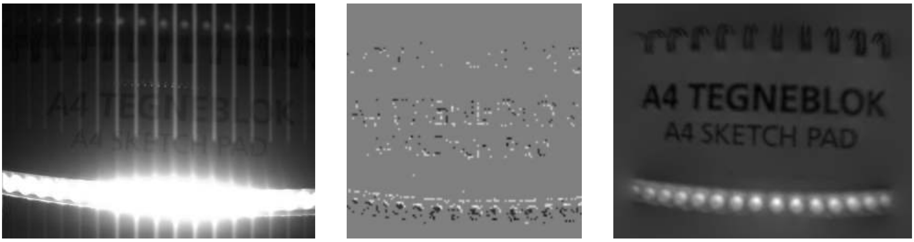
4.3 High dynamic range reconstruction

Another key characteristic of the event camera is its sensitivity over a very high dynamic range (*e.g.* 120dB for DVS). Our algorithm can build mosaics which make use of this range, to deal with scenes where there are large intensity difference between the brightest and darkest parts. We created a scene with a very high range of light intensity by placing a row of bright LED lights on top of a poorly lit sketch pad. A standard global shutter camera generates an image which is partly saturated, partly very dark and also has smearing effects (Figure 7(a)). However, the event camera and our algorithm are able to reconstruct the high dynamic range log intensity image in Figure 7(c) where all elements are clear.



(a) (b)

Figure 6: High resolution reconstruction: (a) a down sampled normal camera image for a comparison; (b) a reconstructed high resolution scene.



(a) (b) (c)

Figure 7: High dynamic range reconstruction: (a) a saturated normal CCD camera image with the smear effect for a comparison; (b) a visualisation of a stream of events from the DVS camera; (c) a reconstructed high dynamic range scene.

5 Conclusion

We believe these are breakthrough results, showing how joint sequential and global estimation permits the great benefits of an event camera to be applied to a real problem of mosaicing, and hopefully opening the door to similar approaches in dense 3D reconstruction in the style of [16] and many other vision problems. It is worth restating the comparison between the data rate of an event camera — typically on the order of 40–180kB/s in our experiments — and for instance a standard monochrome VGA video feed at 30Hz: 10MB/s. The only information that is important for tracking and reconstruction is how edges move, and the event camera gives us directly that information and nothing else, while removing the problems of blur, low dynamic range and limited resolution which standard cameras have.

Acknowledgments

Hanme Kim was supported by an EPSRC DTA studentship. We thank Jacek Zienkiewicz and other colleagues at Imperial College London for many useful discussions.

References

[1] A. Agrawal, R. Chellappa, and R. Raskar. An Algebraic Approach to Surface Reconstruction from Gradient Fields. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2005.

[2] A. Agrawal, R. Raskar, and R. Chellappa. What is the Range of Surface Reconstructions from a Gradient Field. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[3] A. N. Belbachir, S. Schraml, M. Mayerhofer, and M. Hofstätter. A Novel HDR Depth Camera for Real-time 3D 360° Panoramic Vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014.

[4] 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.

[5] K Boahen. Neuromorphic Chips. *Scientific American*, 2005.

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

[7] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R.J. 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.

[8] 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.

[9] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[10] P. Gemeiner, A. J. Davison, and M. Vincze. Improving Localization Robustness in Monocular SLAM Using a High-Speed Camera. In *Proceedings of Robotics: Science and Systems (RSS)*, 2008.

[11] A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison. Real-Time Camera Tracking: When is High Frame-Rate Best? In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[12] G. Klein and D. W. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

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

[14] S. J. Lovegrove and A. J. Davison. Real-Time Spherical Mosaicing using Whole Image Alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[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] 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.

[18] J. Tumblin, A. Agrawal, and R. Raskar. Why I want a Gradient Camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[19] D. Weikersdorfer, R. Hoffmann, and J. Conradt. Simultaneous Localization and Mapping for event-based Vision Systems. In *International Conference on Computer Vision Systems (ICVS)*, 2013.

[20] D. Weikersdorfer, D. B. Adrian, D. Cremers, and J. Conradt. Event-based 3D SLAM with a depth-augmented dynamic vision sensor. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.