Real-Time Intensity-Image Reconstruction for Event Cameras Using Manifold Regularisation

Christian Reinbacher[1]
reinbacher@icg.tugraz.at
Gottfried Graber[1]
graber@icg.tugraz.at
Thomas Pock[1,2]
pock@icg.tugraz.at

[1] Graz University of Technology
Institute for Computer Graphics and Vision
[2] Austrian Institute Of Technology
Vienna

Abstract

Event cameras or neuromorphic cameras mimic the human perception system as they measure the per-pixel *intensity change* rather than the actual *intensity level*. In contrast to traditional cameras, such cameras capture new information about the scene at MHz frequency in the form of sparse events. The high temporal resolution comes at the cost of losing the familiar per-pixel intensity information. In this work we propose a variational model that accurately models the behaviour of event cameras, enabling reconstruction of intensity images with arbitrary frame rate in real-time. Our method is formulated on a per-event-basis, where we explicitly incorporate information about the asynchronous nature of events via an *event manifold* induced by the relative timestamps of events. In our experiments we verify that solving the variational model on the manifold produces high-quality images without explicitly estimating optical flow.

1 Introduction

In contrast to standard CMOS digital cameras that operate on frame basis, neuromorphic cameras such as the Dynamic Vision Sensor (DVS) [17] work asynchronously on a pixel level. Each pixel measures the incoming light intensity and fires an *event* when the absolute change in intensity is above a certain threshold (which is why those cameras are also often referred to as *event cameras*). The time resolution is in the order of μs. Due to the sparse nature of the events, the amount of data that has to be transferred from the camera to the computer is very low, making it an energy efficient alternative to standard CMOS cameras for the tracking of very quick movement [8, 27]. While it is appealing that the megabytes per second of data produced by a digital camera can be compressed to an asynchronous stream of events, these events can not be used directly in computer vision algorithms that operate on a frame basis. In recent years, the first algorithms have been proposed that transform the problem of camera pose estimation to this new domain of time-continuous events *e.g.* [3, 9, 12, 20, 21, 26], unleashing the full potential of the high temporal resolution and low latency

© 2016. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

Pages 9.1-9.12
DOI: https://dx.doi.org/10.5244/C.30.9

(a) Raw Events (b) Reconstructed Image (c) Event Manifold

Figure 1: Sample results from our method. The image (a) shows the raw events and (b) is the result of our reconstruction. The time since the last event has happened for each pixel is depicted as a surface in (c) with the positive and negative events shown in green and red respectively.

of event cameras. The main drawback of the proposed methods are specific assumptions on the properties of the scene or the type of camera movement.

Contribution In this work we aim to bridge the gap between the time-continuous domain of events and frame-based computer vision algorithms. We propose a simple method for intensity reconstruction for neuromorphic cameras (see Fig. 1 for a sample output of our method). In contrast to very recent work on the same topic by Bardow *et al*. [1], we formulate our algorithm on an event-basis, avoiding the need to simultaneously estimate the optical flow. We cast the intensity reconstruction problem as an energy minimisation, where we model the camera noise in a data term based on the *generalised Kullback-Leibler divergence*. The optimisation problem is defined on a manifold induced by the timestamps of new events (see Fig. 1(c)). We show how to optimise this energy using variational methods and achieve real-time performance by implementing the energy minimisation on a graphics processing unit (GPU). We release software to provide live intensity image reconstruction to all users of DVS cameras[1]. We believe this will be a vital step towards a wider adoption of this kind of cameras.

2 Related Work

Neuromorphic or event-based cameras receive increasing interest from the computer vision community. The low latency compared to traditional cameras make them particularly interesting for tracking rapid camera movement. Also more classical low-level computer vision problems are transferred to this new domain like optical flow estimation, or image reconstruction as proposed in this work. In this literature overview we focus on very recent work that aims to solve computer vision tasks using this new camera paradigm. We begin our survey with a problem that benefits the most from the temporal resolution of event cameras: camera pose tracking. Typical simultaneous localisation and mapping (SLAM) methods need to perform image feature matching to build a map of the environment and localise the camera within [11]. Having no image to extract features from means, that the vast majority of visual SLAM algorithms can not be readily applied to event-based data. Milford *et al*. [19]

[1] https://github.com/VLOGroup/dvs-reconstruction

show that it is possible to extract features from images that have been created by accumulating events over time slices of 1000 ms to perform large-scale mapping and localisation with loop-closure. While this is the first system to utilise event cameras for this challenging task, it trades temporal resolution for the creation of images like Fig. 1(a) to reliably track camera movement.

A different line of research tries to formulate camera pose updates on an event basis. Cook *et al*. [7] propose a biologically inspired network that simultaneously estimates camera rotation, image gradients and intensity information. An indoor application of a robot navigating in 2D using an event camera that observes the ceiling has been proposed by Weikersdorfer *et al*. [26]. They simultaneously estimate a 2D map of events and track the 2D position and orientation of the robot. Similarly, Kim *et al*. [12] propose a method to simultaneously estimate the camera rotation around a fixed point and a high-quality intensity image only from the event stream. A particle filter is used to integrate the events and allow a reconstruction of the image gradients, which can then be used to reconstruct an intensity image by Poisson editing. All methods are limited to 3 DOF of camera movement. A full camera tracking has been shown in [20, 21] for rapid movement of an UAV with respect to a known 2D target and in [9] for a known 3D map of the environment.

Benosman *et al*. [3] tackle the problem of estimating optical flow from an event stream. This work inspired our use of an event manifold to formulate the intensity image reconstruction problem. They recover a motion field by clustering events that are spatially and temporally close. The motion field is found by locally fitting planes into the event manifold. In experiments they show that flow estimation works especially well for low-textured scenes with sharp edges, but still has problems for more natural looking scenes. Very recently, the first methods for estimating intensity information from event cameras without the need to recover the camera movement have been proposed. Barua *et al*. [2] use a dictionary learning approach to map the sparse, accumulated event information to infer image gradients. Those are then used in a Poisson reconstruction to recover the log-intensities. Bardow *et al*. [1] proposed a method to simultaneously recover an intensity image and dense optical flow from the event stream of a neuromorphic camera. The method does not require to estimate the camera movement and scene characteristics to reconstruct intensity images. In a variational energy minimisation framework, they concurrently recover optical flow and image intensities within a time window. They show that optical flow is necessary to recover sharp image edges especially for fast movements in the image. In contrast, in this work we show that intensities can also be recovered without explicitly estimating the optical flow. This leads to a substantial reduction of complexity: In our current implementation, we are able to reconstruct > 500 frames per second. While the method is defined on a per-event-basis, we can process blocks of events without loss in image quality. We are therefore able to provide a true live-preview to users of a neuromorphic camera.

3 Image Reconstruction from Sparse Events

We have given a time sequence of events $(e^n)_{n=1}^N$ from a neuromorphic camera, where $e^n = \{x^n, y^n, \theta^n, t^n\}$ is a single event consisting of the pixel coordinates $(x^n, y^n) \in \Omega \subset \mathbb{R}^2$, the polarity $\theta^n \in \{-1, 1\}$ and a monotonically increasing timestamp t^n.

A positive θ^n indicates that at the corresponding pixel the intensity has increased by a certain threshold $\Delta^+ > 0$ in the log-intensity space. Vice versa, a negative θ^n indicates a drop in intensity by a second threshold $\Delta^- > 0$. Our aim is now to reconstruct an intensity

image $u^n : \Omega \to \mathbb{R}_+$ by integrating the intensity changes indicated by the events over time.

Taking the $\exp(\cdot)$, the update in intensity space caused by one event e^n can be written as

$$f^n(x^n, y^n) = u^{n-1}(x^n, y^n) \cdot \begin{cases} c_1 & \text{if } \theta^n > 0 \\ c_2 & \text{if } \theta^n < 0 \end{cases}, \tag{1}$$

where $c_1 = \exp(\Delta^+)$, $c_2 = \exp(-\Delta^-)$. Starting from a known u^0 and assuming no noise, this integration procedure will reconstruct a perfect image (up to the radiometric discretisation caused by $\Delta^\pm$). However, since the events stem from real camera hardware, there is noise in the events. Also the initial intensity image u^0 is unknown and can not be reconstructed from events alone. Therefore the reconstruction of u^n from f^n can not be solved without imposing some regularity in the solution. We therefore formulate the intensity image reconstruction problem as the solution of the optimisation problem

$$u^n = \underset{u \in C^1(\Omega, \mathbb{R}_+)}{\operatorname{argmin}} \; [E(u) = D(u, f^n) + R(u)] \;, \tag{2}$$

where $D(u, f^n)$ is a *data term* that models the camera noise and $R(u)$ is a *regularisation term* that enforces some smoothness in the solution. In the following section we will show how we can utilise the timestamps of the events to define a manifold which guides a variational model and detail our specific choices for data term and regularisation.

3.1 Variational Model on the Event Manifold

Moving edges in the image cause events once a change in logarithmic intensity is bigger than a threshold. The collection of all events $(e^n)_{n=1}^N$ can be recorded in a spatiotemporal volume $V \subset \Omega \times T$. V is very sparsely populated, which makes it infeasible to directly store it. To alleviate this problem, Bardow et al. [1] operate on events in a fixed time window that is sliding along the time axis of V. They simultaneously optimise for optical flow and intensities, which are tightly coupled in this volumetric representation.

Regularisation Term As in [3], we observe that events lie on a lower-dimensional manifold within V, defined by the most recent timestamp for each pixel $(x, y) \in \Omega$. A visualisation of this manifold for a real-world scene can be seen in Fig. 1(c). Benosman *et al.* [3] fittingly call this manifold the *surface of active events*. We propose to incorporate the surface of active events into our method by formulating the optimisation *directly on the manifold*. Our intuition is, that parts of the scene that have no or little texture will not produce as many events as highly textured areas. Regularising an image reconstructed from the events should take into account the different "time history" of pixels. In particular, we would like to have strong regularisation across pixels that stem from events at approximately the same time, whereas regularisation between pixels whose events have very different timestamps should be reduced. This corresponds to a grouping of pixels in the time domain, based on the timestamps of the recorded events. Solving computer vision problems on a surface is also known as *intrinsic image processing* [14], as it involves the intrinsic (*i.e.* coordinate-free) geometry of the surface, a topic studied by the field of differential geometry. Looking at the body of literature on intrinsic image processing on surfaces, we can divide previous work into two approaches based on the representation of the surface. Implicit approaches [6, 13] use an implicit surface (e.g. through the zero level set of a function), whereas explicit approaches

[18, 24] construct a triangular mesh representation. Our method uses the same underlying theory of differential geometry, however we note that because the surface of active events is defined by the timestamps which are monotonically increasing, the class of surfaces is effectively restricted to $2\frac{1}{2}$D. This means that there exists a simple parameterisation of the surface and we can perform all computations in a local euclidean coordinate frame (*i.e.* the image domain Ω). In contrast to [14], where the authors deal with arbitrary surfaces, we avoid the need to explicitly construct a representation of the surface. This has the advantage that we can straightforwardly make use of GPU-accelerated algorithms to solve the large-scale optimisation problem. A similar approach was proposed recently in the context of variational stereo [10].

We start by defining the surface $S \subset \mathbb{R}^3$ as the graph of a scalar function $t(x,y)$ through the mapping $\varphi : \Omega \to S$

$$X = \varphi(x,y) = \begin{bmatrix} x, & y, & t(x,y) \end{bmatrix}^T, \tag{3}$$

where $X \in S$ denotes a 3D-point on the surface. $t(x,y)$ is simply an image that records for each pixel (x,y) the time since the last event. The partial derivatives of the parameterisation φ define a basis for the tangent space $T_X\mathcal{M}$ at each point X of the manifold $\mathcal{M}$, and the dot product in this tangent space gives the *metric* of the manifold. In particular, the *metric tensor* is defined as the symmetric 2×2 matrix

$$g = \begin{bmatrix} \langle \varphi_x, \varphi_x \rangle & \langle \varphi_x, \varphi_y \rangle \\ \langle \varphi_x, \varphi_y \rangle & \langle \varphi_y, \varphi_y \rangle \end{bmatrix}, \tag{4}$$

where subscripts denote partial derivatives and $\langle \cdot, \cdot \rangle$ denotes the scalar product. Starting from the definition of the parameterisation Eqn. (3), straightforward calculation gives $\varphi_x = \begin{bmatrix} 1 & 0 & t_x \end{bmatrix}^T$, $\varphi_y = \begin{bmatrix} 0 & 1 & t_y \end{bmatrix}^T$ and

$$g = \begin{bmatrix} 1+t_x^2 & t_x t_y \\ t_x t_y & 1+t_y^2 \end{bmatrix} \tag{5a}$$

$$g^{-1} = \frac{1}{G} \begin{bmatrix} 1+t_y^2 & -t_x t_y \\ -t_x t_y & 1+t_x^2 \end{bmatrix}, \tag{5b}$$

where $G = \det(g)$.

Given a smooth function $\tilde{f} \in C^1(S, \mathbb{R})$ on the manifold, the gradient of $\tilde{f}$ is characterised by $\mathrm{d}\tilde{f}(Y) = \langle \nabla_g \tilde{f}, Y \rangle_g \quad \forall Y \in T_X\mathcal{M}$ [16]. We will use the notation $\nabla_g \tilde{f}$ to emphasise the fact that we take the gradient of a function defined on the surface (i.e. under the metric of the manifold). $\nabla_g \tilde{f}$ can be expressed in local coordinates as

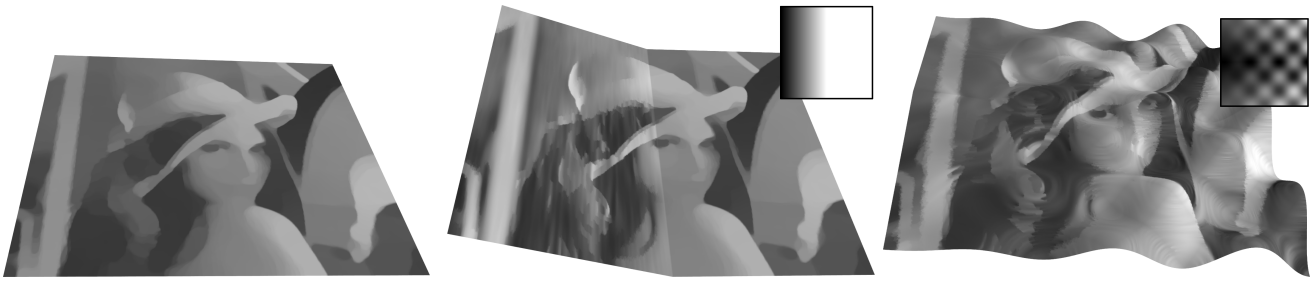
$$\nabla_g \tilde{f} = \left(g^{11} \tilde{f}_x + g^{12} \tilde{f}_y\right) \varphi_x + \left(g^{21} \tilde{f}_x + g^{22} \tilde{f}_y\right) \varphi_y, \tag{6}$$

where g^{ij}, $i,j = 1,2$ denotes the components of the inverse of g (the so-called pull-back). Inserting g^{-1} into Eqn. (6) gives an expression for the gradient of a function $\tilde{f}$ on the manifold in local coordinates

$$\nabla_g \tilde{f} = \frac{1}{G} \left[\left(\left(1+t_y^2\right) \tilde{f}_x - t_x t_y \tilde{f}_y \right) \begin{bmatrix} 1 & 0 & t_x \end{bmatrix}^T + \left(\left(1+t_x^2\right) \tilde{f}_y - t_x t_y \tilde{f}_x \right) \begin{bmatrix} 0 & 1 & t_y \end{bmatrix}^T \right]. \tag{7}$$

Equipped with these definitions, we are ready to define our regularisation term. It will be a variant of the total variation (TV) norm insofar that we take the norm of the gradient of $\tilde{f}$ on the manifold

$$TV_g(\tilde{f}) = \int_S |\nabla_g \tilde{f}| \, \mathrm{d}s. \tag{8}$$



(a) Flat surface (b) Ramp surface (c) Sine surface

Figure 2: ROF denoising on different manifolds. A flat surface (a) gives the same result as standard ROF denoising, but more complicated surfaces (b)(c) significantly change the result. The graph function $t(x,y)$ is depicted in the upper right corner. We can see that a ramp surface (b) produces regularisation anisotropy due to the fact that the surface gradient is zero in y-direction but non-zero in x-direction. The same is true for the sine surface (c), where we can see strong regularisation along level sets of the surface and less regularisation across level sets.

It is easy to see that if we have $t(x,y) = const$, then g is the 2×2 identity matrix and $TV_g(\tilde{f})$ reduces to the standard TV. Also note that in the definition of the TV_g we integrate over the surface. Since our goal is to formulate everything in local coordinates, we relate integration over S and integration over Ω using the pull-back

$$\int_S |\nabla_g \tilde{f}| \, \mathrm{d}s = \int_\Omega |\nabla_g \tilde{f}| \sqrt{G} \, \mathrm{d}x\mathrm{d}y, \tag{9}$$

where $\sqrt{G}$ is the differential area element that links distortion of the surface element $\mathrm{d}s$ to local coordinates $\mathrm{d}x\mathrm{d}y$. In the same spirit, we can pull back the data term defined on the manifold to the local coordinate domain Ω. In contrast to the method of Graber *et al.* [10] which uses the differential area element as regularization term, we formulate the full variational model on the manifold, thus incorporating spatial as well as temporal information.

To assess the effect of TV_g as a regularisation term, we depict in Fig. 2 results of the following variant of the ROF denoising model [23]

$$\min_u \int_\Omega |\nabla_g u| \sqrt{G} + \tfrac{\lambda}{2} |u - f|^2 \sqrt{G} \mathrm{d}x\mathrm{d}y, \tag{10}$$

with different $t(x,y)$, *i.e.* ROF-denoising on different manifolds. We see that computing the TV norm on the manifold can be interpreted as introducing anisotropy based on the surface geometry (see Fig. 2(b),2(c)). We will use this to guide regularisation of the reconstructed image according to the surface defined by the event time.

Data Term The data term $D(u, f^n)$ encodes the deviation of u from the noisy measurement f^n Eqn. (1). Under the reasonable assumption that a neuromorphic camera sensor suffers from the same noise as a conventional sensor, the measured update caused by one event will contain noise. In computer vision, a widespread approach is to model image noise as zero-mean additive Gaussian. While this simple model is sufficient for many applications, real sensor noise is dependent on scene brightness and should be modelled as a Poisson distribution [22]. We therefore define our data term as

$$D(u, f^n) := \lambda \int_S (u - f^n \log u) \, \mathrm{d}s = \lambda \int_\Omega (u - f^n \log u) \sqrt{G} \, \mathrm{d}x\mathrm{d}y \quad \text{s.t. } u(x,y) \in [u_{\min}, u_{\max}] \tag{11}$$

whose minimiser is known to be the correct ML-estimate under the assumption of Poisson-distributed noise between u and f^n [15]. Note that, in contrast to [10], we also define the data term to lie on the manifold. Eqn. (11) is also known as *generalised Kullback-Leibler divergence* and has been investigated by Steidl and Teuber [25] in variational image restoration methods. Furthermore, the data term is convex, which makes it easy to incorporate into our variational energy minimisation framework. We restrict the range of $u(x,y) \in [u_{\min}, u_{\max}]$ since our reconstruction problem is defined up to a gray value offset caused by the unknown initial image intensities.

Discrete Energy In the discrete setting, we represent images of size $M \times M$ as matrices in $\mathbb{R}^{M\times M}$ with indices $(i,j) = 1\ldots M$. Derivatives are represented as linear maps $L_x, L_y : \mathbb{R}^{M\times M} \to \mathbb{R}^{M\times M}$, which are simple first order finite difference approximations of the derivative in x- and y-direction [4]. The discrete version of ∇_g, defined in Eqn. (7), can then be represented as a linear map $L_g : \mathbb{R}^{M\times M} \to \mathbb{R}^{M\times M\times 3}$ that acts on u as follows

$$
\begin{aligned}
(L_g u)_{ij1} &= \tfrac{1}{G_{ij}}\left((1+(L_y t)_{ij}^2)(L_x u)_{ij} - (L_x t)_{ij}(L_y t)_{ij}(L_y u)_{ij}\right)\\
(L_g u)_{ij2} &= \tfrac{1}{G_{ij}}\left((1+(L_x t)_{ij}^2)(L_y u)_{ij} - (L_x t)_{ij}(L_y t)_{ij}(L_x u)_{ij}\right)\\
(L_g u)_{ij3} &= \tfrac{1}{G_{ij}}\left((L_x t)_{ij}(L_x u)_{ij} + (L_y t)_{ij}(L_y u)_{ij}\right)
\end{aligned}
$$

Here, $G \in \mathbb{R}^{M\times M}$ is the pixel-wise determinant of g given by $G_{ij} = 1 + (L_x t)_{ij}^2 + (L_y t)_{ij}^2$. The discrete data term follows from Eqn. (11) as $D(u, f^n) := \lambda \sum_{i,j}(u_{ij} - f_{ij}^n \log u_{ij})\sqrt{G_{ij}}$. This yields the complete discrete energy

$$
\min_u \|L_g u\|_g + \lambda \sum_{i,j}\left(u_{ij} - f_{ij}^n \log u_{ij}\right)\sqrt{G_{ij}} \quad \text{s.t. } u_{ij} \in [u_{\min}, u_{\max}], \tag{13}
$$

with the g-tensor norm defined as $\|A\|_g = \sum_{i,j}\sqrt{G_{ij}\sum_l (A_{ijl})^2} \quad \forall A \in \mathbb{R}^{M\times M\times 3}$.

3.2 Minimising the Energy

We minimise (13) using the Primal-Dual algorithm [5]. Dualising the g-tensor norm yields the primal-dual formulation

$$
\min_u \max_p \left[D(u, f^n) + \langle L_g u, p\rangle - R^*(p)\right], \tag{14}
$$

where $u \in \mathbb{R}^{M\times M}$ is the discrete image, $p \in \mathbb{R}^{M\times M\times 3}$ is the dual variable and R^* denotes the convex conjugate of the g-tensor norm. A solution of Eqn. (14) is obtained by iterating

$$
\begin{aligned}
u_{k+1} &= (I + \tau\partial D)^{-1}(u_k - \tau L_g^* p_k)\\
p_{k+1} &= (I + \sigma\partial R^*)^{-1}(p_k + \sigma L_g(2u_{k+1} - u_k)),
\end{aligned}
$$

where L_g^* denotes the adjoint operator of L_g. The proximal maps for the data term and the regularisation term can be solved in closed form, leading to the following update rules

$$
\begin{aligned}
\hat{u} &= \operatorname{prox}_{\tau D}(\bar{u}) &&\Leftrightarrow \hat{u}_{ij} = \operatorname*{clamp}_{u_{\min},u_{\max}}\left(\tfrac{1}{2}\left(\bar{u}_{ij} - \beta_{ij} + \sqrt{(\bar{u}_{ij} - \beta_{ij})^2 + 4\beta_{ij} f_{ij}^n}\right)\right)\\
\hat{p} &= \operatorname{prox}_{\sigma R^*}(\bar{p}) &&\Leftrightarrow \hat{p}_{ijl} = \frac{\bar{p}_{ijl}}{\max\{1, \|\bar{p}_{ij,\cdot}\|/\sqrt{G_{ij}}\}},
\end{aligned}
$$

with $\beta_{ij} = \tau\lambda\sqrt{G_{ij}}$. The time-steps τ, σ are set according to $\tau\sigma \leq 1/\|L_g\|^2$, where we estimate the operator norm as $\|L_g\|^2 \leq 8 + 4\sqrt{2}$. Since the updates are pixel-wise independent, the algorithm can be efficiently parallelised on GPUs. Moreover, due to the low number of events added in each step, the algorithm usually converges in $k \leq 50$ iterations.

4 Experiments

We perform our experiments using a DVS128 camera with a spatial resolution of 128×128 and a temporal resolution of $1\,\mu s$. The parameter λ is kept fixed for all experiments. The thresholds Δ^+, Δ^- are set according to the chosen camera settings. In practice, the timestamps of the recorded events can not be used directly as the manifold defined in Section 3.1 due to noise. We therefore denoise the timestamps with a few iterations of a TV-L1 denoising method. We compare our method to the recently proposed method of [1] on sequences provided by the authors. Furthermore, we will show the influence of the proposed regularisation on the event manifold using a few self-recorded sequences.

4.1 Timing

In this work we aim for a real-time reconstruction method. We implemented the proposed method in C++ and used a Linux computer with a 3.4 GHz processor and a NVidia Titan X GPU[2]. Using this setup we measure a wall clock time of **1.7 ms** to create one single image, which amounts to ≈ 580 fps. While we can create a new image for each new event, this would create a tremendous amount of images due to the number of events (≈ 500.000 per second on natural scenes with moderate camera movement). Furthermore one is limited by the monitor refresh rate of 60 Hz to actually display the images. In order to achieve real-time performance, one has two parameters: the number of events that are integrated into one image and the number of frames skipped for display on screen. The results in the following sections have been achieved by accumulating 500 events to produce one image, which amounts to a time resolution of 3-5 ms.

4.2 Influence of the Event Manifold

We have captured a few sequences around our office with a DVS128 camera. In Fig. 3 we show a few reconstructed images as well as the raw input events and the time manifold. For comparison, we switched off the manifold regularisation (by setting $t(x,y) = const$), which results in images with notably less contrast.

4.3 Comparison to Related Methods

In this section we compare our reconstruction method to the method proposed by Bardow *et al.* [1]. The authors kindly provided us with the recorded raw events, as well as intensity image reconstructions at regular timestamps $\delta t = 15$ms. Since we process shorter event packets, we search for the nearest neighbour timestamp for each image of [1] in our sequences. We visually compare our method on the sequences *face*, *jumping jack* and *ball* to

[2] We note that the small image size of 128×128 is not enough to fully load the GPU such that we measured almost the same wall clock time on a NVidia 780 GTX Ti.

Figure 3: Sample results from our method. The columns depict raw events, time manifold, result without manifold regularisation and finally with our manifold regularisation. Notice the increased contrast in weakly textured regions (especially around the edge of the monitor).

the results of [1]. We point out that no ground truth data is available so we are limited to purely qualitative comparisons.

In Fig. 4 we show a few images from the sequences. Since we are dealing with highly dynamic data, we point the reader to the included supplementary video[3] which shows whole sequences of several hundred frames.

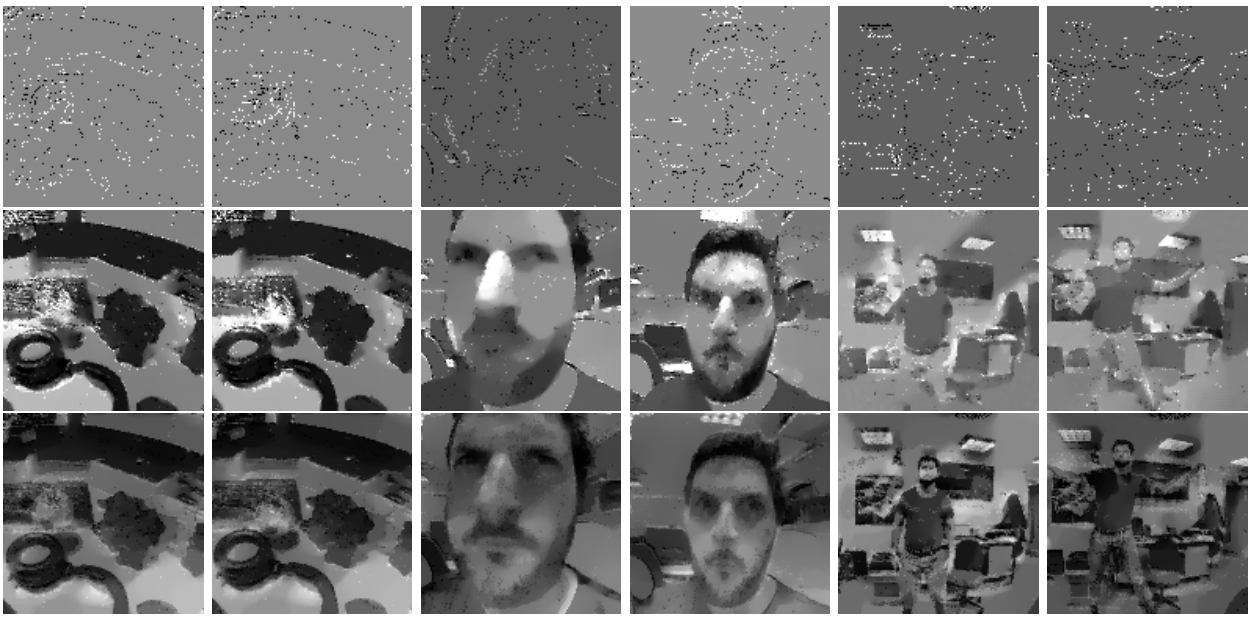


Figure 4: Comparison to the method of [1]. The first row shows the raw input events that have been used for both methods. The second row depicts the results of Bardow *et al.*, and the last row shows our result. We can see that out method produces more details (e.g. face, beard) as well as more graceful gray value variations in untextured areas, where [1] tends to produce a single gray value.

4.4 Comparison to Standard Cameras

We have captured a sequence using a DVS128 camera as well as a Canon EOS60D DSLR camera to compare the fundamental differences of traditional cameras and event-based cameras. As already pointed out by [1], rapid movement results in motion blur for conventional

[3] https://www.youtube.com/watch?v=rvB2URrGT94

cameras, while event-based cameras show no such effects. Also the dynamic range of a DVS is much higher, which is also shown in Fig. 5.

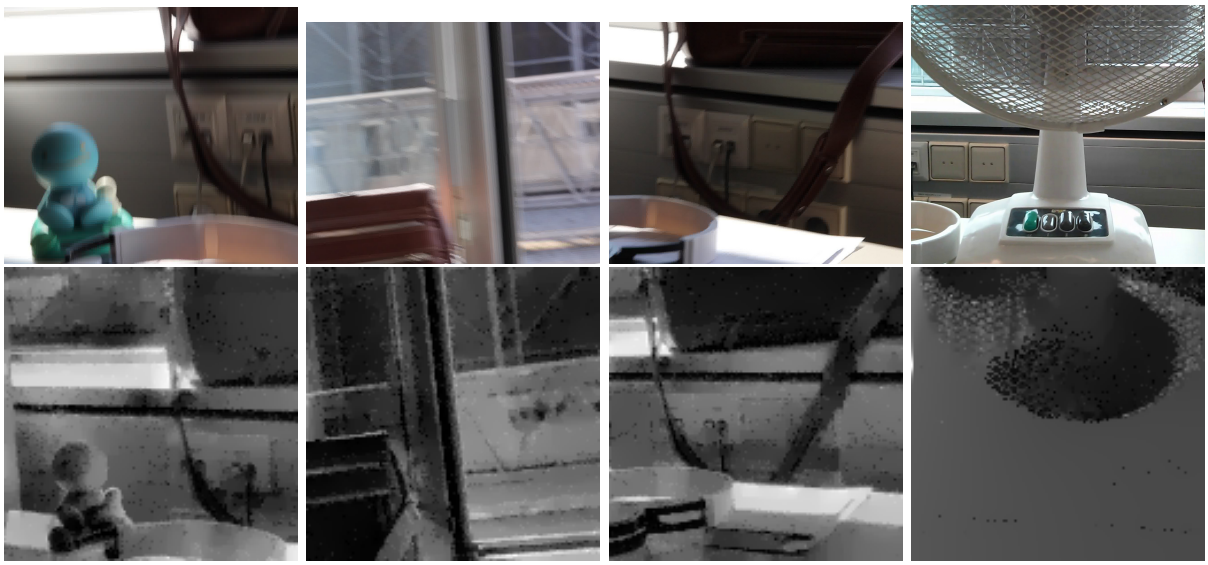


Figure 5: Comparison to a video captured with a modern DSLR camera. Notice the rather strong motion blur in the images of the DSLR (top row), whereas the DVS camera can easily deal with fast camera or object movement (bottom row).

5 Conclusion

In this paper we have proposed a method to recover intensity images from neuromorphic or event cameras in real-time. We cast this problem as an iterative filtering of incoming events in a variational denoising framework. We propose to utilise a manifold that is induced by the timestamps of the events to guide the image restoration process. This allows us to incorporate information about the relative ordering of incoming pixel information without explicitly estimating optical flow like in previous works. This in turn enables an efficient algorithm that can run in real-time on currently available PCs.

Future work will include the study of the proper noise characteristic of event cameras. While the current model produces natural-looking intensity images, a few noisy pixels appear that indicate a still non-optimal treatment of sensor noise within our framework. Also it might be beneficial to look into a local minimisation of the energy on the manifold (*e.g.* by coordinate-descent) to further increase the processing speed.

Acknowledgements

This work was supported by the research initiative Mobile Vision with funding from the AIT and the Austrian Federal Ministry of Science, Research and Economy HRSM programme (BGBl. II Nr. 292/2012).

References

[1] Patrick Bardow, Andrew Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *CVPR*, 2016.

[2] S. Barua, Y. Miyatani, and A. Veeraraghavan. Direct face detection and video reconstruction from event cameras. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, March 2016. doi: 10.1109/WACV.2016.7477561.

[3] R. Benosman, C. Clercq, X. Lagorce, S. H. Ieng, and C. Bartolozzi. Event-based visual flow. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):407–417, 2014.

[4] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

[5] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1), 2011.

[6] Li-Tien Cheng, Paul Burchard, Barry Merriman, and Stanley Osher. Motion of curves constrained on surfaces using a level set approach. *J. Comput. Phys*, 175:2002, 2000.

[7] M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger. Interacting maps for fast visual interpretation. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 770–776, July 2011. doi: 10.1109/IJCNN.2011.6033299.

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

[9] Guillermo Gallego, Christian Forster, Elias Mueggler, and Davide Scaramuzza. Event-based camera pose tracking using a generative event model. *CoRR*, abs/1510.01972, 2015.

[10] Gottfried Graber, Jonathan Balzer, Stefano Soatto, and Thomas Pock. Efficient minimal-surface regularization of perspective depth maps in variational stereo. In *CVPR*, 2015.

[11] J. Hartmann, J. H. Klüssendorff, and E. Maehle. A comparison of feature descriptors for visual slam. In *European Conference on Mobile Robots*, 2013.

[12] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew Davison. Simultaneous mosaicing and tracking with an event camera. In *BMVC*, 2014.

[13] Matthias Krueger, Patrice Delmas, and Georgy L. Gimel'farb. Active contour based segmentation of 3d surfaces. In *ECCV*, 2008.

[14] Rongjie Lai and Tony F. Chan. A framework for intrinsic image processing on surfaces. *Computer Vision and Image Understanding*, 115(12):1647 – 1661, 2011. Special issue on Optimization for Vision, Graphics and Medical Imaging: Theory and Applications.

[15] Triet Le, Rick Chartrand, and Thomas J. Asaki. A variational approach to reconstructing images corrupted by poisson noise. *J. Math. Imaging Vision*, 27:257–263, 2007.

[16] John Marshall Lee. *Riemannian manifolds: an introduction to curvature*. Graduate Texts in Mathematics. Springer, New York, 1997. ISBN 0-387-98322-8.

[17] 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*, 43(2): 566–576, 2008.

[18] Lok Ming Lui, Xianfeng Gu, Tony F. Chan, and Shing-Tung Yau. Variational method on riemann surfaces using conformal parameterization and its applications to image processing. *Methods Appl. Anal.*, 15(4):513–538, 12 2008.

[19] Michael Milford, Hanme Kim, Stefan Leutenegger, and Andrew Davison. Towards visual slam with event-based cameras. In *The Problem of Mobile Sensors Workshop in conjunction with RSS*, 2015.

[20] E. Mueggler, B. Huber, and D. Scaramuzza. Event-based, 6-dof pose tracking for high-speed maneuvers. In *International Conference on Intelligent Robots and Systems*, 2014.

[21] Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. Continuous-time trajectory estimation for event-based vision sensors. In *Robotics: Science and Systems*, 2015.

[22] N. Ratner and Y. Y. Schechner. Illumination multiplexing within fundamental limits. In *CVPR*, 2007.

[23] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259 – 268, 1992.

[24] Jos Stam. Flows on surfaces of arbitrary topology. *ACM Trans. Graph.*, 22(3):724–731, July 2003.

[25] G. Steidl and T. Teuber. Removing multiplicative noise by douglas-rachford splitting methods. *Journal of Mathematical Imaging and Vision*, 36(2):168–184, 2010.

[26] David Weikersdorfer, Raoul Hoffmann, and Jörg Conradt. Simultaneous localization and mapping for event-based vision systems. In *International Conference on Computer Vision Systems*, 2013.

[27] G. Wiesmann, S. Schraml, M. Litzenberger, A. N. Belbachir, M. Hofstätter, and C. Bartolozzi. Event-driven embodied system for feature extraction and object recognition in robotic applications. In *CVPR Workshops*, 2012.