International Journal of Computer Vision (2018) 126:1381–1393
https://doi.org/10.1007/s11263-018-1106-2

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

Gottfried Munda[1] · **Christian Reinbacher**[1] · **Thomas Pock**[1]

Received: 26 January 2017 / Accepted: 18 April 2018 / Published online: 4 July 2018

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. This paper is an extended version of our previous work (Reinbacher et al. in British machine vision conference (BMVC), 2016) and contains additional details of the variational model, an investigation of different data terms and a quantitative evaluation of our method against competing methods as well as synthetic ground-truth data.

Keywords Event camera · Denoising · Convex optimisation · Variational methods

1 Introduction

In contrast to standard CMOS digital cameras that operate on frame basis, neuromorphic cameras such as the Dynamic Vision Sensor (DVS) (Lichtsteiner et al. 2008) 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 movements (Delbruck and Lichtsteiner 2007; Wiesmann et al. 2012). The asynchronous stream of events brings a significant reduction in transmission bandwidth compared to the megabytes per second produced by a traditional frame-based sensor. However, the paradigm of a continuous event stream also requires new algorithms, since the majority of computer vision methods operates under the assumption that there exists an image with intensity information for every pixel. 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. Benosman et al. (2014), Gallego et al. (2015), Kim et al. (2014), Mueggler et al. (2014), Mueggler et al. (2015) and Weikersdorfer et al. (2013), tapping into the full potential of the high temporal resolution and low latency 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

Communicated by Xiaoou Tang.

✉ Gottfried Munda
Munda@icg.tugraz.at

Christian Reinbacher
Reinbacher@icg.tugraz.at

Thomas Pock
Pock@icg.tugraz.at

[1] Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria

Fig. 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 (Color figure online)

for a sample output of our method). In contrast to very recent work on the same topic by Bardow et al. (2016), 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, and investigate different data-terms for modeling the camera noise. The optimisation problem is defined on a manifold induced by the timestamps of new events (see Fig. 1c). 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 our software to provide live intensity image reconstruction to all users of DVS cameras.[1]

We emphasize that we do not endorse the approach of reconstructing intensity images with the purpose of running traditional frame based computer vision algorithms. In fact, we feel the appropriate way to deal with this new camera paradigm is to formulate algorithms directly in the event domain. Many recent methods that adapt classical computer vision problems for event cameras seem to agree with this point of view, see the overview in the related work (Sect. 2). However, a visualization of the raw events is not very informative, as depicted in Fig. 1a. Therefore, we also feel that there is a need for simple methods to generate a (live) preview that can be used to foster a deeper understanding of neuromorphic cameras, e.g. by demonstrating/verifying high time resolution, superior dynamic range, etc. We believe this will be a vital step towards a wider adoption of this kind of cameras. We also point out that our method stays true to the asynchronous nature of the event stream, since it is formulated on a per-event basis. In particular, our algorithm does not require the accumulation of events over a time interval.

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

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 (Hartmann et al. 2013). 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. (2015) 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. 1a to reliably track camera movement.

A different line of research tries to formulate camera pose updates on an event basis. Cook et al. (2011) 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. (2013). They simultaneously estimate a 2D map of events and track the 2D position and orientation

of the robot. Similarly, Kim et al. (2014) 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. These methods are limited to 3 DOF of camera movement. A full camera tracking has been shown in Mueggler et al. (2014) and Mueggler et al. (2015) for rapid movement of an UAV with respect to a known 2D target and in Gallego et al. (2015) for a known 3D map of the environment. Very recently, works combining 6 DOF tracking and sparse 3D reconstruction into a full SLAM system started to appear (Kim et al. 2016; Rebecq et al. 2016, 2017).

Benosman et al. (2014) 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. (2016) 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. (2016) 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 are 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. We denote the result of the integration by f^n, since it will turn out that integration alone is not enough to recover the intensity image u^n because of noise and other nuisances.

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

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

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.

4 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. (2016) 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.

4.1 Regularisation Term

As in Benosman et al. (2014), 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. 1c. Benosman et al. (2014) 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* (Lai et al. 2011), 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 (Krueger et al. 2008; Cheng et al. 2000) use an implicit surface (e.g. through the zero level set of a function), whereas explicit approaches (Lui et al. 2008; Stam 2003) construct a triangular mesh representation.

One of the difficulties of intrinsic image processing is that in many cases very little is known about the surface. This means that algorithms need to be able to deal with arbitrarily complex surfaces, which can have a high number of foldings and/or high genus. In our case, the situation is different: we observe that the surface of active events is defined by the timestamps which are monotonically increasing. Thus, 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 Lai et al. (2011), 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 (Graber et al. 2015).

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

$$X = \varphi(x, y) = \left[x,\ y,\ t(x, y)\right]^T, \tag{3}$$

where $X \in S$ denotes a 3D-point on the surface. $t(x, y)$ is the most recent timestamp for each pixel (x, y).

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 Eq. (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 the metric tensor and its inverse compute to

$$g = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, \qquad g^{-1} = \frac{1}{G} \begin{bmatrix} c & -b \\ -b & a \end{bmatrix}, \tag{5}$$

where $G = \det(g)$ and $a = 1 + t_x^2$, $b = t_x t_y$ and $c = 1 + t_y^2$.

Given a smooth function $\tilde{f} \in C^1(S, \mathbb{R})$ on the manifold, the gradient of $\tilde{f}$ is characterised by $\mathrm{d}\tilde{f}(Y) = \langle \nabla_g \tilde{f}, Y \rangle_g \quad \forall Y \in T_X\mathcal{M}$ (Lee 1997). 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 g^{-1} (see Eq. 5), the so-called pull-back. Inserting g^{-1} into Eq. (6) gives an expression for the gradient of a function $\tilde{f}$ on the manifold in local coordinates

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

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

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

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

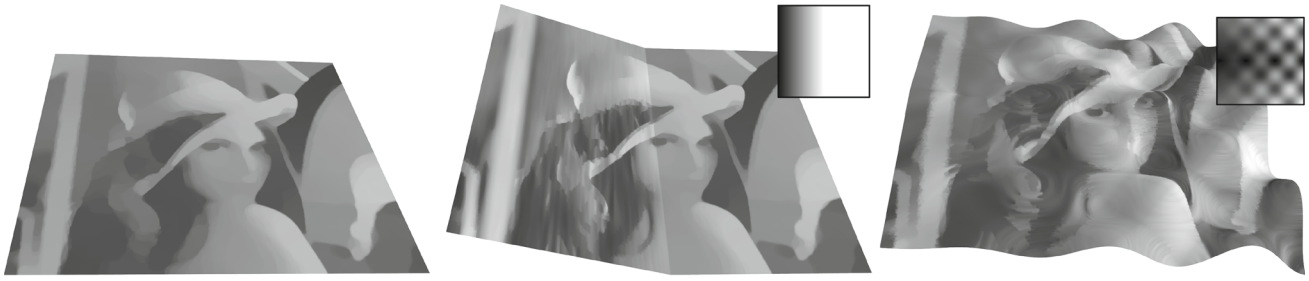

Fig. 2 ROF denoising on different manifolds. A flat surface **a** gives the same result as standard ROF denoising, but more complicated surfaces **b**, **c** significantly change the result. The graph function $t(x, y)$ is depicted in the upper right corner. We can see that a ramp surface **b** produces 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

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. (2015) 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 (Rudin et al. 1992)

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

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. 2b, c). We will use this to guide regularisation of the reconstructed image according to the surface defined by the event time.

4.2 Discretising the 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 (Chambolle 2004). For example, the x-derivative at index (i, j) of a function $u \in \mathbb{R}^{M\times M}$ can be written symbolically as $(L_x u)_{ij}$.

To define the discrete version of ∇_g, we start from the continuous definition Eq. (7). The gradient vector $\nabla_g \tilde{f}$ will have three components denoted by $(\nabla_g \tilde{f})_l$, $l = 1 \ldots 3$. It is easy to see that the first two components are given by

$$(\nabla_g \tilde{f})_1 = \frac{1}{G} \left((1 + t_y^2) \tilde{f}_x - t_x t_y \tilde{f}_y \right) \tag{11a}$$

$$(\nabla_g \tilde{f})_2 = \frac{1}{G} \left((1 + t_x^2) \tilde{f}_y - t_x t_y \tilde{f}_x \right), \tag{11b}$$

since in each case one of the terms in Eq. (7) multiplies with 0 respectively. The last component computes as follows

$$\begin{aligned}
(\nabla_g \tilde{f})_3 &= \frac{1}{G} \left\{ \left((1 + t_y^2) \tilde{f}_x - t_x t_y \tilde{f}_y \right) t_x \right. \\
&\qquad \left. + \left((1 + t_x^2) \tilde{f}_y - t_x t_y \tilde{f}_x \right) t_y \right\} \\
&= \frac{1}{G} \left(\tilde{f}_x t_x + \tilde{f}_x t_x t_y^2 - \tilde{f}_y t_x^2 t_y \right. \\
&\qquad \left. + \tilde{f}_y t_y + \tilde{f}_y t_x^2 t_y - \tilde{f}_x t_x t_y^2 \right) \\
&= \frac{1}{G} \left(\tilde{f}_x t_x + \tilde{f}_y t_y \right)
\end{aligned} \tag{12}$$

In the discrete setting, we replace the derivatives in Eqs. (11a), (11b) and (12) with multiplication by the linear operators L_x, L_y. Then the discrete version of ∇_g can be represented as a linear map $L_g : \mathbb{R}^{M\times M} \to \mathbb{R}^{M\times M\times 3}$ that acts on a function u as follows

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

Here, $G \in \mathbb{R}^{M\times M}$ is the pixel-wise determinant of g given by $G_{ij} = 1 + (L_x t)_{ij}^2 + (L_y t)_{ij}^2$. This yields the complete discrete energy

$$\min_u \|L_g u\|_g + \lambda \sum_{i,j} D_{ij}(u, f)\sqrt{G_{ij}} \quad \text{s.t. } u_{ij} \in [u_{\min}, u_{\max}], \tag{13}$$

with the g-tensor norm defined as

$$\|A\|_g = \sum_{i,j} \sqrt{G_{ij} \sum_l (A_{ijl})^2} \quad \forall A \in \mathbb{R}^{M\times M\times 3}. \tag{14}$$

We restrict the range of $u_{ij} \in [u_{\min}, u_{\max}]$ since our reconstruction problem is defined up to a grey value offset caused by the unknown initial image intensities.

The discretised data term $D_{ij}(u, f)$ will be described in Sect. 4.4. In this paper we compare the performance of different data terms. We investigate $\mathcal{L}_2$ in intensity-space, $\mathcal{L}_2$ in log-space and the *generalised Kullback-Leibler divergence*, and show how they can be incorporated in our energy minimisation framework, detailed in Sect. 4.3.

4.3 Minimising the Energy

We minimise Eq. (13) using the Primal-Dual algorithm (Chambolle and Pock 2011). Dualising the g-tensor norm yields the primal-dual formulation

$$\min_u \max_p \left\{ D(u, f^n) + \langle L_g u, p\rangle - R^*(p) \right\}, \tag{15}$$

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 Eq. (15) is obtained by iterating

$$u^{k+1} = (I + \tau \partial D)^{-1}(u^k - \tau L_g^* p^k) \tag{16a}$$

$$p^{k+1} = (I + \sigma \partial R^*)^{-1}(p^k + \sigma L_g(2u^{k+1} - u^k)), \tag{16b}$$

where L_g^* denotes the adjoint operator of L_g. The time-steps τ, σ are set according to $\tau\sigma \leq 1/\|L_g\|^2$, where we estimate the operator norm as $\|L_g\|^2 \leq 8 + 4\sqrt{2}$.

The proximal map for the regularisation term can be solved in closed form, leading to the following update rule for the dual

$$\hat{p} = \text{prox}_{\sigma R^*}(\bar{p}) \quad \Leftrightarrow \quad \hat{p}_{ijl} = \frac{\bar{p}_{ijl}}{\max\{1, \|\bar{p}_{ij,\cdot}\|/\sqrt{G_{ij}}\}}.$$

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.

4.4 Data Terms

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. The data term $D(u, f^n)$ penalises the deviation of u from the noisy measurement f^n in Eq. (1). Therefore the data term should be modelled according to the expected noise distribution in the input data. In contrast to Reinbacher et al. (2016) we will now investigate different choices for the data term. We qualitatively compare the data terms in Fig. 3 and later in quantitative experiments in Sect. 5.1.

4.4.1 $\mathcal{L}_2$ in Intensity-Space

Let us assume that at a certain time n the noise between the accumulated image f^n and u is Gaussian distributed

$$\left(u - f^n\right) \sim \mathcal{N}\left(0, \sigma^2\right). \tag{17}$$

We therefore choose the following data fidelity term that is suitable for Gaussian distributed noise:

$$D_{ij}(u, f^n) := \frac{1}{2}\left(u_{ij} - f_{ij}^n\right)^2. \tag{18}$$

The proximal operator needed for Eq. (16a) can be easily written in closed form as

$$\hat{u} = \text{prox}_{\tau D}(\bar{u}) \Leftrightarrow \hat{u}_{ij} = \underset{u_{\min}, u_{\max}}{\text{clamp}}\left(\frac{\bar{u}_{ij} + \tau f_{ij}^n}{1 + \tau}\right), \tag{19}$$

where $\underset{x_{\min}, x_{\max}}{\text{clamp}}(x) = \max(x_{\min}, \min(x_{\max}, x))$.

4.4.2 $\mathcal{L}_2$ in Log-Space

We know that the event camera operates in log-space rather than intensity space like most cameras. We therefore modify our assumption on the noise to be affecting the log images as

$$\left(\log u - \log f^n\right) \sim \mathcal{N}\left(0, \sigma^2\right). \tag{20}$$

Our modified data term therefore reads as

$$D_{ij}(u, f^n) := \frac{1}{2}\left(\log u_{ij} - \log f_{ij}^n\right)^2 = \frac{1}{2}\left(\log \frac{u_{ij}}{f_{ij}^n}\right)^2 = d\left(\frac{u_{ij}}{f_{ij}^n}\right), \tag{21}$$

(a) (b) (c)

Fig. 3 Qualitative comparison of the investigated data terms on a self recorded table top scene. In **a** the $\mathcal{L}_2$ data term in intensity space, described in Sect. 4.4.1 is used. **b** Shows the output of $\mathcal{L}_2$ data term in log space, described in Sect. 4.4.2. **c** Shows the output of the Kullback–Leibler divergence data term, described in Sect. 4.4.3

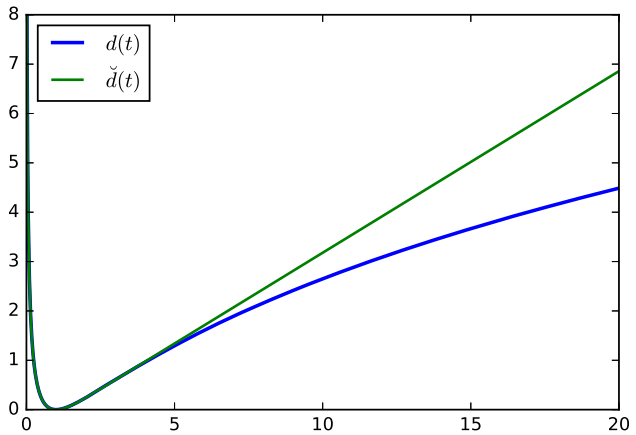

Fig. 4 Comparison of log $\mathcal{L}_2$ data term $d(t)$ and its convex approximation $\check{d}(t)$. The approximation error is small for $t \in [0, 5]$. Since in our case $t = \frac{u_{ij}}{f_{ij}^n}$, this means that u_{ij} can be up to 5 times as large as f_{ij}^n. As the value of u^{ij} is typically close to f_{ij}^n, in most cases the condition $t \in [0, 5]$ should be fulfilled

where $d(t) = \frac{1}{2}\log(t)^2,\ t > 0$.

In contrast to $\mathcal{L}_2$ in intensity-space, this data term is non-convex. We see that it is convex for $\log t < 1$ by looking at the zero-crossing of the second derivative $d''(t) = \frac{1-\log t}{t^2}$.

We propose to use a convex approximation of $d(t)$ given by

$$\check{d}(t) = \begin{cases} d(t) & \text{if } \log t < 1 \\ d(e) + d'(e)(t-e) = -\frac{1}{2} + \frac{t}{e} & \text{else} \end{cases} \tag{22}$$

which replaces the non-convex part with a first-order Taylor expansion around $t = e$. A visualization of $d(t)$ and its convex approximation $\check{d}(t)$ is depicted in Fig. 4.

The proximal operator can not be calculated in closed form, therefore we propose to solve

$$\hat{u} = \text{prox}_{\tau D}(\bar{u}) \Leftrightarrow \min_u \check{d}(u) + \frac{\|u - \bar{u}\|^2}{2\tau} \tag{23}$$

using a few Gauss–Newton iterations

$$\hat{u}^{n+1} = \hat{u}^n - \frac{\hat{u}^n - \bar{u} + \tau \check{d}'(\hat{u}^n)}{1 + \tau \check{d}''(\hat{u}^n)}, \tag{24}$$

with $\hat{u}^0 = \bar{u}$.

4.4.3 Kullback–Leibler Divergence

While the previously presented models are sufficient for many applications, real sensor noise is dependent on scene brightness and should be modelled as a Poisson distribution (Ratner and Schechner 2007). We therefore define our data term to be

$$D_{ij}(u, f^n) := u_{ij} - f_{ij}^n \log u_{ij} \tag{25}$$

whose minimiser is known to be the correct ML-estimate under the assumption of Poisson-distributed noise between u and f^n (Le et al. 2007). Note that, in contrast to Graber et al. (2015), we also define the data term to lie on the manifold. Equation (25) is also known as *generalised Kullback–Leibler divergence* and has been investigated by Steidl and Teuber (2010) in variational image restoration methods. Furthermore, the data term again is convex, which makes it easy to incorporate into our variational energy minimisation framework. The proximal operator needed in Eq. (16a) can be calculated in closed form as

$$\hat{u} = \text{prox}_{\tau D}(\bar{u}) \Leftrightarrow \hat{u}_{ij} = \underset{u_{\min}, u_{\max}}{\text{clamp}} \left(\frac{1}{2} \left(\bar{u}_{ij} - \beta_{ij} + \sqrt{\left(\bar{u}_{ij} - \beta_{ij}\right)^2 + 4\beta_{ij} f_{ij}^n} \right) \right) \tag{26}$$

with $\beta_{ij} = \tau\lambda\sqrt{G_{ij}}$.

4.4.4 Discussion

In Fig. 3 we compare the output of out method for different choices of data terms on a self-recorded desktop scene. As can be seen from the pictures, $\mathcal{L}_2$ in intensity space is not very well suited for this type of camera noise. Isolated black and white pixels remain after reconstruction. $\mathcal{L}_2$ in log space and Kullback–Leibler divergence (KLD) perform similarly, with a slightly higher contrast and general image quality in the case of log $\mathcal{L}_2$. In practice, we use log $\mathcal{L}_2$ since it results in the highest image quality at a small additional computational cost ($\approx 10\%$ slower than KLD).

5 Experiments

We perform our experiments using a DVS128 camera with a spatial resolution of 128×128 and a recently proposed dataset that has been acquired using a DAVIS240 with a resolution of 240×180 (Mueggler et al. 2016). 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 Sect. 4 due to noise. We therefore denoise the timestamps with a few iterations of a TV-L1 denoising method. We compare our method to the method of Bardow et al. (2016) on sequences provided by the authors. Furthermore, we will show the influence of the proposed regularisation on the event manifold using synthetic data from Mueggler et al. (2016).

5.1 Influence of Hyperparameters

We begin our evaluation with quantitative experiments on synthetic sequences of a recently proposed dataset for event-based computer vision applications (Mueggler et al. 2016). The dataset consists of 27 sequences of lengths between 2 s and 2 min with associated ground-truth camera pose. Two of those sequences have been generated by a DAVIS simulator that uses rendered scenes created by Blender. We will use them to validate our image reconstruction approach and also reveal the relationship between the different hyperparameters of your method.

The synthetic sequences are 2 s long and consist of an asynchronous event stream and 2000 frames rendered by Blender. The provided frames allow us to compare the reconstruction output of our method to the frames that were used to generate the events. Each ground-truth frame is registered to the events via a timestamp. In order to compare our output, we search for the output frame which is closest in time to each ground-truth frame.

As error measure, we use the *Feature Similarity Index* (FSIM), proposed by Zhang et al. (2011). FSIM uses the phase congruency as the primary feature and the image gradient magnitude as secondary feature. It achieves a high consistency with subjective image quality impression. We chose FSIM because it is invariant to a global grey-value offset (which can not be recovered from events alone).

In this first experiment we verify the ability of our method to recover the initial intensity image u_0. The synthetic nature of the test data allows us to provide our method the correct u_0 instead of starting from an uniform image. In Fig. 5 we plot the FSIM value for each frame of the test sequence. In Fig. 5a an initial image is provided, whereas in Fig. 5b the method is initialized with $u_0 = const$. For both variants, λ has been fixed to 100. In the first row of Fig. 5, we plot the FSIM value for the $\mathcal{L}_2$ data term in log-space (as defined in Sect. 4.4.2) for a varying number of events per reconstructed frame. As can be seen from this plot, our method is rather agnostic regarding the number of events per image. For the second row of Fig. 5 we have set the number of events per reconstructed frame to 300.

5.2 Influence of the Event Manifold

In the previous section we investigated the optimal parameter setting for the proposed method. In this second experiment we show the influence of defining the reconstruction problem on the manifold induced by the timestamps of the events. For that, we switch off the event manifold by setting the timestamps of all events to a constant. Therefore the metric tensor defined in Eq. (4) reduces to the identity matrix and the subsequent optimization is carried out in image space.

Quantitative results using the best-performing parameter settings are reported in Table 1. The increase in performance regarding to the used error metric is small. To give the reader a better impression of the impact in real-world scenes, we have captured a few sequences around our office with a DVS128 camera. In Fig. 6 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 which results in images with notably less contrast.

5.3 Comparison to Related Methods

In this section we compare our reconstruction method to the method proposed by Bardow et al. (2016). The authors kindly provided us with the recorded raw events, as well as intensity image reconstructions at regular timestamps $\delta t = 15$ ms.

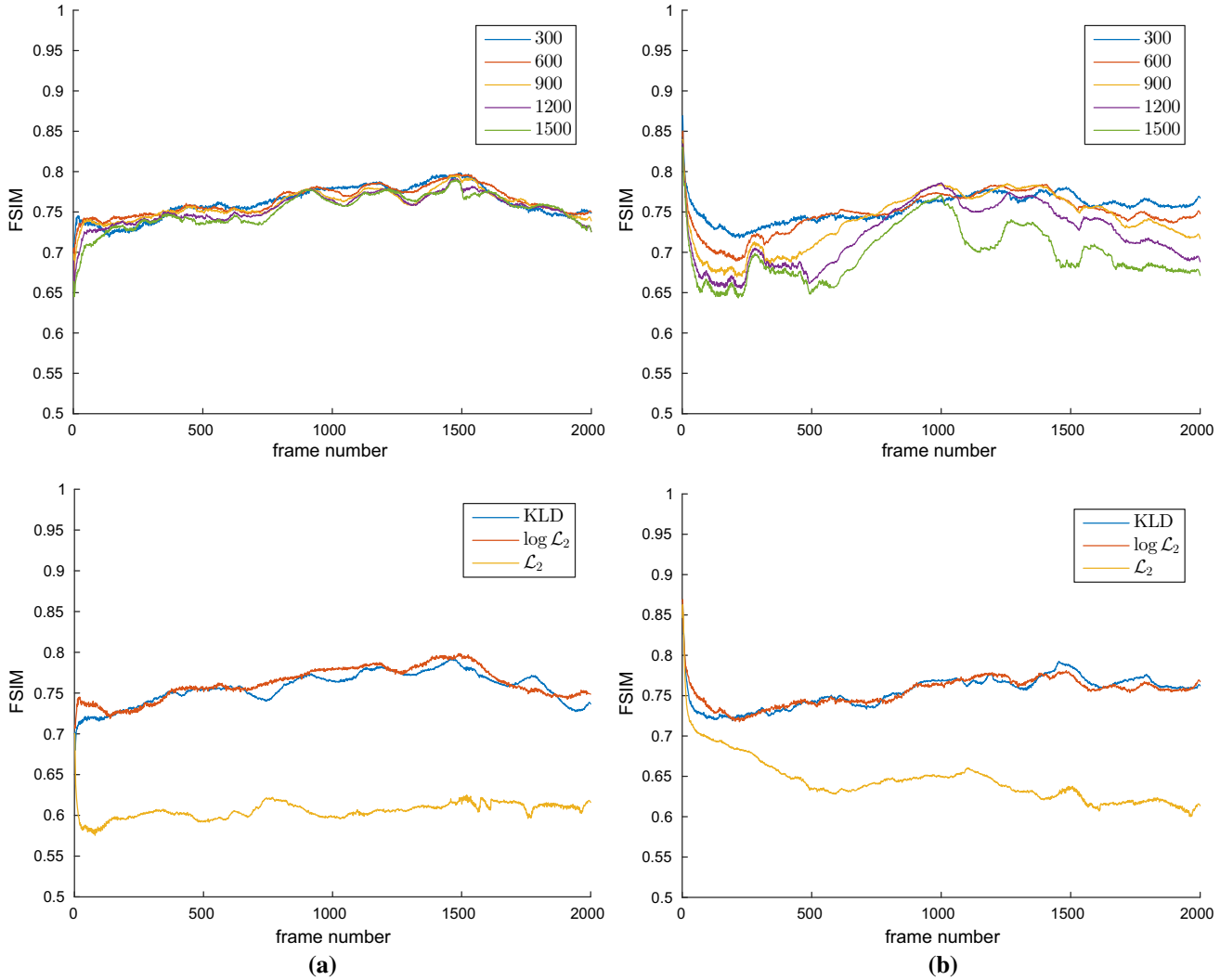

Fig. 5 Image quality (FSIM measure) related to hyperparameter setting. In the first row we fix the data term to $\mathcal{L}_2$ in log-space and vary the number of events per reconstructed frame. In the second row we fix the number of events per reconstructed frame to 300 and show the results using different data terms. **a** Unknown u_0, **b** known u_0

Table 1 Quantitative evaluation of the manifold regularisation

Method	Known u_0	Unknown u_0
OURS no manifold	0.7622 ± 0.0197	0.7548 ± 0.0165
OURS full	0.7724 ± 0.0216	0.7656 ± 0.0128

The reported numbers are mean and standard deviation of the FSIM measure (higher = better) applied to all reconstructed frames of the sequences

Since we process shorter event packets, we search for the nearest neighbour timestamp for each image of Bardow et al. (2016) in our sequences. We visually compare our method on the sequences *face*, *jumping jack* and *ball* to the results of Bardow et al. (2016) in Fig. 7. For this experiment we chose the *Kullback–Leibler Divergence* as data term. Since we are dealing with highly dynamic data, we point the reader to the included supplementary video[2] which shows whole sequences of several hundred frames.

We point out that no ground truth data is available. In order to also provide a quantitative evaluation, we use the BRISQUE score, proposed in Mittal et al. (2012). BRISQUE is a no-reference image quality measure that allows to quantify the "naturalness" of an image, in case no ground-truth image is available. The values reported by BRISQUE range from 0 (=very unnatural) to 100 (=very high quality). We compared the output of our method on the sequences *face*, *jumping jack* and *ball* to the results of Bardow et al. (2016) in Table 2. The reported numbers are the mean and standard

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

Fig. 6 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) and the more natural shading on the wall with light from the overhead window. **a** Events, **b** manifold, **c** w/o MR, **d** with MR

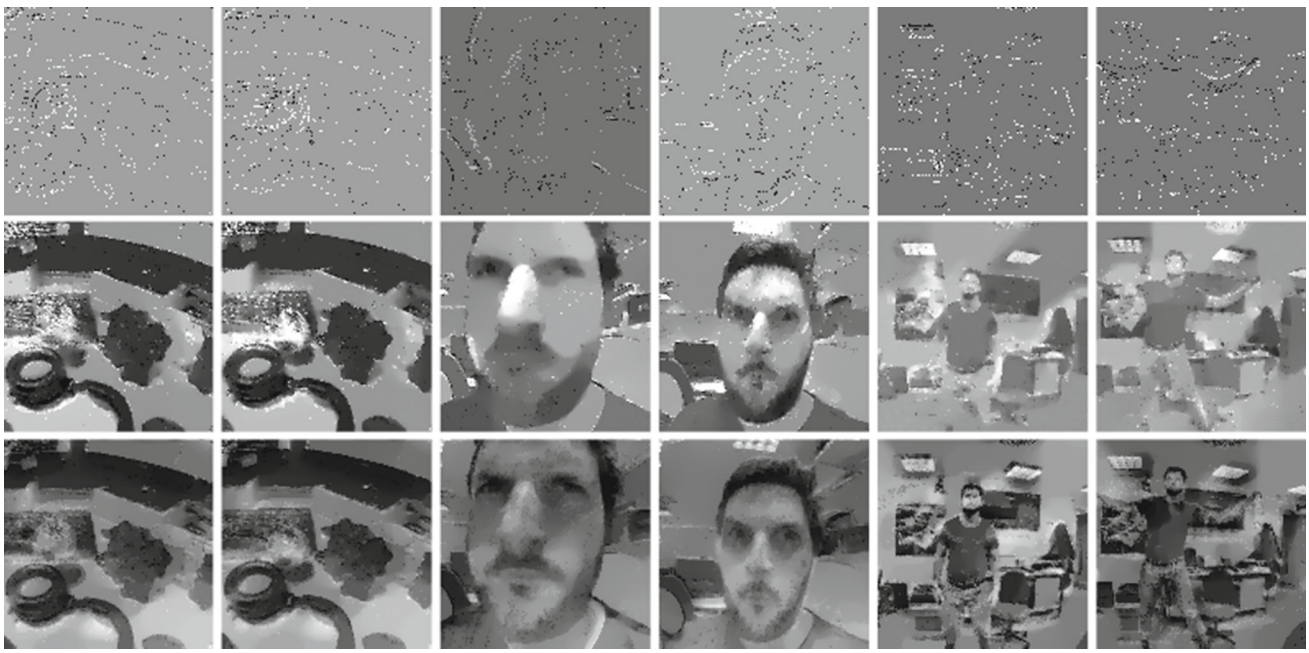

Fig. 7 Comparison to the method of Bardow et al. (2016). The first row shows the raw input events that have been used for both methods. The second row depicts the results of Bardow et al., and the last row shows our result. We can see that out method produces more details (e.g. face, beard) as well as more graceful gray value variations in untextured areas, where Bardow et al. (2016) tends to produce a single gray value

Table 2 Quantitative comparison to the method of Bardow et al. (2016)

Method	Face	Jumping	Ball
Bardow et al. (2016)	22.27 ± 8.81	29.39 ± 7.27	29.37 ± 9.61
OURS	27.29 ± 7.27	48.18 ± 6.70	34.98 ± 9.31

The reported numbers are mean and standard deviation of the BRISQUE measure applied to all reconstructed frames of the sequences

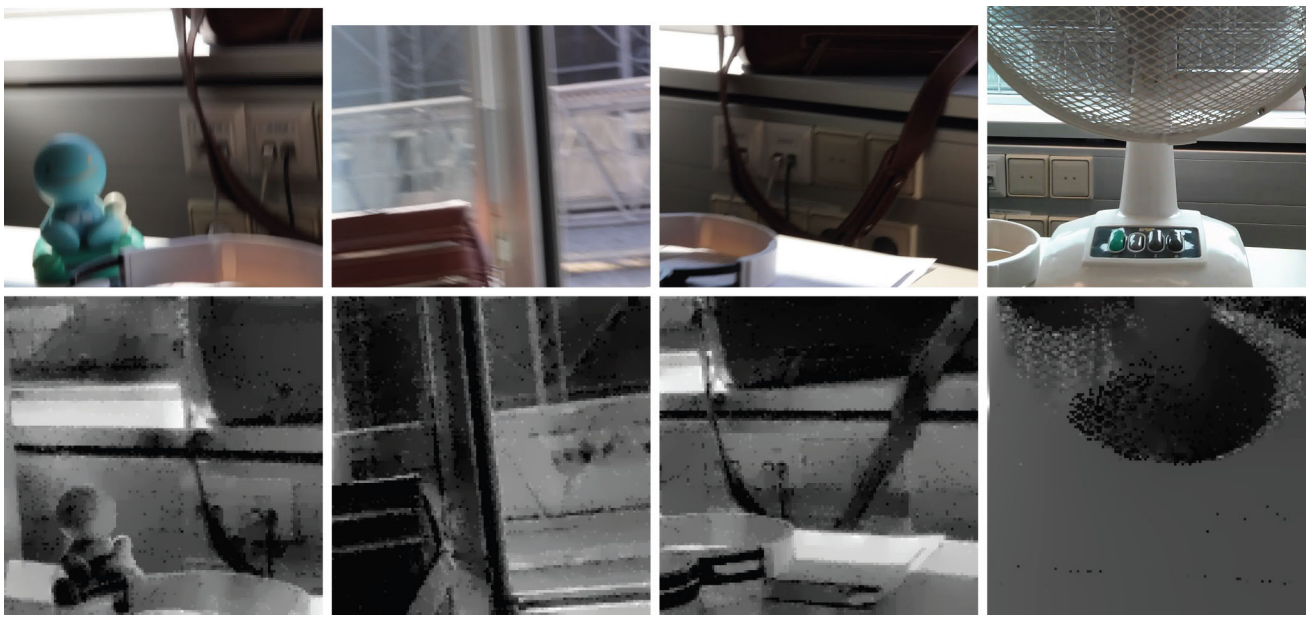

Fig. 8 Comparison to a video captured with a modern DSLR camera. Notice the rather strong motion blur in the images of the DSLR (top row), whereas the DVS camera can easily deal with fast camera or object movement (bottom row). Even fast moving objects such as the fan blades in the last column can be reconstructed from the sparse event stream (including the protecting grill in front of the blades)

deviation over the whole sequences. The results correspond to the visual impression in Fig. 7. While our result on the ball sequence is very similar to Bardow et al. (2016), our output on the jumping jack sequence features much more details, resulting in considerably higher score.

5.4 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 Bardow et al. (2016), rapid movement results in motion blur for conventional 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. 8.

5.5 Timing

In this paper 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.[3] Using this setup and KLD as data term (see Sect. 4.4.3), we measure a wall clock time of 1.7 ms to create one single image, which amounts to $\approx$ 580 fps. While we could create a new image for each new event, this would create a tremendous amount of images due to the number of events ($\approx$ 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. Accumulating 1000 events to produce one image amounts to a time resolution of 3–5 ms and allows us to achieve real-time performance.

[3] 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.

6 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.

We have evaluated the method both quantitatively and qualitatively by comparing it to related methods as well as ground-truth data from a simulator.

We have investigated three different data terms to model the noise characteristic of event cameras. While the current models 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

Bardow, P., Davison, A., & Leutenegger, S. (2016). Simultaneous optical flow and intensity estimation from an event camera. In *CVPR*.

Barua, S., Miyatani, Y., & Veeraraghavan, A. (2016). Direct face detection and video reconstruction from event cameras. In *2016 IEEE winter conference on applications of computer vision (WACV)* (pp. 1–9). https://doi.org/10.1109/WACV.2016.7477561.

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

Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, *20*(1–2), 89–97.

Chambolle, A., & Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, *40*(1), 120–145.

Cheng, L. T., Burchard, P., Merriman, B., & Osher, S. (2000). Motion of curves constrained on surfaces using a level set approach. *Journal of Computational Physics*, *175*, 2002.

Cook, M., Gugelmann, L., Jug, F., Krautz, C., & Steger, A. (2011). Interacting maps for fast visual interpretation. In *The 2011 international joint conference on neural networks (IJCNN)* (pp. 770–776). https://doi.org/10.1109/IJCNN.2011.6033299

Delbruck, T., & Lichtsteiner, P. (2007). Fast sensory motor control based on event-based hybrid neuromorphic-procedural system. In *International symposium on circuits and systems*.

Gallego, G., Forster, C., Mueggler, E., & Scaramuzza, D. (2015). Event-based camera pose tracking using a generative event model. CoRR arXiv:1510.01972.

Graber, G., Balzer, J., Soatto, S., & Pock, T. (2015). Efficient minimal-surface regularization of perspective depth maps in variational stereo. In *CVPR*.

Hartmann, J., Klüssendorff, J. H., & Maehle, E. (2013). A comparison of feature descriptors for visual slam. In *European conference on mobile robots*.

Kim, H., Handa, A., Benosman, R., Ieng, S. H., & Davison, A. (2014). Simultaneous mosaicing and tracking with an event camera. In *BMVC*.

Kim, H., Leutenegger, S., & Davison, A. (2016). Real-time 3d reconstruction and 6-dof tracking with an event camera. In *Proceedings of European conference on computer vision*.

Krueger, M., Delmas, P., & Gimel'farb, G. L. (2008). Active contour based segmentation of 3d surfaces. In *ECCV*.

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

Le, T., Chartrand, R., & Asaki, T. J. (2007). A variational approach to reconstructing images corrupted by poisson noise. *Journal of Mathematical Imaging and Vision*, *27*, 257–263.

Lee, J. M. (1997). *Riemannian manifolds: An introduction to curvature*., Graduate Texts in Mathematics New York: Springer.

Lichtsteiner, P., Posch, C., & Delbruck, T. (2008). A 128 × 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, *43*(2), 566–576.

Lui, L. M., Gu, X., Chan, T. F., & Yau, S. T. (2008). Variational method on riemann surfaces using conformal parameterization and its applications to image processing. *Methods and Applications of Analysis*, *15*(4), 513–538.

Milford, M., Kim, H., Leutenegger, S., & Davison, A. (2015). Towards visual slam with event-based cameras. In *The problem of mobile sensors workshop in conjunction with RSS*.

Mittal, A., Moorthy, A. K., & Bovik, A. C. (2012). No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, *21*(12), 4695–4708. https://doi.org/10.1109/TIP.2012.2214050.

Mueggler, E., Gallego, G., & Scaramuzza, D. (2015). Continuous-time trajectory estimation for event-based vision sensors. In *Robotics: science and systems*.

Mueggler, E., Huber, B., & Scaramuzza, D. (2014). Event-based, 6-dof pose tracking for high-speed maneuvers. In *International conference on intelligent robots and systems*.

Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., & Scaramuzza, D. (2016). The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. arXiv:1610.08336.

Ratner, N., & Schechner, Y. Y. (2007). Illumination multiplexing within fundamental limits. In *CVPR*.

Rebecq, H., Gallego, G., & Scaramuzza, D. (2016). EMVS: Event-based multi-view stereo. In *Proceedings of the british machine vision conference, BMVC*.

Rebecq, H., Horstschaefer, T., Gallego, G., & Scaramuzza, D. (2017). EVO: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, *2*(2), 593–600.

Reinbacher, C., Graber, G., & Pock, T. (2016). Real-time intensity-image reconstruction for event cameras using manifold regularisation. In *British machine vision conference (BMVC)*.

Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, *60*(1), 259–268.

Stam, J. (2003). Flows on surfaces of arbitrary topology. *ACM Transactions on Graphics*, *22*(3), 724–731.

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

Weikersdorfer, D., Hoffmann, R., & Conradt, J. (2013). Simultaneous localization and mapping for event-based vision systems. In *International conference on computer vision systems*.

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

Zhang, L., Zhang, L., Mou, X., & Zhang, D. (2011). Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, *20*(8), 2378–2386. https://doi.org/10.1109/TIP.2011.2109730.