

Event-based Mosaicing Bundle Adjustment

Shuang Guo¹ and Guillermo Gallego^{1,2}

¹ TU Berlin and Robotics Institute Germany, Berlin, Germany

² Einstein Center Digital Future and SCIOI Excellence Cluster, Berlin, Germany

Abstract. We tackle the problem of mosaicing bundle adjustment (i.e., simultaneous refinement of camera orientations and scene map) for a purely rotating event camera. We formulate the problem as a regularized non-linear least squares optimization. The objective function is defined using the linearized event generation model in the camera orientations and the panoramic gradient map of the scene. We show that this BA optimization has an exploitable block-diagonal sparsity structure, so that the problem can be solved efficiently. To the best of our knowledge, this is the first work to leverage such sparsity to speed up the optimization in the context of event-based cameras, without the need to convert events into image-like representations. We evaluate our method, called EMBA, on both synthetic and real-world datasets to show its effectiveness (50% photometric error decrease), yielding results of unprecedented quality. In addition, we demonstrate EMBA using high spatial resolution event cameras, yielding delicate panoramas in the wild, even without an initial map. The code is available at <https://github.com/tub-rip/emba>.

1 Introduction

Event cameras are novel bio-inspired visual sensors that measure per-pixel brightness changes [11, 28, 40]. In contrast to the images/frames captured by standard cameras, the output of an event camera is a stream of asynchronous events. This unique working principle endows event cameras with great potential in the tasks of camera motion estimation and scene reconstruction, especially in scenarios of high dynamic range (HDR), low power consumption and/or fast motion [12].

Bundle Adjustment (BA) is the problem of jointly refining the camera motion and the reconstructed scene map that best fit the visual data through a given objective function [2, 43] (e.g., reprojection or photometric error). It is a paramount topic in photogrammetry, computer vision and robotics, enabling accurate positioning and measurement technology for applications such as image stitching [5], visual odometry (VO) [9], simultaneous localization and mapping (SLAM) [2, 6] and AR/VR [10]. BA with frame-based cameras is a mature topic [2, 21, 26, 42]. In contrast, BA with event cameras is still in its infancy, which limits the maturity of the above-mentioned applications for event cameras.

A key problem for BA and SLAM-related tasks is data association, i.e., establishing correspondences between measurements to identify which pixels observe the same scene point [17, 22]. So far, event-based BA has been applied

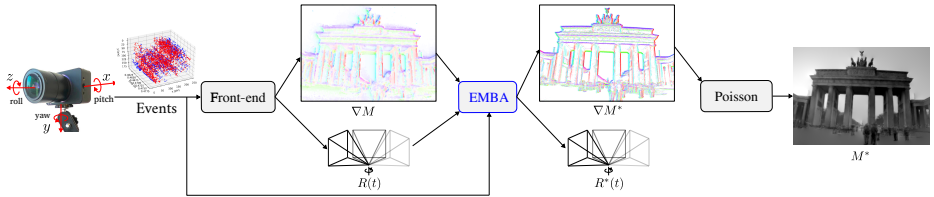


Fig. 1: Our back-end module EMBA jointly refines the camera rotations and panoramic gradient map. The intensity map can be recovered by solving Poisson’s equation.

in a feature-based (i.e., indirect) manner, i.e., extracting sparse keypoints from image-like event representations and associating them over time (e.g., [7, 44]). However, this discards the large amount of information contained in the events (as shown in image reconstruction [29, 34, 41, 46]) and/or quantizes their high temporal resolution. Instead, recent development in direct methods with event cameras [20, 22, 35] suggest that it should be possible to achieve BA while exploiting the unique characteristics of events, namely that they are continuously (asynchronously) triggered by edges as the camera moves, and that each event is a relative brightness measurement (i.e., an increment if using logarithmic scale).

This paper proposes an event-based mosaicing bundle adjustment (EMBA) method to tackle the photometric BA problem for event cameras (Fig. 1). Rotational motion is a rich and practical scenario, as shown by previous works [3, 7, 8, 16, 20, 24, 25, 35]. It is essential to many applications: panorama creation (e.g., in smartphones), star tracking [3], and VO/SLAM in dominantly-rotational motion cases (e.g., satellites [7]). We leverage the linearized event generation model (LEGM) to formulate the problem as a regularized non-linear least squares (NLLS) optimization in the high-dimensional space of camera motions and panoramic gradient maps. Due to the sparse property of event data, only a portion of pixels of the panoramic map are observed are consequently refined, which naturally leads to a semi-dense gradient map. Moreover, the LEGM also yields a block-diagonal sparsity pattern within the system equations, which we leverage to design an efficient second-order solver.

Therefore, to the best of our knowledge, EMBA is novel. In the experiments, we run EMBA to refine the camera motion trajectories and maps obtained by four state-of-the-art event-based rotation estimation front-end methods [16, 24, 25, 35], on both synthetic and real-world datasets. The results show notable improvements in terms of both camera motion and map quality, revealing previously hidden scene details. We also demonstrate the application of EMBA to generate high-quality panoramas in outdoor scenes with high resolution event cameras, without requiring an initial map. That is, EMBA just needs a set of initial camera rotations (e.g., provided by an IMU or some front-end) to recover a delicate panorama from scratch while jointly refining the camera motion.

Our contributions can be summarized as follows:

- We propose a novel event-only mosaicing bundle adjustment method, which refines an event-camera’s trajectory orientation and gradient map, produc-

Table 1: *VO/SLAM systems that use event data.* The columns indicate the number of degrees of freedom (DOFs), the type of method (**D**irect or **I**ndirect –feature-based), whether there is a refinement step (back-end [6]), and whether the method exploits the event generation model (linearized –LEGM– or not).

System	Year	DOFs	Refine	D/I	EGM	Remarks
Weikersdorfer et al. [45]	2013	3	✗	D	✗	Edge map; 2D scenario
PF-SMT [24]	2014	3	✗	D	✓	LEGM. Brightness map
RTPT [35]	2017	3	✗	D	✗	Probabilistic map
CMax- ω [16]	2017	3	✗	D	✗	The map is a local IWE
EKF-SMT [23]	2018	3	✗	D	✓	LEGM. Brightness map
Chin et al. [7]	2019	3	✓	I	✗	Converts events into frames
CMax-GAE [25]	2021	3	✗	D	✗	The map is a growing 3D-point set
CMax-SLAM [20]	2024	3	✓	D	✗	The map is a panoramic IWE
This work	2024	3	✓	D	✓	LEGM. Event-only photometric BA

ing a high quality grayscale panoramic map of the scene (Sec. 3). Its key ingredients are: formulating the BA problem as a regularized NLLS optimization and leveraging the block-diagonal sparsity pattern induced by the chosen parameterization to implement an efficient solver (Sec. 3.2).

- We conduct a comprehensive evaluation on synthetic and real-world datasets (Sec. 4) using four state-of-the-art front-ends for initialization. We demonstrate the method using high-resolution event cameras (VGA and HD), obtaining remarkable panoramas without map initialization (Sec. 4.5).
- We make the source code publicly available.

2 Related Work

Table 1 summarizes some of the VO/SLAM methods operating on event data.

Event-based Rotation Estimation. Several works have demonstrated the capabilities of event cameras to estimate rotational motion in challenging scenarios (e.g., high speed, HDR). Kim *et al* [24] proposed a 3-DOF simultaneous mosaicing and tracking (SMT) method consisting of two Bayesian filters operating in parallel (PF-SMT – particle filter SMT); it estimated the camera motion and a grayscale intensity map of the scene. Later, the tracker was replaced by a Kalman filter [13], yielding EKF-SMT [23]. Although EMBA uses a similar measurement model (LEGM) as SMT, the latter only performs local-time estimation since it is filter-based. Conversely, EMBA is optimization-based, so it can perform global refinement in both time and map domains. Also working in parallel, a real-time panoramic tracking and probabilistic mapping was presented in [35] (RTPT), where the panoramic map of the scene stored the spatial event rate at each point (instead of intensity). Using contrast maximization (CMax), [16] proposed to estimate the camera’s angular velocity by warping events on the image plane and aligning them via a focus function [14]. The work has been extended to jointly estimate angular velocity and orientation in [25] (CMax-GAE).

Bundle Adjustment. All above-mentioned methods are short-term, i.e., front-ends of SLAM systems. They lack a BA refinement module, i.e., a SLAM

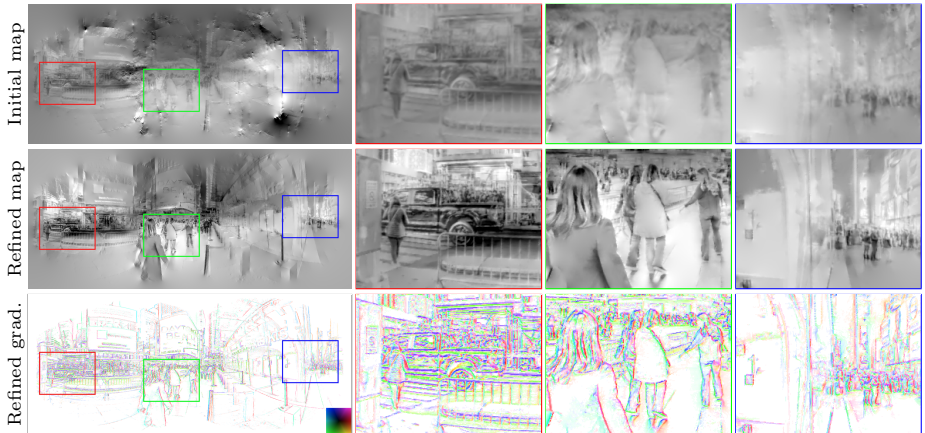


Fig. 2: Initial intensity map (top), and final map M (middle), via the refined gradient map ∇M (bottom), for the *street* data from [20]. Three insets are also shown.

back-end [6], which is desirable to improve accuracy and consistency. Surveying the literature, [7] introduced a BA approach for an event-based system; but it was feature-based and tested only on synthetic star-tracking data. Guo et al. [20] augmented [16] with a back-end, but the map was obtained as a by-product of camera trajectory refinement, resulting in a panoramic edgemap (no intensity map). Expanding the survey to 6-DOF motions, USLAM [36] fused events, frames and IMU data: keypoints were extracted from motion-compensated event-images and frames, and fed to a classical back-end [27]. Recent work [22] (EDS) proposed an event-aided direct VO system, in which event data was leveraged to track the camera motion during the blind time between frames. The system borrowed the photometric BA module from direct methods like DSO [9], which works on images. Current stereo methods are semi-dense and lack a back-end [47], or have a back-end but are feature-based (i.e., indirect) [44]. Therefore, to the best of our knowledge, event-only photometric (i.e., direct) BA is still an unexplored topic, which we address.

3 Event-based Mosaicing Bundle Adjustment

3.1 Event Generation Model (EGM)

EGM on the sensor. Each pixel of an event camera measures brightness changes independently, producing an event $e_k \doteq (\mathbf{x}_k, t_k, s_k)$ when the logarithmic intensity change ΔL at the pixel reaches a preset contrast threshold C [12]:

$$\Delta L \doteq L(\mathbf{x}_k, t_k) - L(\mathbf{x}_k, t_k - \Delta t_k) = s_k C, \quad (1)$$

where the event polarity $s_k \in \{+1, -1\}$ indicates the sign of the intensity change, and Δt_k is the time elapsed since the last event at the same pixel $\mathbf{x}_k = (x_k, y_k)^\top$.

Assuming brightness constancy (i.e., optical flow constraint), one can further linearize (1) to obtain the “linearized event generation model” (**LEGM**) [12]:

$$\Delta L \approx -\nabla L(\mathbf{x}_k, t_k) \cdot \mathbf{v} \Delta t_k = s_k C. \quad (2)$$

It states that the brightness change ΔL is caused by an edge ∇L moving with velocity \mathbf{v} during Δt over a displacement $\Delta \mathbf{x} = \mathbf{v} \Delta t$. The dot product captures the condition that no event is triggered if the motion is parallel to the edge.

EGM on the scene map. Following [24], we may model the scene map using a mosaic, $M : \mathbb{R}^2 \rightarrow \mathbb{R}$ (e.g., Fig. 2), where each map point \mathbf{p} holds the logarithmic intensity of the 3D world point viewed in the direction of \mathbf{p} . As the camera rotates, the correspondence between camera pixels \mathbf{x} and map points \mathbf{p} varies. This warp (i.e., geometric transformation) depends on the camera orientation $\mathbf{R}(t)$, intrinsic calibration \mathbf{K} and type of projection model π (e.g., equirectangular) used to represent the map: $\mathbf{x} \mapsto \mathbf{p}$, i.e.,

$$\mathbf{p}(t) \doteq \mathbf{W}(\mathbf{x}; \mathbf{R}(t), \mathbf{K}, \pi). \quad (3)$$

Given this correspondence, we may reformulate (2) in terms of the map:

$$\Delta L \approx \nabla M(\mathbf{p}(t_k)) \cdot \Delta \mathbf{p}(t_k) = s_k C, \quad (4)$$

where $\Delta \mathbf{p}(t_k) \doteq \mathbf{p}(t_k) - \mathbf{p}(t_k - \Delta t_k)$ is the map displacement “traveled” by the pixel \mathbf{x}_k as the camera moves during Δt_k . Hence, the LEGM (4) naturally associates each event e_k with the brightness gradient at one map point, $\nabla M(\mathbf{p}(t_k))$.

3.2 Problem Formulation

Objective or Loss Function. Stemming from (4), each event represents a brightness change of predefined size C , which can be modeled in terms of the camera motion $\mathbf{R}(t)$ and the scene texture ∇M . Hence, assuming C is known, a natural design choice consists of formulating the BA problem as finding the motion and scene parameters \mathbf{P} that minimize the sum of square errors

$$g(\mathbf{P}) \doteq \sum_{k=1}^{N_e} (\hat{\Delta L}_k(\mathbf{P}) - \Delta L_k)^2 \quad (5)$$

where $\Delta L_k \doteq s_k C$ is the measurement, $\hat{\Delta L}_k(\mathbf{P}) \doteq \nabla M \cdot \Delta \mathbf{p}$ is its prediction, and N_e is the number of events considered. Stacking the per-event error terms into a vector $(\mathbf{e})_k \doteq \hat{\Delta L}_k(\mathbf{P}) - \Delta L_k$, we may rewrite the problem as:

$$\min_{\mathbf{P}} g(\mathbf{P}), \quad \text{with} \quad g = \|\mathbf{e}\|^2 = \mathbf{e}^\top \mathbf{e}, \quad (6)$$

where $\mathbf{e}(\mathbf{P}) \in \mathbb{R}^{N_e}$ is the photometric error (or “residual”) vector. This is a non-linear least squares (NLLS) function of the state \mathbf{P} . It admits the probabilistic interpretation of maximum likelihood estimation under the assumption of zero-mean Gaussian noise in the temporal contrast ΔL , which is a sensible choice according to empirical evidence [28, Fig.6].

Solution Approach. The standard and effective approach to minimize NLLS objectives like (6) is Gauss-Newton’s (GN) method and its variations, e.g., Levenberg-Marquardt (LM) [4, 21]. They linearize the errors in terms of the parameters, solve the normal equations and update the model parameters $\Delta \mathbf{P}^*$, iterating until local convergence. For GN, assuming an “operating point” \mathbf{P}_{op} (in a high dimensional space) and a perturbation $\Delta \mathbf{P}$ around this operating point, the errors are linearized in terms of the parameters:

$$\mathbf{e} \approx \mathbf{e}_{\text{op}} + \mathbf{J}_{\text{op}} \Delta \mathbf{P}, \quad (7)$$

where $\mathbf{e}_{\text{op}} = \mathbf{e}(\mathbf{P}_{\text{op}})$, and \mathbf{J}_{op} is the derivative of the error with respect to \mathbf{P} . Inserting (7) into (6), differentiating with respect to $\Delta \mathbf{P}$ and setting the result equal to zero yields the necessary optimality condition. The optimal perturbation $\Delta \mathbf{P}^*$ satisfies the system of normal equations:

$$\mathbf{J}_{\text{op}}^\top \mathbf{J}_{\text{op}} \Delta \mathbf{P}^* = -\mathbf{J}_{\text{op}}^\top \mathbf{e}_{\text{op}} \quad \Leftrightarrow \quad \mathbf{A} \Delta \mathbf{P}^* = \mathbf{b} \quad (8)$$

The optimal perturbation is used to update the “operating point” and iterate.

While this approach may appear as a classic one, several challenges are involved: (i) designing a meaningful and well-behaved loss, (ii) identifying a suitable parametrization, (iii) finding efficient approximations and solvers for an actual implementation. We tackle these challenges in the upcoming paragraphs.

Parameterization. The two unknowns of the problem are the camera trajectory $\mathbf{R}(t)$ and the scene map M . The continuous-time trajectory is approximated using splines that interpolate $\mathbf{R}(t)$ linearly between two neighboring poses $\{\mathbf{R}_i, \mathbf{R}_{i+1}\} \subset \boldsymbol{\alpha}$. Thus the parameters $\boldsymbol{\alpha}$ represent the discrete “control poses” that specify the trajectory [20]. The map M is approximated by a panoramic intensity image (Fig. 2). Since the error terms $(\mathbf{e})_k$ depend directly on the intensity gradient ∇M , we use its N_p pixels $\boldsymbol{\beta}$ as the parameters to optimize in (6).

The computation of the linearized errors (7) in terms of the camera and scene parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is given in the supplementary. We use a Lie Group sensible LM approach [4] to linearize and update camera rotations. The perturbation $\Delta \mathbf{P}$ of the parameters has two parts, corresponding to the camera trajectory $\Delta \mathbf{P}_{\boldsymbol{\alpha}} \in \mathbb{R}^{3N_{\text{poses}}}$ (with 3 DOFs per control pose), and the map pixels $\Delta \mathbf{P}_{\boldsymbol{\beta}} \in \mathbb{R}^{2N_p}$ (with 2 values/channels per map gradient pixel).

Partitioning and Sparsity. For problems of moderate size, with millions of events (N_e), thousands of pixels (N_p), and hundreds of control poses (N_{poses}), it is intractable to store the Jacobian matrix $\mathbf{J}_{\text{op}} \in \mathbb{R}^{N_e \times (3N_{\text{poses}} + 2N_p)}$. Even in sparse format, accessing its non-zero entries is time-consuming because it does not have a simple sparsity pattern. Instead, we directly calculate the matrix in the normal equations in an efficient way (see the supplementary). This matrix only depends on the number of unknowns, which is significantly smaller than the size of \mathbf{J}_{op} , and has a simpler sparsity pattern.

According to the parameterization, the state \mathbf{P} of the BA problem has two parts: the camera rotations and the scene map. This allows us to partition the

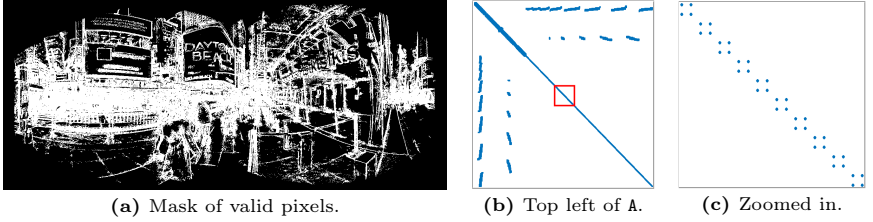


Fig. 3: *Sparsity illustration.* (a) Mask of valid pixels; (b) 1000×1000 block at the top left of matrix \mathbf{A} ; (c) Zoomed-in version of \mathbf{A}_{22} showing its block-diagonal structure.

perturbation vector and the normal equations (8) in blocks:

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^\top & \mathbf{A}_{22} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{P}_\alpha^* \\ \Delta \mathbf{P}_\beta^* \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}, \quad (9)$$

where $\mathbf{A}_{11} \doteq \mathbf{J}_{\text{op},\alpha}^\top \mathbf{J}_{\text{op},\alpha}$ only depends on the derivatives w.r.t. the camera poses, $\mathbf{A}_{22} \doteq \mathbf{J}_{\text{op},\beta}^\top \mathbf{J}_{\text{op},\beta}$ only depends on the derivatives w.r.t. the scene map, and $\mathbf{A}_{12} \doteq \mathbf{J}_{\text{op},\alpha}^\top \mathbf{J}_{\text{op},\beta}$. There is a large size difference: the size of \mathbf{A}_{11} (poses) is significantly smaller than that of \mathbf{A}_{22} (map pixels), as shown in Fig. 3b. This fact can be leveraged when using well-known tools for solving block-partitioned systems.

Additionally, we can exploit sparsity to implement an efficient LM solver for this problem, as follows. Due to the sparsity of event data, only a portion of map points is observed. We select sufficiently measured map points (e.g., receiving more than five events) as “valid pixels” in the optimization, as shown in Fig. 3a. Furthermore, (6) states that each error term $(\mathbf{e})_k$ only depends on the gradient at one map point, which leads to a block-diagonal structure of \mathbf{A}_{22} (with blocks of size 2×2), as depicted in Fig. 3c. This makes \mathbf{A}_{22} easy to invert. We can leverage this property, together with the block-partitioning structure of the normal equations (9) to solve them efficiently via the Schur complement [4].

Map Regularization (Loss). The fact that each error term $(\mathbf{e})_k$ only depends on the gradient at one map pixel is beneficial for speed, but it causes instabilities: during optimization the values of β at some pixels may grow rapidly, suppressing the update of other pixels. To mitigate this, we add a map prior to (5), so that map pixels evolve with regularization, yielding the objective:

$$\min_{\{\mathbf{R}_i\}, \nabla M} \|\mathbf{e}(\{\mathbf{R}_i\}, \nabla M)\|^2 + \eta \|\nabla M\|^2, \quad (10)$$

where $\{\mathbf{R}_i\} \equiv \alpha$ are the control poses of the camera trajectory, and $\eta > 0$ is the weight of the L^2 regularizer $\|\nabla M\|^2 \equiv \|\beta\|^2$, which encourages smoothness of the estimated map. Consequently, the normal equations of (10) become:

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & 0 \\ \mathbf{A}_{12}^\top & \mathbf{A}_{22} + \eta \mathbf{1} & 0 \\ 0 & 0 & \eta \mathbf{1} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{P}_\alpha^* \\ \Delta \mathbf{P}_{\beta_1}^* \\ \Delta \mathbf{P}_{\beta_2}^* \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 - \eta \nabla M_{\text{op},\beta_1} \\ -\eta \nabla M_{\text{op},\beta_2} \end{pmatrix}, \quad (11)$$

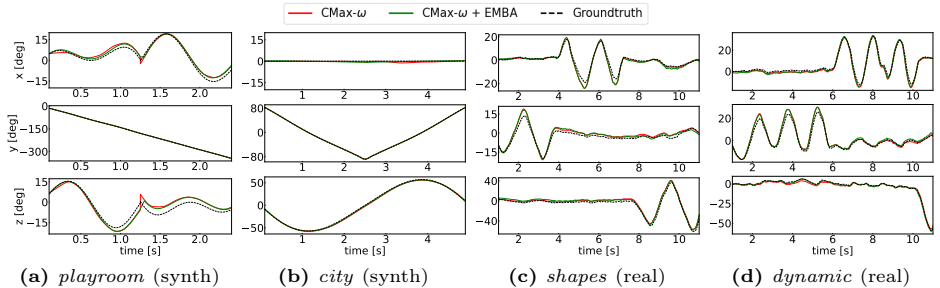


Fig. 4: Camera trajectory degrees-of-freedom (DOFs) before (“CMax- ω ”) and after (“CMax- ω +EMBA”) refinement, for some synthetic and real sequences from [20, 30].

where $\mathbf{1}$ is identity matrix, and we distinguish “valid” and “invalid” pixels using β_1 and β_2 , respectively. Eq. (11) says that “invalid” pixels are updated only using the L^2 regularization, which just sets their gradients to zero. For valid pixels, the L^2 regularization adds a scaled identity matrix to \mathbf{A}_{22} , which does not spoil its block-diagonal structure. Hence, it is still cheap to solve (11).

Poisson Reconstruction. Having obtained the optimized gradient map β by solving the above NLLS problem, we can recover the corresponding intensity map M by solving the well-know Poisson’s equation [1, 24]: $\nabla^2 M = \frac{\partial g_x}{\partial x} + \frac{\partial g_y}{\partial y}$, where g_x and g_y are the two channels of image $\beta \equiv \nabla M$.

4 Experiments

4.1 Experimental Setup

Datasets. We test EMBA on publicly available data: six synthetic sequences from [20] and four real-world sequences from [30]. All sequences contain events, frames (not used), IMU data (not used) and groundtruth (GT) poses.

The synthetic sequences were obtained with a simulator [33], with input panoramas from the Internet. The panoramas covered indoor, outdoor, day-light, night, human-made and natural scenarios, with varying resolution, from 2K (*playroom*), 4K (*bicycle*), 6K (*city* and *street*), to 7K (*town* and *bay*). *playroom* was created with a DVS128 camera model (128×128 px) and a duration of 2.5s, while the other five sequences were created with a DAVIS240C camera model (240×180 px) and a duration of 5 s.

The Event Camera Dataset (ECD) [30] contains four dominantly-rotational motion sequences (*shapes*, *poster*, *boxes* and *dynamic*) that have been commonly used for benchmarking [14–16, 18, 20, 25, 31, 32, 35]. They feature indoor scenes with various amounts of texture complexity and motion. A motion capture system (mocap) outputs accurate GT poses at 200 Hz. We use the ECD data from 1 to 11 s for evaluation, where the camera translation is small.

Initialization. To obtain camera rotations and gradient maps to initialize EMBA, we first run the four front-end methods (EKF-SMT [23], RTPT [35],

Table 2: Absolute rotation RMSE [$^{\circ}$] on synthetic sequences. The best results per sequence are in bold. “-” means the method fails on that sequence, and “N/A” indicates that EMBA is not applicable because the corresponding front-end failed on this sequence. RTPT is not shown because it fails on all sequences.

Sequence	playroom		bicycle		city		street		town		bay	
	before	after	before	after	before	after	before	after	before	after	before	after
EKF-SMT	5.86	6.09	1.47	1.18	1.69	1.68	3.44	3.46	4.32	4.40	2.50	2.41
CMax-GAE	4.63	4.42	1.65	1.50	-	N/A	-	N/A	4.66	4.53	-	N/A
CMax- ω	3.22	2.86	1.69	0.92	1.53	0.97	0.97	0.74	1.91	0.86	1.80	1.41

Table 3: Squared photometric error [$\cdot 10^6$] on synthetic data.

Sequence	playroom		bicycle		city		street		town		bay	
	before	after	before	after	before	after	before	after	before	after	before	after
EKF-SMT	0.35	0.23	0.52	0.30	2.62	2.13	1.82	1.52	1.88	1.51	2.26	1.96
CMax-GAE	0.35	0.19	0.53	0.31	-	N/A	-	N/A	1.90	1.54	-	N/A
CMax- ω	0.33	0.15	0.55	0.30	2.71	1.98	1.90	1.34	1.92	1.43	2.30	1.83

CMax-GAE [25] and CMax- ω [16]) on all sequences. Then we feed these front-end rotations and the event data into the mapping module of EKF-SMT to obtain the initial maps (e.g., top row of Fig. 2). The front-end rotations are interpolated at 1 kHz and aligned to the GT ones at $t = t_0$ ($t_0 = 0.1$ s for synthetic data and $t_0 = 1$ s for real data) before they are used to obtain initial gradient maps and bootstrap EMBA. Unless otherwise specified, the map size is set to 1024×512 px and the control pose frequency f is set to 20 Hz.

Evaluation Metrics. We evaluate EMBA using two metrics:

Absolute Rotation Error (ARE). The ARE measures the accuracy of the estimated camera rotations. At timestamp t_k , the rotation error between the estimated rotation \mathbf{R}_k and the corresponding GT rotation \mathbf{R}'_k (obtained by linear interpolation), is given by the angle of their difference $\Delta \mathbf{R}_k = \mathbf{R}'_k^T \mathbf{R}_k$ [4]. Since each front-end method outputs rotations at a different rate, we calculate the errors at such timestamps and aggregate them using the Root Mean Square (RMS). The refined rotations share the same control pose timestamps (regardless of the front-ends), hence errors are calculated on them.

Photometric Error (PhE). The PhE, defined by (5), measures the goodness of fit between the data and the estimated variables in the model, and is the most straightforward criterion to assess the effect of BA algorithms.

4.2 Experiments on Synthetic Data

Figures 4a and 4b compare the initial and refined CMax- ω rotations on *playroom* and *city*. The refined orientations using EMBA agree better with the GT than the initial ones. This is further elaborated in Tab. 2, using all front-ends: the errors decrease on almost all synthetic sequences, which is most salient when



Fig. 5: *EMBA results on synthetic data.* Panoramic maps have 2048×1024 px. Initial camera rotations are obtained using CMax- ω [16].

initialized by CMax- ω . For *city*, the rotation RMSE of the CMax- ω trajectory is reduced from 1.53° to 0.97° , and that of *town* decreases from 1.91° to 0.86° .

While the plots in Fig. 4 show small differences between the DOF curves, and the numbers in Tab. 2 also report apparently small differences (of $\approx 1^\circ$ RMSE), the improvement effect of EMBA is most noticeable in the photometric error PhE (Tab. 3) and the visual quality of the maps (Figs. 2 and 5). In all trials, the PhE values are significantly reduced (Tab. 3); the maximal relative decrease reaches 54.5% (when refining the CMax- ω rotations on *playroom*).

Moreover, EMBA is also able to refine higher resolution maps: Fig. 5 compares initial and refined maps of 2048×1024 px size. The large improvements of EMBA refinement are visually obvious: blurred regions become sharper, and subtle textures hidden at initialization are revealed, such as the wheels in *bicycle*, the billboards in *city* and the windows and tree leaves in *town*.

In short, EMBA achieves a compelling refinement on synthetic data in terms of rotation accuracy (Tab. 2), map quality (Fig. 5) and photometric error (Tab. 3).

4.3 Experiments on Real Data

The main difficulty of real-world evaluation lies in finding real data that is compatible with the purely rotational motion assumption of the problem [20]. Real-world sequences from established datasets [30] are recorded hand-held, hence they contain some translational motion (see top row of Fig. 6). However, such a translation cannot be removed from the input events. Hence, by design, all rotational motion estimation methods explain the translation in the data using only rotational DOFs. If the translation is non-negligible, comparing the rotations that explain additional DOFs to the rotational component of the GT provided by a 6-DOF motion-capture system [30] can be misleading. Therefore, in the context of photometric BA, the PhE becomes a sensible figure of merit.

Table 4: Results on real data. Top: absolute rotation error (ARE) [$^{\circ}$], in RMSE form. Bottom: squared PhE [$\cdot 10^5$]. EKF-SMT is not shown since it fails on all sequences.

	Sequence	shapes		poster		boxes		dynamic	
		before	after	before	after	before	after	before	after
ARE	RTPT	2.19	2.85	3.80	3.96	1.74	2.32	2.00	2.29
	CMax-GAE	2.51	2.69	3.63	4.09	2.02	2.40	1.70	2.00
	CMax- ω	4.11	4.44	4.07	4.20	3.22	2.87	3.13	2.79
PhE	RTPT	0.68	0.37	4.69	2.58	4.46	2.30	3.29	2.24
	CMax-GAE	0.61	0.38	5.03	3.07	4.52	2.93	3.16	2.39
	CMax- ω	0.58	0.36	4.37	2.58	3.92	2.25	3.05	2.13

Plots 4c and 4d compare the CMax- ω rotations before and after refinement on *shapes* and *dynamic*; the differences are small at this scale. The top part of Tab. 4 reports the errors using all front-ends. The ARE slightly decreases in some trials while it increases in others; there are no big differences between initial and refined trajectory errors because the estimated camera rotation contains compensation for the translational component. The benefits of EMBA are demonstrated in terms of the PhE (bottom part of Tab. 4) and the maps (Fig. 6). EMBA considerably reduces the PhE on real-world data (Tab. 4), around 30% to 50% reduction. Visually, Fig. 6 shows that a remarkable improvement is attained after refinement. Just for comparison, we fed the GT rotations from the mocap into the mapping part of EKF-SMT, and displayed the reconstructed maps in the top row in Fig. 6, which are blurred due to the presence of translation. In the EMBA-refined maps (bottom row), the fine textures on the stones in *poster* are revealed, and the HDR lights in the roof in *dynamic* are also recovered clearly.

In summary, although the real-world evaluation presents difficulties, the effectiveness of EMBA is still proved by the PhE criterion and the map quality.

4.4 Relationship with CMax-SLAM

On the topic of event-based rotational bundle adjustment, the closest relevant work is CMax-SLAM [20]. This section clarifies the differences between EMBA and CMax-SLAM, and demonstrates the potential of their combination.

First of all, they have different objectives and produce different types of map. The objective of CMax-SLAM is to find the camera rotations that maximize the contrast of the panoramic IWE. Hence, the optimization only involves the camera rotations; the scene map is obtained as a secondary result and it is an edge map (Fig. 7a). The problem is well-posed, not suffering from “event collapse” [37–39]. Conversely, EMBA aims at minimizing the event-based photometric error by refining both camera rotations and an intensity panorama. The intensity map is explicitly modeled as a problem unknown (i.e., it is not a by-product).

In terms of mode of operation, CMax-SLAM works in a sliding-window manner, whereas EMBA processes all events in batch. A sliding window means that rotations far away in time are not refined; hence if CMax-SLAM runs for a

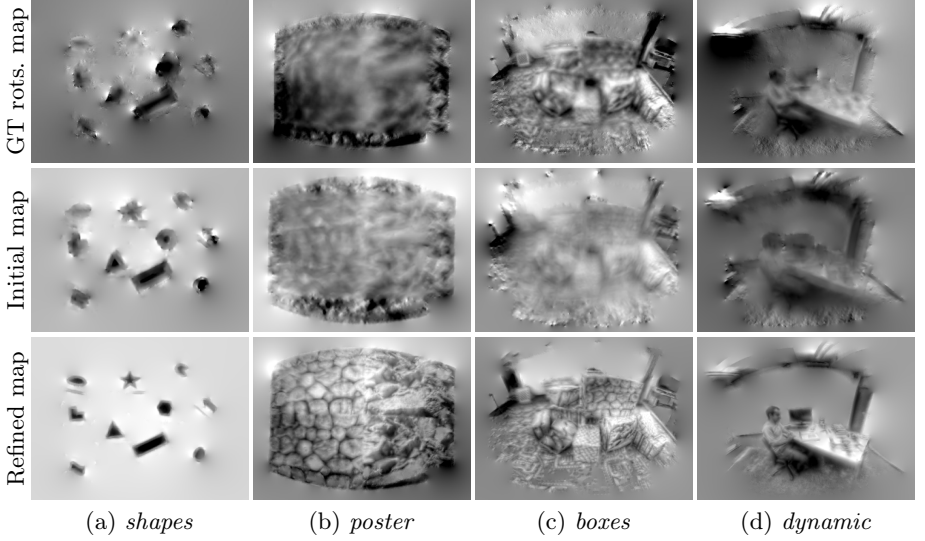


Fig. 6: *EMBA results on real-world data from [30].* The maps in the top two rows are obtained using the mapping module of SMT [24], by feeding the GT camera rotations or the rotations estimated using CMax- ω , respectively. The refined maps are produced with our method. These are central crops from 1024×512 px panoramic maps.

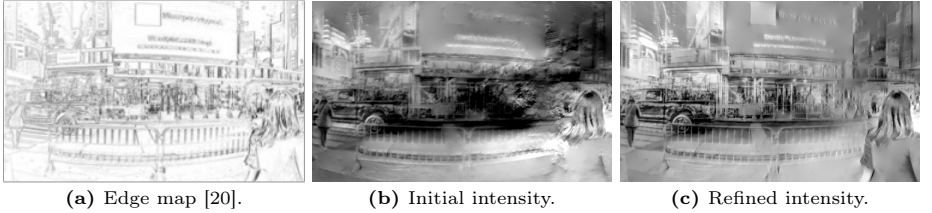


Fig. 7: Results of initializing EMBA with CMax-SLAM [20] (*street scene*).

very long time interval, some events might align to wrong edges, which does not happen in EMBA. Last but not least, both methods are actually complementary: CMax-SLAM can be used to initialize EMBA and get a clean photometric map of the scene, as shown in Fig. 7. Here, the ARE decreases from 0.470° (see Tab. II in [20]) to 0.377° . Hence, EMBA can further refine the rotations from CMax-SLAM while jointly reconstructing a precise intensity panorama.

4.5 Experiments with VGA and HD event cameras

An immediate application of EMBA is panoramic imaging (mosaicing), in particular using the latest event cameras, which produce massive event rates due to their high spatial resolution. To this end, we show results on sequences with a DVXplorer (VGA, 640×480 px [40]) and a Prophesee EVK4 (HD, 1280×720

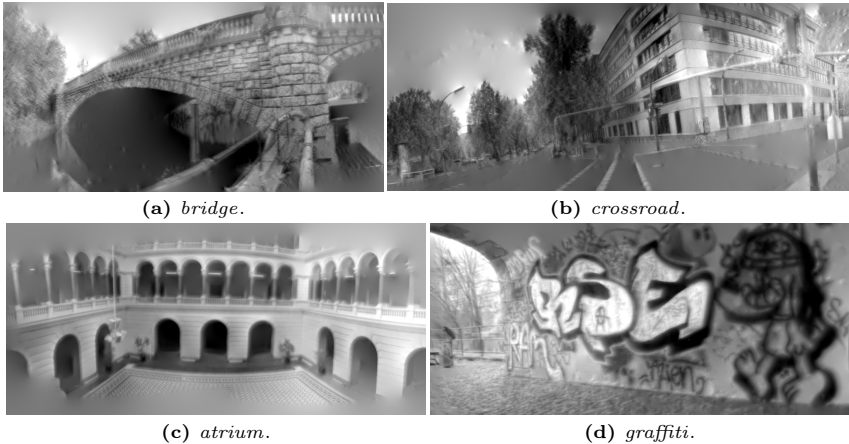


Fig. 8: *Panoramas obtained from scratch.* (a) and (b): DVXplorer data. (c) and (d): Prophesee’s EVK4 (1 Mpixel camera) data. Crops from 4K panoramic maps.

px [11]). For the DVXplorer, which is equipped with an IMU, EMBA is initialized by IMU angular velocity dead-reckoning [4]. For the Prophesee EVK4, which does not have an IMU, we feed the events into [16, 20] to provide initial camera orientations. EMBA recovers the gradient maps from scratch while refining the camera motion parameters. The output panoramas in Fig. 8 have high quality, demonstrating the capabilities of EMBA to handle sequences in the wild.

4.6 Complexity Analysis and Runtime

EMBA has three main steps. First, evaluating the objective function and its derivatives, whose complexity is $O(N_e)$. Second, forming the normal equations, whose complexity is also $O(N_e)$. Third, solving the (LM-augmented) normal equations, whose main complexity lies in working with A_{22} . Due to the block-diagonal structure, the cost of inverting A_{22} is linear with the number of blocks, i.e., $O(N_p)$, where N_p is the number of valid pixels. Overall, EMBA is lightweight and efficient, compared to the other event-based algorithms.

To support the above statements, a runtime evaluation is carried out. Table 5 reports the average runtime of each step for different scenes (e.g., texture complexity), on a standard laptop (Intel Core i7-1165G7 CPU @ 2.80GHz). The most expensive step is evaluating the objective function and its derivatives. Obviously, the runtime of EMBA increases with N_e . Leveraging the block-diagonal sparsity, the cost of solving the normal equations using the Schur complement increases slowly as the texture complexity grows (order: *shapes* < *dynamic* < *boxes* < *poster*). For comparison, we also implement EMBA with Eigen’s [19] built-in conjugate gradient (CG) solver, which does not directly exploit sparsity. We find that the Schur solver is even faster than the CG solver when N_p is large.

Table 5: Runtime evaluation of the three main steps of EMBA [s].

ECD sequence	shapes	poster	boxes	dynamic
Obj. func. evaluation	1.114	8.873	7.436	5.837
Forming Normal Eqs.	0.300	2.366	2.106	1.574
Solving Normal Eqs. (Schur)	0.429	2.013	2.006	1.656
Solving Normal Eqs. (CG)	0.267	3.127	3.561	2.056
N_p (valid pixels)	6913	50738	49357	41313
N_e (number of events)	1.78M	12.59M	10.76M	8.80M

4.7 Limitations

Event cameras rely on scene texture to produce data. Too little texture usually leads to tracking failure (EMBA initialization failure), while high texture triggers too many events, which slows down the algorithm in spite of the linear complexity of EMBA. This limitation, shared by most event-based algorithms, might be overcome by adapting the camera’s C value and/or downsampling events.

All surveyed event-based rotational SLAM methods assume static scenarios and brightness constancy. Events triggered by moving objects and flickering lights may cause inaccuracy or failure if they are plentiful. Also, the linearization in (2) due to Taylor’s approximation in the LEGM model [12] is another source of inaccuracies. However, this was chosen because it endowed matrix \mathbf{A}_{22} in the normal equations with a highly beneficial block-diagonal sparsity pattern.

The Levenberg-Marquardt method has its limitations, e.g., local convergence. EMBA may get stuck in local minima of the very large search space if the initialization is not good. This is also a problem in BA for frame-based cameras.

5 Conclusion

We have introduced EMBA, an event-only mosaicing bundle adjustment approach to jointly refine the orientations of a rotating camera and the panoramic gradient map of the scene. We have leveraged the LEGM to formulate the BA problem as a regularized NLLS optimization with a beneficial sparsity pattern so that it can be efficiently solved by exploiting well-developed tools in BA, such as the LM method. To the best of our knowledge, no previous work has constructed and utilized such a useful sparsity for event-based BA without converting events into image-like representations. Through a comprehensive evaluation, the proposed method achieves remarkable results in terms of photometric error (50% decrease), camera poses and map quality. In addition, we have demonstrated the application of EMBA to mosaicing with high-resolution event cameras, of relevance for smartphone applications, even without map initialization. We release the code and hope that our work helps bring maturity to event-based SLAM and related applications.

Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC 2002/1 “Science of Intelligence” – project number 390523135.

References

1. Agrawal, A., Raskar, R., Chellappa, R.: What is the range of surface reconstructions from a gradient field? In: Eur. Conf. Comput. Vis. (ECCV). pp. 578–591 (2006). https://doi.org/10.1007/11744023_45
2. Alismail, H., Browning, B., Lucey, S.: Photometric bundle adjustment for vision-based SLAM. In: Asian Conf. Comput. Vis. (ACCV). pp. 324–341 (2016). https://doi.org/10.1007/978-3-319-54190-7_20
3. Bagchi, S., Chin, T.J.: Event-based star tracking via multiresolution progressive hough transforms. In: IEEE Winter Conf. Appl. Comput. Vis. (WACV). pp. 2132–2141 (2020). <https://doi.org/10.1109/WACV45572.2020.9093309>
4. Barfoot, T.D.: State Estimation for Robotics - A Matrix Lie Group Approach. Cambridge University Press (2015). <https://doi.org/10.1017/9781316671528>
5. Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vis.* **74**(1), 59–73 (Dec 2006). <https://doi.org/10.1007/s11263-006-0002-3>
6. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I.D., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **32**(6), 1309–1332 (2016). <https://doi.org/10.1109/TRO.2016.2624754>
7. Chin, T.J., Bagchi, S., Eriksson, A.P., van Schaik, A.: Star tracking using an event camera. In: IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW). pp. 1646–1655 (2019). <https://doi.org/10.1109/CVPRW.2019.00208>
8. Cook, M., Gugelmann, L., Jug, F., Krautz, C., Steger, A.: Interacting maps for fast visual interpretation. In: Int. Joint Conf. Neural Netw. (IJCNN). pp. 770–776 (2011). <https://doi.org/10.1109/IJCNN.2011.6033299>
9. Engel, J., Koltun, V., Cremers, D.: Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(3), 611–625 (Mar 2018). <https://doi.org/10.1109/TPAMI.2017.2658577>
10. Engel, J., et al.: Project Aria: A new tool for egocentric multi-modal ai research (2023)
11. Finateu, T., Niwa, A., Matolin, D., Tsuchimoto, K., Mascheroni, A., Reynaud, E., Mostafalu, P., Brady, F., Chotard, L., LeGoff, F., Takahashi, H., Wakabayashi, H., Oike, Y., Posch, C.: A 1280x720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μ m pixels, 1.066Geps readout, programmable event-rate controller and compressive data-formatting pipeline. In: IEEE Int. Solid-State Circuits Conf. (ISSCC). pp. 112–114 (2020). <https://doi.org/10.1109/ISSCC19947.2020.9063149>
12. Gallego, G., Delbruck, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A., Conradt, J., Daniilidis, K., Scaramuzza, D.: Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(1), 154–180 (2022). <https://doi.org/10.1109/TPAMI.2020.3008413>

13. Gallego, G., Forster, C., Mueggler, E., Scaramuzza, D.: Event-based camera pose tracking using a generative event model (2015), arXiv:1510.01972
14. Gallego, G., Gehrig, M., Scaramuzza, D.: Focus is all you need: Loss functions for event-based vision. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 12272–12281 (2019). <https://doi.org/10.1109/CVPR.2019.01256>
15. Gallego, G., Rebecq, H., Scaramuzza, D.: A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 3867–3876 (2018). <https://doi.org/10.1109/CVPR.2018.00407>
16. Gallego, G., Scaramuzza, D.: Accurate angular velocity estimation with an event camera. IEEE Robot. Autom. Lett. **2**(2), 632–639 (2017). <https://doi.org/10.1109/LRA.2016.2647639>
17. Gehrig, D., Rebecq, H., Gallego, G., Scaramuzza, D.: EKLT: Asynchronous photometric feature tracking using events and frames. Int. J. Comput. Vis. **128**, 601–618 (2020). <https://doi.org/10.1007/s11263-019-01209-w>
18. Gu, C., Learned-Miller, E., Sheldon, D., Gallego, G., Bideau, P.: The spatio-temporal Poisson point process: A simple model for the alignment of event camera data. In: Int. Conf. Comput. Vis. (ICCV). pp. 13495–13504 (2021). <https://doi.org/10.1109/ICCV48922.2021.01324>
19. Guennebaud, G., Jacob, B., et al.: Eigen v3. <http://eigen.tuxfamily.org> (2010)
20. Guo, S., Gallego, G.: CMax-SLAM: Event-based rotational-motion bundle adjustment and SLAM system using contrast maximization. IEEE Trans. Robot. **40**, 2442–2461 (2024). <https://doi.org/10.1109/TRO.2024.3378443>
21. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2003). <https://doi.org/10.1017/CBO9780511811685>, 2nd Edition
22. Hidalgo-Carrió, J., Gallego, G., Scaramuzza, D.: Event-aided direct sparse odometry. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 5781–5790 (Jun 2022). <https://doi.org/10.1109/CVPR52688.2022.00569>
23. Kim, H.: Real-time visual SLAM with an event camera. Ph.D. thesis, Imperial College London (2018)
24. Kim, H., Handa, A., Benosman, R., Ieng, S.H., Davison, A.J.: Simultaneous mosaicing and tracking with an event camera. In: British Mach. Vis. Conf. (BMVC) (2014). <https://doi.org/10.5244/C.28.26>
25. Kim, H., Kim, H.J.: Real-time rotational motion estimation with contrast maximization over globally aligned events. IEEE Robot. Autom. Lett. **6**(3), 6016–6023 (2021). <https://doi.org/10.1109/LRA.2021.3088793>
26. Klenk, S.: Photometric Bundle Adjustment for Globally Consistent Mapping. Master’s thesis, TU Munich (2020)
27. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P.: Keyframe-based visual–inertial odometry using nonlinear optimization. Int. J. Robot. Research **34**(3), 314–334 (2015). <https://doi.org/10.1177/0278364914554813>
28. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128×128 120 dB $15 \mu\text{s}$ latency asynchronous temporal contrast vision sensor. IEEE J. Solid-State Circuits **43**(2), 566–576 (2008). <https://doi.org/10.1109/JSSC.2007.914337>
29. Mostafavi I, S., Wang, L., Yoon, K.J.Y.: Learning to reconstruct HDR images from events, with applications to depth and flow prediction. Int. J. Comput. Vis. **129**(4), 900–920 (Apr 2021). <https://doi.org/10.1007/s11263-020-01410-2>
30. Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., Scaramuzza, D.: The event-camera dataset and simulator: Event-based data for pose estimation, visual odom-

- etry, and SLAM. *Int. J. Robot. Research* **36**(2), 142–149 (2017). <https://doi.org/10.1177/0278364917691115>
31. Nunes, U.M., Demiris, Y.: Robust event-based vision model estimation by dispersion minimisation. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(12), 9561–9573 (2022). <https://doi.org/10.1109/TPAMI.2021.3130049>
 32. Peng, X., Gao, L., Wang, Y., Kneip, L.: Globally-optimal contrast maximisation for event cameras. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(7), 3479–3495 (2022). <https://doi.org/10.1109/TPAMI.2021.3053243>
 33. Rebecq, H., Gehrig, D., Scaramuzza, D.: ESIM: an open event camera simulator. In: *Conf. on Robotics Learning (CoRL)*. *Proc. Machine Learning Research*, vol. 87, pp. 969–982. PMLR (2018)
 34. Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D.: High speed and high dynamic range video with an event camera. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(6), 1964–1980 (2021). <https://doi.org/10.1109/TPAMI.2019.2963386>
 35. Reinbacher, C., Munda, G., Pock, T.: Real-time panoramic tracking for event cameras. In: *IEEE Int. Conf. Comput. Photography (ICCP)*. pp. 1–9 (2017). <https://doi.org/10.1109/ICCPHOT.2017.7951488>
 36. Rosinol Vidal, A., Rebecq, H., Horstschaefer, T., Scaramuzza, D.: Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios. *IEEE Robot. Autom. Lett.* **3**(2), 994–1001 (Apr 2018). <https://doi.org/10.1109/LRA.2018.2793357>
 37. Shiba, S., Aoki, Y., Gallego, G.: Event collapse in contrast maximization frameworks. *Sensors* **22**(14), 1–20 (2022). <https://doi.org/10.3390/s22145190>
 38. Shiba, S., Aoki, Y., Gallego, G.: A fast geometric regularizer to mitigate event collapse in the contrast maximization framework. *Adv. Intell. Syst.* p. 2200251 (2022). <https://doi.org/10.1002/aisy.202200251>
 39. Shiba, S., Klose, Y., Aoki, Y., Gallego, G.: Secrets of event-based optical flow, depth, and ego-motion by contrast maximization. *IEEE Trans. Pattern Anal. Mach. Intell.* pp. 1–18 (2024). <https://doi.org/10.1109/TPAMI.2024.3396116>
 40. Son, B., Suh, Y., Kim, S., Jung, H., Kim, J.S., Shin, C., Park, K., Lee, K., Park, J., Woo, J., Roh, Y., Lee, H., Wang, Y., Ovsianikov, I., Ryu, H.: A 640x480 dynamic vision sensor with a 9 μ m pixel and 300Meps address-event representation. In: *IEEE Int. Solid-State Circuits Conf. (ISSCC)*. pp. 66–67 (2017). <https://doi.org/10.1109/ISSCC.2017.7870263>
 41. Stoffregen, T., Scheerlinck, C., Scaramuzza, D., Drummond, T., Barnes, N., Kleeman, L., Mahony, R.: Reducing the sim-to-real gap for event cameras. In: *Eur. Conf. Comput. Vis. (ECCV)*. pp. 534–549 (2020). https://doi.org/https://doi.org/10.1007/978-3-030-58583-9_32
 42. Szeliski, R.: *Computer Vision: Algorithms and Applications*. Texts in Computer Science, Springer (2010)
 43. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment – a modern synthesis. In: Triggs, W., Zisserman, A., Szeliski, R. (eds.) *Vision Algorithms: Theory and Practice*. LNCS, vol. 1883, pp. 298–372. Springer Berlin Heidelberg (2000). https://doi.org/10.1007/3-540-44480-7_21
 44. Wang, J., Gammell, J.D.: Event-based stereo visual odometry with native temporal resolution via continuous-time Gaussian process regression. *IEEE Robot. Autom. Lett.* **8**(10), 6707–6714 (2023). <https://doi.org/10.1109/LRA.2023.3311374>
 45. Weikersdorfer, D., Hoffmann, R., Conradt, J.: Simultaneous localization and mapping for event-based vision systems. In: *Int. Conf. Comput. Vis. Syst. (ICVS)*. pp. 133–142 (2013). https://doi.org/10.1007/978-3-642-39402-7_14

46. Zhang, Z., Yezzi, A., Gallego, G.: Formulating event-based image reconstruction as a linear inverse problem with deep regularization using optical flow. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022). <https://doi.org/10.1109/TPAMI.2022.3230727>
47. Zhou, Y., Gallego, G., Shen, S.: Event-based stereo visual odometry. *IEEE Trans. Robot.* **37**(5), 1433–1450 (2021). <https://doi.org/10.1109/TRO.2021.3062252>