MC-VEO: A Visual-Event Odometry With Accurate 6-DoF Motion Compensation

Jiafeng Huang, Shengjie Zhao, *Senior Member, IEEE*, Tianjun Zhang, and Lin Zhang, *Senior Member, IEEE*

***Abstract*—Nowadays, robust and accurate odometries, as the foundation technology of navigation systems, gains significance in autonomous driving and robotic navigation fields. Although odometries, especially visual odometries (VOs), have made substantial progress, their application scenarios are still limited by the normal cameras' frame rate limitations and their low robustness to motion blur. The event camera, a recently proposed bionic sensor, seeks to tackle these challenges, offering new possibilities for VO solutions to overcome extreme environments. However, integrating event cameras into VO faces challenges like the RGB-event modality gap and the requirement for efficient event processing. To address these research gaps to some extent, we propose a novel visual-event odometry, namely MC-VEO (Motion Compensated Visual-Event Odometry). Specifically, by introducing the temporal Gaussian weight into the standard contrast maximization framework, we propose the first effective 6-DoF motion compensation method that generates deblurred event frames from event data without additional sensors. The generated frames then be aligned with the RGB images through Event Generation Model (EGM) in MC-VEO, so as to overcome the RGB-event modality gap. Additionally, during the optimization of the EGM-based motion estimation algorithm, our decoupling and pre-calculation, matrix representation, and parallel solving further accelerate the per-point processing of events, which enables MC-VEO to show satisfactory speed performance even when facing large amounts of events and candidate points. The superior performance of MC-VEO is evaluated by both qualitative and quantitative experimental results. To ensure that our results are fully reproducible, all the relevant data and codes have been released publicly.**

***Index Terms*—Visual-event odometry, SLAM, contrast maximization, motion compensation, data fusion.**

Manuscript received 15 August 2023; revised 19 September 2023; accepted 6 October 2023. Date of publication 10 October 2023; date of current version 23 February 2024. This work was supported in part by the National Natural Science Foundation of China under Grants 62272343, 61973235, and 61936014, in part by the Shanghai Science and Technology Innovation Plan under Grant 20510760400, in part by the Shuguang Program of Shanghai Education Development Foundation and Shanghai Municipal Education Commission under Grant 21SG23, and in part by the Fundamental Research Funds for the Central Universities. *(Corresponding author: Lin Zhang.)*

The authors are with the School of Software Engineering, Tongji University, Shanghai 201804, China, and also with the Engineering Research Center of Key Software Technologies for Smart City Perception and Planning, Ministry of Education, Shanghai 201804, China (e-mail: 2010195@tongji.edu.cn; shengjiezhao@tongji.edu.cn; 1911036@tongji.edu.cn; cslinzhang@tongji.edu.cn).

https://cslinzhang.github.io/MC-VEO/MC-VEO.html.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2023.3323378.

Digital Object Identifier 10.1109/TIV.2023.3323378

I. Introduction

AS IMPORTANT tools in fields such as autonomous driving [1], robot navigation [2], virtual reality [3] and augmented reality [4], accurate and reliable odometries play important roles in autonomous navigation systems [5], [6]. Among them, Visual Odometry (VO) refers to the technique that estimates the location and motion of a camera by extracting and tracking visual features in consecutive images, which has the advantages of low sensor cost and wide applicability [7]. Unfortunately, due to the poor robustness of traditional optical cameras to motion blur, VO systems often perform poorly in fast motion environments. Unlike traditional cameras that capture intensity images at a fixed rate, event cameras [8], [9], [10] are biomimetic sensors that captures pixel-level brightness changes in real time, providing high temporal resolution and low latency for applications requiring fast and dynamic visual perception. An event camera can measure the asynchronous brightness changes of each pixel, called "event". Such an operating mechanism provides excellent performance characteristics for event cameras, such as low latency, high time resolution (microseconds) and low power consumption (milliwatts instead of watts). Numerous studies have investigated the significant potential of event cameras in addressing visual tracking [11], [12], [13] and other related problems [14], [15], [16], [17] in challenging scenes.

Currently, the use of event cameras to design or to improve odometries has become a research hotspot, but creating an event-based odometry presents significant challenges. The main difficulties lie in two aspects: the motion compensation and the speed-accuracy balance. On the aspect of motion compensation, for most of the event-based odometries, the motion blur due to the camera's self-motion [18] usually occurs during the accumulation of events to create event frames [19], [20], [21], [22], [23]. The differences between the distributions of the raw events and the motion compensated ones are shown in Fig. 1. It can be seen that the event frame formed by directly accumulating the raw events is blurry and has trailing edges, which seriously affects the accuracy of subsequent applications such as event-based VOs. Unfortunately, at present, existing 6-DoF compensation methods for 3D motion most rely on the assistance of additional sensors such as Inertial Measurement Unit (IMU) [24], [25], and methods that only use the event camera are still lacking. On the other aspect, event odometries based on geometric alignment [15], [26], [27], [28] have relatively low localization accuracy, while several recent works [17], [29], [30] have

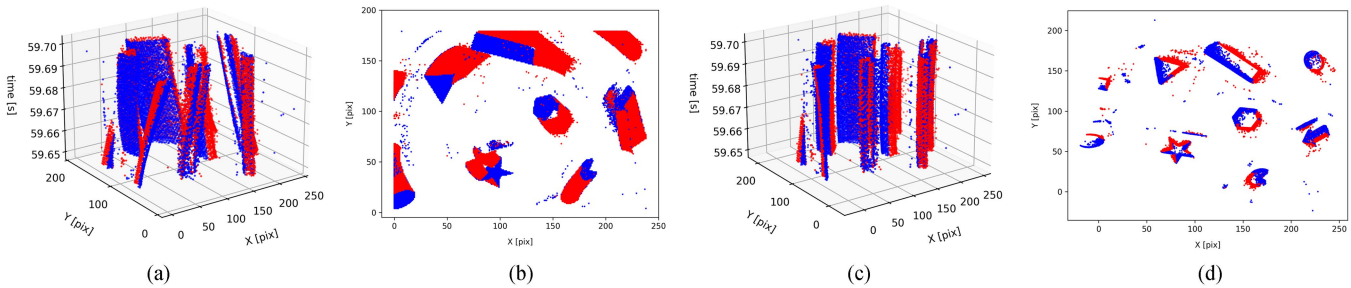


Fig. 1. Differences between the distributions of the raw events and the motion compensated ones (blue: positive events; red: negative events). The event camera rotates around the main optical axis of the lens, and the scene observed is a wall with some simple shapes. (a) Is the distribution of the raw events in the spatio-temporal space. (b) Is the event frame formed by the accumulation of raw events. (c) and (d) are the motion compensated results correspond to (a) and (b), respectively.

demonstrated the use of Event Generation Model (EGM) [8], [31], which describes the photometric relationship between absolute brightness and brightness changes (i.e. events), brings significant performance gain in accuracy [14], [17]. However, the motion estimation algorithm using EGM is computationally expensive, resulting in a relatively slow processing speed for related solutions. Currently, there are no solutions that can strike a good balance between accuracy and speed.

As an attempt to fill in the aforementioned research gaps to some extent, a novel visual-event odometry, namely MC-VEO, is proposed in this article. Specifically, the first pure event-based 6-DoF motion compensation method is proposed to address the problem of motion blur in event frames generation. Unlike the approaches mentioned in [24], [25], our method no longer relies on additional sensors. Furthermore, an efficient EGM-based motion estimation algorithm is proposed, which can align the generated clear event frames with image frames and avoid the problem of high computational complexity [17], [30], resulting in a balance between accuracy and efficiency in MC-VEO. Our contributions are summarized as follows:

1) The first pure event-based 6-DoF motion compensation method based on improved contrast maximization framework is proposed. As far as we know, this is the first 6-DoF motion compensation method that relies solely on events and does not require additional sensors. In addition to event-based odometry, this method can also be applied to a variety of other research fields that require clear event frames, such as event-based feature tracking [26], motion segmentation [32], ego-motion estimation [18], video reconstruction [33], and more.
2) An efficient motion estimation algorithm based on EGM is designed. The optimization process of the objective function is decoupled into two stages, which reduces the computational cost of the overall solution by precalculating the "independent variables". The per-point processing of events is accelerated by our matrix representation and parallel solving method. This strategy significantly shortens the time required for the convergence of the motion estimation algorithm.
3) A novel visual-event odometry, namely MC-VEO, that incorporates our 6-DoF motion compensation method is proposed. MC-VEO exhibits excellent localization accuracy performance, particularly on high-resolution fast-motion sequences. Extensive qualitative and quantitative experiments on multiple benchmark datasets show that our MC-VEO outperforms SOTA event-based or image-based VOs.

The remainder of this article is organized as follows. Section II introduces the related work. Section III presents the overall framework of our odometry. Details for evaluation on publicly available datasets are presented in Section IV. Section V concludes the article.

II. Related Work

Motion compensation is usually an important component of raw data processing and pose estimation in event-based visual odometry, and the contrast maximization framework provides a theoretical basis for event-based motion compensation. In this section, we review the studies related to motion compensation, contrast maximization and event-based visual odometry in three subsections, respectively.

A. Motion Compensation

As discussed in Section I, there are two main types of motion compensation algorithms: IMU-assisted ones and event-only ones. The former ones are usually used in event-based Visual-Inertial Odometry (VIO). For example, in [24], [25], the IMU measurement values and their linear interpolation are utilized to correct the coordinates of each event, resulting in the generation of motion compensated event frames. This type of method relies on the data from accurately calibrated IMU and cannot be applied in pure event-based visual odometry. The latter ones often employ the contrast maximization framework to estimate and correct the motion of the event camera. For example, in [18], the authors only consider the rotating motion of the event camera and estimate the constant angular velocity by maximizing the variance of the image brightness. In [45], the contrast maximization framework is utilized to estimate the 8-DoF constant motion parameters, considering the homographic motion in the plane scene. The idea of contrast maximization provides a theoretical foundation for event-based motion compensation algorithms. A more accurate contrast maximization framework

TABLE I
OVERVIEW OF EXISTING EVENT-BASED VISUAL ODOMETRIES

Reference	Year	DoF	Track	Depth	Scene	Sensor	Additional requirements
Cook [34]	2011	3-DoF	yes	no	natural scenes	event camera	rotation only
Weikersdorfer [35]	2013	3-DoF	yes	no	B&W line patterns	event camera	plane motion
Censi [36]	2014	6-DoF	yes	no	B&W line patterns	event camera, RGB-D sensor	-
Weikersdorfer [37]	2014	6-DoF	yes	yes	natural scenes	event camera&RGB-D sensor	-
Mueggler [38]	2014	6-DoF	yes	no	B&W line patterns	event camera	3D map of lines
Kueng [26]	2016	6-DoF	yes	yes	natural scenes	event camera, frame-based camera	-
Kim [14]	2016	6-DoF	yes	yes	natural scenes	event camera	-
Gallego [18]	2017	3-DoF	yes	no	natural scenes	event camera	rotation only
Reinbacher [39]	2017	3-DoF	yes	no	natural scenes	event camera	rotation only
Rebecq [15]	2017	6-DoF	yes	yes	natural scenes	event camera	-
Kim [12]	2018	3-DoF	yes	no	natural scenes	event camera	rotation only
Gallego [13]	2018	6-DoF	yes	no	natural scenes	event camera, RGB-D sensor	3D map of the scene
Rebecq [40]	2018	6-DoF	no	yes	natural scenes	event camera	camera pose
Zhu [41]	2019	3-DoF	yes	yes	in-vehicle scenes	event camera	plane motion
Gehrig [42]	2020	3-DoF	yes	no	synthetic scenes	event camera	rotation only
Kim [11]	2021	3-DoF	yes	no	natural scenes	event camera	rotation only
Liu [43]	2021	3-DoF	yes	no	natural scenes	event camera	rotation only
Wang [44]	2022	3-DoF	yes	yes	natural scenes	event camera	plane motion
Zuo [16]	2022	6-DoF	yes	yes	natural scenes	event camera, RGB-D sensor	-
Hidalgo-Carrió [17]	2022	6-DoF	yes	yes	natural scenes	event camera, frame-based camera	-

and motion model can enhance the effectiveness of motion compensation and further improve the accuracy of subsequent algorithms.

B. Contrast Maximization

The idea of contrast maximization is first proposed by Gallego and Scaramuzza [18] when introducing their event camera rotation estimation algorithm. This method helps to accurately estimate the angular velocity of the event camera under the condition of high-speed motion (close to 1,000 degrees/second). The concrete theoretical concept of the contrast maximization is firstly summarized in [45], where the unified framework is also proposed to solve various computer vision problems based on event cameras, such as motion estimation and optical flow tracking. In this framework, a group of events is warped to the image plane along the motion trajectory, and the trajectory is estimated and optimized by evaluating the resulting Image of Warped Events to restore the trajectory most suitable for the original event set. Zhu et al. [28] employ contrast maximization as a loss function to train unsupervised neural networks to estimate optical flow, depth, and self-motion. Stoffregen and Kleeman [46] examine the selection of reward function for contrast maximization, propose the classification of different rewards, and show how to build a more robust reward for noise and aperture uncertainty. Furthermore, Peng et al. [47] apply global contrast maximization to the front-parallel motion estimation of an event camera and derive the global optimal solution of this general non-convex problem. Later, Peng et al. [48] use the branch-and-bound method to derive the recursion upper and lower bounds for six different contrast estimation functions, which are utilized to solve the optimal solution of the global contrast maximization problem.

C. Event-Based Visual Odometry

Event-based Visual Odometry (VO) technology is still in an immature stage, and the number of relevant schemes is limited. Some representative works in this field are summarized in Table I. The table columns indicate following aspects: 1) the year of the proposal; 2) the DoF of the motion; 3) whether tracking is performed; 4) whether depth estimation is performed; 5) the type of applied scenes; 6) the sensor configuration; and 7) any additional constraints or requirements. From an applicability standpoint, early event-based VO works typically have strict limitations on camera motion styles and scene scales. However, recent works have gradually released these limitations and shifted towards general natural 3D scenes, in which the flexible 6-DoF motion is supported.

In 2016, Kim et al. [14] propose the earliest complete event-based visual odometry, which includes three probabilistic filters that predict camera motion, light brightness gradient, and inverse depth of the scene, respectively. This work assumes constant brightness and linear gradient, and achieves simultaneous motion tracking and 3D scene reconstruction, but requires GPU due to the huge computational burden. Rebecq et al. [15] propose EVO, an event-based VO that does not require light brightness reconstruction and can work on a simple CPU. EVO [15] includes two parallel pipelines: the tracking pipeline based on image-to-model alignment and the mapping one for event-based 3D reconstruction. Unfortunately, the localization and mapping accuracy of it are relatively poor due to its unstable event-based bootstrap and feature detection method. Zuo et al. [16] propose DEVO, which generates semi-dense depth maps by warping the corresponding depth values of the extrinsically calibrated depth camera and updates the camera pose through geometric 3D-2D edge alignment. It shows comparable performances to state-of-the-art RGB-D camera-based alternatives in regular conditions. Hidalgo-Carrió et al. [17] propose EDS, the first scheme to use direct method for 6-DoF VO using both event and color information. EDS [17] uses sparse 3D points to predict the brightness increments of pixels through an ordinary optical camera and estimates camera motion by comparing them with event-based brightness increments in error. While its use of EGM results in higher tracking accuracy compared to other schemes, it

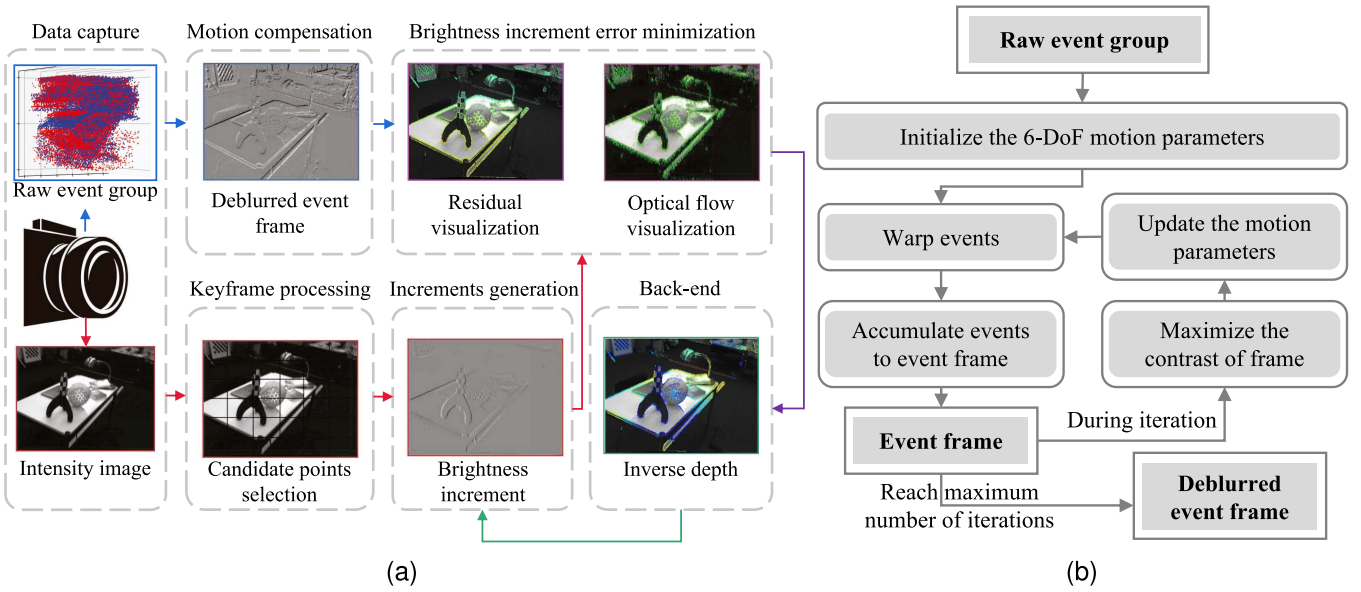


Fig. 2. (a) Is the overall pipeline of our proposed MC-VEO. The events obtained from the event camera are divided into groups, and after motion compensation, clear event frames are formed. The images obtained from the color camera go through keyframe judgment and candidate point selection to predict and form the brightness increments. The event generative model is used to correlate measurements from events and images. The front-end predicts camera motion by minimizing the brightness increment error of both two kinds of measurements. The camera pose and velocities as well as the depth of sparse candidate points are refined by photometric bundle adjustment at the back-end to sustain the VO system's good performance. (b) Is the detailed flowchart of the key component (motion compensation module) in MC-VEO.

also results in a larger computational load, making optimization much slower.

III. Pipeline of MC-VEO

This section describes the overall pipeline of our MC-VEO, which is summarized in Fig. 2(a). First, we briefly review the working principle of the event camera in Section III-A. Next, we introduce our event-based 6-DoF motion compensation method and the preprocess on image frames in Section III-B and III-C, respectively. Then, how the efficient motion estimation algorithm based on EGM works in the front-end is introduced in Section III-D. Finally, the back-end of the MC-VEO system is introduced in Section III-E.

A. The Working Principle of Event Camera

The observation of each pixel by the event camera is independent and asynchronous. When the event camera detects that the logarithmic brightness intensity of a pixel $\boldsymbol{u}_k = [x_k, y_k]^T$ changes exceeding a specified amount C (called contrast sensitivity) [8] at timestamp t_k, it will produces an event $e_k \doteq (\boldsymbol{u}_k, t_k, p_k)$:

$$\Delta I\left(\boldsymbol{u}_k, t_k\right) \doteq I\left(\boldsymbol{u}_k, t_k\right) - I\left(\boldsymbol{u}_k, t_k'\right) = p_k C, \tag{1}$$

where I is the logarithmic brightness intensity, the polarity $p_k \in \{+1, -1\}$ represents the sign of the brightness change (Brighten or darken), and t_k' is the timestamp of the last event generated on $\mathbf{u}_k$. It is worth mentioning that the event timestamps t_k usually have a resolution of microseconds. Different from ordinary optical cameras, an event camera does not output images at a constant rate but instead a stream of asynchronous events in spatio-temporal space.

B. Event-Based 6-DoF Motion Compensation

Our proposed pure event 6-DoF motion correction method is implemented based on an improved contrast maximization framework, which is described as follows.

Improved contrast maximization framework: Given the location and timestamp of each event in a group and initialized warping parameters, each event can be warped backward along a point-trajectory into a reference view with a timestamp t_{ref}. Since events are more likely to appear near high-gradient edges, the correct warping parameters can be found by adjusting the event frame in the reference view, which is called the Image of Warped Events (IWE), to form the sharpest possible edge graph. Unlike [18] and [11], we extend the warping function from rotational-only motion to the more general Euclidean transformation.

To demonstrate this approach, let $\mathcal{E} \doteq \{e_k\}_{k=1}^{N_e}$ be a set of events within a time interval $\mathcal{T} \doteq \{t_k\}_{k=1}^{N_e}$, where N_e represents the event group size. We define $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ as the angular and linear velocities of the event group, respectively. The warping function for a single event in such a group is defined as follows:

$$\boldsymbol{w}\left(\boldsymbol{u}_k, \boldsymbol{\omega}, \boldsymbol{\theta}, \delta t_k \,|\, \mathbf{T}\right) = \mathbf{T} \begin{bmatrix} \exp_{\mathfrak{so}(3)}(\hat{\boldsymbol{\omega}} \delta t_k) & \boldsymbol{\theta} \delta t_k \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_k' \\ 1 \end{bmatrix}. \tag{2}$$

Here, δt_k is the time difference between the event timestamp t_k of the event e_k and the reference timestamp t_{ref}, i.e., $\delta t_k = t_k - t_{\text{ref}}$, $\mathbf{T}$ is the Euclidean transformation matrix representing

camera pose in t_{ref}, the hat operator $\hat{\boldsymbol{\omega}} \in \mathbb{R}^{3\times3}$ represents the cross product matrix of $\boldsymbol{\omega}$, $\exp_{\mathfrak{so}(3)}(\cdot)$ refers to exponential mapping from $\mathfrak{so}(3)$ to $SO(3)$, and $\boldsymbol{u}'_k = [x'_k, y'_k, z'_k]^T \in \mathbb{R}^3$ is an inverse-projected point from $\boldsymbol{u}_k$ to the camera coordinate system. To abbreviate the notation, we omitted the back projection operation on the homogeneous coordinates of the event points.

The IWE is generated by accumulating warped events at each discrete pixel position for the time interval $[t_1, t_{N_e}]$:

$$\boldsymbol{I}^{\text{raw}}(\boldsymbol{u}, \boldsymbol{\omega}, \boldsymbol{\theta} \mid \mathbf{T}) = \sum_{k=1}^{N_e} \alpha_k p_k \delta_d \left(\boldsymbol{u} - \boldsymbol{w}\left(\boldsymbol{u}_k, \boldsymbol{\omega}, \boldsymbol{\theta}, \delta t_k \mid \mathbf{T}\right)\right), \tag{3}$$

where the Dirac delta function δ_d selects the appropriate pixel. It is worth mentioning that here we omitted the projection function for projecting events from camera coordinates to image coordinates for brevity. Unlike conventional accumulation methods, we accumulate weighted polarities $\alpha_k p_k \leftarrow p_k$ to obtain IWE, where

$$\alpha_k = \frac{3}{\sqrt{\pi} N_e} \mathrm{e}^{-0.5\left(\frac{t_k - t_{\text{ref}}}{N_e/6}\right)^2}. \tag{4}$$

Based on the assumption that events with closer timestamps have more similar warping velocities, the Gaussian weights emphasize events whose timestamp is close to t_{ref}, resulting in a more accurate IWE compared to the unweighted case ($\alpha_k = 1$). Subsequently, the values of motion parameters $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ can be accurately measured by evaluating the contrast of the improved IWE.

Motion compensation: To compensate the motion blur in the event frame, an accurate IWE is needed. In the front-end of our MC-VEO, we choose to maximize the squared Frobenius norm of the IWE $\boldsymbol{I}^{\text{raw}}$ for the interval $[t_1, t_{N_e}]$ by optimizing the velocities $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$ using the Root Mean Square Propagation (RMS-prop) optimizer:

$$\underset{\boldsymbol{\omega}, \boldsymbol{\theta}}{\text{maximize}} \left\| \boldsymbol{I}^{\text{raw}}(\boldsymbol{\omega}, \boldsymbol{\theta} \mid \mathbf{T}) \right\|^2, \tag{5}$$

where $\|\cdot\|^2$ denotes the squared Frobenius norm. The cost function measures the contrast of the IWE, which is equivalent to the sum of squares reward in [46]. Its Jacobian is computed as follows (6) shown at the bottom of this page. Here, $\nabla \boldsymbol{I}(\boldsymbol{u}_k, \boldsymbol{\omega}, \boldsymbol{\theta})$ is the gradient of brightness $\boldsymbol{I}(\boldsymbol{u}_k, \boldsymbol{\omega}, \boldsymbol{\theta})$ at coordinate $\boldsymbol{u}_k$, f_x and f_y are the focal lengths of the event camera, and $\bar{x}_k$ and $\bar{y}_k$ are the normalized coordinates satisfying $\bar{x}_k = x'_k/z'_k$ and $\bar{y}_k = y'_k/z'_k$. Through our motion compensation process, accurate event frames are generated as pseudo measurements of brightness increments. The pseudocodes of the motion compensation are given in Algorithm 1.

Algorithm 1: Algorithm for Motion Compensation.

Input: The set of events $\mathcal{E} \doteq \{e_k\}_{k=1}^{N_e}$, The intrinsic matrix $\mathbf{K}$, The maximum number of iterations $maxIter$
Output: A contrast maximized event frame $\boldsymbol{I}$

initial $\boldsymbol{\omega} = 0$, $\boldsymbol{\theta} = 0$, $it = 0$, $\nu = 0$
set $t_{\text{ref}} = t_1$, $dr = 0.995$, $lr = 0.05$, $\epsilon = 10^{-8}$
while $it \leq maxIter$ **do**
  for each k$\in [1, N_e]$ **do**
    compute $\boldsymbol{p}_k$ by back projecting $\boldsymbol{u}_k$ of event e_k
    $\delta t_k = t_k - t_{\text{ref}}$
    $\boldsymbol{R} = \exp_{\mathfrak{so}(3)}(\hat{\boldsymbol{\omega}} \delta t_k)$
    $\boldsymbol{t} = \boldsymbol{\theta} \delta t_k$
    $\boldsymbol{p}'_k = \boldsymbol{R}\boldsymbol{p}_k + \boldsymbol{t}$
    compute $\boldsymbol{w}(\boldsymbol{u}_k, \boldsymbol{\omega}, \boldsymbol{\theta}, \delta t_k)$ by projecting $\boldsymbol{p}'_k$
    $\alpha_k = \frac{3}{\sqrt{\pi} N_e} \mathrm{e}^{-0.5(\frac{t_k - t_{\text{ref}}}{N_e/6})^2}$
    compute $\boldsymbol{I}$ by (3)
  end for
  compute $\nabla \boldsymbol{I}$ by using Sobel operation on $\boldsymbol{I}$
  compute Jacobian $\boldsymbol{J}$ by (6)
  $\boldsymbol{G} = \boldsymbol{J}^T \delta t$
  $\nu = dr \times \nu + (1 - dr) \boldsymbol{G}^T \boldsymbol{G}$
  $\begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{\theta} \end{bmatrix} - \frac{lr \boldsymbol{G}}{\sqrt{\nu + \epsilon}}$
  $it = it + 1$
end while
return $\boldsymbol{I}$

C. The Preprocess on Image Frames

For RGB images, the preprocess performed by MC-VEO mainly includes two steps, candidate points selection and keyframe selection.

Candidate points selection: When preprocessing RGB images, in order to reduce the occupation of computational resources while maintaining accuracy in image processing, it is important to carefully select informative pixels from keyframes. This can be achieved by selecting pixels with high gradients, which are typically associated with the edges of the scene within an image. To ensure that these selected pixels are evenly distributed across the image plane, the image is divided into rectangular blocks of a fixed size, and a certain percentage of pixels with the highest brightness gradient within each block are chosen (usually between $10\% - 15\%$ of the total number of pixels in the image). The pixels with the strongest gradients coincide with those where events are triggered, as events correspond to moving edges. As the camera moves, the set of keyframe pixels and event pixels begin to diverge, and the camera motion and the scene depth would be estimated to maintain their correspondence.

$$\frac{d\left\|\boldsymbol{I}(\boldsymbol{\omega}, \boldsymbol{\theta})\right\|^2}{d(\boldsymbol{\omega}, \boldsymbol{\theta})} = \sum_{k=1}^{N_e} 2\boldsymbol{I}(\boldsymbol{u}_k, \boldsymbol{\omega}, \boldsymbol{\theta}) \nabla \boldsymbol{I}^T(\boldsymbol{u}_k, \boldsymbol{\omega}, \boldsymbol{\theta}) \boldsymbol{M} \delta t_k,$$

$$\boldsymbol{M} = \begin{bmatrix} -\bar{x}_k \bar{y}_k \mathrm{f}_x & \left(1 + \bar{x}_k^2\right) \mathrm{f}_x & -\bar{y}_k \mathrm{f}_x & \mathrm{f}_x / z'_k & 0 & -\bar{x}_k \mathrm{f}_x / z'_k \\ -\left(1 + \bar{y}_k^2\right) \mathrm{f}_y & \bar{x}_k \bar{y}_k \mathrm{f}_y & \bar{x}_k \mathrm{f}_y & 0 & \mathrm{f}_y / z'_k & -\bar{y}_k \mathrm{f}_y / z'_k \end{bmatrix}. \tag{6}$$

Keyframe selection: Our VO system generates a keyframe when either the number of selected points decreases by 20%–30% since some points fall out of the field of view (FOV) or the event camera's relative rotation with respect to the keyframe exceeds a predetermined threshold.

After the creation of a new keyframe, we populate its inverse depth estimates using those from past keyframes. The set of selected pixels is first back-projected to the 3D space and then projected onto the new keyframe. For the remaining pixels, we initialize their inverse depth values using nearest neighbors with a k-d tree, which has proven to be a simple and effective approach [17]. In the back-end of our VO system, inverse depth refinement is carried out to improve the accuracy of the estimations.

D. The Efficient Motion Estimation Algorithm Based on EGM

In the front-end of our MC-VEO, we create pseudo measurements from the events through event-based 6-DoF motion compensation introduced in Section III-B. Additionally, we further utilize (8) to build measurements from the preprocessed image frames. Our primary objective in the front-end is to estimate the VO state by minimizing the brightness increment error between these two measurements according to EGM. As shown in Fig. 2(a), The proposed efficient motion estimation algorithm based on EGM is composed of increments generation and brightness increment error minimization.

Event Generation Model: EGM describes the relationship between events and image brightness as follows. Directly accumulating the set of events $\mathcal{E}$ in the period of time interval $\mathcal{T}$ can generate a brightness increment frame $\Delta \boldsymbol{I}(\boldsymbol{u})$:

$$\Delta \boldsymbol{I}(\boldsymbol{u}) = \sum_{t_k \in \mathcal{T}} p_k C \delta_d \left(\boldsymbol{u} - \boldsymbol{u}_k\right), \tag{7}$$

which is very close to the generated IWE, with only the difference in contrast sensitivity C.

From the perspective of intensity images, when the time interval $\Delta t = t_{N_e} - t_1$ is small, Taylor's expansion can be used to approximate the increment in (1). Further substituting the brightness constancy assumption gives that the change in image brightness $\Delta \widetilde{\boldsymbol{I}}$ is caused by brightness gradients $\nabla \widetilde{\boldsymbol{I}}$ moving with velocity $\boldsymbol{v}$ on the image plane [29], [49]:

$$\Delta \widetilde{\boldsymbol{I}}(\boldsymbol{u}) \approx -\nabla \widetilde{\boldsymbol{I}}^T(\boldsymbol{u}) \Delta \boldsymbol{u} = -\nabla \widetilde{\boldsymbol{I}}^T(\boldsymbol{u}) \boldsymbol{v}(\boldsymbol{u}) \Delta t. \tag{8}$$

Brightness increments from image frames: To build the measurements from the image frame, the increments generation component of MC-VEO selects only pixels on the image contours at the keyframes for brightness increments calculation as given in (8). The spatial gradient of the logarithmic normalized brightness of the keyframe $\widetilde{I}$ is computed using the Sobel operator. The 2D image-point velocity $\boldsymbol{v}(\boldsymbol{u})$ in (8) can be expressed in terms of the camera's angular and linear velocities $\boldsymbol{V}$, and the depth $d_{\mathbf{u}}$ of the 3D point with respect to the camera [50]:

$$\boldsymbol{v}(\boldsymbol{u}) = \boldsymbol{J}(\boldsymbol{u}, d_{\boldsymbol{u}}) \boldsymbol{V}. \tag{9}$$

Here, $\boldsymbol{J}(\boldsymbol{u}, d_{\boldsymbol{u}})$ is the 2×6 feature sensitivity matrix, defined as follows (10) shown at the bottom of this page. The definitions of variables in this matrix are consistent with (6). Merging (9) and (8) gives the predicted brightness change as,

$$\Delta \widetilde{\boldsymbol{I}}(\boldsymbol{u}) \approx -\nabla \widetilde{\boldsymbol{I}}^T(\boldsymbol{u}) \boldsymbol{J}(\boldsymbol{u}, d_{\boldsymbol{u}}) \boldsymbol{V} \Delta t. \tag{11}$$

It is worth mentioning that the velocity $\boldsymbol{V}$ are global variables shared by all image pixels $\boldsymbol{u}$ of the same keyframe. As shown in Fig. 2(a), the depth $d_{\boldsymbol{u}}$ is an input to the front-end from the back-end. The initialization process for the depth estimate will be described in Section III-E later.

Acceleration: Unlike [30] and [17], we accelerate the iterative optimization process to minimize computational complexity so as to improve the optimization speed. Firstly, for all sparse keypoints on a single image frame, we make brightness increment prediction during the iteration process as (11), which is the product of the gradient vector $\nabla \widetilde{\boldsymbol{I}}^T(\boldsymbol{u})$, the feature sensitivity matrix $\boldsymbol{J}(\boldsymbol{u}, d_{\boldsymbol{u}})$, the velocity vector $\boldsymbol{V}$ and the time difference Δt. Along with the optimization evolvement, only the velocity $\boldsymbol{V}$ is optimized, while the image brightness gradient values $\nabla \widetilde{\boldsymbol{I}}(\boldsymbol{u})$ and matrix $\boldsymbol{J}$ are set fixed. Thus, the product of $\nabla \widetilde{\boldsymbol{I}}^T(\boldsymbol{u})$ and $\boldsymbol{J}$ of all utilized pixels can be precomputed via the tensor product before the optimization to avoid repetitive calculations.

In addition, in the event part, each event point should be warped with a different δt_k corresponding to each event in (2), which brings a high computational load. For speed up, we describe an accurate warping method in a matrix form that does not require computation of (2) for each event. For k-th event $e_k \doteq (\boldsymbol{u}_k, t_k, p_k)$, $\delta t_k = t_k - t_{\text{ref}}$. In order to compute the warping accurately, we adopt the Rodrigues Formula and the second-order approximation of the warping function can be achieved by the Taylor expansion of trigonometric functions as follows:

$$\begin{aligned} \overrightarrow{\boldsymbol{w}}(\boldsymbol{u}_k, \boldsymbol{\omega}, \boldsymbol{\theta}, \delta t_k \,|\, \mathbf{T}) = \mathbf{R} \Bigg\{ & \cos(|\boldsymbol{\omega}| \delta t_k) \boldsymbol{I} + \sin(|\boldsymbol{\omega}| \delta t_k) \frac{\hat{\boldsymbol{\omega}}}{|\boldsymbol{\omega}|} \\ & + (1 - \cos(|\boldsymbol{\omega}| \delta t_k)) \left(\frac{\hat{\boldsymbol{\omega}}}{|\boldsymbol{\omega}|}^2 + \boldsymbol{I} \right) \Bigg\} \boldsymbol{u}'_k \\ & + \mathbf{t} + \boldsymbol{\theta} \delta t_k \\ \approx \mathbf{R} & \left(\boldsymbol{I} + \delta t_k \hat{\boldsymbol{\omega}} + \frac{1}{2} \delta t_k^2 \hat{\boldsymbol{\omega}}^2 \right) \boldsymbol{u}'_k + \mathbf{t} + \boldsymbol{\theta} \delta t_k, \end{aligned} \tag{12}$$

where $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and the translation vector corresponding to the Euclidean transformation matrix $\mathbf{T}$, respectively, the hat operator $\hat{\boldsymbol{\omega}} \in \mathbb{R}^{3\times3}$ represents the cross product matrix of $\boldsymbol{\omega}$, and $\boldsymbol{u}'_k = [x'_k, y'_k, z'_k]^T \in \mathbb{R}^3$ is an inverse-projected point from $\boldsymbol{u}_k$ to the camera coordinate system. The

$$\boldsymbol{J}(\boldsymbol{u}, d_{\boldsymbol{u}}) = \begin{bmatrix} \bar{x}_k \bar{y}_k & -\left(1 + \bar{x}_k^2\right) & \bar{y}_k & -1/d_{\boldsymbol{u}} & 0 & \bar{x}_k / d_{\boldsymbol{u}} \\ \left(1 + \bar{y}_k^2\right) & -\bar{x}_k \bar{y}_k & -\bar{x}_k & 0 & -1/d_{\boldsymbol{u}} & \bar{y}_k / d_{\boldsymbol{u}} \end{bmatrix}. \tag{10}$$

proposed warping function (12) can be easily implemented in matrix representation based parallel computing, avoiding per-event calculation in (2).

Brightness increment error minimization: Camera tracking by means of minimizing errors in brightness increment is carried out on the latest keyframe. The events are partitioned into groups, and event frames as given in (7) are generated using the motion compensation method. The camera tracking is formulated as a joint optimization problem over the camera motion parameters, including the 6-DoF pose $\boldsymbol{T}$ and its velocity $\boldsymbol{V}$:

$$(\delta \boldsymbol{T}^*, \boldsymbol{V}^*) = \arg\min_{\delta \boldsymbol{T}, \boldsymbol{V}} \left\| \frac{\Delta \widetilde{\boldsymbol{I}}}{\|\Delta \widetilde{\boldsymbol{I}}\|_2} - \frac{\Delta \boldsymbol{I}}{\|\Delta \boldsymbol{I}\|_2} \right\|_\gamma , \tag{13}$$

where the error is the Huber norm γ of the difference between normalized brightness increments ($\Delta \boldsymbol{I}$ from the events and $\Delta \widetilde{\boldsymbol{I}}$ from the keyframe). Norms are computed over a sparse set of pixels, which are the aforementioned selected pixels in the keyframe. The camera pose $\boldsymbol{T}$ is parameterized using a 3-vector and a quaternion, while the velocity $\boldsymbol{V}$ is parameterized using a 6-vector. The Levenberg–Marquardt algorithm is used to minimize (13).

The output of the front-end consists of the coarse estimated camera motion (relative to the keyframe) of each event packet. These outputs are then passed on to the back-end for further nonlinear refinement (as discussed in Section III-E).

E. Back-End

The back-end of our MC-VEO system is responsible for refining the camera poses and 3D structure through photometric bundle adjustment (PBA). The objective function that we aim to minimize is composed of the sum of errors between the pixel intensities of a given keyframe and those of other keyframes in which the same 3D point is visible:

$$\sum_{i \in \boldsymbol{KF}} \sum_{\boldsymbol{u} \in \boldsymbol{P}_i} \sum_{j \in \boldsymbol{KF}_{\Pi(\boldsymbol{u})}} \|\boldsymbol{KF}_i(\boldsymbol{u}) - \boldsymbol{KF}_j(\boldsymbol{u}')\|_\gamma . \tag{14}$$

Here, i runs over all keyframes $\boldsymbol{KF}$, $\boldsymbol{u}$ runs over all selected pixels $\boldsymbol{P}_i$ in keyframe i, and j runs over all keyframes in which point $\boldsymbol{u}$ is visible ($\Pi(\boldsymbol{u})$ refers to the keyframe where the 3D point corresponding to $\boldsymbol{u}$ is visible). $\boldsymbol{u}'$ is the corresponding point to $\boldsymbol{u}$ on the j-th keyframe. The Huber norm γ is used to downweight the influence of outliers, by which errors exceeding a certain threshold are discarded. Errors are computed within an 8-pixel patch centered on each image point, under the assumption of a uniform depth estimate for all pixels within the patch. To achieve a balance between accuracy and computational efficiency, we employ a sliding window estimator that keeps seven keyframes at a time [51]. On average, the back-end utilizes between 2,000 and 8,000 points. Motivated by [51], a coarse depth initializer is also utilized for system bootstrap.

IV. Experiments

A. Methods for Comparison

We compared our proposed MC-VEO with the following competing methods.

- *EVO* [15] is a semi-dense VO approach that combines an event-based tracking approach based on image-to-model alignment with an event-based 3D reconstruction algorithm in a parallel fashion. EVO [15] does not exploit EGM since it neither uses frames nor recovers image brightness.
- *USLAM* [25] is an indirect monocular method that fuses events, frames and IMU measurements. Its front-end converts events into frames by motion compensation using the IMU's gyroscope and the median scene depth. Then FAST corners [52] are extracted and tracked separately on the event frames and the grayscale frames, and then passed to a geometric feature-based back-end. The IMU is tightly used in the front-end for event frame creation, and so removing it is not possible without breaking the robustness of the method.
- *EDS* [17] is a direct monocular visual odometry using events and frames. Its front-end predicts per-pixel brightness increments and compares them to the events via the brightness increment error to estimate camera motion. The method recovers a semi-dense 3D map using photometric bundle adjustment in its back-end.
- *ORB-SLAM3* [53] is the system able to perform visual, visual-inertial and multi-map SLAM with monocular, stereo and RGB-D cameras, using pin-hole and fish-eye lens models. It is not only a feature-based tightly-integrated visual-inertial SLAM system that fully relies on Maximum-A-Posteriori (MAP) estimation but a multiple map system that relies on a novel place recognition method with improved recall.
- *DSO* [51] is a direct and sparse formulation for visual odometry. It combines a fully direct probabilistic model (minimizing a photometric error) with consistent and joint optimization of all model parameters, including geometry represented as inverse depth in a reference frame and camera motion.
- *VINS-MONO* [54] is a monocular visual-inertial state estimator. A tightly-coupled, nonlinear optimization-based visual-inertial odometry is designed by fusing pre-integrated IMU measurements and feature observations. It also performs four DoF pose graph optimization to enforce global consistency.
- *DROID-SLAM* [55] is a deep learning based visual SLAM system. It consists of recurrent iterative updates of camera pose and pixelwise depth through a Dense Bundle Adjustment layer.

Among the methods compared, *EDS* [17] and *ORB-SLAM3* [53] are the state-of-the-art approaches for the event odometry and the visual odometry, respectively.

B. Datasets and Metrics

Our MC-VEO and compared methods (introduced in Section IV-A) were tested on two datasets, including the RPG Stereo DAVIS[1] and the EDS dataset[2]. The RPG Stereo DAVIS [56] was

[1]The RPG Stereo DAVIS: https://rpg.ifi.uzh.ch/ECCV18_stereo_davis.html [56].

[2]The EDS dataset: https://rpg.ifi.uzh.ch/eds.html [17].

collected with a hand-held DAVIS-240 C (240×180) stereo event camera in an indoor environment, and the EDS dataset [17] was collected using a custom-made beamsplitter with a Prophesee Gen3 (640×480) event camera and a FLIR color camera. The ground truth poses of these two datasets were both given by motion capture systems. The RPG Stereo DAVIS dataset [56] is a benchmark widely used in related studies, while the EDS dataset [17] is a recently proposed high-resolution large-scale challenging dataset that includes cases of violent camera shaking and no texture scenes (wall corners, pure white ceiling, etc.).

To assess the performance of the full VO method, we report ego-motion estimation results using standard metrics: RMSE (Root Mean Squared Error) of Absolute Translation Error (cm) and Rotation Error (deg). The Absolute Translation Error can be given as,

$$ATE_{RMSE} = \left(\frac{1}{M} \sum_{i=1}^{M} \left\| \mathrm{trans}\left(\mathbf{Q}_i^{-1} \mathbf{P}_i \right) \right\|^2 \right)^{\frac{1}{2}}, \quad \mathbf{Q}_i, \mathbf{P}_i \in SE(3), \tag{15}$$

where $\{\mathbf{Q}_i\}_{i=1}^{M}$ and $\{\mathbf{P}_i\}_{i=1}^{M}$ are groundtruth and estimated poses, respectively. The $\mathrm{trans}(\cdot)$ represents the translation part of the pose. M is the total number of poses in each trajectory. The Rotation Error can be given as,

$$RE_{RMSE} = \left(\frac{1}{M} \sum_{i=1}^{M} | \ \mathrm{angle}_i \ |^2 \right)^{\frac{1}{2}}, \quad \mathrm{angle}_i = \log_{SO(3)} \left(\mathrm{rot}\left(\mathbf{Q}_i^{-1} \mathbf{P}_i \right) \right), \quad \mathbf{Q}_i, \mathbf{P}_i \in SE(3). \tag{16}$$

The $\mathrm{rot}(\cdot)$ represents the rotation part of the pose. The $\log_{SO(3)}(\cdot)$ is the inverse of $\exp_{\mathfrak{so}(3)}(\cdot)$ (Rodrigues Formula). The angle_i represents the i-th rotation angle. We used the toolbox from [57] to evaluate the poses given by different odometry solutions. The experimental platform is a laptop with a CPU model of AMD Ryzen 5 4600 U with Radeon Graphics.

C. Qualitative Results

Fig. 3 shows the comparisons of event frames generated by our motion compensation method and some other competitors. The raw intensity frame and the brightness increment are shown on the left for reference. The brightness increment frame here is predicted using (11) to align with the event frame in the system for the pose estimation. On the right, from top to bottom, event frames generated by direct accumulation, edge lighten in EDS [17] and our method are listed. It can be seen that in the case of direct accumulation, the event frames exhibit severe edge blurring (such as the edges of dolls and tables), which is clear in the brightness increment. The edge lighten strategy in EDS [17] can alleviate this situation to some extent, while our motion compensation scheme restores the event frame closest to the brightness increment of the raw image.

Fig. 4 shows the qualitative comparison of our MC-VEO and some representative odometries on three test sequences from [56] (first three columns) and a sequence from [17] (last column). From Fig. 4, it can be seen that, EVO [15] gives a sparse depth estimate and contains obvious outliers. DSO [51] generates a complete semi-dense depth map, but in sequence "all_characters", it shows severe estimation errors (the relative depth of the foreground and background is not correct). In comparison, MC-VEO forms a highly colorful semi-dense depth structure and can accurately estimate the depth at most contour pixels.

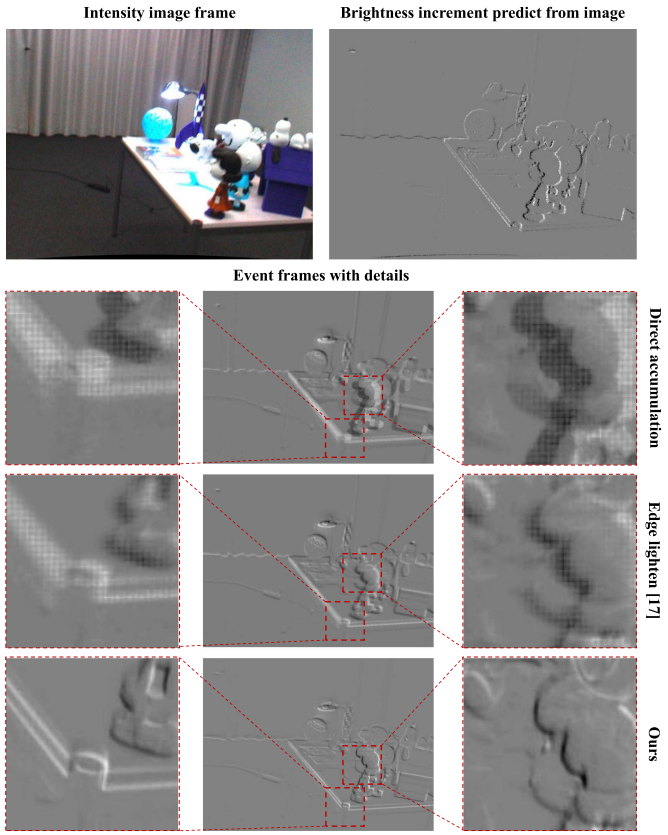


Fig. 3. Comparisons of a typical event frame generated by different methods. The intensity image frame and the brightness increment of the image are listed on the top for reference. Below them, from top to bottom, the event frames generated by direct accumulation, edge lighten in EDS [17] and our motion compensation method are listed, respectively. The event density is approximately 0.6 event per pixel.

D. Quantitative Results

Table II and Table III report quantitative results of the comparison of our method with other VOs in term of Absolute Translation Error (cm) and Rotation Error (deg), respectively, on sequences from the datasets [17], [56]. The optimal results are highlighted in bold, while the runner-up results are highlighted with an underline. The input data for the odometries may be events (E), image frames (F) or inertial measurements (I). The first four rows of both tables are results from The RPG Stereo DAVIS dataset [56] while the last three rows from the EDS dataset [17]. Due to the severe distortion of sensor output in challenging scenarios of the EDS dataset [17], all compared works cannot fully complete the sequence. In order to obtain

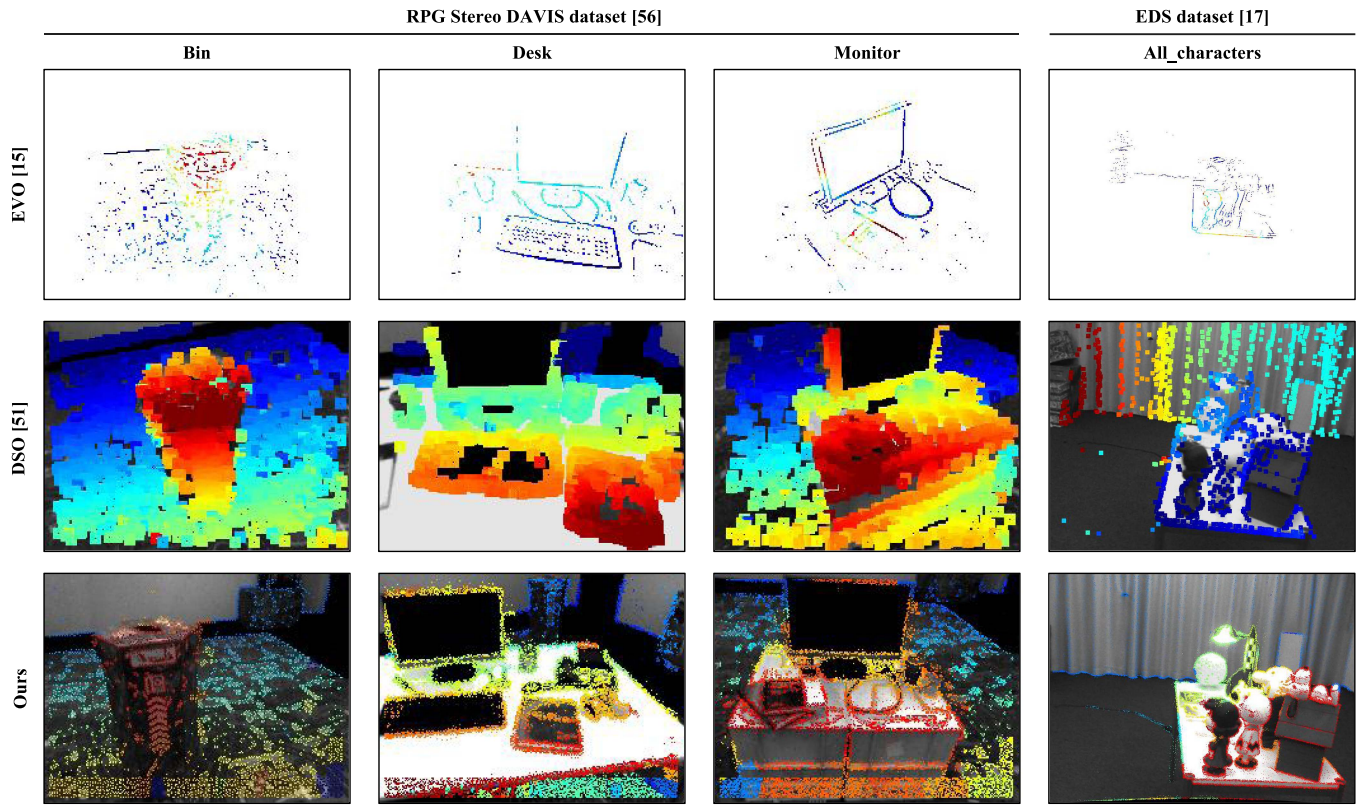


Fig. 4. Qualitative comparison on three test sequences from [56] (bin, desk, and monitor) and a sequence from [17] (all_characters). Each row depicts the pseudo-colored inverse depth maps generated by corresponding methods (red represents near and blue for far). It is worth mentioning that since the timestamps of event frames formed by different methods are not completely aligned, we chose to show results with relatively close perspectives.

TABLE II
ABSOLUTE TRANSLATION ERRORS (CM) OF MC-VEO AND COMPARED VOS ON THE DATASETS [56] AND [17]

Input	USLAM [25] E+F+I	EVO [15] E	EDS [17] E+F	ORB-SLAM3 [53] F	DSO [51] F	VINS-MONO [54] F+I	DROID-SLAM [55] F	MC-VEO (**Ours**) E+F
bin	**0.858**	10.698	1.149	1.661	7.431	1.185	4.464	1.138
boxes	5.218	12.068	2.144	2.482	2.117	6.355	2.958	**2.085**
desk	1.935	13.080	**1.584**	2.749	7.182	1.745	4.272	**1.584**
monitor	**0.875**	25.105	0.963	29.235	0.966	1.825	4.838	0.953
*all_characters**	139.399	151.092	2.417	15.758	23.985	63.329	36.300	**1.894**
*rocket_earth_light**	fail	80.991	9.019	**1.435**	4.478	142.366	29.262	4.159
*rpg_building**	50.352	fail	6.121	35.568	77.736	34.074	37.993	**3.774**

TABLE III
ROTATION ERRORS (DEG) OF MC-VEO AND COMPARED VOS ON THE DATASETS [56] AND [17]

Input	USLAM [25] E+F+I	EVO [15] E	EDS [17] E+F	ORB-SLAM3 [53] F	DSO [51] F	VINS-MONO [54] F+I	DROID-SLAM [55] F	MC-VEO (**Ours**) E+F
bin	1.218	2.085	1.512	1.109	2.729	1.203	3.050	**1.053**
boxes	2.398	2.788	4.855	2.483	4.591	**2.286**	2.811	2.466
desk	1.893	2.375	2.155	1.527	4.263	1.885	2.720	**1.131**
monitor	**1.308**	2.971	1.864	1.491	2.622	1.812	3.459	1.355
*all_characters**	1.869	6.031	0.957	1.047	4.061	1.647	9.429	**0.238**
*rocket_earth_light**	fail	22.703	0.730	0.460	2.193	9.963	8.474	**0.440**
*rpg_building**	2.858	fail	0.742	0.750	3.880	2.136	10.139	**0.676**

reliable quantitative comparison results on this high-resolution dataset, we can only use fragments that can be completed by most methods. The sequence marked with * represents taking partial fragments of the sequence to enable most methods to successfully complete.

From Table II, it can be seen that MC-VEO outperforms all other baseline methods on translation accuracy, even without assistance of inertial measurements. It is worth mentioning that the USLAM [25] and the EVO [15] cannot be completed on sequence rocket_earth_light and rpg_building, respectively. The

TABLE IV
ABLATION STUDIES ON THE ACCURACY OF MC-VEO

	Input	None	Compensation	Acceleration	Both
ATE	*all_characters**	2.417	1.898	2.415	**1.894**
	*rocket_earth_light**	8.442	4.478	8.410	**4.159**
	*rpg_building**	6.320	4.175	6.186	**3.774**
RE	*all_characters**	0.872	0.267	0.866	**0.238**
	*rocket_earth_light**	0.633	0.450	0.620	**0.440**
	*rpg_building**	0.938	0.683	0.921	**0.676**

The optimal results are highlighted in bold.

TABLE V
ABLATION STUDIES ON THE SPEED OF MC-VEO

	Input	None	Compensation	Acceleration	Both	Ratio
Time [s]	*bin*	287.286	239.838	261.732	**195.968**	-31.8%
	boxes	513.967	322.632	419.092	**276.099**	-46.3%
	desk	413.748	212.215	328.384	**162.519**	-60.7%
	monitor	320.470	280.169	230.059	**219.449**	-31.5%
	*all_characters**	1,062.992	897.067	977.067	**744.225**	-30.0%
	*rocket_earth_light**	2,792.026	2,084.415	2,402.231	**1,961.957**	-30.7%
	*rpg_building**	387.858	364.745	331.323	**332.305**	-14.3%

The optimal results are highlighted in bold.

former suffers from insufficient feature points which leads to loss of tracking, while the latter cannot smoothly bootstrap from the event stream. Besides, MC-VEO is also superior to all other baseline methods in terms of rotation error according to Table III. In high-resolution and challenging sequences (last three rows), MC-VEO is far ahead of other methods due to the integration of our 6-DoF motion compensation and EGM-based motion estimation. In these three sequences, the motion of the camera is fast, and the insufficient frame rate of the RGB camera affects the performance of frame-based VO method. For other existing event-based methods, they face serious challenges due to the low Signal Noise Ratio of event cameras, as high-resolution sequences bring more event noise than low-resolution ones [58], [59]. Our MC-VEO effectively eliminates the influence of event noise and distinguishes outliers when aligning events in motion compensation, while also utilizing EGM-based method to fully utilize event information to estimate camera's fast motion. On low-resolution sequences, MC-VEO achieves rotational accuracy comparable to the frame-based SOTA method, ORB-SLAM3 [53].

E. Ablation Study

In order to verify the performance gain brought by our motion compensation module and accelerated iterative solving module on the accuracy and speed, ablation experiments on these two modules were conducted. The experimental results are presented in Tables IV (in term of Absolute Translation Error (cm) and Rotation Error (deg)) and V (in term of total time cost (s)). For each column in these two tables, "None" refers to not using these two modules, "Compensation" refers to using only the motion compensation module, "Acceleration" refers to using only the accelerated iterative solving module, and "Both" refers to using both of them. Besides, in Table V, "Ratio" refers to the total time reduction rate.

Table IV shows the impact of the two aforementioned modules on system accuracy. Only results on high-resolution sequences are presented, as the differences in accuracy are not significant on low-resolution ones. Table V shows the impact of these two modules on speed by comparing the total time taken to complete tracking on each sequence. From Table IV and Table V, it can be seen that by integrating our motion compensation module and accelerated iterative solving module, both the accuracy and the speed of our MC-VEO can be significantly improved, corroborating the effectiveness of them.

F. Limitations and Discussions

Although our MC-VEO has achieved pleasing results in evaluated datasets, we found that the textures of the scene have an obvious influence on MC-VEO's performance. When the scene texture is weak, the information contained in the image is relatively scarce. In this case, the candidate point selection strategy of MC-VEO would be affected and could not be able to extract high-quality feature points, resulting in relatively poor performance. Thus we will continue to devote efforts in finding more robust feature extraction method to alleviate such a negative impact.

V. CONCLUSION

In this article, we proposed a novel visual-event odometry solution, namely MC-VEO. In MC-VEO, a novel 6-DoF motion compensation method based on an improved contrast maximization framework is utilized to create pseudo measurements from the event. By minimizing the brightness increment errors between these measurements and the measurements predicted from image frames, the relative poses of event frames and keyframes are estimated. Thanks to our novel motion compensation method and accelerated iterative solving module, MC-VEO achieved a good balance between accuracy and speed. Experimental results corroborated MC-VEO's superiority over the state-of-the-art competitors in this area.

REFERENCES

[1] C. Shu and Y. Luo, "Multi-modal feature constraint based tightly coupled monocular visual-LiDAR odometry and mapping," *IEEE Trans. Intell. Veh.*, vol. 8, no. 5, pp. 3384–3393, May 2023.
[2] H. Guo, J. Zhu, and Y. Chen, "E-LOAM: LiDAR odometry and mapping with expanded local structural information," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1911–1921, Feb. 2023.
[3] Y. Tian and M. Compere, "A case study on visual-inertial odometry using supervised, semi-supervised and unsupervised learning methods," in *Proc. IEEE Int. Conf. Artif. Intell. Virtual Reality*, 2019, pp. 203–2034.
[4] T. Schöps, J. Engel, and D. Cremers, "Semi-dense visual odometry for AR on a smartphone," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2014, pp. 145–150.
[5] M. Ramezani and K. Khoshelham, "Vehicle positioning in GNSS-deprived urban areas by stereo visual-inertial odometry," *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 208–217, Jun. 2018.
[6] S. Liang, Z. Cao, C. Wang, and J. Yu, "Hierarchical estimation-based LiDAR odometry with scan-to-map matching and fixed-lag smoothing," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1607–1623, Feb. 2023.
[7] C. Cadena et al., "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
[8] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 × 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008.
[9] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, Jan. 2011.

[10] C. Brandli, R. Berner, M. Yang, S. C. Liu, and T. Delbruck, "A 240 × 180 130 dB 3 μs latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, Oct. 2014.

[11] H. Kim and H. J. Kim, "Real-time rotational motion estimation with contrast maximization over globally aligned events," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 6016–6023, Jul. 2021.

[12] H. Kim, A. Handa, R. Benosman, S. Ieng, and A. Davison, "Simultaneous mosaicing and tracking with an event camera," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–12.

[13] G. Gallego, J. E. A. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza, "Event-based, 6-DOF camera tracking from photometric depth maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2402–2412, Oct. 2018.

[14] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3D reconstruction and 6-DoF tracking with an event camera," in *Proc. 14th Eur. Conf. Comput. Vis.*, 2016, pp. 349–364.

[15] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza, "EVO: A geometric approach to event-based 6-DoF parallel tracking and mapping in real time," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 593–600, Apr. 2017.

[16] Y. Zuo, J. Yang, J. Chen, X. Wang, Y. Wang, and L. Kneip, "DEVO: Depth-event camera visual odometry in challenging conditions," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 2179–2185.

[17] J. Hidalgo-Carrió, G. Gallego, and D. Scaramuzza, "Event-aided direct sparse odometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5771–5780.

[18] G. Gallego and D. Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 632–639, Apr. 2017.

[19] J. Kogler, C. Sulzbachner, and W. Kubinger, "Bio-inspired stereo vision system with silicon retina imagers," in *Proc. Int. Conf. Comput. Vis. Syst.*, 2009, pp. 174–183.

[20] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5419–5427.

[21] A. Nguyen, T. Do, D. G. Caldwell, and N. G. Tsagarakis, "Real-time 6-DoF pose relocalization for event cameras with stacked spatial LSTM networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshop*, 2019, pp. 1638–1645.

[22] G. Gallego, M. Gehrig, and D. Scaramuzza, "Focus is all you need: Loss functions for event-based vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12272–12281.

[23] A. Mitrokhin, C. Ye, C. Fermüller, Y. Aloimonos, and T. Delbruck, "EV-IMO: Motion segmentation dataset and learning pipeline for event cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6105–6112.

[24] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 16.1–16.12.

[25] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.

[26] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, "Low-latency visual odometry using event-based feature tracks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 16–23.

[27] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1433–1450, Oct. 2021.

[28] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 989–997.

[29] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, "EKLT: Asynchronous photometric feature tracking using events and frames," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 601–618, 2020.

[30] S. Bryner, G. Gallego, H. Rebecq, and D. Scaramuzza, "Event-based, direct camera tracking from a photometric 3D map using nonlinear optimization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 325–331.

[31] G. Gallego et al., "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[32] T. Stoffregen, G. Gallego, T. Drummond, L. Kleeman, and D. Scaramuzza, "Event-based motion segmentation by motion compensation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7243–7252.

[33] Z. W. Wang, P. Duan, O. Cossairt, A. Katsaggelos, T. Huang, and B. Shi, "Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1606–1616.

[34] M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger, "Interacting maps for fast visual interpretation," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2011, pp. 770–776.

[35] D. Weikersdorfer, R. Hoffmann, and J. Conradt, "Simultaneous localization and mapping for event-based vision systems," in *Proc. 9th Int. Conf. Comput. Vis. Syst.*, 2013, pp. 133–142.

[36] A. Censi and D. Scaramuzza, "Low-latency event-based visual odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 703–710.

[37] D. Weikersdorfer, D. B. Adrian, D. Cremers, and J. Conradt, "Event-based 3D SLAM with a depth-augmented dynamic vision sensor," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 359–364.

[38] E. Mueggler, B. Huber, and D. Scaramuzza, "Event-based, 6-DOF pose tracking for high-speed maneuvers," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2761–2768.

[39] C. Reinbacher, G. Munda, and T. Pock, "Real-time panoramic tracking for event cameras," in *Proc. IEEE Int. Conf. Comput. Photogr.*, 2017, pp. 1–9.

[40] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, "EMVS: Event-based multi-view stereo-3D reconstruction with an event camera in real-time," *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1394–1414, 2018.

[41] D. Zhu et al., "Neuromorphic visual odometry system for intelligent vehicle application with bio-inspired vision sensor," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2019, pp. 2225–2232.

[42] M. Gehrig, S. B. Shrestha, D. Mouritzen, and D. Scaramuzza, "Event-based angular velocity regression with spiking networks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4195–4202.

[43] D. Liu, A. Parra, and T. J. Chin, "Spatiotemporal registration for event-based visual odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4935–4944.

[44] Y. Wang et al., "Visual odometry with an event camera using continuous ray warping and volumetric contrast maximization," *Sensors*, vol. 22, no. 15, 2022, Art. no. 5687.

[45] G. Gallego, H. Rebecq, and D. Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3867–3876.

[46] T. Stoffregen and L. Kleeman, "Event cameras, contrast maximization and reward functions: An analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12292–12300.

[47] X. Peng, Y. Wang, L. Gao, and L. Kneip, "Globally-optimal event camera motion estimation," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 51–67.

[48] X. Peng, L. Gao, Y. Wang, and L. Kneip, "Globally-optimal contrast maximisation for event cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3479–3495, Jul. 2022.

[49] G. Gallego, C. Forster, E. Mueggler, and D. Scaramuzza, "Event-based camera pose tracking using a generative event model," 2015, *arXiv:1510.01972*. [Online]. Available: https://arxiv.org/abs/1510.01972v1

[50] P. I. Corke and O. Khatib, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, vol. 73. Berlin, Germany: Springer, 2011.

[51] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[52] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. 9th Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.

[53] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[54] T. Qin, P. Li, and S. Shen, "VINS-MONO: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[55] Z. Teed and J. Deng, "DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 16558–16569.

[56] Y. Zhou, G. Gallego, H. Rebecq, L. Kneip, H. Li, and D. Scaramuzza, "Semi-dense 3D reconstruction with a stereo event camera," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 235–251.

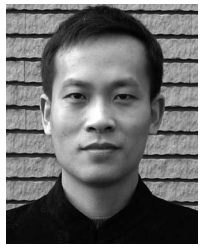
[57] M. Grupp, "evo: Python package for the evaluation of odometry and SLAM," 2017. [Online]. Available: https://github.com/MichaelGrupp/evo

[58] A. Khodamoradi and R. Kastner, "$\mathcal{O}(\mathcal{N})$-space spatiotemporal filter for reducing noise in neuromorphic vision sensors," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 1, pp. 15–23, Jan.–Mar. 2021.
[59] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," *Sci. Robot.*, vol. 5, no. 40, 2020, Art. no. eaaz9712.

Jiafeng Huang received the B.S. degree in 2017 from the School of Software Engineering, Tongji University, Shanghai, China, where he is currently working toward the Ph.D. degree. His research interests include event camera, SLAM systems, and computer vision.

Shengjie Zhao (Senior Member, IEEE) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 1988, the M.S. degree in electrical and computer engineering from the China Aerospace Institute, Beijing, China, in 1991, and the Ph.D. degree in electrical and computer engineering from Texas A&M University, College Station, TX, USA, in 2004. He is currently a Professor with the School of Software Engineering, Tongji University, Shanghai, China. In previous postings, he conducted research at Lucent Technologies, Whippany, NJ, USA, and China Aerospace Science and Industry Corporation, Beijing. His research interests include Big Data, wireless communications, image processing, and signal processing. He is a Fellow of the Thousand Talents Program of China.

Tianjun Zhang received the B.S. degree in 2019 from the School of Software Engineering, Tongji University, Shanghai, China, where he is currently working toward the Ph.D. degree. His research interests include collaborative SLAM, computer vision, and sensor calibration.



Lin Zhang (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Department of Computer Science and Engineering, Shanghai Jiao-Tong University, Shanghai, China, in 2003 and 2006, respectively, and the Ph.D. degree from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, in 2011. From March 2011 to August 2011, he was a Research Associate with the Department of Computing, The HongKong Polytechnic University. In 2011, he joined the School of Software Engineering, Tongji University, Shanghai, China, where he is currently a Full Professor. His research interests include environment perception of intelligent vehicle, pattern recognition, computer vision, and perceptual image/video quality assessment. He is an Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS and *Journal of Visual Communication and Image Representation*. He was awarded as a Young Scholar of Changjiang Scholars Program, Ministry of Education, China.