공학박사학위논문

Motion and Depth Estimation for Event and Frame Cameras

이벤트 프레임 카메라를 위한 모션 및 깊이 추정

2023년 2월

서울대학교 대학원
기계항공공학부
김 하 람

Motion and Depth Estimation for Event and Frame Cameras

이벤트-프레임 카메라를 위한 모션 및 깊이 추정

지도교수 김현진

이 논문을 공학박사 학위논문으로 제출함

2022년 12월

서울대학교 대학원

기계항공공학부

김하람

김하람의 공학박사 학위논문을 인준함

2022년 12월

위원장 : 김유단

부위원장 : 김현진

위원 : 박찬국

위원 : 이현범

위원 : 김표진

Motion and Depth Estimation for Event and Frame Cameras

A Dissertation

by

Haram Kim

Presented to the Faculty of the Graduate School of

Seoul National University

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Department of Mechanical and Aerospace Engineering

Seoul National University

Supervisor : Professor H. Jin Kim

FEBRUARY 2023

Motion and Depth Estimation for Event and Frame Cameras

Haram Kim

Department of Mechanical and Aerospace Engineering

Seoul National University

APPROVED:

Youdan Kim, Chair, Ph.D.

H. Jin Kim, Ph.D.

Chan Gook Park, Ph.D.

Hyeonbeom Lee, Ph.D.

Pyojin Kim, Ph.D.

to my

FAMILY

with love

Abstract

Motion and Depth Estimation for Event and Frame Cameras

Haram Kim
Department of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

Event cameras can stably measure visual information in high-dynamic-range and high-speed environments that are challenging for conventional cameras. However, conventional vision algorithms could not be directly employed to the event data, because of the frameless and asynchronous characteristics of event data. For several years, various applications for event cameras have been studied such as motion and depth estimation, image reconstruction with high-temporal resolution and object segmentation. Here, I propose the rotational motion estimation method with contrast maximization under high-speed motion environments. The proposed rotational motion estimation method runs in real-time and can handle the drift error accumulation, which the existing contrast maximization methods have not dealt with.

However, it is still difficult for event cameras to replace frame cameras in non-challenging normal scenarios. In order to leverage the advantages of event and frame cameras, I conduct a study for the heterogeneous stereo camera system which employs both an event and a frame camera. The proposed system estimates the semi-dense disparity in real-time by matching heterogeneous data of an event and a frame camera in stereo. I propose an accurate, intuitive and efficient way to align events with 6-DOF camera motion, by suggesting the maximum shift distance method. The aligned event image shows high similarity to the edge image of the frame camera. The proposed depth estimation method runs in real-time and can estimate poses of an event camera and depth of events in a few frames, which can speed up the initialization of the event camera system. Additionally, I propose a feature tracking and a pose estimation methods that can operate in a hetero-stereo camera when the frame camera fails. The codes are released to the public on my project page, and I expect to contribute to the event camera community: https://haram-kim.github.io

Keywords: Event cameras, pose estimation, depth estimation, hetero-stereo event-frame camera, contrast maximization

Student Number: 2019-30458

Table of Contents

Page

Chapter

Appendix

List of Tables

List of Figures

1
Introduction

Event cameras, also known as dynamic vision sensors or neuromorphic cameras, are bio-inspired vision sensors that record events rather than capture frame images. Event cameras have a different information recording mechanism from ordinary cameras as in Fig. 1.1. Event cameras asynchronously record the brightness change of a pixel, and their detailed properties can be found in [3]. Generally, one event data (event point) includes the spatio-temporal information as follows.

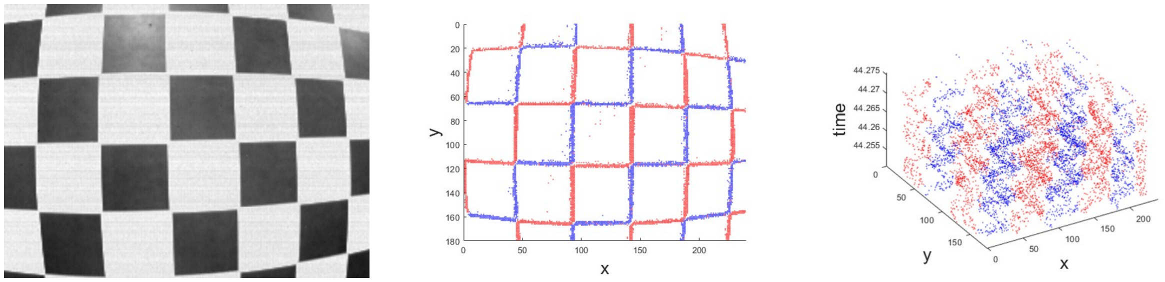
$$e_k = (\mathbf{x}_k, t_k, p_k), \quad \text{where } \mathbf{x}_k = (x_k, y_k). \tag{1.1}$$

For an event point e_k, $\mathbf{x}_k$ is the pixel coordinate, t_k is the time when the event occurred and $p_k \in \{+1, -1\}$ is the event polarity (brightness increase over threshold: $+1$, decrease: -1).

When the brightness changes at a certain pixel of event camera, event data is recorded as Eq. (1.1) if the following Eq. (1.2) is satisfied.

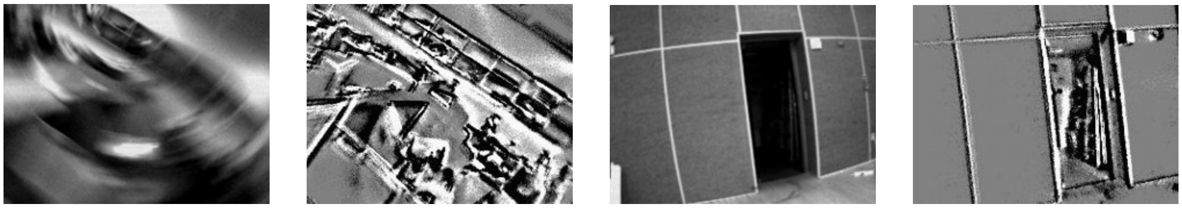
$$\Delta \ln \mathrm{I}(\mathbf{x}, t) := \ln \mathrm{I}(\mathbf{x}, t) - \ln \mathrm{I}(\mathbf{x}, t_{prev}) > p \cdot C_{th}, \tag{1.2}$$

where $\mathrm{I}(\mathbf{x}, t)$ is the intensity at a pixel $\mathbf{x}$, and C_{th} is the event contrast threshold, and t_{prev} is the



(a) (b) (c)

Figure 1.1: Frame and event data examples: (a) frame image (b) events top view (c) events diagonal view.



(a) (b) (c) (d)

Figure 1.2: Frame and event data in challenging environments: (a) frame image with fast motion, (b) aligned event image with fast motion, (c) frame image in HDR, (d) aligned event image in HDR.

time of the most recent event recorded at the very pixel coordinate. If $\Delta \ln \mathrm{I}(\mathbf{x},t) > C_{th}$, the event data will be recorded as $e = (\mathbf{x},t,+1)$ (ON-event). Else if $\Delta \ln \mathrm{I}(\mathbf{x},t) < -C_{th}$, the event data will be $e = (\mathbf{x},t,-1)$ (OFF-event).

Event cameras have characteristics of high dynamic range (HDR), much less motion blur, energy efficiency and low latency which could broaden the applicability of computer vision in challenging environments. Thus, event cameras can show better performance than frame cameras in challenging scenarios such as fast-moving environments or high-dynamic-range scenes. In order to take advantages of event cameras, various applications have been studied for event cameras [4–9]. In the dissertation, I cover the methods for motion and depth estimation using event and frame cameras.

1.1 Literature Survey

1.1.1 Frame image reconstruction from events

There have been attempts to adapt the existing vision algorithm to the event camera, by reconstructing the frame images from events only: [10] and [11]. The author of [12] reconstruct frame images with high temporal resolution, by using the frame and event cameras. However, unwanted artifacts, such as bleeding effect or local black region, remain and if a still image is included or once the algorithm fails to reconstruct the image. The artifacts will continue to adversely affect later reconstruction performance.

1.1.2 Event based Motion Estimation

Some studies [13–22] estimated the camera ego-motion using event cameras. The study of [13] firstly performed a 6-degree of freedom (DOF) motion estimation through the extended Kalman filter (EKF) using an event camera only. The authors of [14] estimated 6-DOF motion using an event camera with a photometric map. They built a photometric depth map by applying the existing dense reconstruction method with an RGB-D camera. The study of [15] also utilized an event camera without any external sensor. The authors of [15] obtained depth information by applying the disparity space image (DSI) in [23], and constructed a 3D edge map with reliable depth. Then, pose tracking was performed through the image-to-model alignment. Although the methods [13–15] can estimate the rotational motion, they are not suitable for pure rotation estimation because they require constraints for depth estimation in the initialization phase such as a planar motion assumption.

Studies such as [1, 2, 24, 25] have been conducted to estimate rotational motion. The authors of [2] proposed real-time panoramic tracking and mapping for event cameras. They utilize the panoramic map (2D spherical mosaic map) as in [24] and estimated the rotational motion by tracking the camera trajectory on the panoramic space. In [1], the authors warped the event points considering the time of the event that occurred. The contrast of images obtained from the warped event points was used as the objective function. The authors estimated the angular velocity by solving the contrast maximization problem. They extended the idea and proposed a method to estimate the depth and optical flow in addition to the rotational motion through [26]. In [25], the global optimal solution is presented to the contrast maximization problem through the branch-and-bound (BnB) approach. Although the theoretical value is high, the heavy computational load limits the actual use.

1.1.3 Event based Depth Estimation

A significant number of vision algorithms employ 3D scene information to demonstrate more diverse applications. Similarly, in order to extend the limited application of a single event camera, stereo event studies [27–38] aimed to estimate the depth. These studies were conducted to utilize event cameras as stereo, and several attempts have been made to extract meaningful features from events. In [28], semi-dense 3d reconstruction was performed employing a stereo event camera. The method estimated the depth on edges where events frequently spiked, considering multiple viewpoints. The authors of [27] estimated the disparity after aligning events with optical flow in consideration of camera motion. Since it is difficult to accurately estimate the camera pose with event data only, the authors used ground truth poses. Even though stereo event cameras can utilize the depth information, there is a fundamental issue that the events do not spike in static situations and the advantage is revealed only in specific scenarios.

In order to extend the usability of event cameras in general scenarios, several attempts [39–43] have recently been made to combine the advantages of frame and event, which can also enable the various applications of frame cameras.

In [39], the author attempted to use stereo cameras and event cameras together. When combining an event camera with a stereo camera or a RGB-D camera, the individual depth of an event is still unknown due to asynchronous characteristic. To combine events and frames, the author conducted the dense and continuous disparity estimation method using camera motion. To accomplish the same goal as [39] in a monocular camera, the authors of [40] estimated depth map in high frame rate from the monocular event camera which also provides frame images. Studies [39–42] use two types of vision data, but cannot operate as stereo because frames and events are acquired from a monocular camera.

So far, there has been a lack of study on stereo event frame studies that can estimate the depth of events in general scenarios. In [44, 45], the authors conveyed a study to perceive 3D scene by firstly configuring an event and a frame camera as stereo. In [44], by using the event-and-frame camera attached to the robot arm, the author estimated the disparity through stereo matching with binary frame edge and binary event edge images. The binary frame edge images are generated from frame images, and the binary event edge images are from the event data with a high-pass filter and non-maximal suppression.

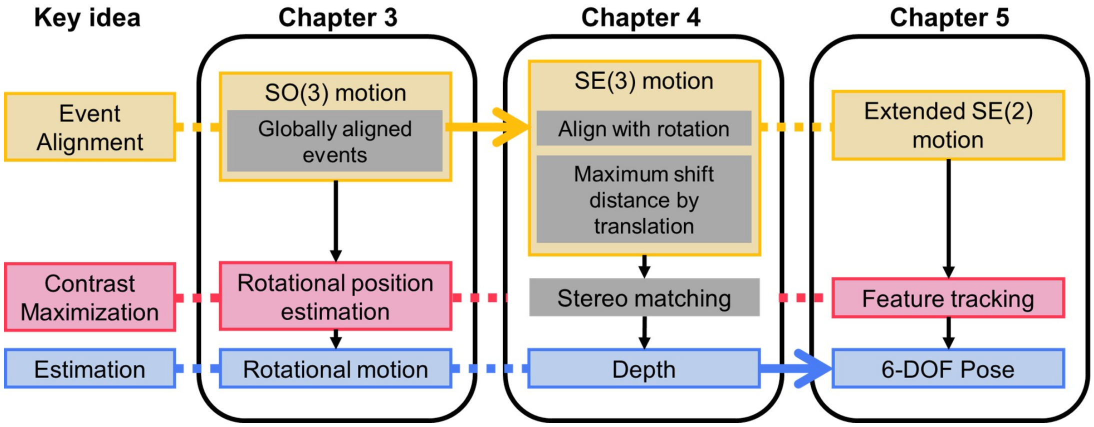


Figure 1.3: Algorithm flowchart.

1.2 Motivation

1.2.1 Real-time Rotational Motion Estimation with Contrast Maximization over Globally Aligned Events

The contrast maximization method [1, 26] showed that event cameras can estimate high-speed motion. Also, inertial measurement units (IMUs) are accurate for estimating angular velocity and can cope with fast motion. However, when measuring the angular position, there is a problem in that a drift error is accumulated along with the integral. Existing contrast maximization cannot handle drift error accumulation because it estimates the parameter (rotational motion, optical-flow, depth) with events only in the current temporal window. Utilizing more events is computationally demanding, and the linearization model cannot be applied to events observed over a long period.

In the dissertation, I propose a method for contrast maximization which utilizes the events observed for a long time through the global events alignment. The proposed method estimates the rotational position by localizing the event points on the globally aligned events, as depicted in Fig. 3.1. The globally aligned events refer to the alignment of all event points to the spatial coordinates of the initial time and are different from the map that represents the intensity of the surrounding environment. I propose the contrast maximization over the globally aligned events with a little additional computational load. I expect that the proposed method can be employed in other applications of contrast maximization framework such as estimating depth and optical-flow.

1.2.2 Real-time Hetero-Stereo Matching for Event and Frame Camera with Aligned Events Using Maximum Shift Distance

In order to extend the applications of the event camera, I tackle the problem of how to associate event data and frame images in stereo event frame camera settings. I present an accurate and intuitive way to align asynchronous event data. In order to accurately describe the edge of the scene, the proposed method warps events by only considering camera motion and disparity. Thus, the proposed event aligning module is parameter-free regardless of the speed of camera motion and data domain. In [44], binary event edge images are obtained from the high-pass filter of [46] with fixed parameters. Such binary edge images do not describe the scene properly on evaluation dataset of Section 4.2. Thus, [44] shows low accuracy on disparity estimation as in Table 4.1.

While the method [27] utilized the ground truth camera pose and assumed that the depth of all events in a short time window only depends on the pixel coordinate (not the temporal coordinate), The proposed method estimates the camera pose from the initial matching method and assume that events have the same depth value only at the reference time. For aligning events with 6-DOF camera motion, I extend the accurate warping method of [47] rather than using the first-order approximation of the warping function.

The proposed method can produce an aligned event image that looks similar to an edge image as in Fig. 4.2.(e), and the edge features are appropriately represented even when uniformly sampled events at 10% of the total are used as in Fig. 4.7. Additionally, I introduce the concept of maximum shift distance to efficiently compute aligned event images. I expect to contribute to the event camera community by suggesting an intuitive and efficient way to present edge images from event data.

1.2.3 Feature Tracking and Pose Estimation for Hetero-Stereo Camera

Existing visual odometry (VO) and simultaneous localization and mapping (SLAM) algorithms can operate on the hetero-stereo camera setup, by utilizing the frame images and the depth images obtained from the proposed stereo matching method. However, under challenging scenarios (i.e. high-dynamic-range and fast camera motion), frame images cannot be used and VO and SLAM algorithms fail. For this situation, I propose a pose estimation and a feature tracking method using an event camera instead of a frame image to build a framework that guarantees robustness.

1.3 Contribution and Outline

1.3.1 Real-time Rotational Motion Estimation with Contrast Maximization over Globally Aligned Events

- The proposed method can accurately estimate the rotational position and velocity during fast movements by introducing the contrast maximization over globally aligned events.
- The proposed method additionally requires less than 50% of the computational load of the existing contrast maximization to use globally aligned events.
- The proposed method operates in real-time, and the source code and the data sets are open to public.

1.3.2 Real-time Hetero-Stereo Matching for Event and Frame Camera with Aligned Events Using Maximum Shift Distance

- I provide the hetero-stereo matching methods in order to associate different data types of event-frame cameras and estimate disparities.
  1. Initial matching method with time-gradient images for fast initialization of the system (Section 4.1.1)
  2. Aligned-event-based matching method with edge images for accurate stereo matching (Section 4.1.3)
- I provide an accurate and intuitive event aligning method to describe the edge-like images, which utilizes internally estimated camera poses (Section 4.1.2).
- I suggest the concept of maximum shift distance to align events efficiently (Section 4.1.4).

1.3.3 Feature Tracking and Pose Estimation for Hetero-Stereo Camera

- For event cameras, I provide the continuous-time feature motion model and propose a feature tracking method with contrast maximization (Section 5.1).

- I propose a pose estimation method using events and depth images (Section 5.2).

2

Background

2.1 Rigid Body Motion

In the dissertation, the 3D rotation and 3D transformation matrix are parameterized as velocity vectors (twist) according to lie group theory. The time resolution of event data is very high. I obtained the linearly approximated 3D motion of individual events through the parameterized velocity.

2.1.1 Lie group for 3D rotation

For the angular velocity $\omega = [\omega_1, \omega_2, \omega_3]$ and the time t, the 3D rotation is represented as follows:

$$R(t) = e^{\hat{\omega} t}, \tag{2.1}$$

where,

$$\begin{aligned} \hat{\omega} &= G_1\omega_1 + G_2\omega_2 + G_3\omega_3 && (2.2) \\ &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \omega_1 + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \omega_2 + \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \omega_3 && (2.3) \\ &= \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} && (2.4) \end{aligned}$$

2.1.2 Lie group for 3D transformation

For the twist (linear velocity) $\xi = [v, \omega] = [v_1, v_2, v_3, \omega_1, \omega_2, \omega_3]$ and the time t, the 3D transformation is represented as follows:

$$\mathbf{T}(t) = \begin{bmatrix} \mathbf{R}(t) & \mathbf{t}(t) \\ 0_{1\times 3} & 1 \end{bmatrix} = e^{\hat{\xi}t}, \qquad (2.5)$$

where,

$$\hat{\xi} = G_1 v_1 + G_2 v_2 + G_3 v_3 + G_4 \omega_1 + G_5 \omega_2 + G_6 \omega_3 \qquad (2.6)$$

$$\hat{\xi} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} v_2 + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} v_3 \tag{2.7}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \omega_1 + \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \omega_2 + \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \omega_3$$

$$= \begin{bmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.8}$$

In the dissertation, I expressed the relative pose from the $n-1$th coordinate to the nth coordinate as $\mathbf{T}^n_{n-1}$ and $\mathbf{T}_{n-1,n}$, which satisfying:

$$\mathbf{T}^n_o = \mathbf{T}^{n-1}_o \mathbf{T}^n_{n-1} \tag{2.9}$$

and

$$\mathbf{T}_{o,n} = \mathbf{T}_{o,n-1} \mathbf{T}_{n-1,n}. \tag{2.10}$$

Let $\mathbf{x}(\tau_n)$ be the position of an event in 3D space at the time of nth coordinate τ_n, then the following equation holds:

$$\mathbf{x}(\tau_{n-1}) = \mathbf{T}^n_{n-1} \mathbf{x}(\tau_n) \tag{2.11}$$

and

$$\mathbf{x}(\tau_{n-1}) = \mathbf{T}_{n-1,n} \mathbf{x}(\tau_n). \tag{2.12}$$

2.2 Rectification

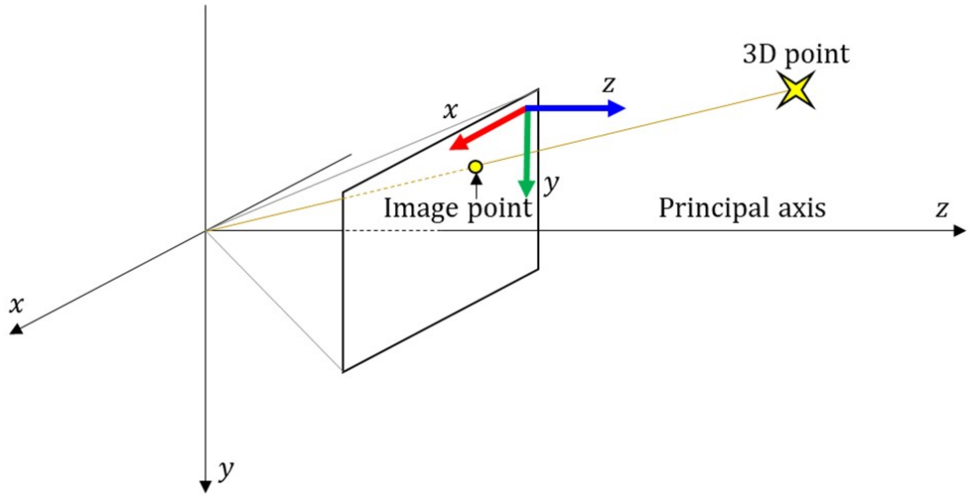


Figure 2.1: The camera model and coordinate.

The data set, which employs stereo event camera and stereo frame camera [48–51], provides rectified projection matrix for the stereo event camera and stereo frame camera, respectively. However, hetero stereo event and frame camera setup is not considered the data set. In this section, I will explain how to compute a calibration matrix with extrinsic parameters.

Let $\mathbf{T}_{o,f}$ be the transformation of world to the frame camera, and $\mathbf{T}_{o,e}$ be the transformation of world to the event camera. The x, y and z-axes are colored as red, green and blue, respectively in Fig. 2.1 and Fig. 2.2. Then, the principal axis is parallel to the blue arrow.

The center of the frame camera is $\mathbf{t}_{o,f}$ and the center of the event camera is $\mathbf{t}_{o,e}$, and the rotation matrix can be decomposed to three vectors as $\mathbf{R} = [\mathbf{r}^x, \mathbf{r}^y, \mathbf{r}^z]$. Then, the x-axis of the rectified coordinate, $\mathbf{r}^x_{o,r}$, has to be in the epipolar plane:

$$\mathbf{r}^x_{o,r} = \frac{\mathbf{t}_{o,e} - \mathbf{t}_{o,f}}{\|\mathbf{t}_{o,e} - \mathbf{t}_{o,f}\|} \tag{2.13}$$

The y and z-axes of the rectified coordinate are not constrained by epipolar plane. Instead, the y and z-axes have to be perpendicular to the x-axis. In order to find two axes, I compute the

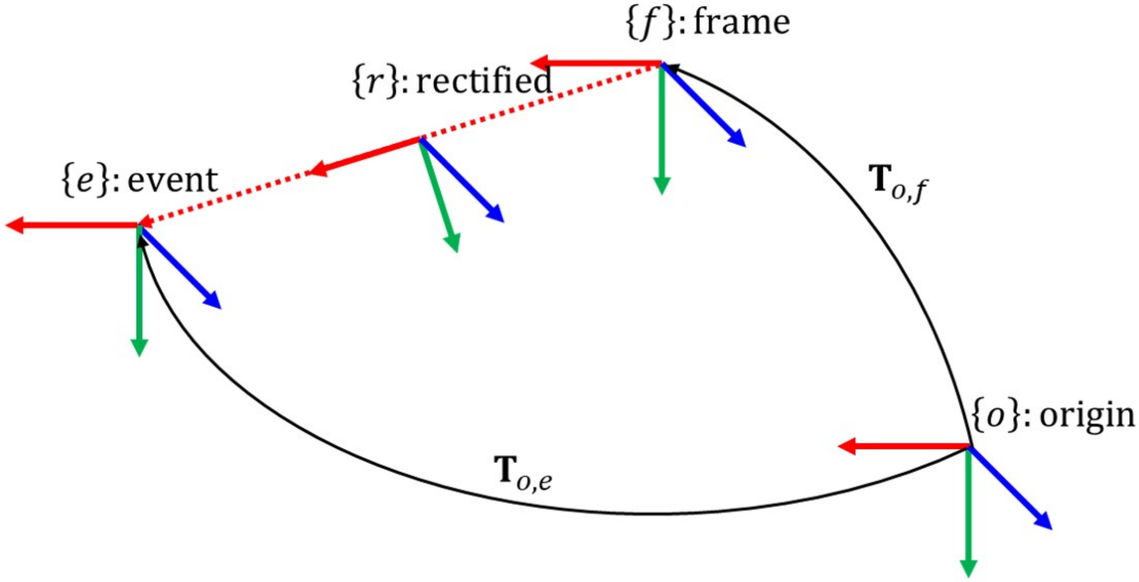
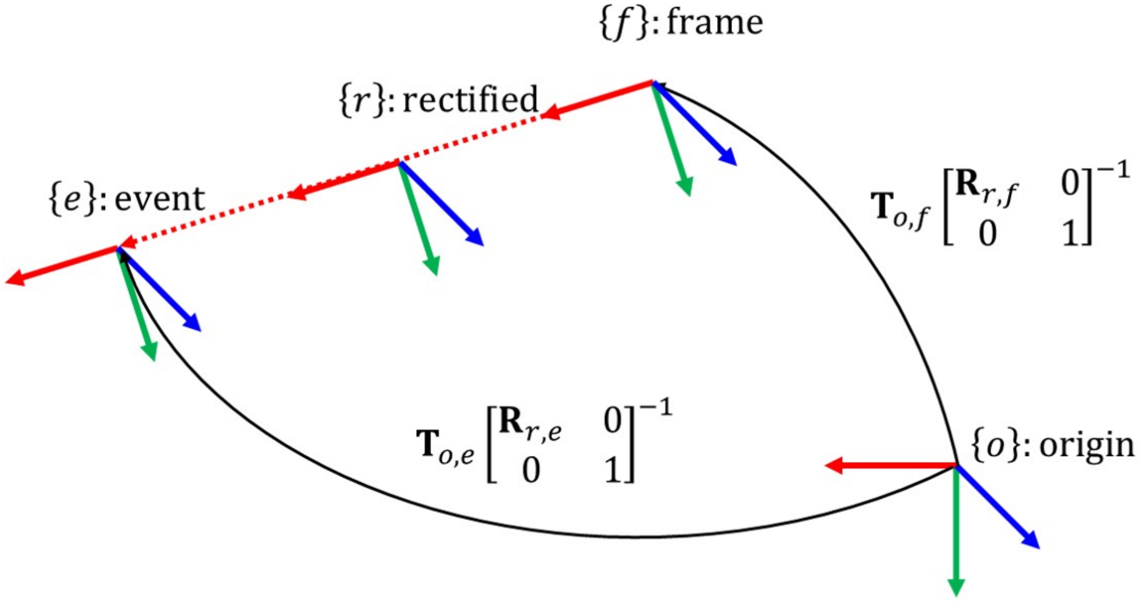


Figure 2.2: The camera coordinates before rectification.



$\{f\}$: frame
$\{r\}$: rectified
$\{e\}$: event
$\mathbf{T}_{o,f}\begin{bmatrix}\mathbf{R}_{r,f} & 0\\ 0 & 1\end{bmatrix}^{-1}$
$\mathbf{T}_{o,e}\begin{bmatrix}\mathbf{R}_{r,e} & 0\\ 0 & 1\end{bmatrix}^{-1}$
$\{o\}$: origin

Figure 2.3: The camera coordinates after rectification.

y-axis and the z-axis by processing cross-product twice to the temporary z-axis. The temporary z-axis determines the y-offset of the rectified image coordinate. If there is a large empty space at the top or bottom of the rectified image, the temporary z-axis should be adjusted. Most simply, the temporary z-axis $\tilde{\mathbf{r}}^z_{o,r}$ can be set as $\mathbf{r}^z_{o,f}$. In this dissertation, I set the z-axis as follows:

$$\tilde{\mathbf{r}}^z_{o,r} = \mathbf{r}^z_{o,f} + \mathbf{r}^z_{o,e} \tag{2.14}$$

Then, $\mathbf{r}^y_{o,r}$ and $\mathbf{r}^z_{o,r}$ are computed as follows:

$$\mathbf{r}^y_{o,r} = \frac{\tilde{\mathbf{r}}^z_{o,r} \times \mathbf{r}^x_{o,r}}{\|\tilde{\mathbf{r}}^z_{o,r} \times \mathbf{r}^x_{o,r}\|} \tag{2.15}$$

and

$$\mathbf{r}^z_{o,r} = \frac{\mathbf{r}^x_{o,r} \times \mathbf{r}^y_{o,r}}{\|\mathbf{r}^x_{o,r} \times \mathbf{r}^y_{o,r}\|} \tag{2.16}$$

The orientation of the rectified coordinate $\mathbf{R}_{o,r} = [\mathbf{r}^x_{o,r}, \mathbf{r}^y_{o,r}, \mathbf{r}^z_{o,r}]$ is now obtained. Finally, the rectified matrices for the frame camera and the event camera are computed as follows:

$$\mathbf{R}_{r,f} = \mathbf{R}^\intercal_{o,r}\mathbf{R}_{o,f} \tag{2.17}$$

$$\mathbf{R}_{r,e} = \mathbf{R}^\intercal_{o,r}\mathbf{R}_{o,e} \tag{2.18}$$

2.3 Non-linear Optimization

The pose estimation and the feature tracking is proposed by using the contrast maximization approach. Various optimizers can be considered to solve contrast maximization, from the basic gradient ascent (or descent) optimizer to Gaussian Newton, Levenberg–Marquardt (LM), Adaptive Gradient (Adagrad), Root Mean Square propagation (RMS-prop) and Adaptive Moment Estimation (Adam).

However, some optimizers are not applicable to contrast maximization problem. The contrast

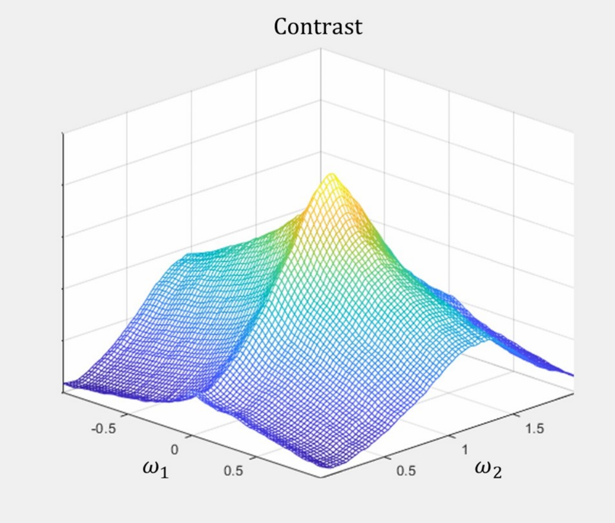


Figure 2.4: Cost function in contrast maximization problem for rotational velocity estimation. In order to visualize cost in 3D space, the cost is depicted with varying the first and second element of the rotational velocity.

cost of the rotational velocity is shown in Fig. 2.4. The Gaussian newton and Levenberg–Marquardt optimizers adjust the step size using the second-order gradient, and it converges fast in convex optimization problems. However, most contrast maximization problems have strong non-linearity and are not convex (for minimization) or concave (for maximization) optimization problems. Therefore, I apply the first-order gradient-based optimizer. Among the first-order gradient methods, the proposed method used the RMS-prop method, which is an optimization method that can converge quickly by considering momentum.

For Jacobian $\nabla_x J(x)$ of the cost $J(x)$, RMS-prop updates the state x as follows.

$$G \leftarrow \gamma G + (1-\gamma)(\nabla_x J(x) * \nabla_x J(x)) \tag{2.19}$$

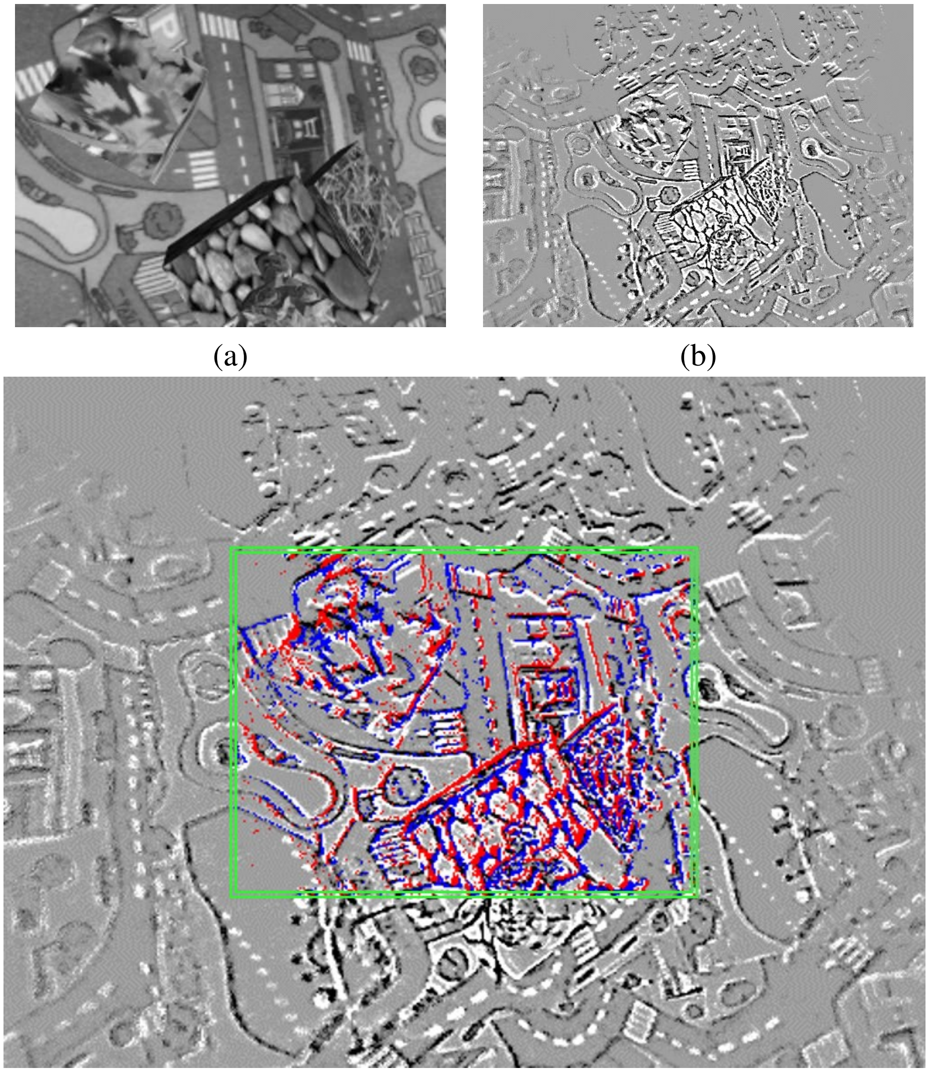
$$x \leftarrow x + \frac{\eta}{\sqrt{G+\varepsilon}} \nabla_x J(x) \tag{2.20}$$

where G is the momentum variable, γ is the smoothing factor ($0 < \gamma < 1$), $*$ is the element-wise product (Hadamard product), η is the step size and $\varepsilon = 1e-12$ is an offset to avoid division by zero problem.

3

Real-time Rotational Motion Estimation with Contrast Maximization over Globally Aligned Events

In this chapter, I propose a method for contrast maximization which utilizes the events observed for a long time through the global events alignment. The proposed method estimates the rotational position by localizing the event points on the globally aligned events, as depicted in Fig. 3.1. The globally aligned events refer to the alignment of all event points to the spatial coordinates of the initial time and are different from the map that represents the intensity of the surrounding environment. I propose the contrast maximization over the globally aligned events with a little additional computational load. I expect that the proposed method can be employed in other applications of contrast maximization framework such as estimating depth and optical-flow.



(a)

(b)

(c)

Figure 3.1: Snapshots of data set ESIM:*OpenGL*. (a) is a gray image for comparison. (b) and colored red and blue in (c) show locally aligned events, and a wide view of a globally aligned event image with estimated poses is shown in grayscale in (c). Existing contrast maximization utilizes the events only in the current temporal window, as in (b). On the other hand, the proposed method calculates the contrast considering globally aligned events as in (c).

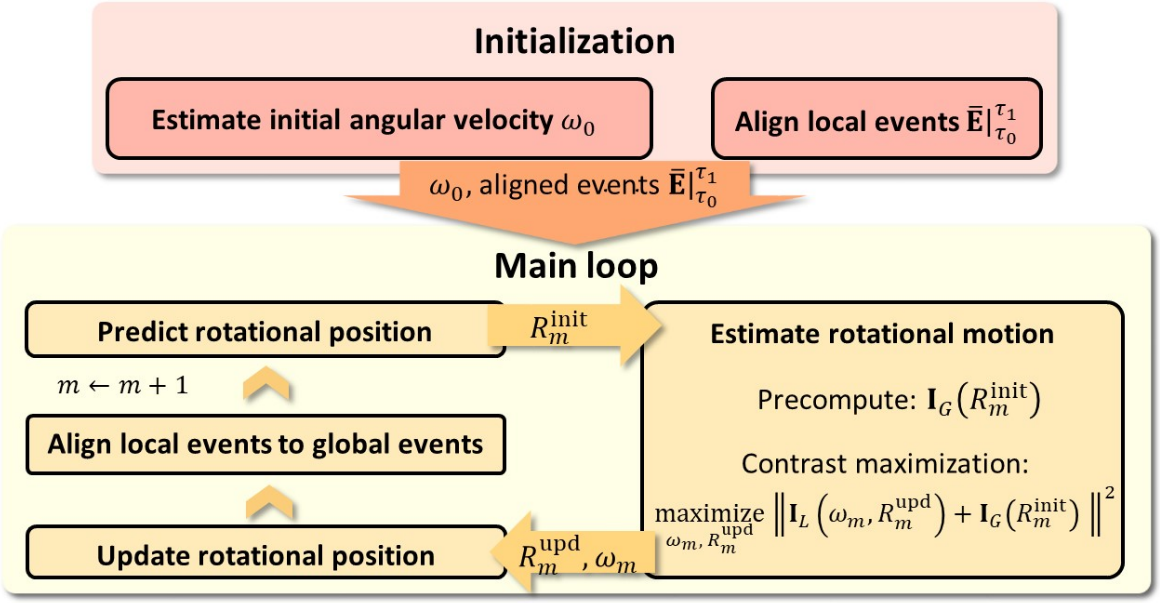

Initialization
Estimate initial angular velocity ω_0
Align local events $\bar{\mathbf{E}}|_{\tau_0}^{\tau_1}$
ω_0, aligned events $\bar{\mathbf{E}}|_{\tau_0}^{\tau_1}$
Main loop
Predict rotational position
R_m^{init}
Estimate rotational motion
Precompute: $\mathbf{I}_G(R_m^{\text{init}})$
Contrast maximization:
$\underset{\omega_m, R_m^{\text{upd}}}{\text{maximize}} \left\| \mathbf{I}_L\left(\omega_m, R_m^{\text{upd}}\right) + \mathbf{I}_G\left(R_m^{\text{init}}\right) \right\|^2$
$m \leftarrow m+1$
Align local events to global events
Update rotational position
$R_m^{\text{upd}}, \omega_m$

Figure 3.2: Overview of the rotational motion estimation algorithm.

3.1 Method

The proposed algorithm simultaneously estimates the rotational motion and aligns all events. This chapter will describe the rotational motion estimation module by dividing the details into the estimation of rotational velocity, global events alignment, and the estimation of rotational position. I will explain the modified rotational velocity estimation method from [1], and then explain a new process for estimating the rotational position using globally aligned events. The block diagram of the proposed method is depicted in Fig. 3.2.

3.1.1 Event Image and Warping Function

It is difficult to obtain enough data from just a single event point to build a meaningful model. Thus, most algorithms process the group of events observed in a certain temporal window or the constant number of events. If the camera moves slowly, a set of events may include multiple rotational motions which cannot be expressed as a single rotational motion. This can be addressed by reducing the time interval or the number of events so that the set of events corresponds to a

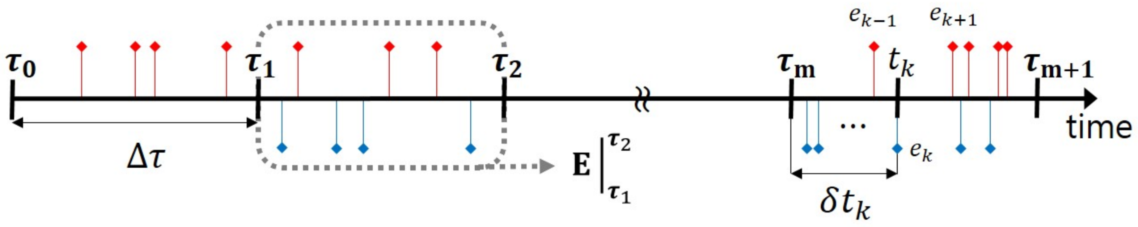


Figure 3.3: Polarity-time graph for the temporal window method.

single motion. However, when the camera moves fast, the method with a small constant number can become inefficient as it may create many event bundles within a short time interval. Thus, I use the constant temporal window with time interval $\Delta\tau$ and represent the bundle of events in a time period $[\tau_m, \tau_{m+1}]$ as

$$\mathbf{E}|_{\tau_m}^{\tau_{m+1}} = \{e_k | \tau_m \leq t_k \leq \tau_{m+1}\}, \tag{3.1}$$

where $\tau_{m+1} = \tau_m + \Delta\tau$. The conceptual illustration is depicted in Fig. 3.3.

Let ω_m be the angular velocity for the event bundle $\mathbf{E}|_{\tau_m}^{\tau_{m+1}}$. Then, the warping function for an event e_k is defined as

$$\mathrm{w}(\mathbf{x}_k, \omega_m, \delta t_k | R) = R \cdot \exp(\hat{\omega}_m \delta t_k)\mathbf{x}'_k. \tag{3.2}$$

Here, δt_k is the time difference between the event point time t_k and the reference time τ_m for the event e_k i.e. $\delta t_k = t_k - \tau_m$. R is the rotation matrix and if it is omitted, the default value of R is the 3×3 identity matrix. The hat operator $\hat{\omega}_m \in \mathbb{R}^{3\times 3}$ represents the cross-product matrix of ω_m, and $\mathbf{x}'_k = [x'_k, y'_k, z'_k]^T \in \mathbb{R}^3$ is an inverse-projected point from $\mathbf{x}_k$ into the camera coordinate.

Then, I obtain the event image aligned with warping function for a pixel $\mathbf{x}$ for the time interval $[\tau_m, \tau_{m+1}]$ as

$$I_m^{\mathrm{raw}}(\mathbf{x}, \omega_m | R) = \sum_{k=1}^{N_m} p_k \cdot \delta_d(\mathbf{x} - \mathrm{w}(\mathbf{x}_k, \omega_m, \delta t_k | R)), \tag{3.3}$$

where δ_d is the Dirac delta function, and N_m is the cardinality of $\mathbf{E}|_{\tau_m}^{\tau_{m+1}} = \{e_k\}_{k=1}^{N_m}$. To abbreviate notation, I omitted the projection function that projects events from the camera coordinate to the image coordinate when obtaining the event image. In the contrast maximization, as convergence in areas with higher intensity is strengthened, events in areas with relatively low intensity are not aligned properly. To address this, the event images are clamped with threshold ρ as

$$I_m(\mathbf{x},\omega_m|R) = \begin{cases} \pm\rho & |I_m^{\mathrm{raw}}(\mathbf{x},\omega_m|R)| > \rho \\ I_m^{\mathrm{raw}}(\mathbf{x},\omega_m|R) & \text{otherwise.} \end{cases} \tag{3.4}$$

I let $\mathbf{I}_m(\omega_m|R)$ denote the matrix whose elements are $I_m(\mathbf{x},\omega_m|R)$.

3.1.2 Estimation Problem of Rotational Velocity

The cost function Eq. (3.5) is used to estimate the angular velocity for the interval $[\tau_m,\tau_{m+1}]$ with the RMS-prop optimizer.

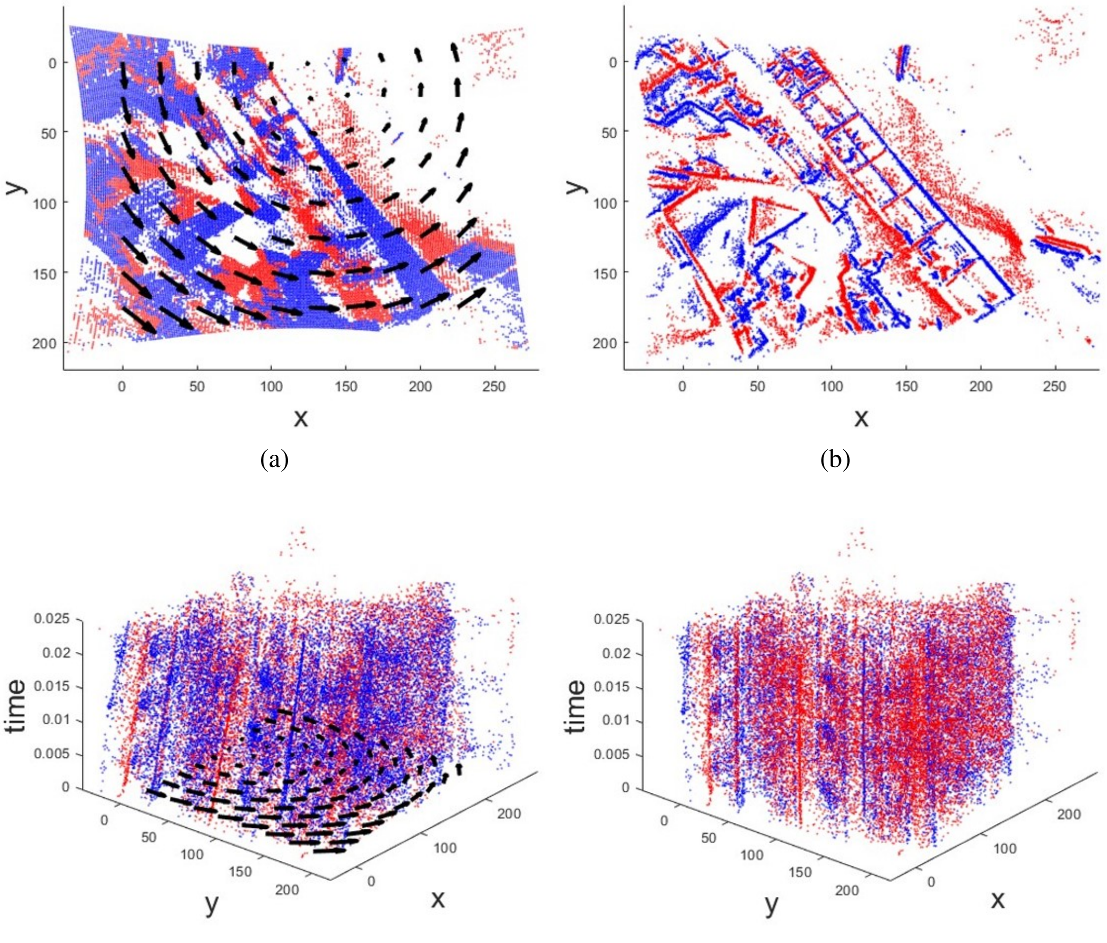
$$\underset{\omega_m}{\text{maximize}}\ J(\omega_m), \tag{3.5}$$

$$\text{where } J(\omega_m) = \|\mathbf{I}_m(\omega_m)\|^2, \tag{3.6}$$

where $\|\cdot\|^2$ denotes the squared Frobenius norm. The cost function $J(\omega_m)$ is the contrast of the event image of the m-th temporal window, which equals the sum of squares reward in [52]. Its Jacobian is computed as follows.

$$\begin{aligned} \frac{dJ(\omega_m)}{d\omega_m} = &\sum_{k=1}^{N_m} 2I_m(\mathbf{x}_k,\omega_m)\ \nabla I_m(\mathbf{x}_k,\omega_m) \\ &\cdot \begin{bmatrix} -\bar{x}_k\bar{y}_k f_x & (1+\bar{x}_k^2)f_x & -\bar{y}_k f_x \\ -(1+\bar{y}_k^2)f_y & \bar{x}_k\bar{y}_k f_y & \bar{x}_k f_y \end{bmatrix} \delta t_k, \end{aligned} \tag{3.7}$$

where $\nabla I_m(\mathbf{x}_k,\omega_m)$ is the gradient of $I_m(\mathbf{x}_k,\omega_m)$, f_x, f_y is the camera focal length in the projection function, and $\bar{x}_k,\bar{y}_k$ is the normalized position satisfying $\bar{x}_k = x'_k/z'_k$, $\bar{y}_k = y'_k/z'_k$.



(a)

(b)

(c)

(d)

Figure 3.4: Example of warping event data. (a), (c) event points before warping. (b), (d) event points warped with the angular velocity which maximizes the event image contrast. The blue and red dots represent different polarities. As a result of warping, events are aligned parallel to the time axis.

3.1.3 Warping Function with Rodrigues' Rotation Formula

When warping as shown in Fig. 3.4, each event point should be warped with different δt_k corresponding to each event in Eq. (3.2), which has a high computational load. In this subsection, I describe an accurate warping in a matrix form that does not require computation of Eq. (3.2) per each event.

Within this subsection, the index of the temporal window m is omitted for the sake of notational simplicity. For each temporal window, $\vec{\delta t} \in \mathbb{R}^{1 \times N_m}$ is the stack of δt_k's which satisfies

$\delta t_k = t_k - \tau_m \leq \Delta\tau$, and $\vec{\mathbf{x}} \in \mathbb{R}^{2\times N_m}$ is the stack of $\mathbf{x}_k$.

The authors in [1] proposed a first-order approximation of the warping function as follows:

$$\vec{\mathrm{w}}(\vec{\mathbf{x}}, \omega, \vec{\delta t}|R) = R \cdot (\vec{\mathbf{x}'} + \hat{\omega} \cdot \vec{\mathbf{x}'} * \vec{\delta t}), \tag{3.8}$$

where $*$ is the element-wise multiplication. Since $\hat{\omega} \in \mathbb{R}^{3\times 3}$, $\vec{\mathbf{x}'} \in \mathbb{R}^{3\times N_m}$ and $\vec{\delta t} \in \mathbb{R}^{1\times N_m}$, the right-hand side of Eq. (3.8) satisfies the dimension $\vec{\mathrm{w}}(\vec{\mathbf{x}}, \omega, \vec{\delta t}|R) \in \mathbb{R}^{3\times N_m}$, while $\mathrm{w}(\mathbf{x}_k, \omega, \delta t_k|R) \in \mathbb{R}^{3\times 1}$.

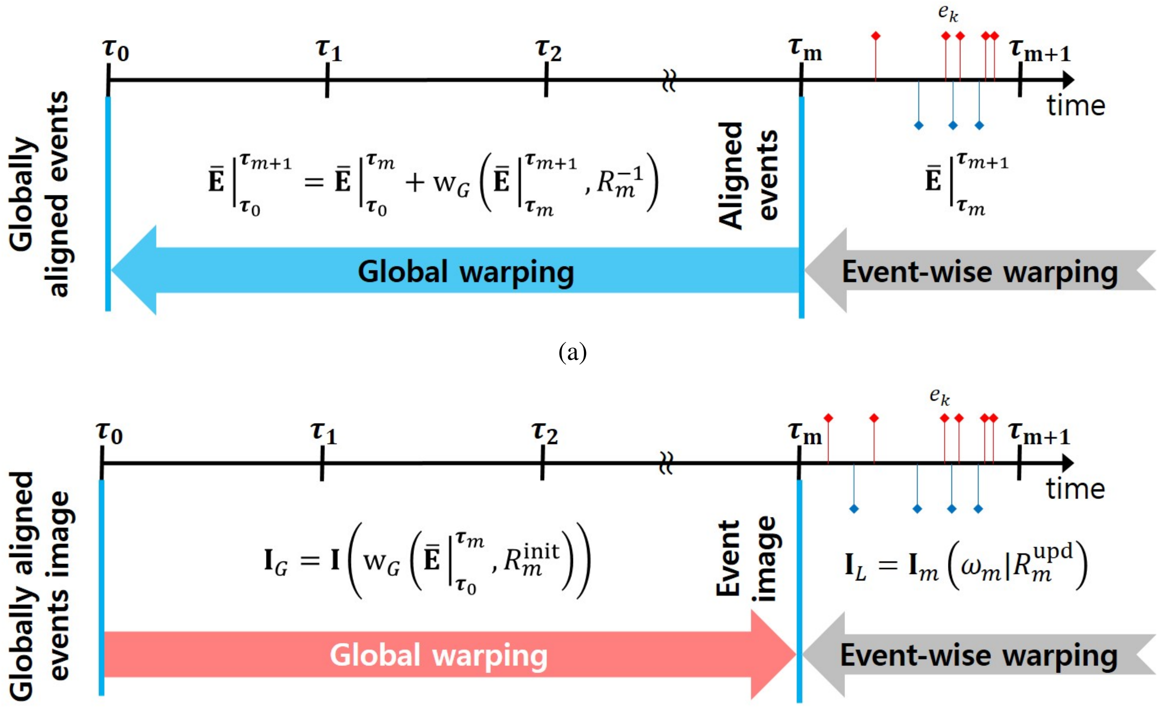
Instead of the linear formula of Eq. (3.8), in order to compute the warping accurately, the Rodrigues' rotation formula is adapted in the proposed warping function Eq. (3.9) for exact warping. Furthermore, the second-order approximation of the warping function can be done by the Taylor expansion of trigonometric functions as in Eq. (3.10).

$$\begin{aligned}
\vec{\mathrm{w}}(\vec{\mathbf{x}}, \omega, \vec{\delta t}|R) = R \cdot \{&\vec{\mathbf{x}'} + \frac{\hat{\omega}}{|\omega|} \cdot \vec{\mathbf{x}'} * \sin(|\omega| * \vec{\delta t}) && (3.9)\\
&+ \frac{\hat{\omega}^2}{|\omega|^2} \cdot \vec{\mathbf{x}'} * (1 - \cos(|\omega| * \vec{\delta t}))\} \\
\approx{}& R \cdot (\vec{\mathbf{x}'} + \hat{\omega} \cdot \vec{\mathbf{x}'} * \vec{\delta t} + \frac{1}{2}\hat{\omega}^2 \cdot \vec{\mathbf{x}'} * \vec{\delta t}^2). && (3.10)
\end{aligned}$$

3.1.4 Global Events Alignment

The proposed algorithm aligns all observed events and uses them to solve the drift error problem in the estimation of rotational position. For the convenience of calculation, The event points are normalized to have the unit distance to the origin. Since the depth of the event point is not necessary for the rotational motion estimation, the normalization does not affect the algorithm. The proposed algorithm continuously aligns events to the initial camera coordinate as shown in Fig. 3.5a. Here, $\mathrm{w}_G(\mathbf{E}, R)$ is the warping function that rotates event points $\mathbf{E}$ by R with respect to the body frame. The algorithm warps the event points $\mathbf{E}|_{\tau_m}^{\tau_{m+1}} = \{e_k\}_{k=1}^{N_m}$ to the camera coordinate at time τ_m as locally aligned events $\bar{\mathbf{E}}|_{\tau_m}^{\tau_{m+1}} = \vec{\mathrm{w}}(\vec{\mathbf{x}}, \omega_m, \vec{\delta t})$, and then warps the locally aligned events $\bar{\mathbf{E}}|_{\tau_m}^{\tau_{m+1}}$ with a rotation matrix R_m to the initial camera coordinate at time τ_0 as $\mathrm{w}_G(\bar{\mathbf{E}}|_{\tau_m}^{\tau_{m+1}}, R_m^{-1})$. The globally aligned events $\bar{\mathbf{E}}|_{\tau_0}^{\tau_m}$ shown in Fig. 3.6 are updated as

$$\bar{\mathbf{E}}|_{\tau_0}^{\tau_{m+1}} \leftarrow \bar{\mathbf{E}}|_{\tau_0}^{\tau_m} + \mathrm{w}_G(\bar{\mathbf{E}}|_{\tau_m}^{\tau_{m+1}}, R_m^{-1}). \tag{3.11}$$

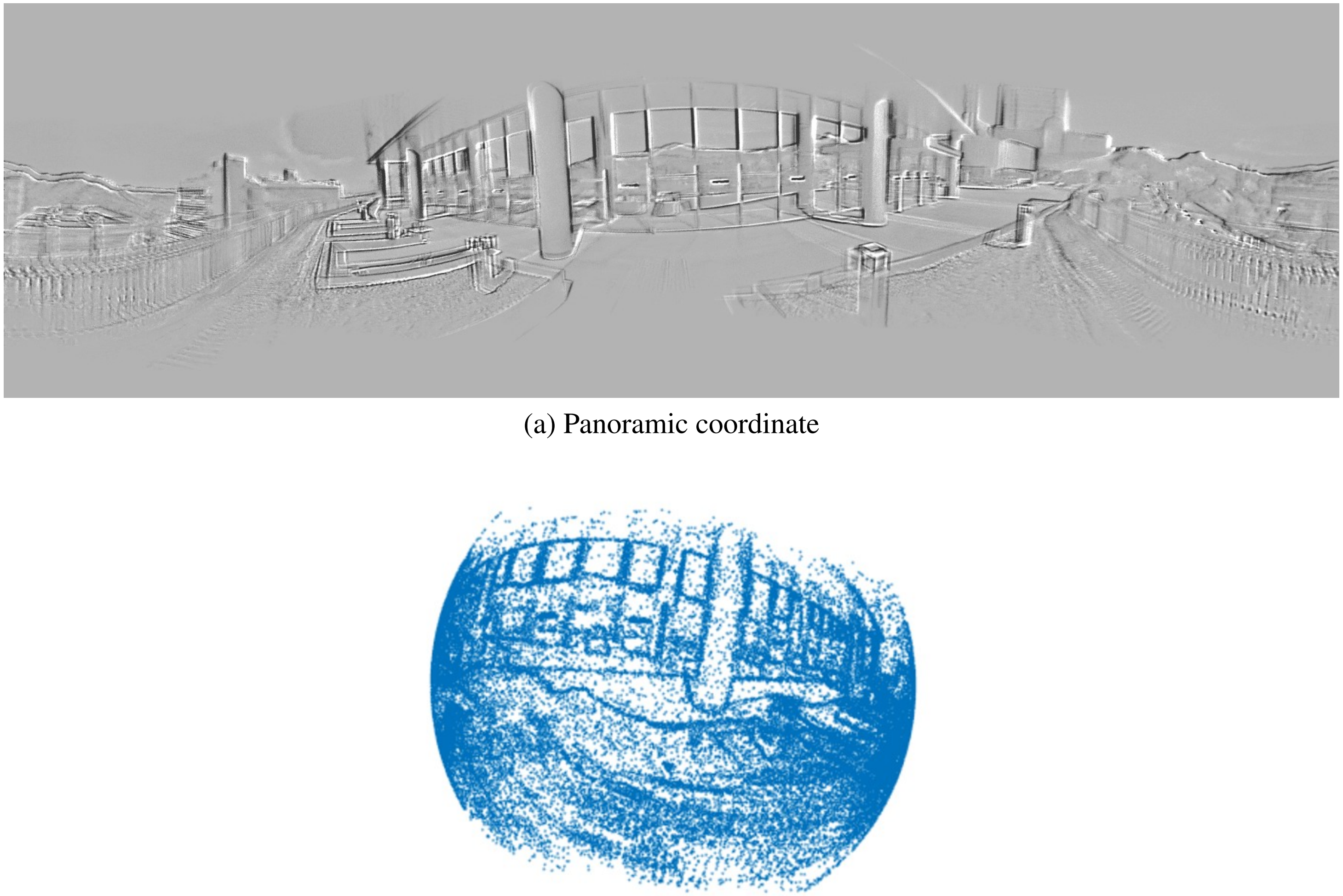

e_k
τ_0
τ_1
τ_2
τ_m
τ_{m+1}
time
Globally aligned events
$\bar{\mathbf{E}}\Big|_{\tau_0}^{\tau_{m+1}} = \bar{\mathbf{E}}\Big|_{\tau_0}^{\tau_m} + \mathrm{w}_G\left(\bar{\mathbf{E}}\Big|_{\tau_m}^{\tau_{m+1}}, R_m^{-1}\right)$
Aligned events
$\bar{\mathbf{E}}\Big|_{\tau_m}^{\tau_{m+1}}$
Global warping
Event-wise warping

(a)

e_k
τ_0
τ_1
τ_2
τ_m
τ_{m+1}
time
Globally aligned events image
$\mathbf{I}_G = \mathbf{I}\left(\mathrm{w}_G\left(\bar{\mathbf{E}}\Big|_{\tau_0}^{\tau_m}, R_m^{\mathrm{init}}\right)\right)$
Event image
$\mathbf{I}_L = \mathbf{I}_m\left(\omega_m | R_m^{\mathrm{upd}}\right)$
Global warping
Event-wise warping

(b)

Figure 3.5: Polarity-time graph for (a) global events alignment and (b) estimation of rotational position.



(a) Panoramic coordinate

(b) Initial camera coordinate

Figure 3.6: Illustrations of globally aligned events for *360° outdoor*.

3.1.5 Estimation of Rotational Position

In this section, I will discuss how to obtain R_m which is used in global alignment. The proposed method estimates the rotational position which maximizes the contrast between the locally aligned event image and the globally aligned event image at the camera coordinate at reference time τ_m as depicted in Fig. 3.5b. In the process of maximization, warping of global events is burdensome to handle many events in real-time. To address this problem, I use a strategy that estimates the update parameter for the initial rotational position. The global events were warped only once for the initial rotational position $R_m^{\text{init}} = R_{m-1}\exp(\hat{\omega}_{m-1}\Delta\tau)$, and the rotational position update parameter R_m^{upd} was estimated by warping the events in the current temporal window in the optimization stage.

The objective function in the proposed contrast maximization of Eq. (3.12) is defined as the contrast of the sum of the locally aligned event image $\mathbf{I}_L$ and the global event image $\mathbf{I}_G$. The image with aligned events $\bar{\mathbf{E}}$ is represented as $\mathbf{I}(\bar{\mathbf{E}})$, so the global event images of the camera coordinate at time τ_m can be represented as $\mathbf{I}(\mathrm{w}_G(\bar{\mathbf{E}}|_{\tau_0}^{\tau_m}, R_m))$.

$$\underset{\omega_m, R_m^{\text{upd}}}{\text{maximize}} \; J(\omega_m, R_m^{\text{upd}}) = \left\| \mathbf{I}_L(\omega_m, R_m^{\text{upd}}) + \mathbf{I}_G(R_m^{\text{init}}) \right\|^2, \tag{3.12}$$

where

$$\mathbf{I}_L(\omega_m, R_m^{\text{upd}}) = \mathbf{I}_m(\omega_m | R_m^{\text{upd}}), \tag{3.13}$$

$$\mathbf{I}_G(R_m^{\text{init}}) = \mathbf{I}(\mathrm{w}_G(\bar{\mathbf{E}}|_{\tau_0}^{\tau_m}, R_m^{\text{init}})). \tag{3.14}$$

Here, the polarity of the local event image and the global event image is ignored for estimating rotational position because the polarity may change according to the camera movement. When estimating the rotational velocity, the polarity of the local event is considered.

The Jacobian matrix for the updating R_m^{upd} is computed as follows.

$$\begin{aligned}\frac{dJ(\omega_m, R_m^{\text{upd}})}{dR_m^{\text{upd}}} &= \sum_{k=1}^{N_m} 2(I_L + I_G)\,(\nabla I_L + \nabla I_G) \\ &\quad \cdot \begin{bmatrix} -\bar{x}_k \bar{y}_k f_x & (1+\bar{x}_k^2) f_x & -\bar{y}_k f_x \\ -(1+\bar{y}_k^2) f_y & \bar{x}_k \bar{y}_k f_y & \bar{x}_k f_y \end{bmatrix}, \end{aligned} \tag{3.15}$$

where I_L, I_G are the intensity of the event image $\mathbf{I}_L$, $\mathbf{I}_G$ at the pixel $\mathbf{x}_k$, ∇I_L and ∇I_G are the intensity of gradient of $\mathbf{I}_L$ and $\mathbf{I}_G$ at the pixel $\mathbf{x}_k$, respectively.

The proposed algorithm solves the cost function Eq. (3.12) with RMS-prop optimizer to estimate the angular velocity and the rotational position simultaneously. Here, $I_G, \nabla I_G$ can be computed in advance, because R_m^{init} does not change. In the optimization process, $I_L, \nabla I_L$ and the last matrix term in Eq. (3.15) are only computed, which are also used computed for the rotational velocity estimation in Eq. (3.7). Then R_m is updated as follows.

$$R_m = R_m^{\text{init}} \cdot R_m^{\text{upd}}. \tag{3.16}$$

Through this strategy, the proposed algorithm runs in real-time by reducing the amount of computation.

3.2 Experimental Results

I evaluate the proposed algorithm on the public data set, the event camera simulator (ESIM) [53], and the data set that I collected.

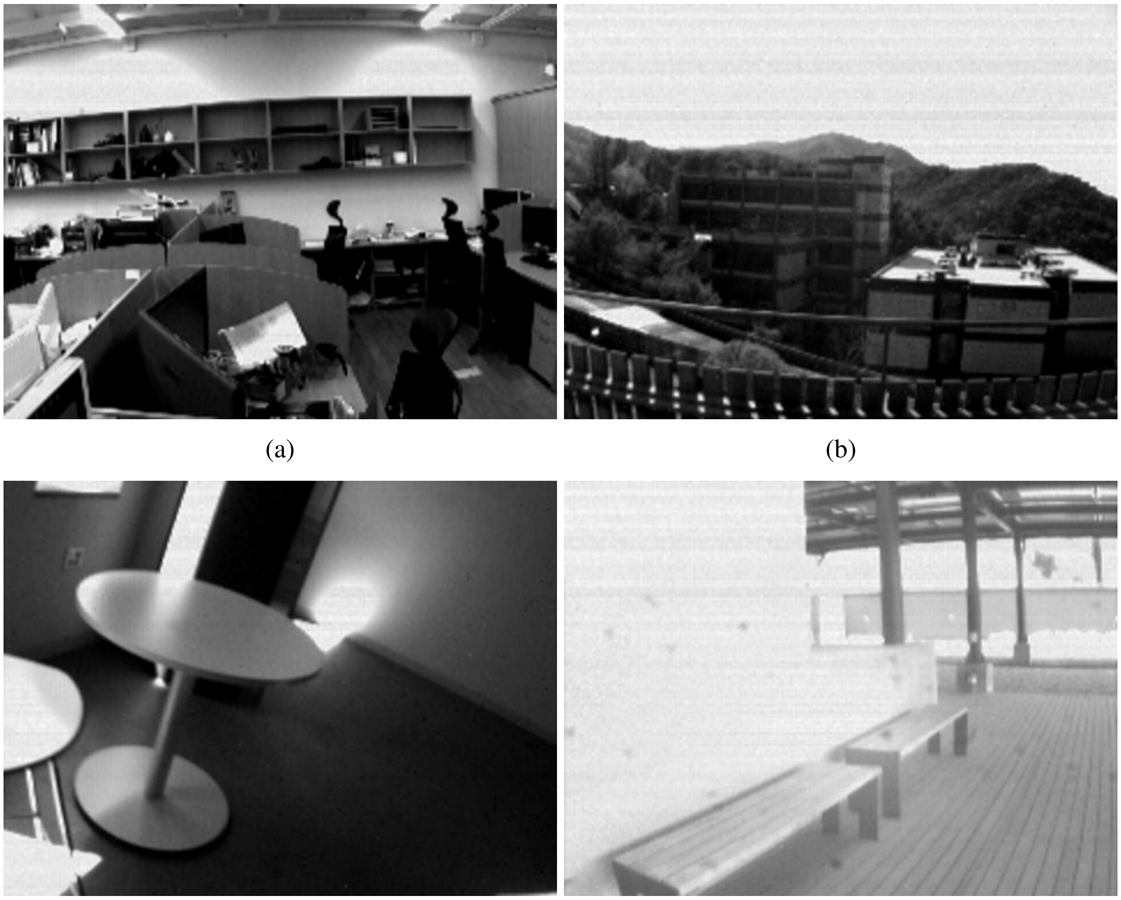
ESIM provides event data for various scenarios. I employed the panoramic renderer and the OpenGL renderer to evaluate the accuracy of the rotational motion estimation. I set IMU measurement noise as a zero-mean Gaussian with 0.2 rad standard deviation and the update rate of the IMU to 1 kHz.

I obtained the rotational motion data set using DAVIS240C and VICON trackers. DAVIS240C can operate as a dynamic vision sensor (DVS, event camera) and an active pixel sensor (APS, gray image camera). I perform camera calibration using the gray image camera. The pixel resolution of the vision sensor is 240×180 and the field of view is 60° on the horizontal axis and 45° on the vertical axis. Also, InvenSense MPU-6150 IMU is supported in DAVIS240C. IMU provides translation acceleration and gyro velocity data at 1kHz. The VICON tracker is used as the ground truth pose, which can accurately estimate the position within the millimeter error range at 100 Hz.

In this section, the angle axes represent the component of the rotation vector with respect to the initial orientation. The coordinate of the initial orientation is shown in Fig. 2.1. Please note that the z-axis is parallel to the principal axis of the initial camera pose.

I set $\rho = 5$, $\Delta\tau = 25$ ms, and compare the proposed algorithm to the integrated gyro velocity data of IMU and the method in [1] in terms of the accuracy of rotational position estimation. For comparison, I re-implemented the method of [1]. I quantitatively evaluated the proposed algorithm with VICON ground truth data. The qualitative evaluation was performed in various environments not limited to the indoor VICON room. The source code and data sets are available at:

https://haram-kim.github.io/Globally_Aligned_Events/



(a)

(b)

(c)

(d)

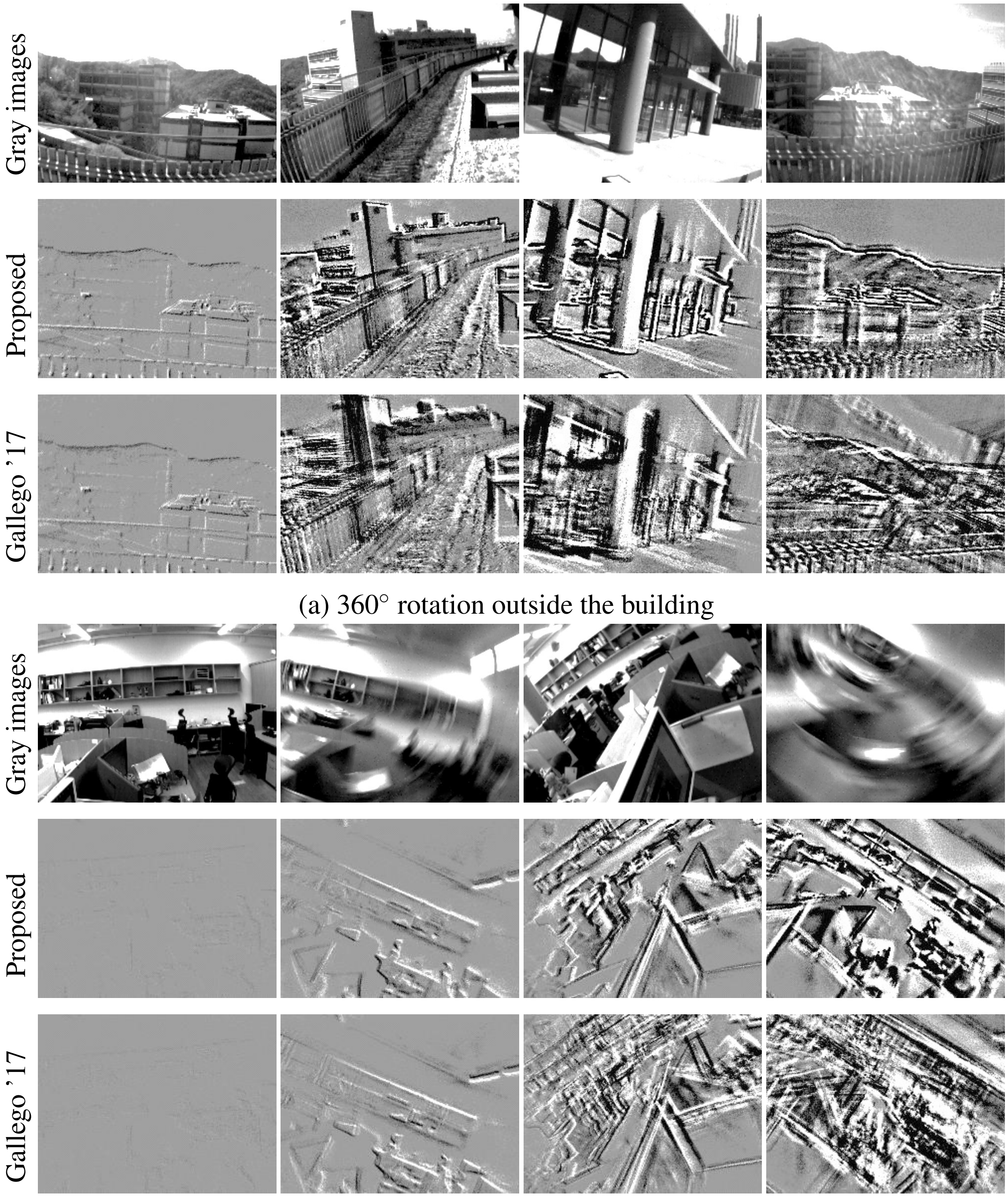
Figure 3.7: Data set acquisition environments: (a) room, (b) outside the building, (c) lobby, (d) rooftop.

3.2.1 Qualitative Evaluation of Global Alignment

I obtained VICON-free data sets from outside the building, in the room, lobby, and at the rooftop Fig. 3.7. The snapshots of two data sets with the most notable results are presented in Fig. 3.8. The polarity is considered for visualization to make the scene easier to understand.

In the data sequence shown in Fig. 3.6 and Fig. 3.8a which is named *360° outdoor*, the camera rotates 360° with respect to Y-axis outside the building. I can qualitatively check the accuracy of rotational motion estimation by checking how well the events are aligned. The first column shows the globally aligned events at the initialization step. In the last column, the edges of the mountain were repeatedly depicted in the proposed method. This is the most impressive result: the initially aligned events from mountain reappear after rotating 360 degrees. This result shows that the rotation drift error is greatly reduced. On the other hand, in the third row, the event images are unclear when the events are aligned by using the integral of the angular velocity of [1]. I acquired data sets handheld, which contain little translational motion. Because of this, in Fig. 3.6, the events of the fences and the poles which are close to the camera are not aligned properly by the estimated rotational motion. On the other hand, the events of distant objects, such as the columns and the mountain reflected in windows, are aligned well. In the last column, the flare phenomenon was detected in the gray image, which is caused by sunlight beyond the photosensitive capacity of the APS sensor, while event cameras are less prone to this problem thanks to the high dynamic range.

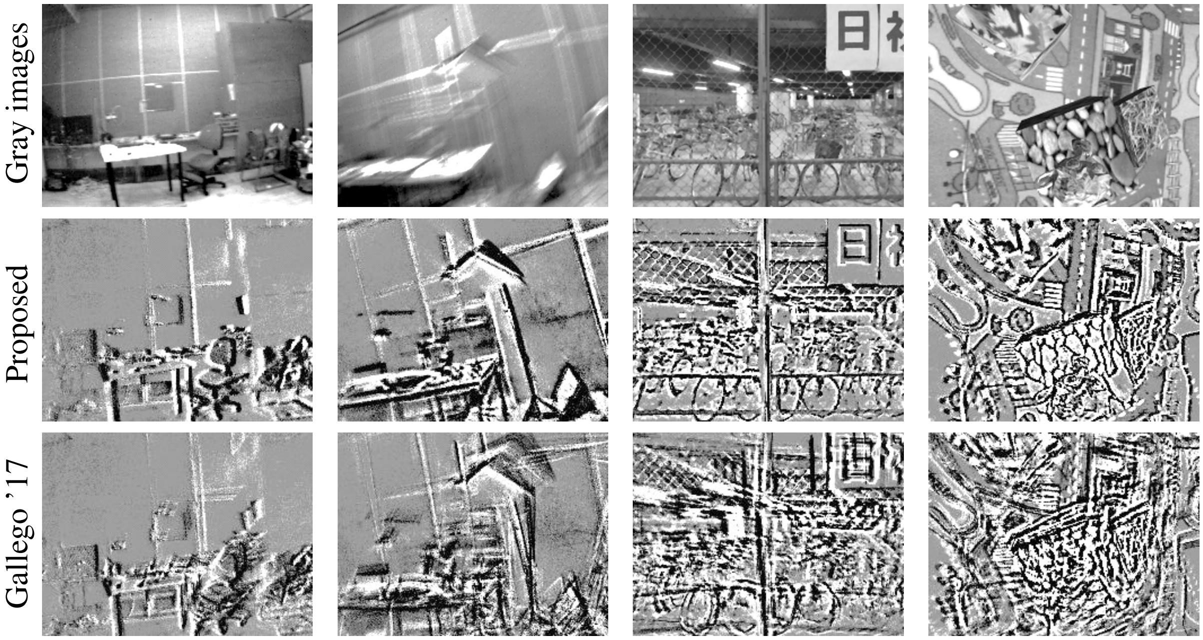
In the *fast roll* data sequence recorded in the room, the event camera rotates repeatedly in the roll direction, up to $1000^{\circ}/s$. Conventional vision algorithms would fail to estimate rotational motion with the gray images of Fig. 3.8b because of the heavy motion blur. However, the proposed algorithm stably aligns all events by accurately estimating the rotational motion even in this fast-moving situation. The last column shows a very fast rotation around $1000^{\circ}/s$, and the proposed method aligned the event points well enough to identify objects such as shelves. In the method obtained by integrating the angular velocity, it is difficult to identify the scene with the aligned event images.



(a) 360° rotation outside the building

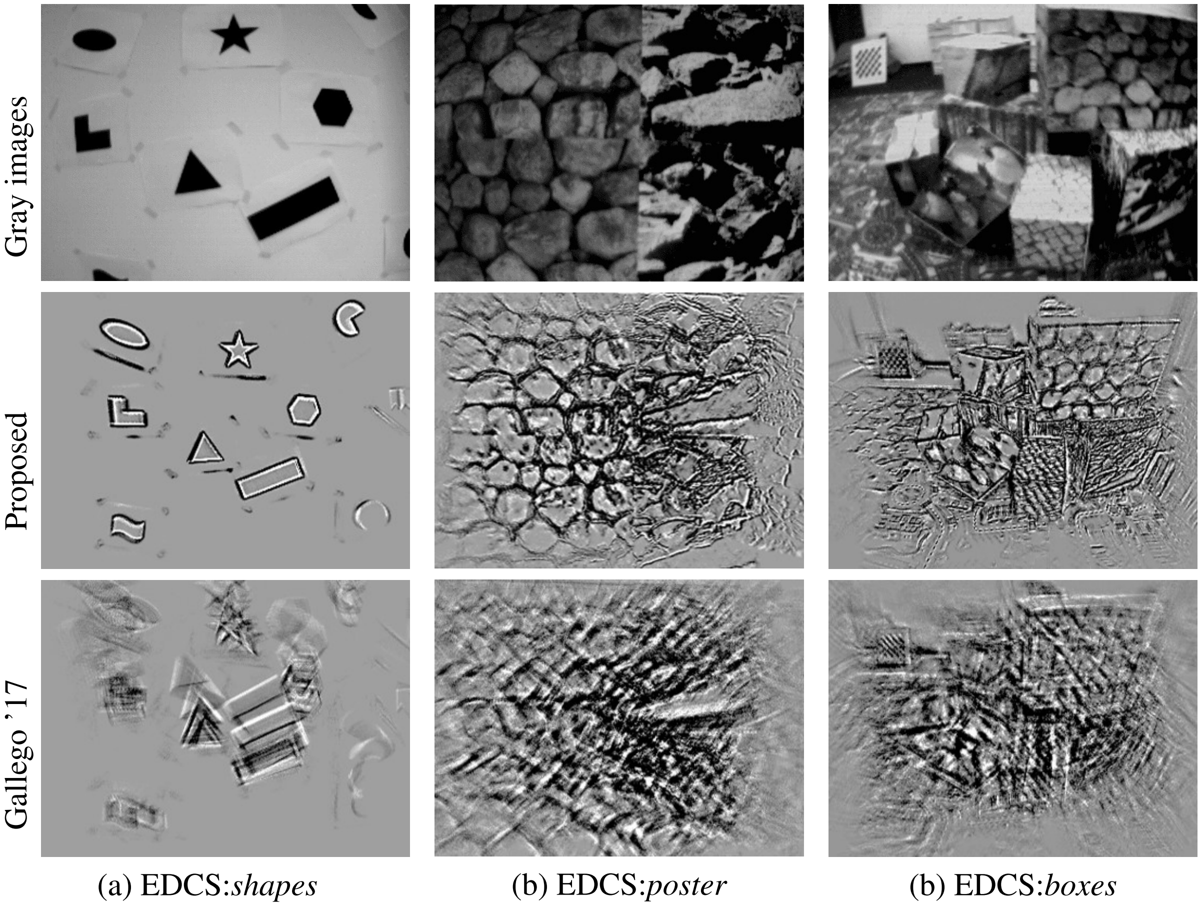
(b) Fast roll rotation in the room

Figure 3.8: Snapshots of the results for *360° outdoor* and *fast roll*. The gray images with the conventional vision sensor (first row), the images of the globally aligned events from the proposed method (second row), and the images of the globally aligned events from the integral of the angular velocity of [1] (third row).



(a) *360° indoor* (b) *fast motion* (c) ESIM: *panorama* (d) ESIM: *OpenGL*

Figure 3.9: Snapshots of the results for *360° indoor*, *fast motion*, *panorama*, *OpenGL*. The gray images with the conventional vision sensor (first row), the images of the globally aligned events from the proposed method (second row), and the images of the globally aligned events from the integral of the angular velocity of [1] (third row).



(a) EDCS:*shapes* (b) EDCS:*poster* (b) EDCS:*boxes*

Figure 3.10: Snapshots of the results for EDCS:*shapes*, *poster*, *boxes*. The gray images with the conventional vision sensor (first row), the images of the globally aligned events from the proposed method (second row), and the images of the globally aligned events from the integral of the angular velocity of [1] (third row).

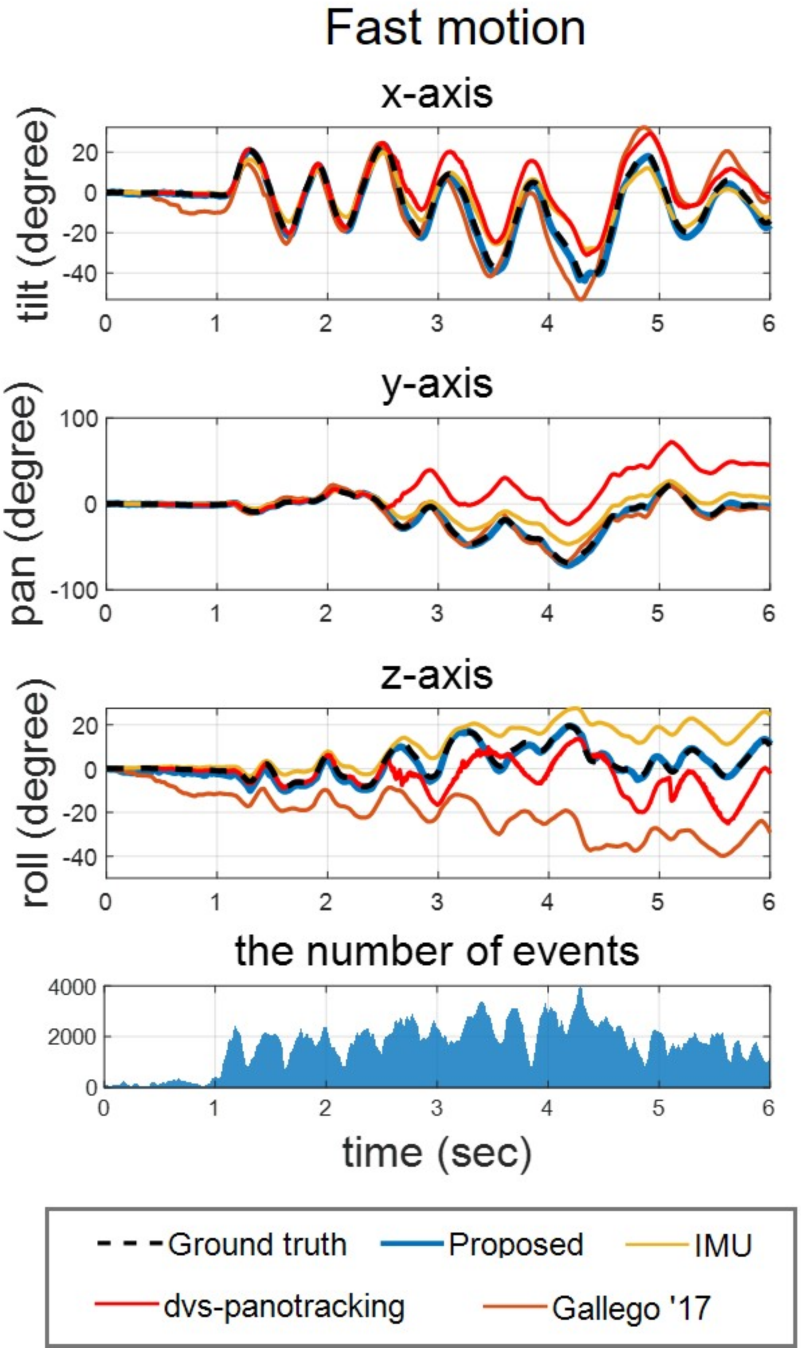


Figure 3.11: Orientation trajectory (first to third row) and the number of events per temporal window used in the proposed method (last row) for *fast motion* data sequence. The trajectories of the proposed method (blue line) are almost identical to the ground truth (black dashed line).

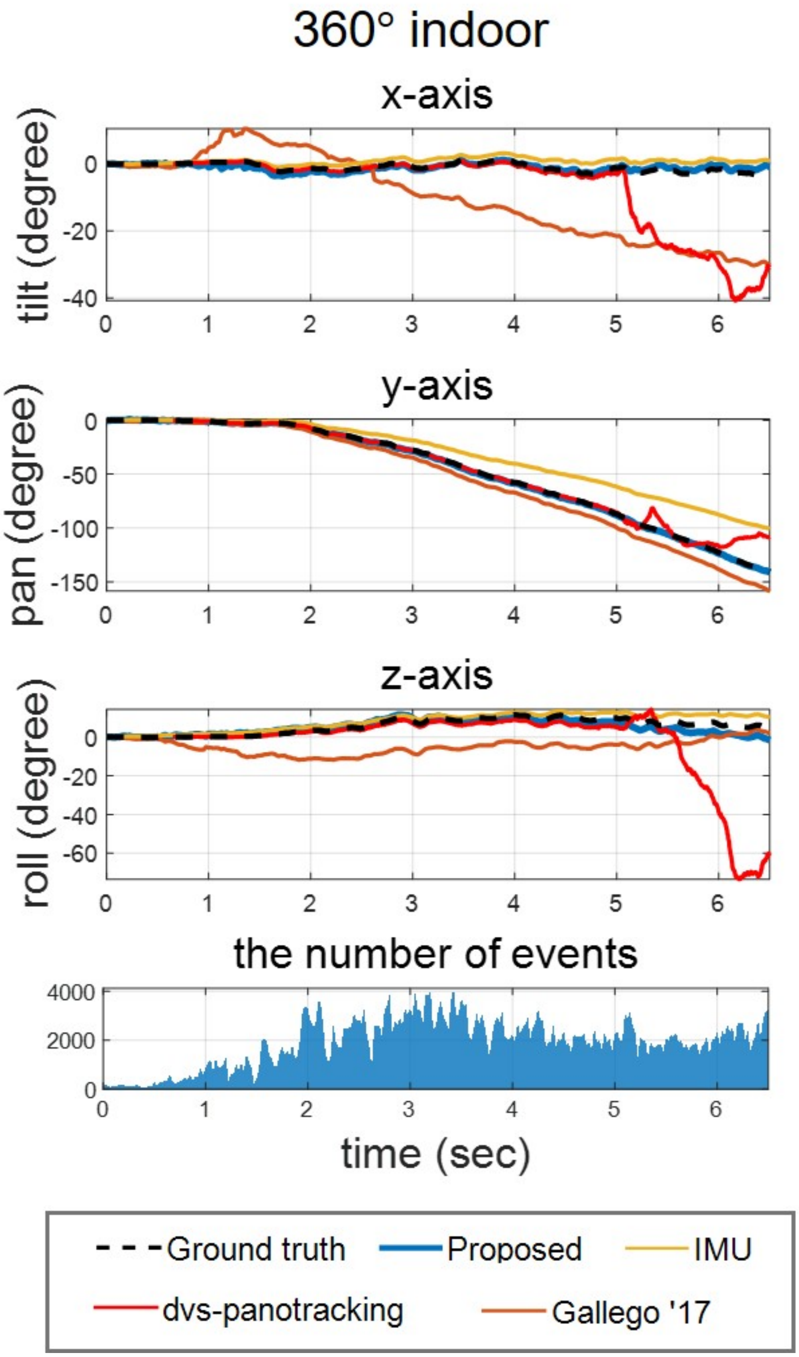


Figure 3.12: Orientation trajectory (first to third row) and the number of events per temporal window used in the proposed method (last row) for *360° indoor* data sequence.

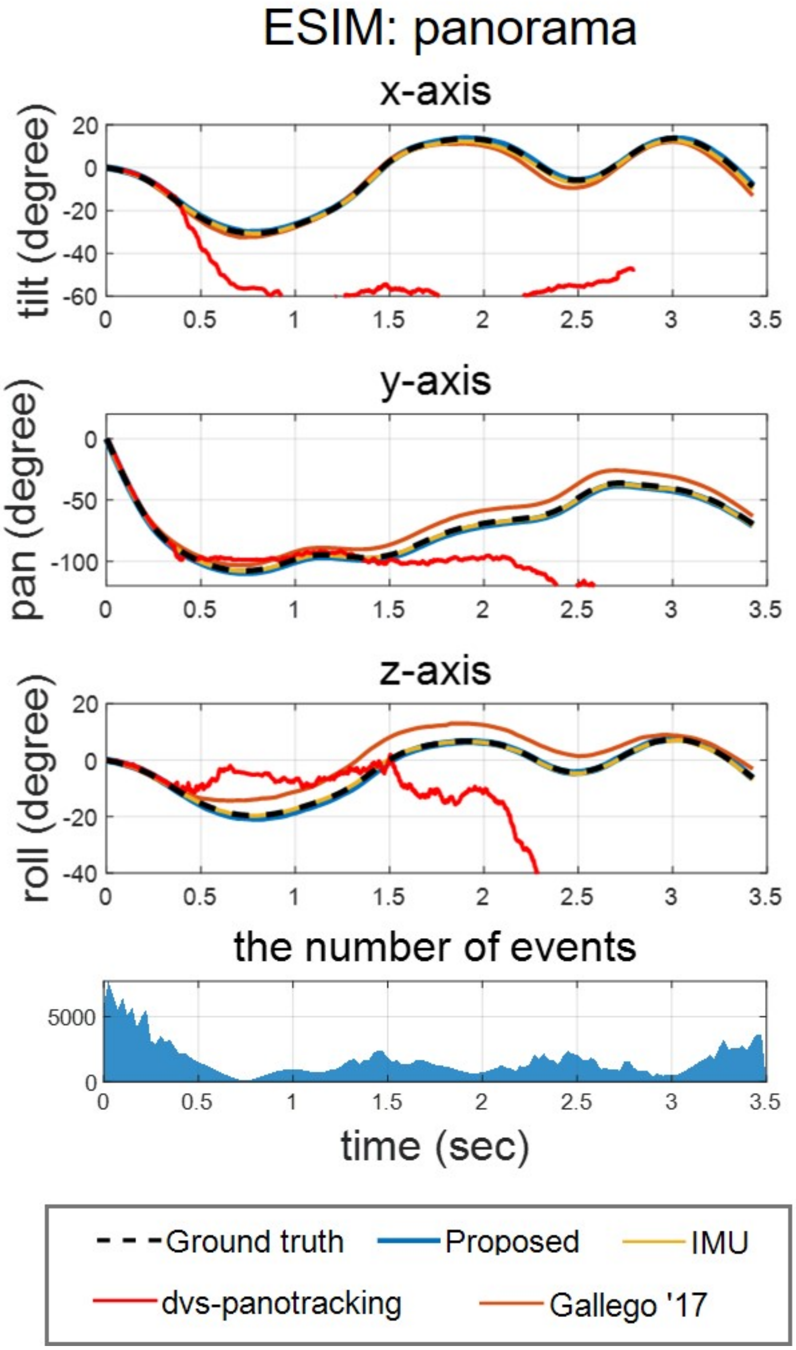


Figure 3.13: Orientation trajectory (first to third row) and the number of events per temporal window used in the proposed method (last row) for ESIM:*panorama* data sequence.

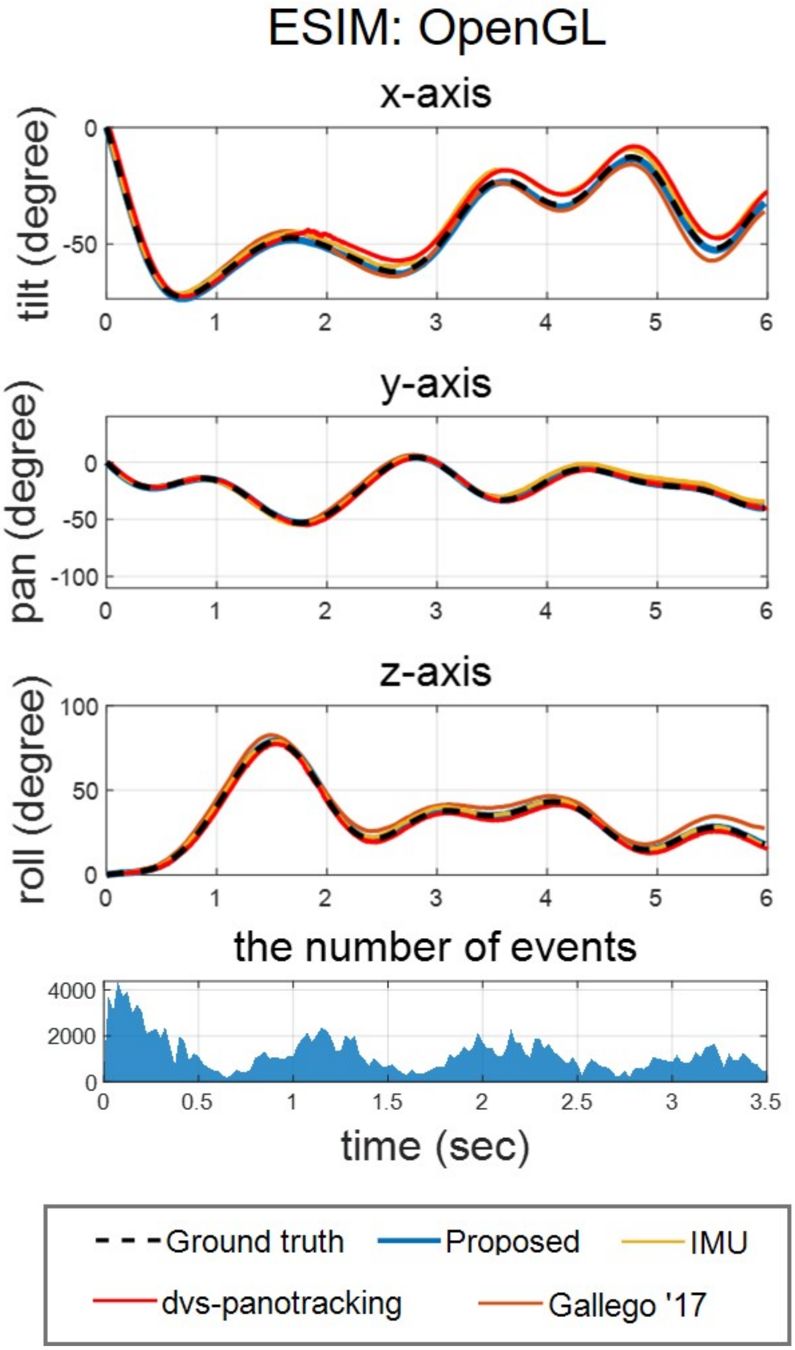


Figure 3.14: Orientation trajectory (first to third row) and the number of events per temporal window used in the proposed method (last row) for ESIM:*OpenGL* data sequence.

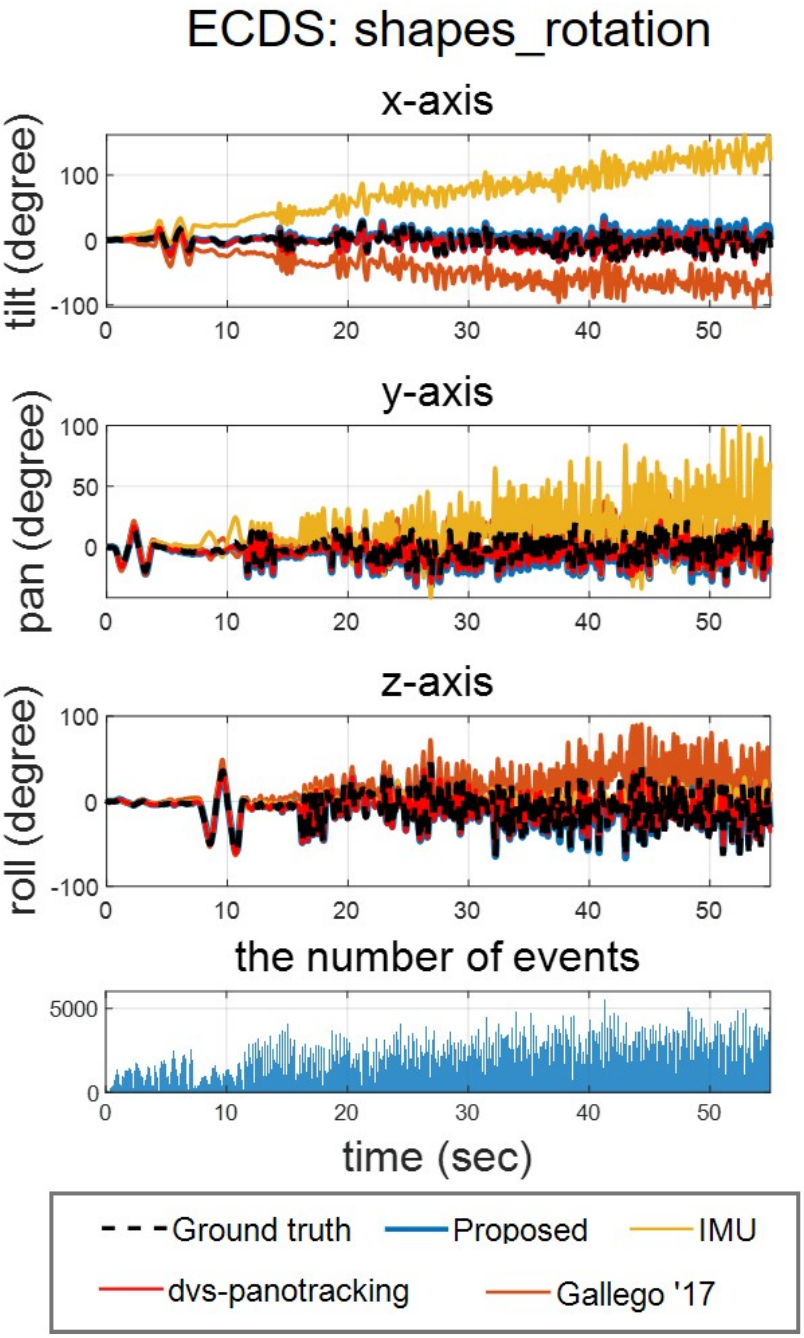


Figure 3.15: Orientation trajectory (first to third row) and the number of events per temporal window used in the proposed method (last row) for ECDS:*shapes* data sequence.

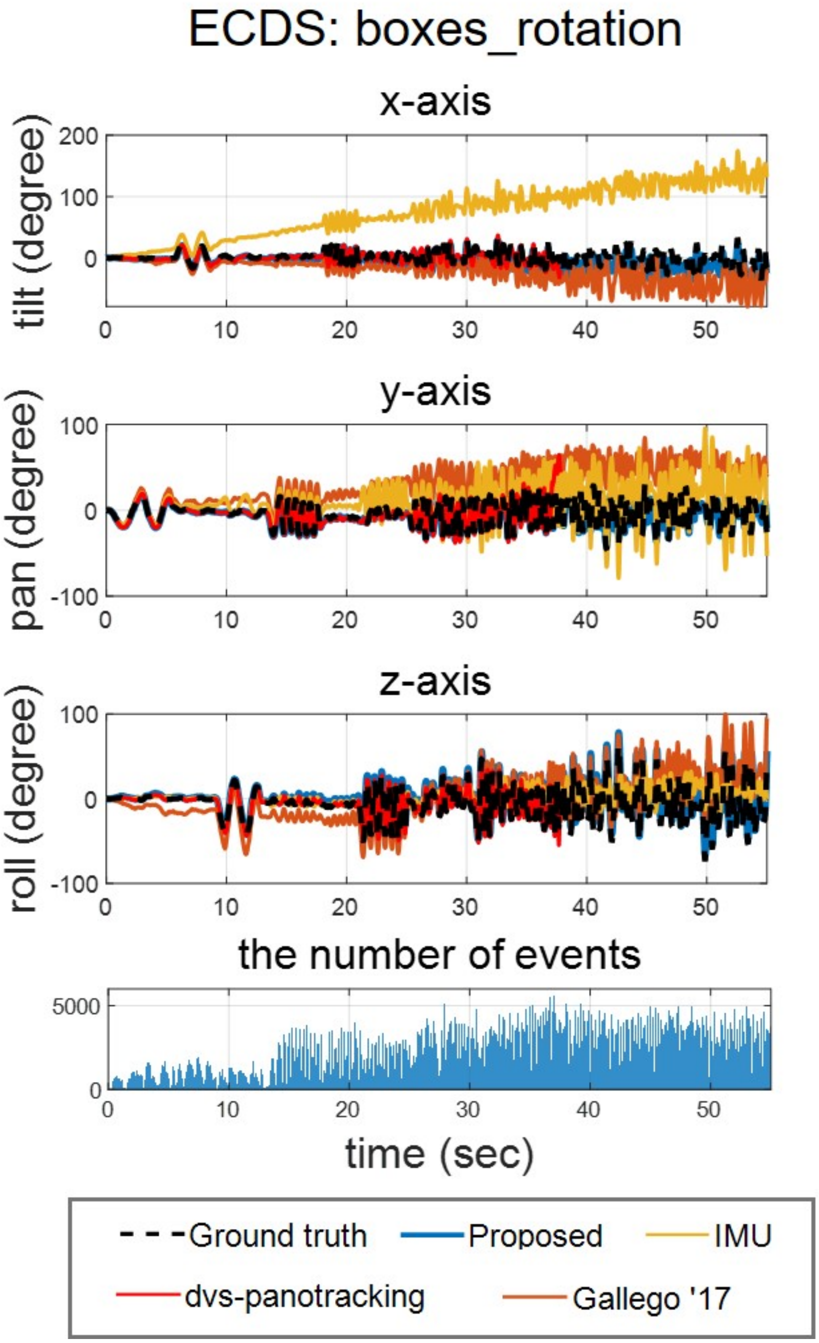


Figure 3.16: Orientation trajectory (first to third row) and the number of events per temporal window used in the proposed method (last row) for ECDS:*boxes* data sequence.

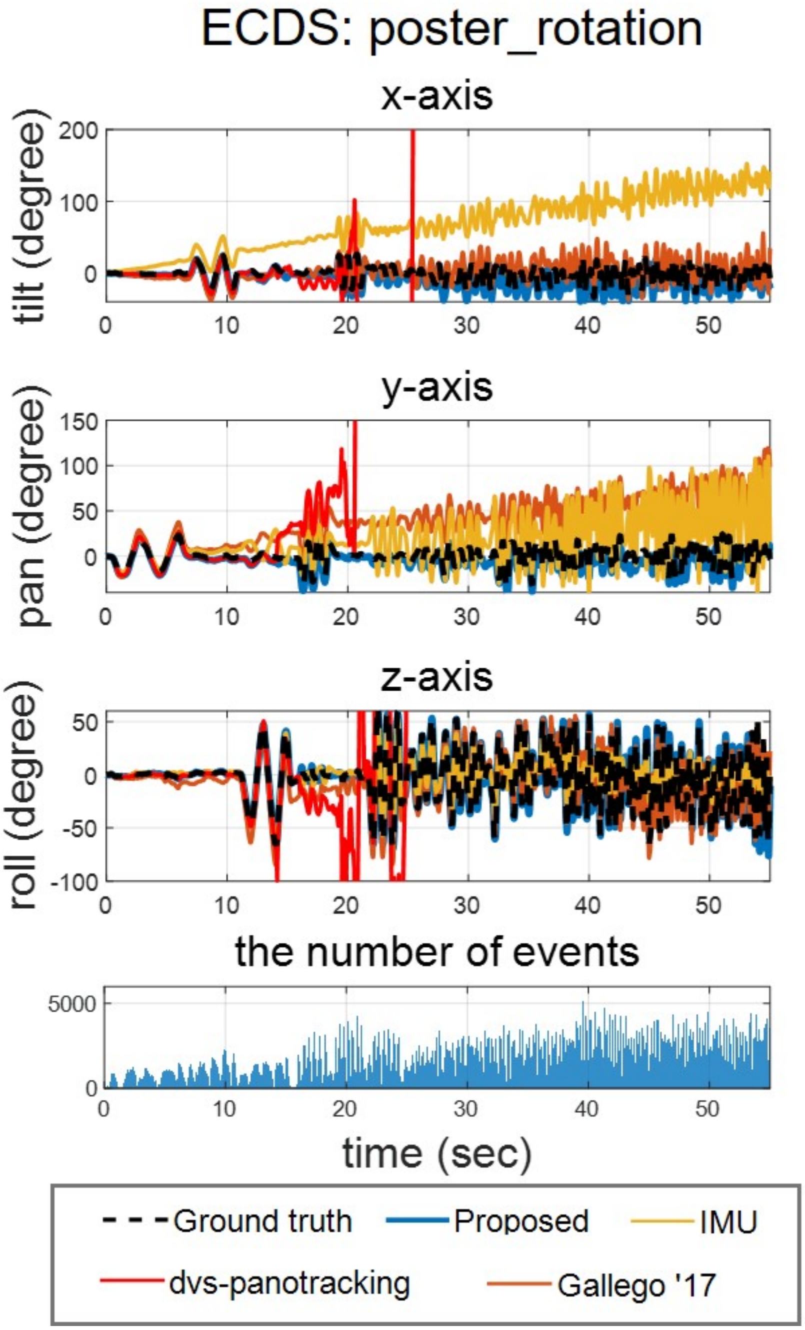


Figure 3.17: Orientation trajectory (first to third row) and the number of events per temporal window used in the proposed method (last row) for ECDS:*poster* data sequence.

The data set from the lobby also contains 360° rotation of the camera, and the data set obtained from the rooftop mainly includes a scene of the ceiling and the floor. The lobby data sequence best demonstrates the HDR advantage of the event camera among the data sets given in Fig. 3.7, and in the rooftop data sequence, the proposed algorithm estimates rotation well in the entire angular range.

3.2.2 Benchmark of Motion Estimation

I evaluate the proposed algorithm with the 360° rotational motion data set with VICON pose data, i.e. the sequence named *360° indoor*, and the high-speed rotational motion sequence *fast motion*, and the event camera simulation sequences ESIM: *panorama*, *OpenGL*. Because the source code for the rotational motion estimation module in [24] is not disclosed, I compare the proposed method with [1, 2] and IMU by integrating the angular velocity. The algorithm of [2] failed and gave not-a-number value in most data sets. I computed the RMS error of the algorithm up to the estimated poses.

I compute the absolute orientation error with the degree unit in a similar concept to the absolute trajectory error (ATE) [55] to evaluate the performance of the algorithm. The root mean square of error was computed for each axis. For example, the X-axis error metric is

$$\mathrm{RMSE(X)} = \sqrt{\frac{1}{T}\sum_{m=1}^{T}(r_m^{\mathrm{esti}}(\mathrm{X}) - r_m^{\mathrm{truth}}(\mathrm{X}))^2}, \tag{3.17}$$

where $r_m = [r_m(X), r_m(Y), r_m(Z)]$ satisfies $R_m = \exp(\hat{r}_m)$, $r_m(\mathrm{X}) \in \mathbb{R}$ is the m-th X-axis angle with degree unit and T is the number of temporal windows in motion estimation. r^{esti} represents the estimated angle, and r^{truth} the ground truth angle. The RMS error and the overall rotational motion plot are shown in Table 3.1 and Fig. 3.11. In the sequence *360° indoor*, the event camera rotates 360° with respect to the Y-axis. There are many events that occurred from the Y-axis rotation, resulting in much smaller drift error on the Y-axis than on the other axes in Table 3.1. In contrast, IMU accumulated more drift error on the Y-axis and less drift error in the other axes.

In the sequence *fast motion*, the proposed method estimates rotation stably despite the high-

RMS error (°)		360° *indoor*	*fast motion*	ESIM [53]: *panorama*	ESIM [53]: *OpenGL*	ECDS [54]: *shapes*	ECDS [54]: *boxes*	ECDS [54]: *poster*
# of events (avg / max)		1917.2 / 4130	1613.0 / 4007	1670.7 / 7736	1235.5 / 4384	1932.2 / 4815	2215.6 / 4897	2022.0 / 5287
Proposed	X	**0.8494**	**1.6556**	**0.3027**	**0.8816**	**8.8142**	**6.0690**	**8.6354**
	Y	**0.7875**	**1.0411**	1.6167	**0.3269**	**2.4661**	**4.6486**	**8.2572**
	Z	**2.5776**	**1.3863**	0.5145	**0.3468**	**7.2347**	**4.3777**	**9.2662**
Gallego [1]	X	16.8523	9.2353	2.2204	2.7918	29.3871	49.0099	17.0044
	Y	10.0389	4.4999	8.0266	2.5701	41.2690	13.8303	59.1020
	Z	11.4191	28.2084	5.0975	4.1400	23.2614	29.7913	14.9381
Reinbacher [2]	X	14.3514 (×)	22.5237 (×)	43.9094 (×)	3.7488	12.9909	10.3854 (×)	142.2918 (×)
	Y	15.2358 (×)	40.1362 (×)	33.7703 (×)	0.7950	12.7940	14.8850 (×)	171.6884 (×)
	Z	38.0566 (×)	11.6826 (×)	25.9544 (×)	2.0672	25.0850	17.3024 (×)	152.0987 (×)
IMU	X	2.3891	5.0420	1.2864*	1.0866*	91.4659	91.6380	86.9751
	Y	20.9446	10.6737	**0.5586***	2.9480*	21.8201	24.0124	30.8752
	Z	3.1554	10.4439	**0.4182***	3.4750*	22.0496	17.4821	16.6411

Table 3.1: Root mean square error of rotational position. Events were uniformly sampled in a given sequence of events to ensure the real-time property. I use the same number of events per temporal window in the proposed method and [1]. # of events in the second row indicates the average and maximum number of events per temporal window. I set the constant number of events in [2] as 1500. * denotes the results of IMU noise setting as $\mathcal{N}(0, 0.2^2\text{rad}^2)$, and (×) denotes that the algorithm failed. I computed the RMS error up to the estimated poses in the failure cases.

speed ego-motion, while [1] and [2] estimate rotation improperly.

In the both sequences, *360° indoor* and *fast motion*, the camera moves very slowly between 0 and 1 seconds as shown in Fig. 3.9, allowing the event camera to detect very few events. In the method from [1], angular velocity estimation frequently falls into the local minimum during the first 1 second. On the other hand, using globally aligned events, the proposed method reliably estimates the rotational motion even though the number of events in the temporal window was small. In the simulation data set, the angular velocity estimation method [1] showed less drift error than in the real-world data sets. In the presence of Gaussian noise in the IMU measurements, the IMU gyro integration method showed a similar level of performance to the proposed method in the ESIM: *panorama*. In ESIM: *OpenGL*, measurement error was accumulated more than in the ESIM: *panorama* data set, resulting in less accurate results than the proposed methods, but the performance is still better than [1].

I verified the performance of the algorithm using the rotation data sets in [54]. The duration of the data sets is about a minute, and the camera moves faster over time. The RMS error evaluated in [54] is shown in Table 3.1. While estimating motion in the sequences of [54] which have longer duration than the other sequences, the RMS error of the proposed method is larger than the other data sets because of the drift error caused by saturation of the contrast image. However, the proposed method showed significant improvements in rotational motion estimation on all the tested data sets. As a future work, I will reduce the drift error even for long-duration data sets by adaptively sampling the globally aligned events to obtain the unsaturated global event image $\mathbf{I}_G$.

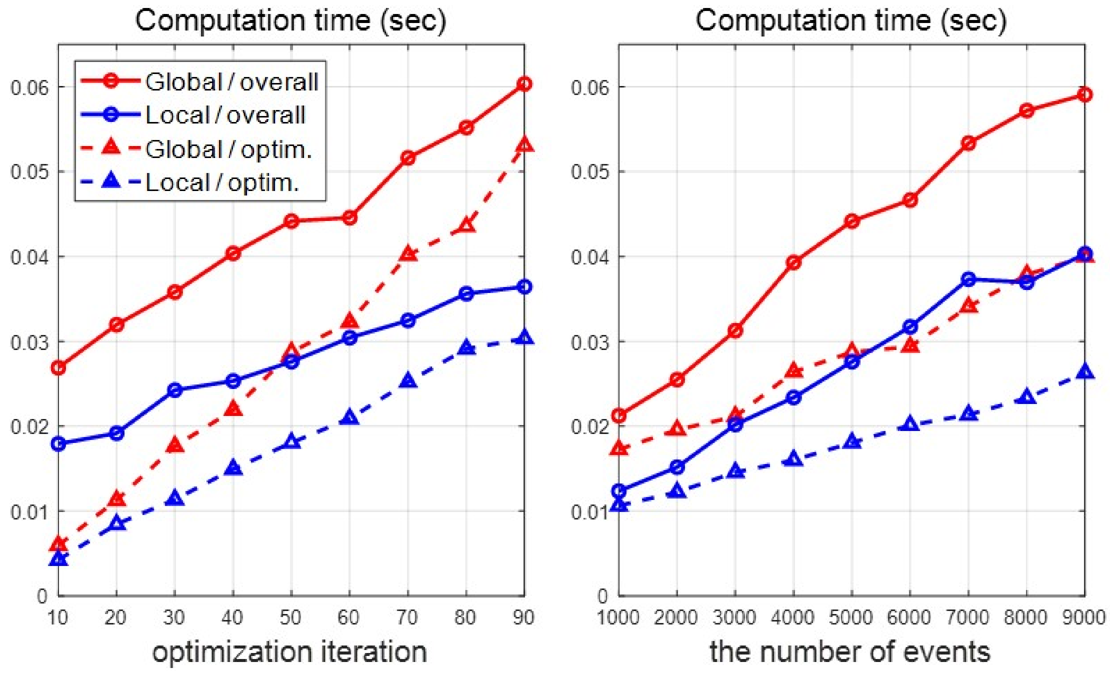
3.2.3 Computation Time

I measured the computation time of the algorithm on a Linux laptop with the i7-7500U@2.7GHz CPU. Even if the temporal window size is constant, the number of events per temporal window can be different depending on motion or surrounding environment, and the calculation time varies according to the number of iterations during optimization. Thus, I measured the computation time varying the number of events and the number of optimization iterations, and analyzed the effect of global alignment and the presence of rotational position estimation on computation time. The results are displayed in Fig. 3.18.

When changing the optimization iteration, the number of events is fixed at 5000, and when changing the number of events, the iteration is fixed at 50. The results show that the time required for the proposed method takes about 50% on average than the existing contrast maximization, and the proposed algorithm achieves angular velocity estimation within tens of ms. The proposed method is highly valuable for contrast maximization frameworks, as drift errors can correct by this computational trade-off.

3.3 Summary

This section, I present the rotational motion estimation method using an event camera only. I use globally aligned events to reliably estimate rotation. The proposed method gives more accurate results than the methods of [1] and [2] in [54] data sets, and shows much higher accuracy than the IMU in real-world data set, within the maximum error of 3 degrees. Also, the algorithm can stably estimate the rotational position in very fast motion. I expect that the proposed method can also be applied to the scale drift problem in monocular depth estimation and the drift problem of the feature tracking with optical flow. In conclusion, the proposed method improves the contrast maximization and runs in real-time with around 50% additional computation only. The provided source code and the data sets will contribute to the community.



(a) (b)

Figure 3.18: Computation time with varying optimization iteration (a), the number of events per temporal window in *360° outdoor data set (b).* The blue line indicates the computation time when only local events (subset events) are processed, and the red line indicates the computation time in the proposed method which utilizes global events. The dashed line represents the time spent in the optimization process, and the solid line represents the total time taken for the contrast maximization. The latency of the system is same as the presented computation time.

4

Real-time Hetero-Stereo Matching for Event and Frame Camera with Aligned Events Using Maximum Shift Distance

The proposed methods estimate the disparity and depth of events by associating the event and frame data. I obtain camera poses from the 3D reconstructed points which are computed from the initial matching method. Then, I estimate accurate disparity and depth by matching the frame edge image to the aligned events using the camera poses. For aligning events, I extend the event alignment module in [47], additionally considering the translation motion and an arbitrary depth in the disparity range. Rather than considering all disparities in range, I efficiently compute the aligned events with maximum shift distance method by considering the representative disparities which produce distinct aligned event images.

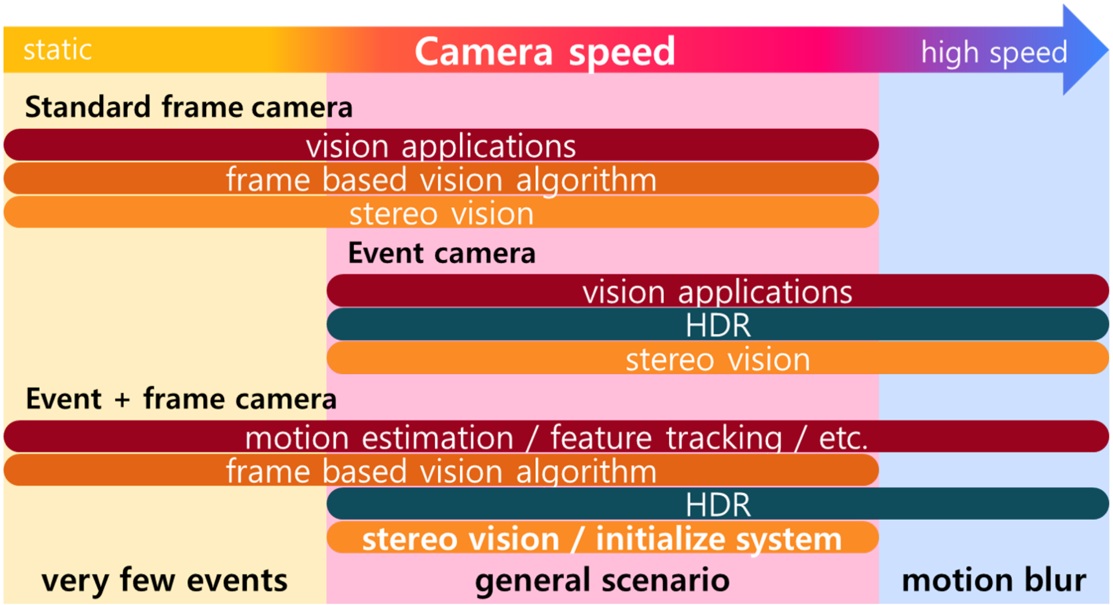


Figure 4.1: Hetero-stereo matching algorithm operating range according to camera speed. The proposed method enables hetero-stereo camera system to apply vision applications (motion estimation, feature tracking, etc.) regardless of camera speed and to have high dynamic range (HDR).

4.1 Hetero Stereo Matching

There are two concepts to associate the event camera with the frame camera. The event data should be reconstructed into frame images [10,56–58], or frame images should be converted into the event camera-like images. Frame reconstruction methods still suffer from unwanted artifacts such as bleeding and local reconstruction error amplification problems. It is confirmed that the artifacts degrade the performance of disparity estimation in Section 4.2. Thus, I will cover the matching method by converting frame images to event images between the two concepts.

For stereo matching, the events and the frame images should be undistorted and rectified. In general for frame images, the inverse mapping technique is used during rectification and undistortion. Likewise, I utilize the inverse mapping to rectify the raw event image for the initial stereo matching of Section 4.1.1. The inverse mapping is an *image-to-image* operation. On the other hand, for aligning events in the Section 4.1.2, The events should be warped *individually* considering their time and camera motion after rectification. This means that warping of individual events

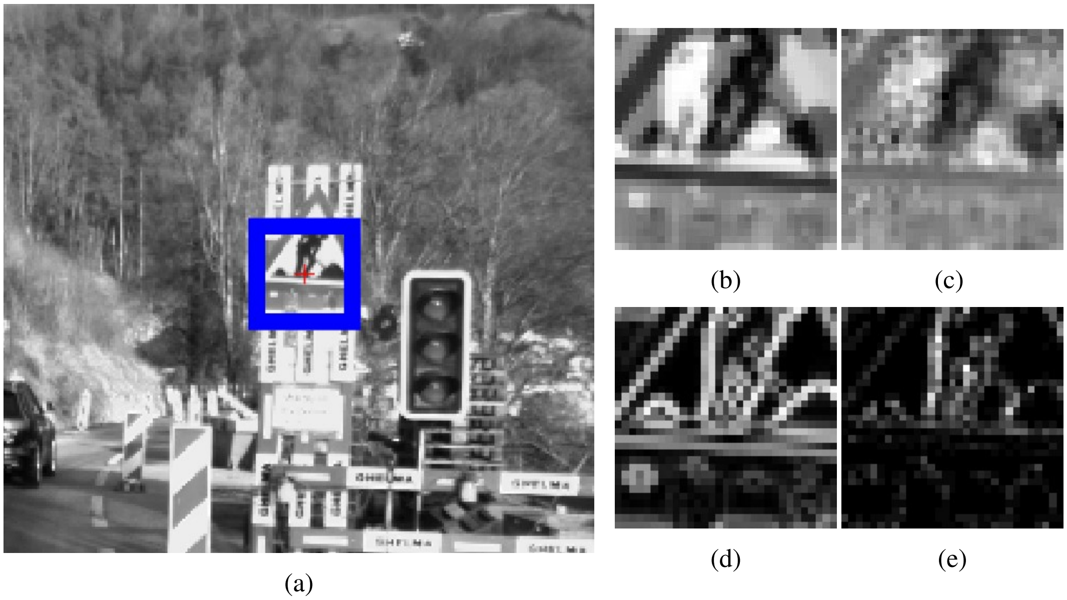

(b) (c) (d) (e) (a)

Figure 4.2: Illustrations of the stereo matching results. (a) frame image. Red cross represents the edge pixel $\mathbf{f}$ and blue square represents patch $\mathbf{P}(\mathbf{f}, {}^F\mathbf{I}_n)$ of the n-th frame image ${}^F\mathbf{I}_n$. (b) and (d) are the frame patch of the temporal gradient image ${}^F\mathbf{I}_n^{\Delta\tau}$ and the edge image ${}^F\mathbf{I}_n^{\Delta\mathbf{x}}$ represented as $\mathbf{P}(\mathbf{f}, {}^F\mathbf{I}_n^{\Delta\tau})$, $\mathbf{P}(\mathbf{f}, {}^F\mathbf{I}_n^{\Delta\mathbf{x}})$, respectively. (c) and (e) are the event patch of the raw event image ${}^E\mathbf{I}_n^{\Delta\tau}$ and the aligned event image ${}^E\mathbf{I}_n^{\Delta\mathbf{x}}$) with the estimated disparity d^* represented as $\mathbf{P}(\mathbf{f}+[d^*,0]^\top, {}^E\mathbf{I}_n^{\Delta\tau})$, $\mathbf{P}(\mathbf{f}+[d^*,0]^\top, {}^E\mathbf{I}_n^{\Delta\mathbf{x}})$, respectively.

cannot be performed after the inverse mapping as in Fig. 4.3 (because the time information of individual events gets lost to perform the image-to-image operation). Thus, I undistort and rectify events in a direct way, before warping the events as in Fig. 4.4.

In the heterogeneous camera case, the resolution of the event camera and the frame camera may be different. If the rectification resolution is set to be different from the resolution of the event camera, the image projected from events will suffer from web-shaped artifacts, because some pixels contain overlapped events or nothing. Likewise, the focal length should not be significantly different from the focal length of the event camera. Thus, I set the resolution and the focal length of rectification coordinate to those of the event camera. In this dissertation, all the event points and frame images are rectified.

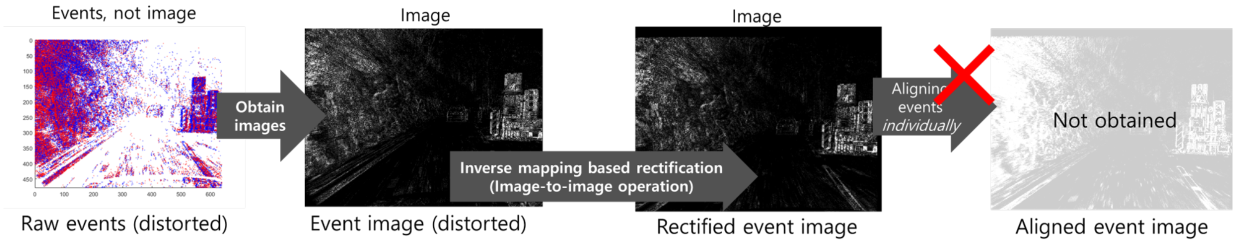


Figure 4.3: Rectification with inverse mapping method.

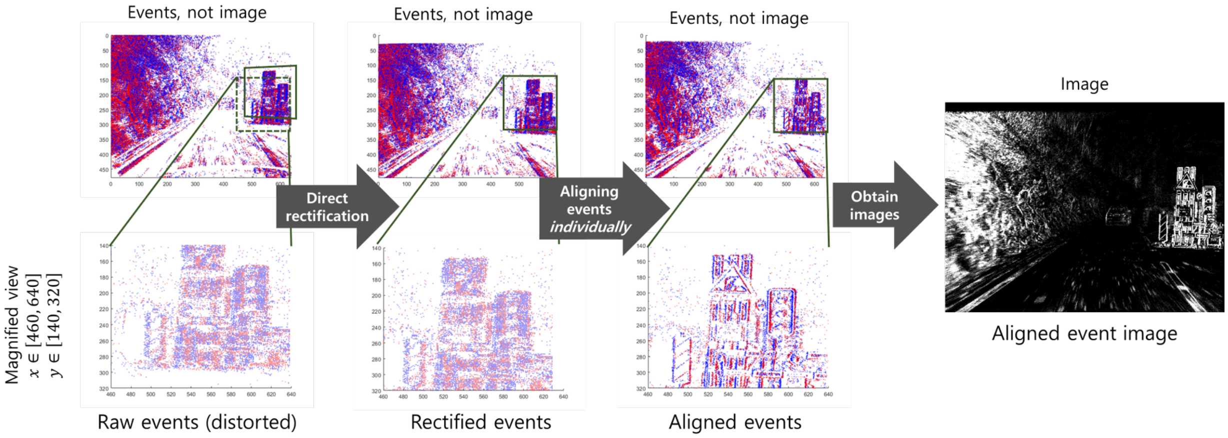
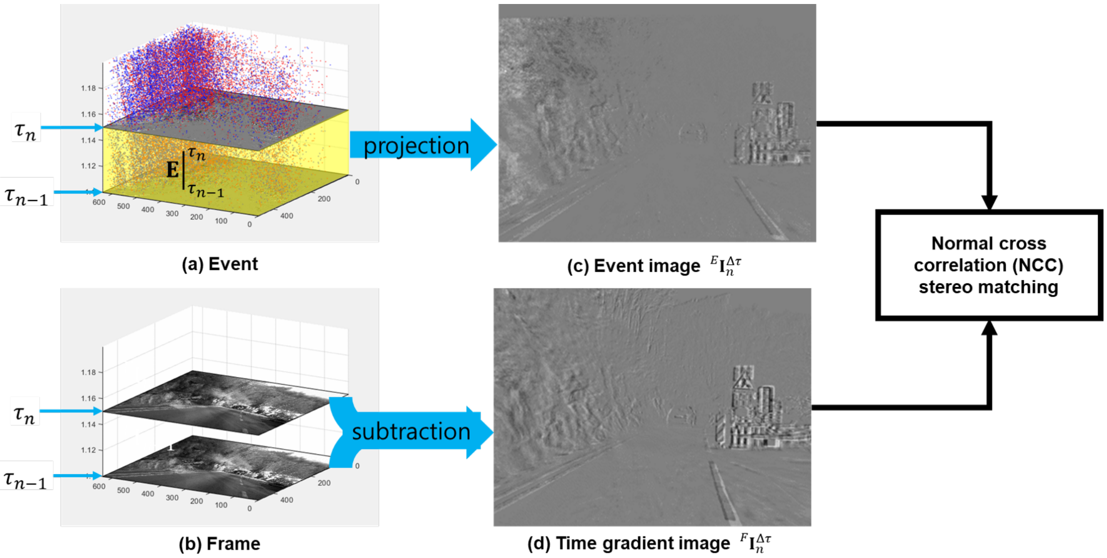


Figure 4.4: Rectification with direct method.

4.1.1 Initial Stereo Matching

Event cameras record polarities of change of log intensity and have a different dynamic range from frame cameras. For associating the frame image ${}^{F}\mathbf{I}_n$ with the event set $\mathbf{E}|_{\tau_{n-1}}^{\tau_n}$ in the initial phase, I use the temporal gradient image ${}^{F}\mathbf{I}_n^{\Delta\tau} = {}^{F}\mathbf{I}_n - {}^{F}\mathbf{I}_{n-1}$ and the event image ${}^{E}\mathbf{I}_n^{\Delta\tau}$ which is computed by accumulating the raw events of $\mathbf{E}|_{\tau_{n-1}}^{\tau_n}$ considering polarity. The frame image ${}^{F}\mathbf{I}_n^{\Delta\tau}$ and the event image ${}^{E}\mathbf{I}_n^{\Delta\tau}$ contain information about intensity changes. Because the above two images have the same tendency but different dynamic range, I utilize the normalized cross-correlation (NCC) cost that is robust to the difference in dynamic range, rather than applying residual cost such as the sum of absolute distance (SAD) and the sum of squared distance (SSD).

Then, the edge pixel $\mathbf{f} \in \mathbb{R}^{2\times 1}$ is extracted from the frame image with Sobel filter and conduct patch matching methods on the temporal gradient image and the event image. I shift the



(a) Event

(c) Event image ${}^{E}\mathbf{I}_n^{\Delta\tau}$

(b) Frame

(d) Time gradient image ${}^{F}\mathbf{I}_n^{\Delta\tau}$

Figure 4.5: Overview of initial stereo matching.

event patch $\mathbf{P}(\mathbf{f}, {}^{E}\mathbf{I}_n^{\Delta\tau})$ along the x coordinate (parallel to epipolar line) and compare $\mathbf{P}(\mathbf{f}, {}^{E}\mathbf{I}_n^{\Delta\tau})$ with the frame patch $\mathbf{P}(\mathbf{f}, {}^{F}\mathbf{I}_n^{\Delta\tau})$ using NCC cost $\mathrm{C}(\cdot,\cdot)$. Additionally, 2D Gaussian-smoothing is applied on NCC cost along the image coordinate to alleviate the noise effect. Then, I estimate the disparity $\hat{d}$ where the Gaussian-smoothed NCC $\mathrm{G}\left(\mathscr{E}_\tau(\mathbf{f},d)\right)$ is maximized as

$$\mathscr{E}_\tau(\mathbf{f},d) = \mathrm{C}\left(\mathbf{P}(\mathbf{f}, {}^{F}\mathbf{I}_n^{\Delta\tau}), \mathbf{P}(\mathbf{f} + [d,0]^\intercal, {}^{E}\mathbf{I}_n^{\Delta\tau})\right), \tag{4.1}$$

$$\hat{d} = \underset{d}{\operatorname{argmax}}\ \mathrm{G}\left(\mathscr{E}_\tau(\mathbf{f},d)\right). \tag{4.2}$$

For the sub-pixel accuracy, I interpolate the disparity with quadratic interpolation as follows:

$$d^* = \hat{d} + 0.5\frac{\mathscr{E}_\tau(\mathbf{f},\hat{d}-1) - \mathscr{E}_\tau(\mathbf{f},\hat{d}+1)}{2\mathscr{E}_\tau(\mathbf{f},\hat{d}) - \mathscr{E}_\tau(\mathbf{f},\hat{d}-1) - \mathscr{E}_\tau(\mathbf{f},\hat{d}+1)}. \tag{4.3}$$

The presented initialization of the system can be performed with two frames if sufficient events are spiked.

4.1.2 Event Alignment

I can expect more accurate matching by using edge images instead of temporal gradient images. In order to match the event data with edge images, I align events with the camera motion. The frame camera pose ${}^{F}\mathbf{T}$ is estimated by applying APnP [59] from the constructed 3D points with the initial disparity, and then compute the event camera pose ${}^{E}\mathbf{T}$ with extrinsic parameters.

When accurately warping event points, it is computationally expensive to consider the time and depth of each event. In order to reduce the computation load of warping function, I assume that the events in $\mathbf{E}|_{\tau_{n-1}}^{\tau_n}$ share the same depth at reference time τ_n, which means that the depth values of events are all the same after warping with the motion ${}^{E}\mathbf{T}_{n-1}^{n}$. Here, the depth at reference time is a given value and is covered in Section 4.1.3, which allows the event to have different depth at the event time t_k. This condition is a more relaxed than [27] which assumes that the depth of the event is the same regardless of the event time.

Even if the events have the same depth at reference time τ_n, an event e_k can have different depth $z_k(t_k)$ before warping. The warping function for the k-th event can be presented as follows.

$$\mathbf{X}_k(\tau_n) = \mathbf{R}_n(\delta t_k)\mathbf{X}_k(t_k) + \mathbf{t}_n(\delta t_k), \tag{4.4}$$

$$\mathbf{X}_k(\tau_n) = z_k(t_k)\mathbf{R}_n(\delta t_k)\bar{\mathbf{X}}_k(t_k) + \mathbf{t}_n(\delta t_k) \tag{4.5}$$

where $\mathbf{X}_k(t) = [x_k(t), y_k(t), z_k(t)]^\top$ is the inverse-projected 3D point of the event e_k in the camera coordinate at time t and $\bar{\mathbf{X}}_k(t) = [\bar{x}_k(t), \bar{y}_k(t), 1]^\top$ is the inverse-projected point from the event pixel $\mathbf{x}_k$, which satisfying $\mathbf{X}_k(t) = z_k(t)\bar{\mathbf{X}}_k(t)$. $z_k(t_k)$ is the exact depth of the event and $z_k(\tau_n)$ is the depth at the reference time. The reference depths $z_k(\tau_n)$ of events in $\mathbf{E}|_{\tau_{n-1}}^{\tau_n}$ are given and have all the same value, as I assumed, while $z_k(t_k)$ differs. δt_k is the time difference between the time of an event t_k and the reference time τ_n i.e. $\delta t_k = \tau_n - t_k$, $\mathbf{R}_n(\delta t_k)$ and $\mathbf{t}_n(\delta t_k)$ is the rotation and the translation matrix of camera motion ${}^{E}\mathbf{T}_{n-1}^{n}$ considering the time difference δt_k.

Then, the depth of the event $z_k(t_k)$ can be inversely computed with given $z_k(\tau_n)$ as follows.

$$z_k(\tau_n) = \begin{bmatrix}0 & 0 & 1\end{bmatrix} \left(z_k(t_k)\mathbf{R}_n(\delta t_k)\bar{\mathbf{X}}_k(t_k) + \mathbf{t}_n(\delta t_k)\right), \tag{4.6}$$

$$z_k(t_k) = \frac{z_k(\tau_n) - \begin{bmatrix}0 & 0 & 1\end{bmatrix}\mathbf{t}_n(\delta t_k)}{\begin{bmatrix}0 & 0 & 1\end{bmatrix}\mathbf{R}_n(\delta t_k)\bar{\mathbf{X}}_k(t_k)}, \tag{4.7}$$

I can get aligned event points with the same depth at reference time by putting $z_k(t_k)$ back into Eq. (4.5).

Since each event has different δt_k, I need to lighten the computational load when computing $\mathbf{R}_n(\delta t_k)$ and $\mathbf{t}_n(\delta t_k)$ for each event. I convert the existing warping function into a matrix operation using the second-order approximation as in [47]. I employ the twist coordinate representation $[v;\omega] \in \mathbb{R}^6$ provided by the Lie algebra $\mathfrak{se}(3)$ associated with the group SE(3), where v is the linear velocity and ω is the angular velocity. The exact warping can be achieved with Rodrigues' formula as follows.

$$\mathbf{R}_n(\delta t_k)\bar{\mathbf{X}}_k(t_k) = \bar{\mathbf{X}}_k(t_k) + \frac{\hat{\omega}}{|\omega|}\bar{\mathbf{X}}_k(t_k)\sin(|\omega|\delta t_k) + \frac{\hat{\omega}^2}{|\omega|^2}\bar{\mathbf{X}}_k(t_k)\left(1-\cos(|\omega|\delta t_k)\right), \tag{4.8}$$

$$\mathbf{t}_n(\delta t_k) = v\delta t_k + \frac{\hat{\omega}v}{|\omega|^2}\left(1-\cos(|\omega|\delta t_k)\right) + \frac{\hat{\omega}^2 v}{|\omega|^3}\left(|\omega|\delta t_k - \sin(|\omega|\delta t_k)\right), \tag{4.9}$$

where $\hat{\omega}$ is the cross-product matrix of ω. By substituting $\cos(|\omega|\delta t_k) \approx 1 - |\omega|^2\delta t_k^2/2$ and $\sin(|\omega|\delta t_k) \approx |\omega|\delta t_k$ in Eq. (4.8) and Eq. (4.9), the rotation and translation can be simplified as follows.

$$\mathbf{R}_n(\delta t_k)\bar{\mathbf{X}}_k(t_k) \approx \bar{\mathbf{X}}_k(t_k) + \hat{\omega}\bar{\mathbf{X}}_k(t_k)\delta t_k + \frac{1}{2}\hat{\omega}^2\bar{\mathbf{X}}_k(t_k)\delta t_k^2, \tag{4.10}$$

$$\mathbf{t}_n(\delta t_k) \approx v\delta t_k + \frac{1}{2}\hat{\omega}v\delta t_k^2. \tag{4.11}$$

Then, the aligned event point $\mathbf{X}_k(\tau_n)$ which has the same depth at reference time can be obtained by substituting Eq. (4.10) and Eq. (4.11) into Eq. (4.7) and Eq. (4.5) as follows.

$$\mathbf{X}_k(\tau_n) = \frac{z_k(\tau_n) - \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{t}_n(\delta t_k)}{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{R}_n(\delta t_k)\bar{\mathbf{X}}_k(t_k)} \cdot \mathbf{R}_n(\delta t_k)\bar{\mathbf{X}}_k(t_k) + \mathbf{t}_n(\delta t_k) \tag{4.12}$$

4.1.3 Stereo Matching with Aligned Events

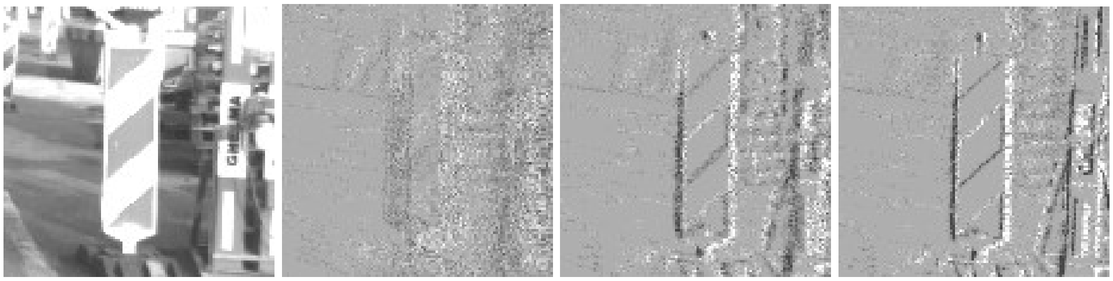
I perform stereo matching using the aligned events to increase the number of inlier disparity and to estimate disparity more accurately than the initial phase. To align the event, the camera pose ${}^{E}\mathbf{T}^{n}_{n-1}$ and the reference depth $z_k(\tau_n)$ are required. Since I slide the patch images in stereo matching, there are several disparity and depth candidates. I set the reference depth candidates by converting the disparity range as $z = f \cdot b/d$ where f is the focal length and b is the length of the baseline, and compute the bunch of aligned events for each depth candidates by assuming that the entire events of $\mathbf{E}|^{\tau_n}_{\tau_{n-1}}$ have the same reference depth. By projecting the aligned events $\mathbf{X}(\tau_n)$ into the image plane, I obtain the aligned event image ${}^{E}\mathbf{I}^{\Delta\mathbf{x}}_{n}(d)$ with disparity d as a parameter as shown in Fig. 4.6. Then, I perform the stereo matching with the aligned event image ${}^{E}\mathbf{I}^{\Delta\mathbf{x}}_{n}(d)$ and the edge image ${}^{F}\mathbf{I}^{\Delta\mathbf{x}}_{n} = \left|\nabla {}^{F}\mathbf{I}_n\right|$ as follows.

$$\mathscr{E}_{\mathbf{x}}(\mathbf{f}, d) = \mathrm{C}\left(\mathbf{P}(\mathbf{f}, {}^{F}\mathbf{I}^{\Delta\mathbf{x}}_{n}), \mathbf{P}\left(\mathbf{f} + [d, 0]^{\intercal}, {}^{E}\mathbf{I}^{\Delta\mathbf{x}}_{n}(d)\right)\right), \tag{4.13}$$

$$\hat{d} = \underset{d}{\operatorname{argmax}}\, \mathrm{G}\left(\mathscr{E}_{\mathbf{x}}(\mathbf{f}, d) \cdot \mathscr{E}_{\tau}(\mathbf{f}, d)\right). \tag{4.14}$$

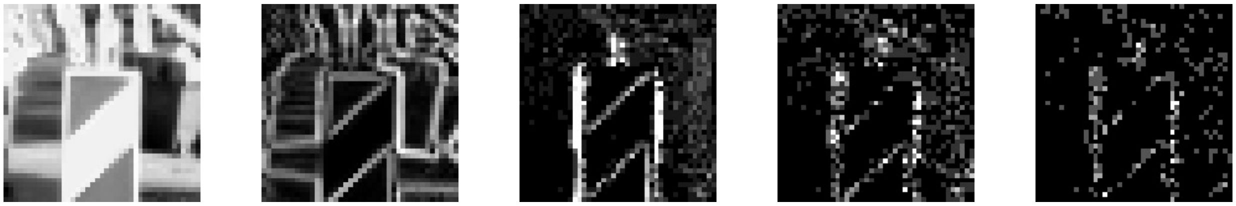
The aligned event images improve stereo matching accuracy by representing clear edges, but similar edges can be mismatched because the edges do not take into account polarity. I alleviate this ambiguity problem by multiplying the initial matching cost that considers polarity as in Eq. (4.14).

As in the initial phase, the interpolated disparity d^{*} is computed as in Eq. (4.3) for sub-pixel accuracy. Due to the aligning events using camera motion, the proposed method can reliably represent edge features even when I use a smaller number of events as in Fig. 4.7.



(a) Reference (b) $d = 0$ (c) $d = 15$ (d) $d = 30$

Figure 4.6: Illustrations of (a) frame image ${}^{F}\mathbf{I}_n$ and (b - d) aligned event images ${}^{E}\mathbf{I}_n^{\Delta\mathbf{x}}(d)$ with varying disparity d. For aligned event images, polarity is considered only in the figures for visibility.



(a) Frame (b) Edge (c) 100% (d) 50% (e) 10%

Figure 4.7: Illustrations of sampling effect on the aligned event image. (a) frame image, (b) edge of frame image, (c) aligned event image without sampling, (d) aligned event image with half of the events and (e) aligned event image with 10% events. Even with a small number of events, the proposed method can describe edges for stereo matching.

4.1.4 Maximum Shift Distance (MSD) by Translational Motion

If there is no translational motion ($\mathbf{t}_n = 0$), the aligned event images are identical, regardless of the varying disparity. In this case, it is inefficient to compute the aligned event image for each disparity, because only one aligned event image is needed. I mitigate this inefficiency by grouping disparities that produce similar aligned events. The disparities are grouped based on the maximum shift distance of events which is related to the magnitude of translation.

In this section, it is assumed that all points are already rotated in order to focus on the effect of translational motion on the maximum shift distance. When points are shifted by the translational motion, the most shifted point exists at the corner of the image due to the characteristics of the pinhole camera model. In the presence of the translational motion $\mathbf{t}$, the 3D point $\mathbf{X}$ which corresponds to the image corner will be warped to $\mathbf{X}'$ with depth z as depicted in Fig. 4.9. The

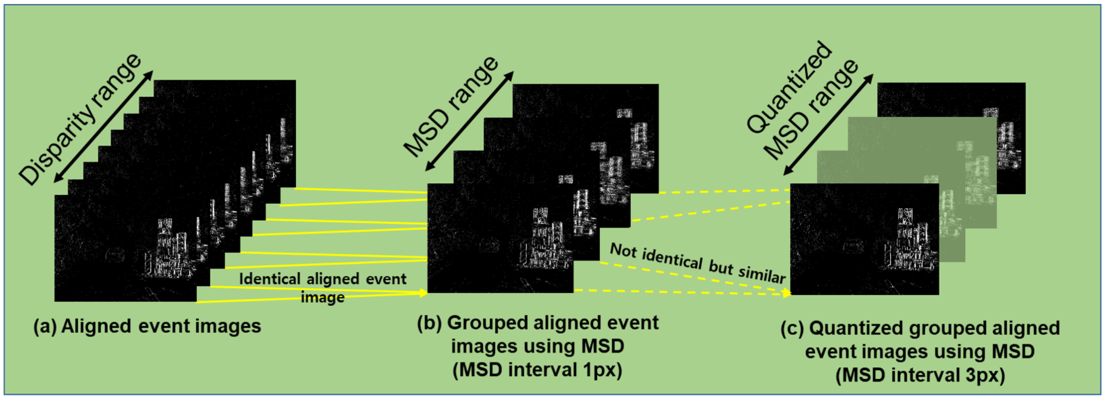

Disparity range
MSD range
Quantized MSD range
Identical aligned event image
Not identical but similar
(a) Aligned event images
(b) Grouped aligned event images using MSD (MSD interval 1px)
(c) Quantized grouped aligned event images using MSD (MSD interval 3px)

Figure 4.8: Overview of maximum shift distance (MSD).

trajectory of the 3D point $\mathbf{X}$ is projected onto the image plane is called the event shift trajectory, denoted as $\mathbf{s}$. It is the orange colored vector in Fig. 4.9.(a).

$$\bar{\mathbf{t}} = \frac{f}{z - |\mathbf{t}_z|}\mathbf{t}, \tag{4.15}$$

where $\bar{\mathbf{t}}$ is the scaling transformation of translation $\mathbf{t}$ for the image plane. $\mathbf{t}_{xy}, \mathbf{t}_z$ are the decomposed vectors of $\mathbf{t}$, perpendicular to the image plane and parallel to the principal axis, respectively.

As shown in Fig. 4.9.(b), the shift $\mathbf{s}$ is computed as the sum of $\mathbf{a}$ and $\mathbf{b}$, where $\mathbf{a}$ is the vector from the corner to the principal point. $\mathbf{b}$ is parallel to $\bar{\mathbf{t}}_{xy} - \mathbf{s}$, and has a similarity ratio of f and $|\bar{\mathbf{t}}_z|$ in the plane perpendicular to $\mathbf{t}_{xy}$. Then, $\mathbf{b}$ is composed of the two vectors $\mathbf{t}_{xy}$ and $\mathbf{s}$ by substituting $\bar{\mathbf{t}}$ with $\mathbf{t}$ as in the second part of Eq. (4.16).

$$\mathbf{b} = \frac{f}{|\bar{\mathbf{t}}_z|}(\bar{\mathbf{t}}_{xy} - \mathbf{s}) = \frac{f}{|\mathbf{t}_z|}\mathbf{t}_{xy} - \left(\frac{z}{|\mathbf{t}_z|} - 1\right)\mathbf{s} \tag{4.16}$$

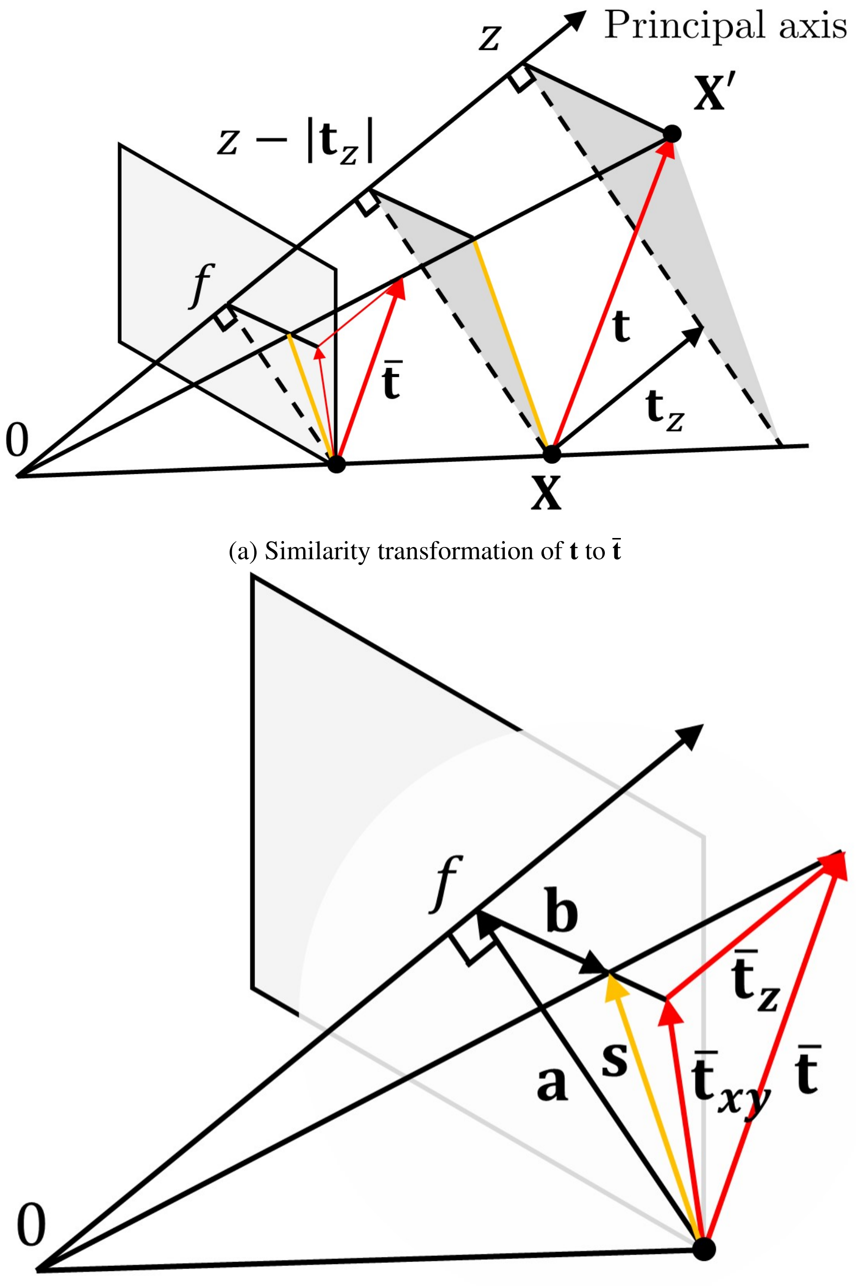
$$\mathbf{s} = \mathbf{a} + \mathbf{b} = \mathbf{a} + \frac{f}{|\mathbf{t}_z|}\mathbf{t}_{xy} + \left(1 - \frac{z}{|\mathbf{t}_z|}\right)\mathbf{s}, \tag{4.17}$$

By substituting $\mathbf{b}$ in Eq. (4.17) with the second part of Eq. (4.16) and rearranging Eq. (4.17) for $\mathbf{s}$, the shift $\mathbf{s}$ can be obtained as Eq. (4.18).

$$\mathbf{s} = \frac{|\mathbf{t}_z|\mathbf{a} + f\mathbf{t}_{xy}}{z} \tag{4.18}$$

By the trigonometric inequality, $|\mathbf{s}| \leq (|\mathbf{a}||\mathbf{t}_z| + f|\mathbf{t}_{xy}|)/z$ is satisfied, where $|\mathbf{a}|$ is the half of the diagonal image size. I set the maximum shift distance as $(|\mathbf{a}||\mathbf{t}_z| + f|\mathbf{t}_{xy}|)/z$. If the depth z is expressed again as disparity d, the maximum shift distance satisfies $s_{\max}(d, \mathbf{t}) := (|\mathbf{a}||\mathbf{t}_z| + f|\mathbf{t}_{xy}|)/(f \cdot b) \cdot d$ which indicates that it is proportional to the disparity d.

I can quantize the aligned event image without loss by grouping disparities whose s_{max} values do not differ by more than 1 px. In addition, I can significantly reduce the number of aligned event bunches by adjusting the interval of pixels where the s_{max} value differs (MSD interval).



(a) Similarity transformation of $\mathbf{t}$ to $\bar{\mathbf{t}}$

(b) Computing maximum shift distance $\mathbf{s}$

Figure 4.9: Illustrations of maximum shift distance. The corner point $\mathbf{X}$ is moved to $\mathbf{X}'$ by translation. The maximum shift distance is the orange colored arrow s, and translation is the red colored arrow $\mathbf{t}$.

4.2 Experimental Results

I evaluate the proposed method on the DSEC dataset [51]. DSEC is the disparity and optical-flow evaluation dataset that records the city driving scenario. It employs high resolution stereo event (640×480) and frame (1440×1080) cameras (4 cameras in total), and provides the ground truth disparity converted from LIDAR.

I compare the proposed method with the initial stereo matching method described in Section 4.1.1, E2VID [10] based method and the implemented version of SHEF [44]. The reconstruction performance of [10] affected by the number of events. I evaluate [10] into the two event grouping manners. E2VID-τ reconstructs frame images from the events in a fixed temporal window ($\tau_n - \tau_{n-1} \approx 50$ ms), $\mathbf{E}|_{\tau_{n-1}}^{\tau_n}$, which I used. E2VID-$N$ uses the fixed number of events ($N = 10^5$). Then, I perform stereo matching on the reconstructed frame with NCC cost as in Eq. (4.2) (E2VID-N/τ). Also, the standard semi-global matching method (SGM) is applied which utilizes residual costs (E2VID-SGM-N/τ).

I set disparity range to 100 px, MSD interval to 10 px, std of Gaussian filter $\mathrm{G}(\cdot)$ to 2 px and the kernel radius of all methods to 12 px.

4.2.1 Performance of Stereo Matching

I verified the matching accuracy with the disparity error in Table 4.1. I used the following metrics:

- root mean squared error within 3 px disparity error (RMSE): $\sqrt{\frac{1}{T}\sum_{\mathrm{p}}(d_{\mathrm{gt}} - d_{\mathrm{p}})^2}$
- mean absolute error within 3 px disparity error(MAE): $\frac{1}{T}\sum_{\mathrm{p}}|d_{\mathrm{gt}} - d_{\mathrm{p}}|$
- percentage of absolute error for the all edge pixels (recall) with threshold δ^*: percentage of d_{p} s.t. $|d_{\mathrm{gt}} - d_{\mathrm{p}}| = \delta < \delta^*$

For depth evaluation, I used RMSE and the following metrics:

- absolute relative distance (ARD): $\frac{1}{T}\sum_{\mathrm{p}}\frac{|z_{\mathrm{gt}} - z_{\mathrm{p}}|}{z_{\mathrm{gt}}}$

Disparity		RMSE	MAE	percentage of absolute error (δ^*)		
				1 px	2 px	3 px
HSM	Prop.	**1.036**	**0.796**	**0.560**	**0.743**	**0.800**
	Init.	1.131	0.895	0.501	0.723	0.791
E2VID-N		1.048	0.807	0.451	0.595	0.644
E2VID-τ		1.768	1.558	0.079	0.175	0.268
E2VID-SGM-N		-	-	0.009	0.116	0.120
E2VID-SGM-τ		-	-	0.002	0.005	0.009
SHEF		1.630	1.386	0.088	0.163	0.227

(a) Disparity

Depth		RMSE	ARD	percentage of relative error (δ^*)		
				1.05	1.05^2	1.05^3
HSM	Prop.	3.092	0.060	**0.444**	**0.664**	**0.743**
	Init.	3.118	0.066	0.350	0.634	0.739
E2VID-N		**2.950**	**0.058**	0.363	0.534	0.599
E2VID-τ		7.490	0.148	0.048	0.100	0.154
E2VID-SGM-N		-	-	0.079	0.108	0.116
E2VID-SGM-τ		-	-	0.001	0.003	0.006
SHEF		6.156	0.112	0.063	0.127	0.170

(b) Depth

Table 4.1: Disparity and depth estimation results for *interlaken_c*. I evaluate the disparity estimation results on edges with root mean squared error, mean absolute error and percentage of absolute error (recall) and evaluate the depth with root mean squared error, absolute relative distance and percentage of relative error (recall). '-' indicates that the algorithm failed with less than 15 % inliers. E2VID-N/τ apply the NCC cost on stereo matching, and E2VID-SGM-N/τ utilize the standard semi-global matching method.

- percentage of relative error (recall) with threshold δ^*: percentage of z_p s.t. $\max(\frac{z_{gt}}{z_p}, \frac{z_p}{z_{gt}}) = \delta < \delta^*$

All error metrics except percentages (disparity RMSE, MAE, depth RMSE and ARD) are only evaluated for pixels with a disparity error within 3 px. This is to prevent large errors from affecting the metric.

The proposed method outperformed other stereo event frame methods (HSM-Init, E2VID-τ [10] and SHEF [44]) for inliers and disparity RMSE and MAE. The standard stereo matching method (E2VID-SGM-N/τ) failed due to the different dynamic range of the reconstructed frame image. NCC cost based E2VID-N [10] methods showed comparable disparity RMSE to the proposed method. The reconstructed image from E2VID methods describes detailed features.

Dataset		*interlaken_c*		*interlaken_d*		*interlaken_e*		*interlaken_f*		*interlaken_g*	
Disparity		RMSE	recall/prec.	RMSE	recall/prec.	RMSE	recall/prec.	RMSE	recall/prec.	RMSE	recall/prec.
HSM	Prop	1.036	0.800/0.806	1.106	0.834/0.838	1.179	0.737/0.742	1.152	0.783/0.788	1.057	0.846/0.850
	Init	1.131	0.791/0.797	1.232	0.838/0.842	1.205	0.748/0.753	1.173	0.791/0.795	1.112	0.837/0.839
E2VID-N		1.048	0.644/0.655	1.093	0.546/0.561	1.348	0.569/0.582	1.384	0.629/0.637	1.267	0.738/0.744
E2VID-τ		1.768	0.268/0.279	1.720	0.167/0.183	1.765	0.271/0.286	1.770	0.372/0.387	1.720	0.371/0.378
SHEF		1.630	0.227/0.230	1.658	0.190/0.196	1.564	0.255/0.278	1.529	0.244/0.282	1.624	0.276/0.299

Table 4.2: Disparity estimation results for *interlaken* sequences. I evaluate the disparity with root mean squared error (px) and the percentage of absolute error within 3 px for all edges (recall manner) and disparity estimated edges (precision manner).

Thus, RMSE error of E2VID methods is small for the inlier disparity matches. However, the percentage of absolute error of E2VID-*N* is 10% to 15% less than the proposed method.

For other data sequences, I evaluate the disparity RMSE and percentage of absolute error within 3 px as in Table 4.2. In E2VID methods, especially E2VID-τ, there exist improperly reconstructed local regions for some data sequences, as already reported in [60]. These local region artifacts appear as black region or squiggle pattern when the events are detected either too little over a long period of time or too many over a short period of time. For this reason, in Fig. 4.10, E2VID-*N* method estimate the disparity of the foreground better than the background. E2VID-τ often failed on reconstruction, because there are too many events in a single input tensor to network. The performance of E2VID methods dropped in the other data sequences due to the reconstruction artifacts. Meanwhile, the proposed method is free from this artifact issue and always showed reliable disparity estimation.

4.2.2 Qualitative Evaluation of Stereo Matching

The results of semi-dense reconstruction is depicted in Fig. 4.11. The proposed method estimate the disparity on edges. For evaluation, I display the disparity which has less than 10 px error on edges. The proposed method can estimate disparity as densely as the ground truth and perform better than E2VID [10] and SHEF [44]. SHEF [44] cannot properly estimate disparity on unclear edge regions where it is difficult to generalize a high-pass filter to build binary edge maps.

I display the detailed patch matching results of the proposed method HSM-Prop and HSM-Init in Fig. 4.17. The last columns of Fig. 4.17.(a) and Fig. 4.17.(a) show examples where HSM-Init failed with the disparity error of more than 10 px.

4.2.3 Computation Time

I run the proposed method on NVIDIA GeForce RTX 3080 Ti GPU and Intel Core i9-12900KF @ 3.20GHz CPU. I compute disparity at 640 × 480 resolution and events are not scaled or sampled at all. The analysis at ‘*interlaken_c*’ is shown in Fig. 4.20 and Table 4.3. I estimate computation

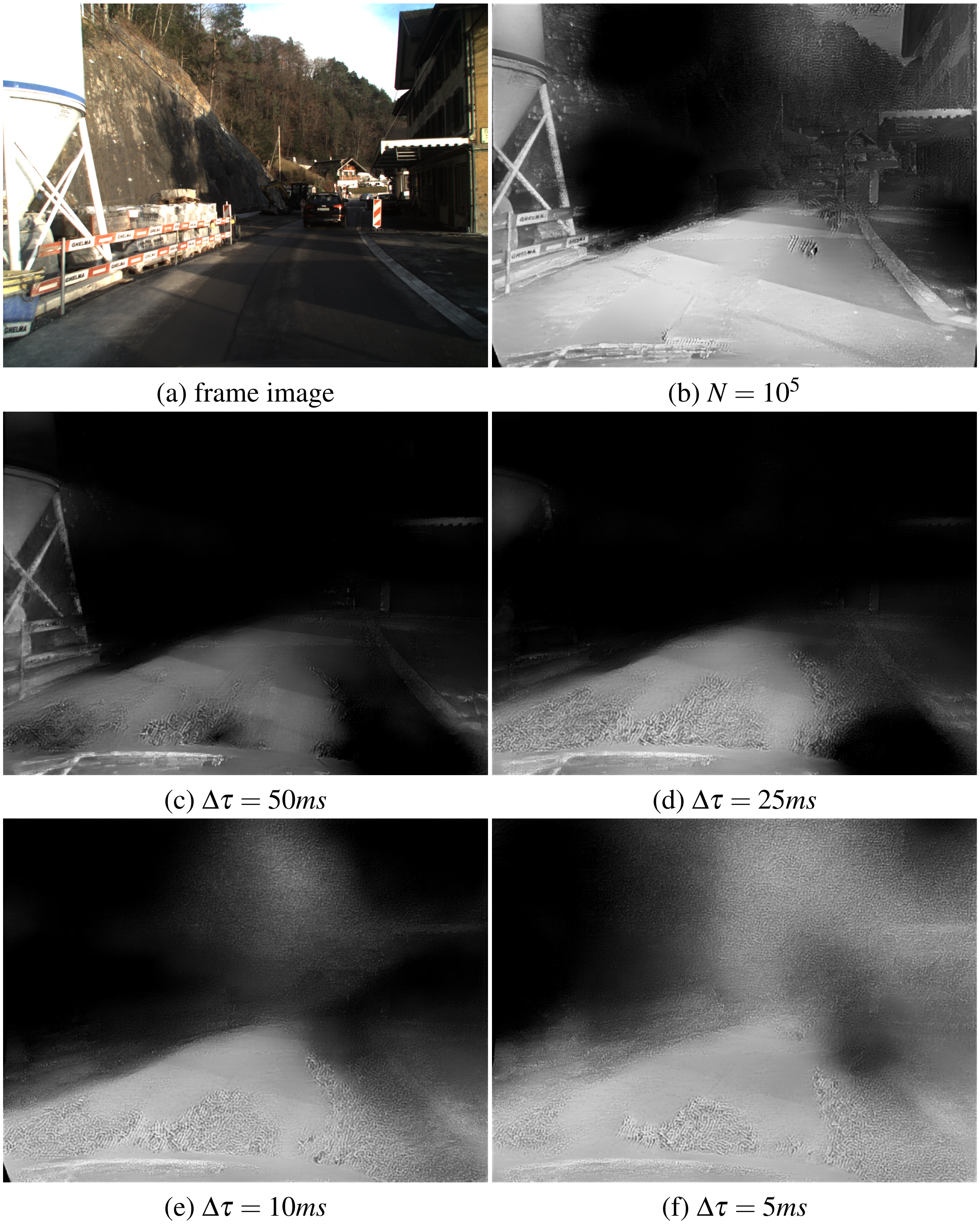
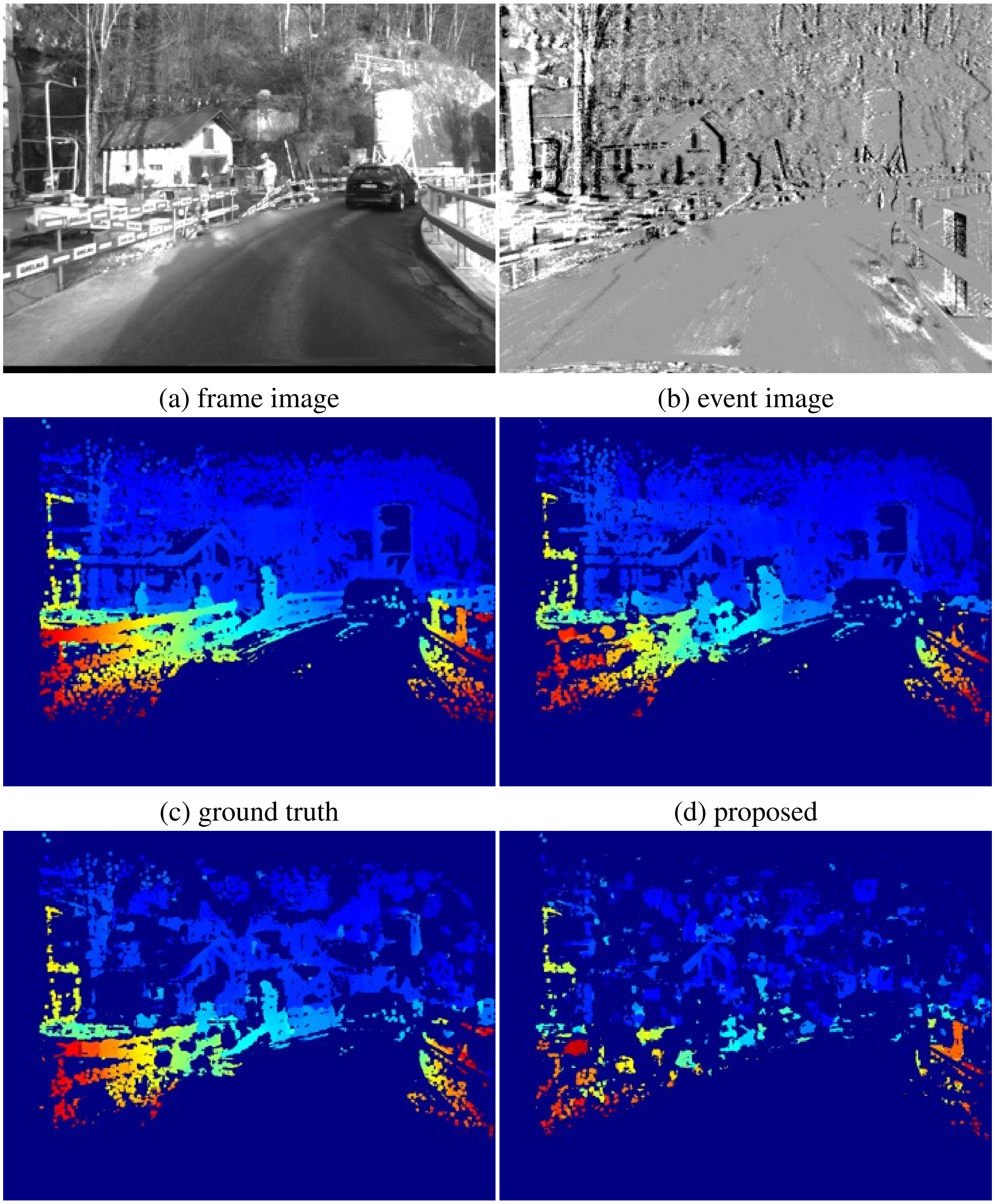


Figure 4.10: E2VID reconstruction results for *interlaken_c* data sequence. (a) is the frame image, (b) is the reconstructed image with fixed number of events, and the others are the reconstructed image with fixed temporal window. The reconstruction suffer from black region for dense event region, and squiggle pattern for sparse event region. The unwanted artifacts degrade the stereo matching performance.



(a) frame image

(b) event image

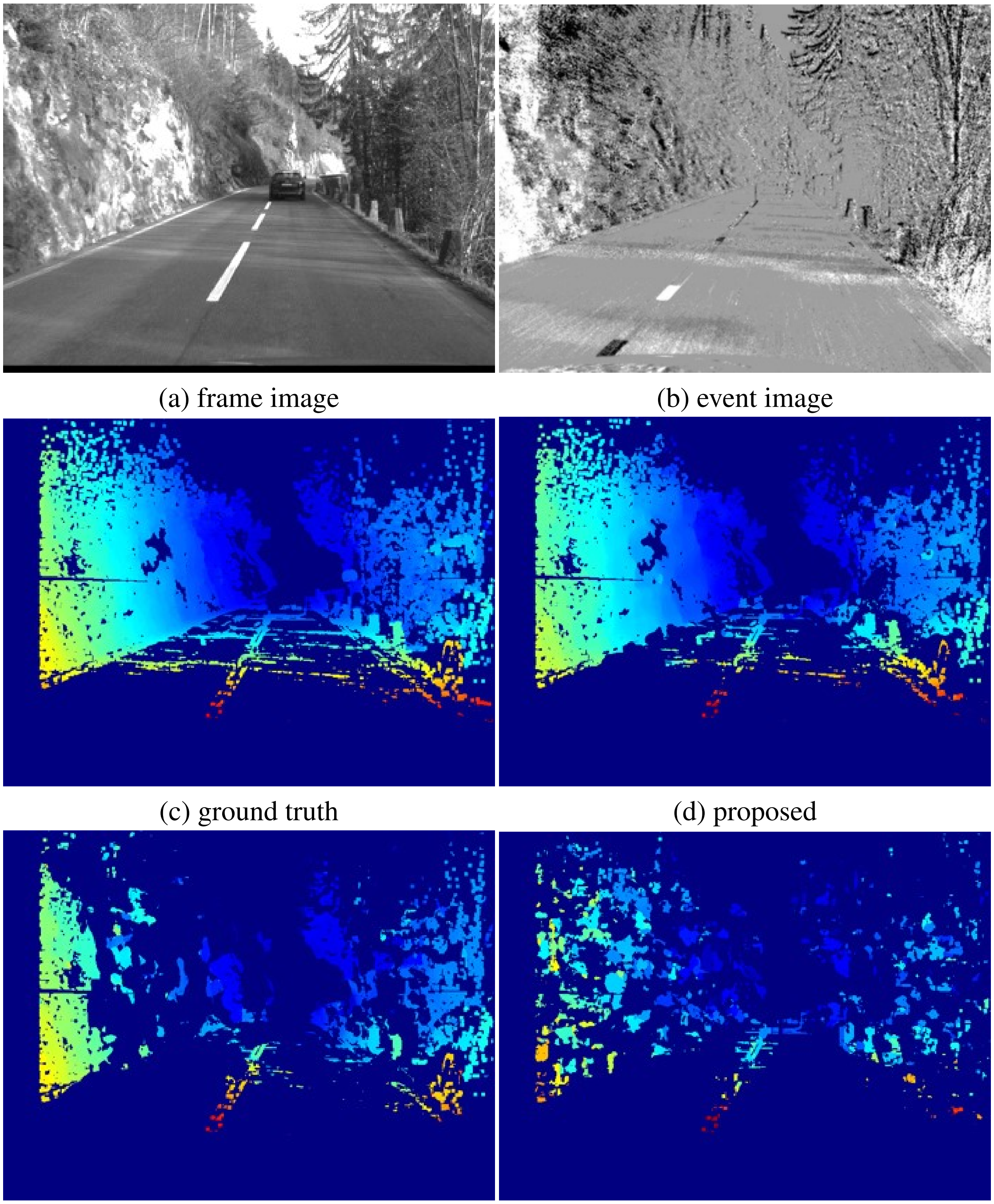
(c) ground truth

(d) proposed

(e) E2VID-*N*

(f) SHEF

Figure 4.11: Snapshots of the semi-dense disparity results for *interlaken_c* data sequence.



(a) frame image

(b) event image

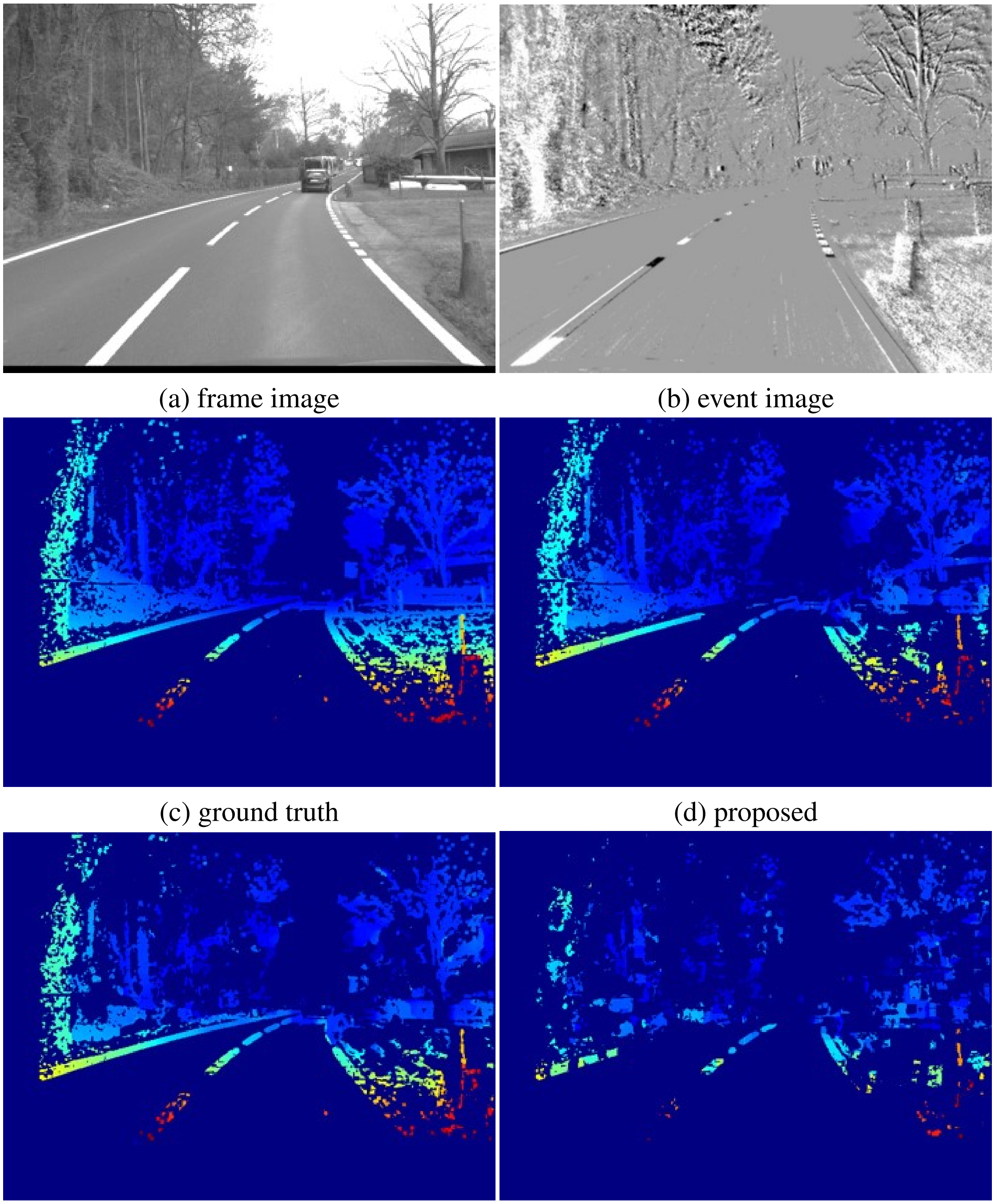
(c) ground truth

(d) proposed

(e) E2VID-*N*

(f) SHEF

Figure 4.12: Snapshots of the semi-dense disparity results for *interlaken_d* data sequence.



(a) frame image

(b) event image

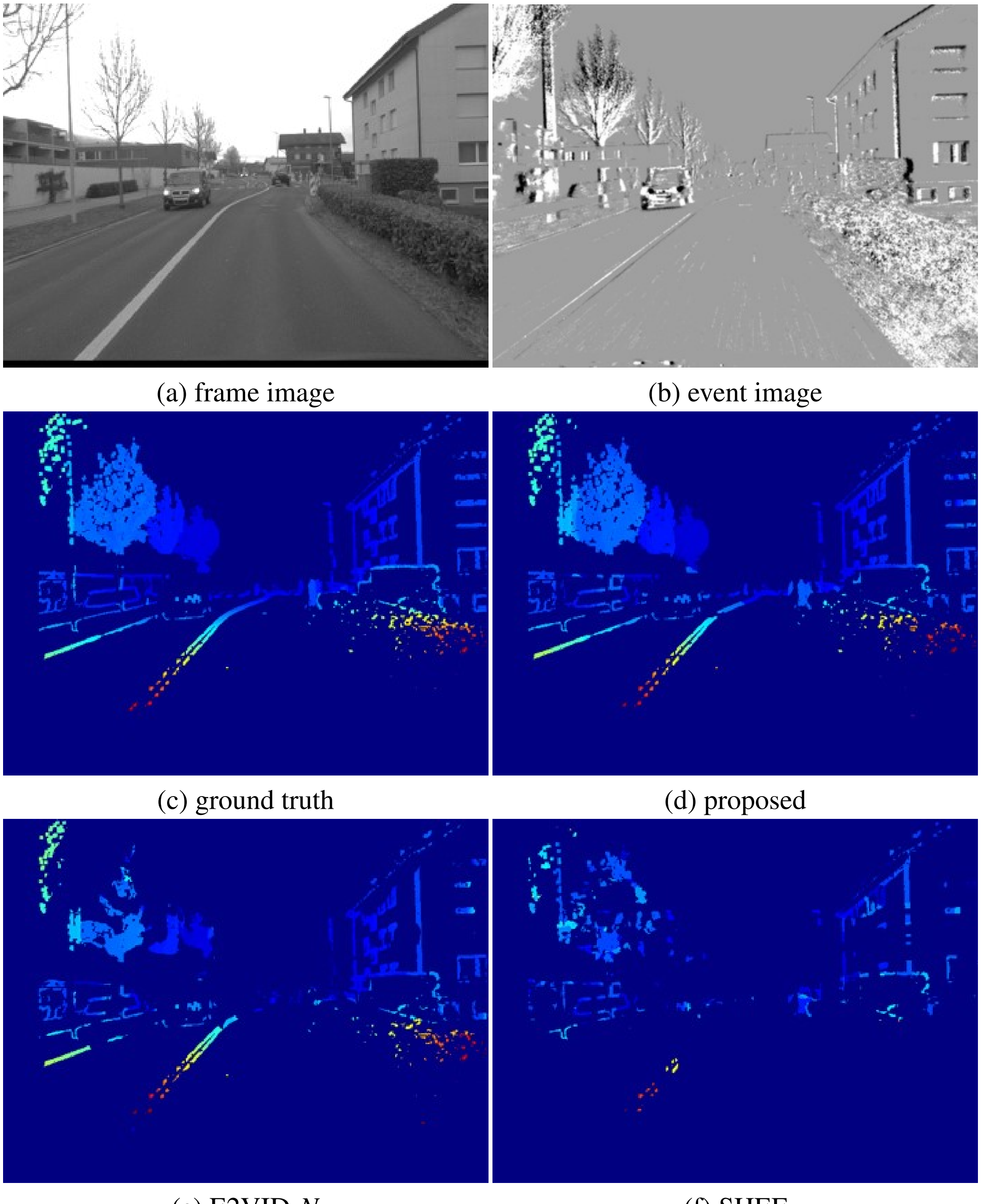
(c) ground truth

(d) proposed

(e) E2VID-*N*

(f) SHEF

Figure 4.13: Snapshots of the semi-dense disparity results for *interlaken_e* data sequence.


(a) frame image

(b) event image

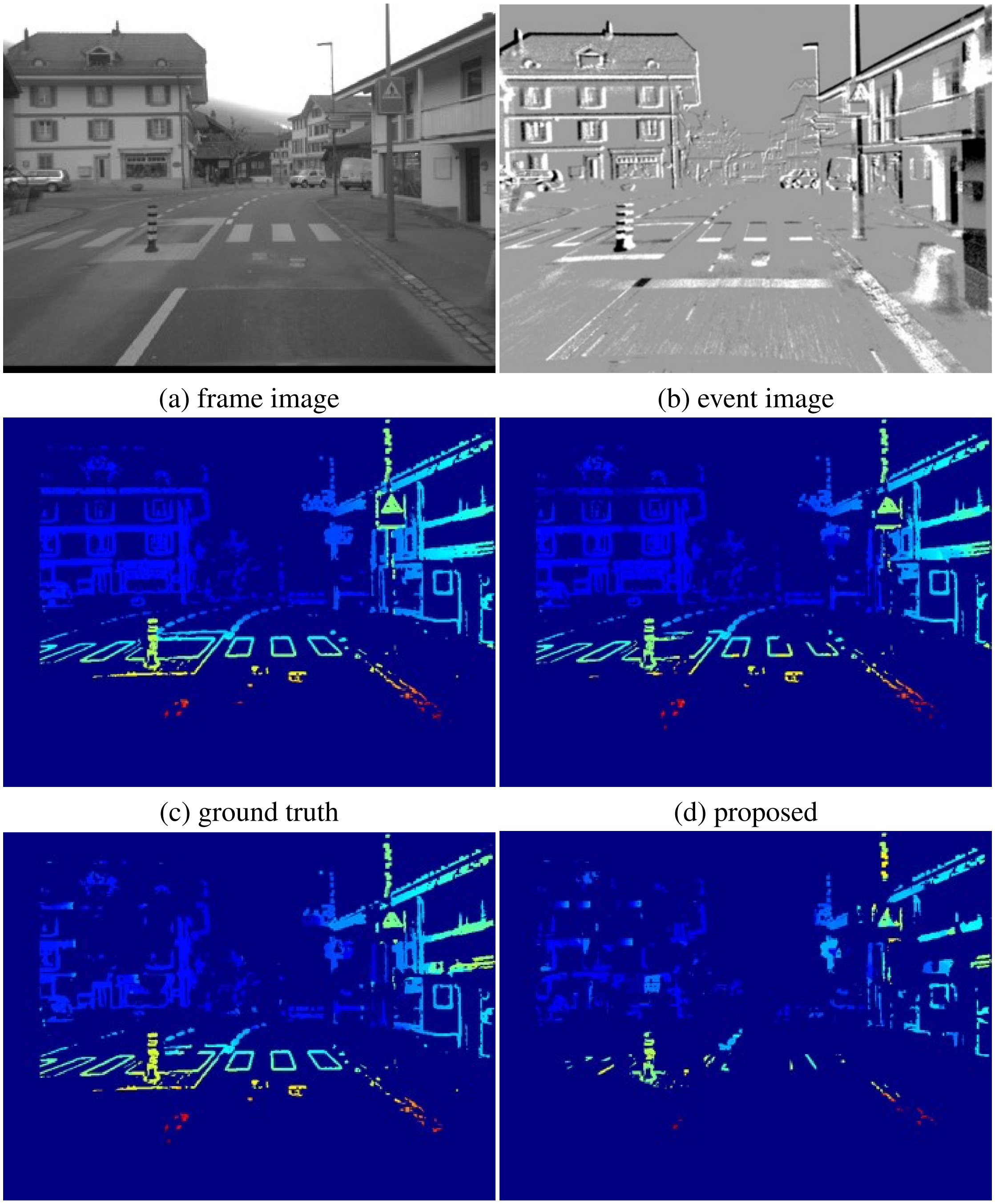
(c) ground truth

(d) proposed

(e) E2VID-*N*

(f) SHEF

Figure 4.14: Snapshots of the semi-dense disparity results for *interlaken_f* data sequence.



(a) frame image

(b) event image

(c) ground truth

(d) proposed

(e) E2VID-*N*

(f) SHEF

Figure 4.15: Snapshots of the semi-dense disparity results for *interlaken_g* data sequence.

HSM-Prop		E2VID-N/τ	
Compute MSD	2.66	Reconstruction	93.26 / 16.36
Compute AEI	3.13		
Stereo NCC AEI	5.00		
Stereo NCC Init	4.79	Stereo NCC	6.04
Stereo postproc	3.12	Stereo postproc	2.42
Etc.	2.41	Etc.	2.08
Total	21.13	Total	103.82 / 26.90

Table 4.3: Computation time per frame (ms). MSD is maximum shift distance, AEI is aligned event images. 'Stereo NCC AEI, Init' correspond to construct cost volume $\mathscr{E}_{\mathbf{x}}(\mathbf{f},d), \mathscr{E}_{\tau}(\mathbf{f},d)$, respectively, and 'Stereo postproc' is the computation process of Eq. (4.14). Except computing MSD, all the processes are computed on GPU. E2VID-N takes 93.26 ms for reconstruction, while E2VID-τ takes 16.36 ms.

time by averaging the computation time of the modules processing 10 times. The proposed stereo matching method achieves real-time performance by taking 21.13 ms per frame, which is less than 50 ms per frame. Such performance has become possible by utilizing the concept of the maximum shift distance (MSD) which lessen the computational load. The maximum shift distance depends on the size of the translation and the pixel interval. I can compute fewer aligned event images by adjusting the MSD interval. Since most pixels are not located on the corner of the image, most events are shifted much less than $s_{\max}(d,\mathbf{t})$. Thus, even if I aligned the events with more sparse disparity values, it does not significantly affect the stereo matching performance as in Table 4.4. Rather, the performance becomes better than default, since MSD supports Gaussian smoothing to work better. When a camera moves fast, $s_{\max}(d,\mathbf{t})$ can be greater than the maximum disparity. In this case, it is necessary to compute the aligned event image for all disparity.

I implemented SHEF [44] without computation time optimization. SHEF reconstructs edge images using the high-pass filter of [46] and non-maximal suppression, which is a similar concept to the Canny edge detection with the time complexity $\mathscr{O}(n\log n)$ where n is the number of image pixels. SHEF requires relatively light computation on reconstruction than the E2VID methods and HSM-Prop. Thus, SHEF can sufficiently operate in real-time.

MSD interval	1 px	2 px	3 px	5 px	10 px	Avg. events per frame
Compute AEI	15.96	10.41	7.06	5.24	3.13	0.8 million
RMSE	1.08	1.07	1.07	1.06	1.04	

Table 4.4: Computation time (ms) to construct aligned event images (AEI) and disparity RMSE (px) with varying maximum shift distance interval.

4.3 Summary

I perform hetero-stereo matching using frame cameras and event cameras, which have different characteristics. I present a method using a temporal gradient image to perform initialization within a few frames to estimate the camera pose, and proposed an accurate, efficient and intuitive method for aligning events utilizing camera motion. I propose the warping method considering the different depth of asynchronous events, and the maximum shift distance method to use fewer aligned event images for real-time performance. The proposed method describes edges using a much smaller number of events through the aligning events with camera motion. I verify the method with several experiments, which confirm that the proposed method outperforms than the other method for the inlier percentage and matching accuracy. I expect that the proposed approach will improve the capability of using frame and event camera and the provided code will contribute to the event camera community.

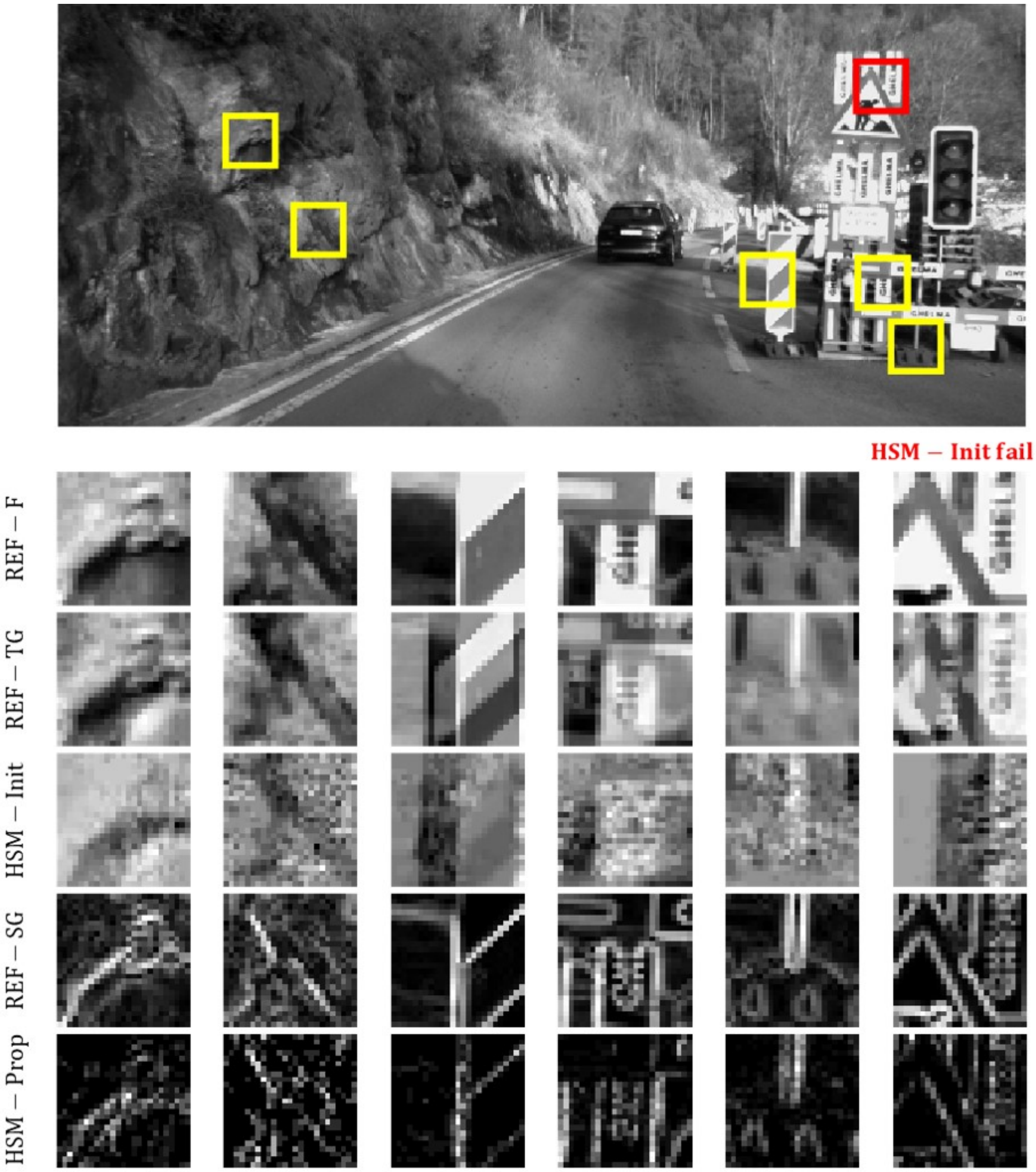


Figure 4.16

Figure 4.17: Snapshots of the hetero-stereo matching results. The reference frame images in the first row (REF-F) are magnified images of the yellow squared patch, and the red squared patches indicate the mismatched patches in HSM-Init method (the last column). The second row (REF-TG) indicates the corresponding reference temporal gradient of the frame images ${}^{F}\mathbf{I}_n^{\Delta\tau}$, and the third row represents the matched event images ${}^{E}\mathbf{I}_n^{\Delta\tau}$. The fourth row (REF-SG) and the last row are showing the edge images ${}^{F}\mathbf{I}_n^{\Delta\mathbf{x}}$ (the norm of spatial gradient image) obtained from the frame images and the aligned event images ${}^{E}\mathbf{I}_n^{\Delta\mathbf{x}}$ of the proposed method, respectively.

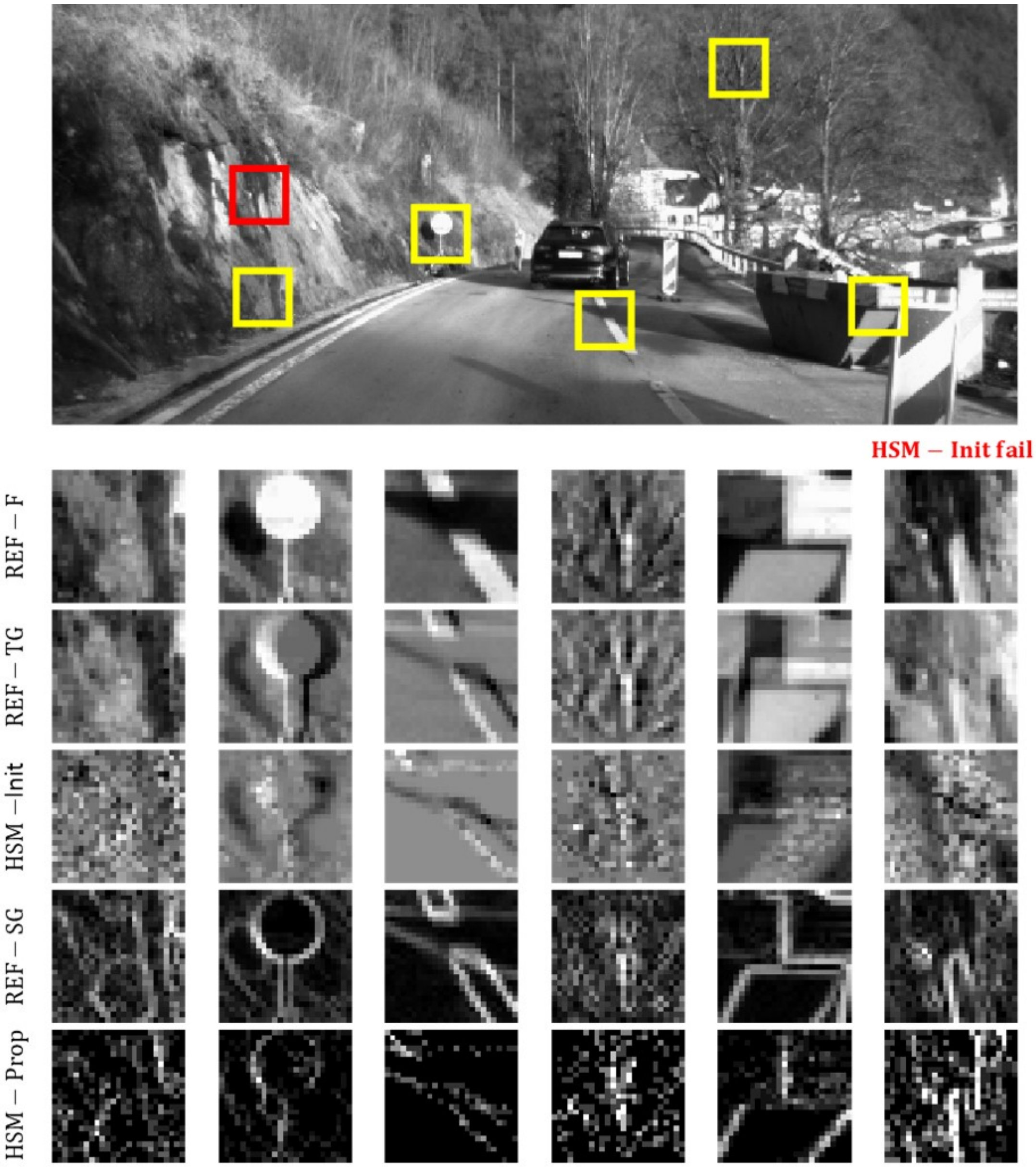


Figure 4.18

Figure 4.19: Snapshots of the hetero-stereo matching results. The reference frame images in the first row (REF-F) are magnified images of the yellow squared patch, and the red squared patches indicate the mismatched patches in HSM-Init method (the last column). The second row (REF-TG) indicates the corresponding reference temporal gradient of the frame images ${}^{F}\mathbf{I}_n^{\Delta\tau}$, and the third row represents the matched event images ${}^{E}\mathbf{I}_n^{\Delta\tau}$. The fourth row (REF-SG) and the last row are showing the edge images ${}^{F}\mathbf{I}_n^{\Delta\mathbf{x}}$ (the norm of spatial gradient image) obtained from the frame images and the aligned event images ${}^{E}\mathbf{I}_n^{\Delta\mathbf{x}}$ of the proposed method, respectively.

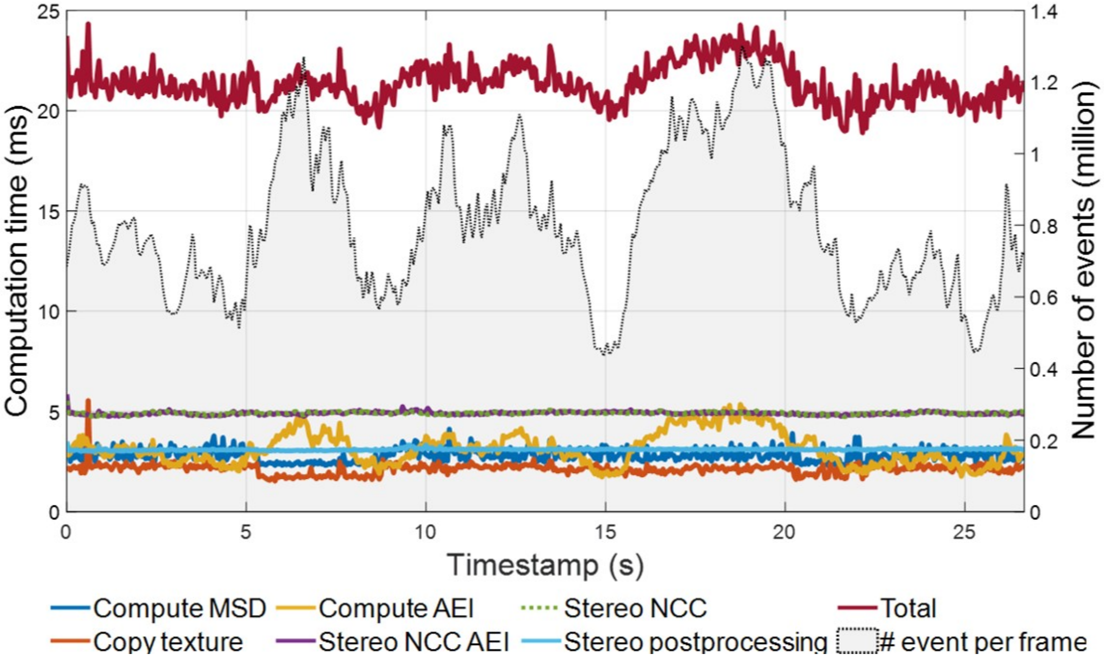


Figure 4.20: Graph of computation time per frame (ms). Computation time to construct aligned event images is proportional to the number of events.

5

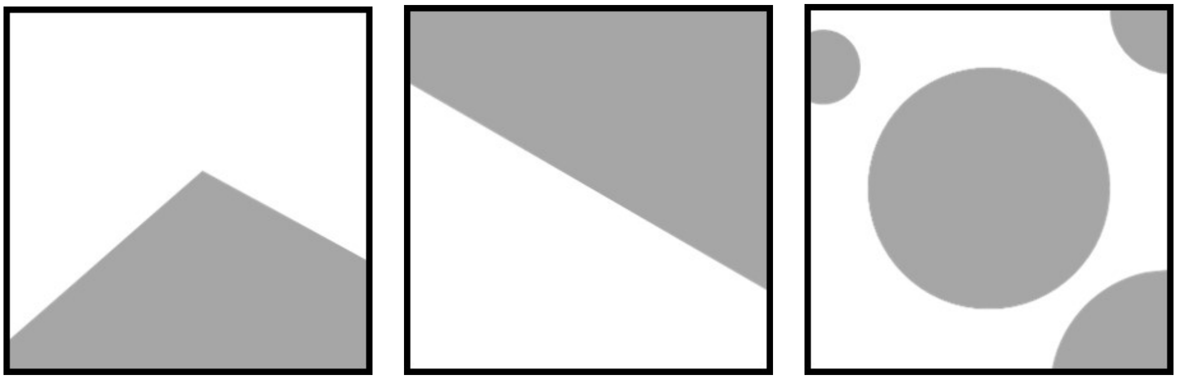
Feature Tracking and Pose Estimation for Hetero-Stereo Camera

The proposed method can estimate depth and can estimate camera pose using ORB-SLAM. Since I provide stereo depth image to the ORB-SLAM, camera pose is estimated up to real-world scale. For challenging environments such as blurred image or bad illumination condition, ORB-SLAM poses are incorrect. In the proposed framework workflow, event camera can assist feature tracking and pose estimation when frame camera fails to track under fast camera motion and high-dynamic-range scene. Here, I propose robust heterogeneous stereo camera system with feature tracking via contrast maximization.

5.1 Feature Tracking

5.1.1 Motion Model

In conventional frame images, feature is a piece of information about the content of an image. The types of feature are edges, corners, blobs and etc. Most of studies utilize the corner and edge



(a) Corner (b) Edge (c) Blob

Figure 5.1: Types of features for frame image data.

features Fig. 5.1.

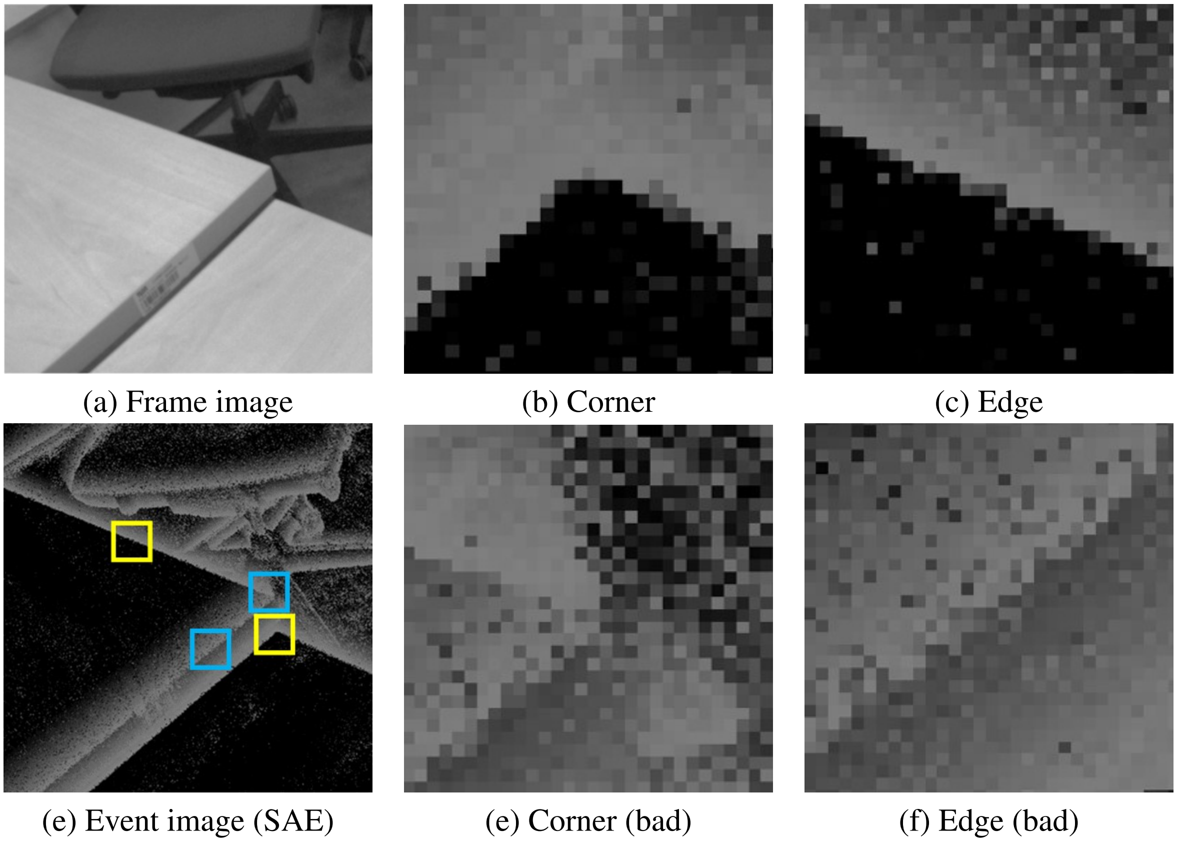
In contrast to the frame images, event data gives high-temporal resolution trajectory of the corners and edges. Thanks to the high-temporal resolution property of event data, it is possible to track features that have moved a lot. Meanwhile, the defining descriptor and matching feature task is difficult for event data. Also it cannot take advantage of the high-temporal resolution property of event data. Thus, feature tracking is more suitable than feature matching on event data.

Since events are consisted of temporal information, event images can show different aspects even for the same corner and edge as in Fig. 5.2. Therefore, I track features by aligning events considering event timestamp. I applied contrast maximization approach as in Chapter 3. Here, corner features are mainly tracked due to the aperture problem of tracking edge features.

In order to apply contrast maximization to the feature tracking task, it is necessary to define the motion model of the feature. In general, SE(2) space is mainly used as a rigid body transform on 2D space. The SE(2) space contains the rotation and movement of the center of the feature.

First, SO(2), Lie group for 2D rotation, and so(2), Lie algebra for 2D rotation, are expressed as follows.

$$\theta \in \mathbb{R}, \tag{5.1}$$



(a) Frame image (b) Corner (c) Edge

(e) Event image (SAE) (e) Corner (bad) (f) Edge (bad)

Figure 5.2: Corner and edge features for event data. The clear features correspond to the yellow patch and ambiguous features correspond to the blue patch.

$$\begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} \in \mathrm{so}(2), \tag{5.2}$$

$$\mathbf{R} = \exp\begin{pmatrix} 0 & -\theta \\ \theta & 0 \end{pmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \in \mathrm{SO}(2), \tag{5.3}$$

where, θ is the rotation angle. Then, SO(2) can be extended by considering translation as SE (2). SE(2), Lie group for 2D transformation, and se(2), Lie algebra for 2D transformation, are expressed as follows.

$$\begin{bmatrix} \theta, u_1, u_2 \end{bmatrix} \in \mathbb{R}^3, \tag{5.4}$$

$$\begin{bmatrix} 0 & -\theta & u_1 \\ \theta & 0 & u_2 \\ 0 & 0 & 1 \end{bmatrix} \in \mathrm{se}(2), \tag{5.5}$$

$$\mathbf{T} = \exp\begin{pmatrix} 0 & -\theta & u_1 \\ \theta & 0 & u_2 \\ 0 & 0 & 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0_{1\times 2} & 1 \end{bmatrix} \in \mathrm{SE}(2), \tag{5.6}$$

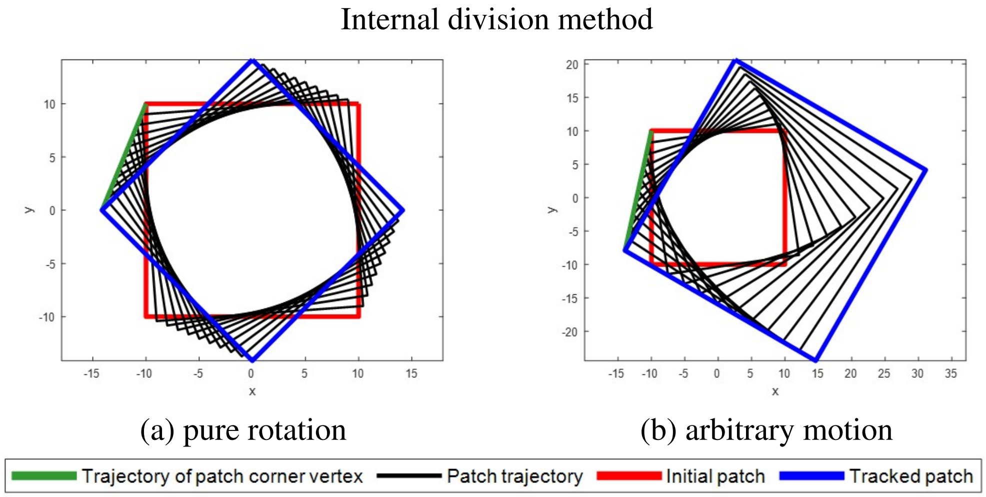
where u_1, u_2 are the translational velocity components. Note that $\mathbf{t} \in \mathbb{R}^2, \mathbf{t} \neq [u_1, u_2]^\intercal$.

Since feature tracking of frame camera processes two frames, the framed based feature tracking module only consider the amount of change of the position and rotation of a target feature. That is to say, frame based feature tracking module uses $\boldsymbol{T}$ directly and $[\theta, u_1, u_2]$ is not calculated.

However, in the case of events that have individual timestamp, I should consider time in the motion model. As a naive method, the motion model at time t_k can be interpolated linearly through internal division as follows.

$$\mathbf{T}(t_k) = \mathbf{I}_{3\times 3} + (\mathbf{T} - \mathbf{I}_{3\times 3}) \cdot \frac{\delta t_k}{\Delta \tau}, \tag{5.7}$$

where $\delta t_k = t_k - \tau_m$ and $\Delta\tau = \tau_{m+1} - \tau_m$ as same in Fig. 3.3. Then, warping can be computed


Internal division method

(a) pure rotation

(b) arbitrary motion

as follows.

$$\mathbf{x}_k(t_k) = \mathbf{T}(t_k) \cdot \mathbf{x}_k(\tau_m), \tag{5.8}$$

$$\begin{bmatrix} x_k(t_k) \\ y_k(t_k) \\ 1 \end{bmatrix} = \mathbf{T}(t_k) \cdot \begin{bmatrix} x_k(\tau_m) \\ y_k(\tau_m) \\ 1 \end{bmatrix}. \tag{5.9}$$

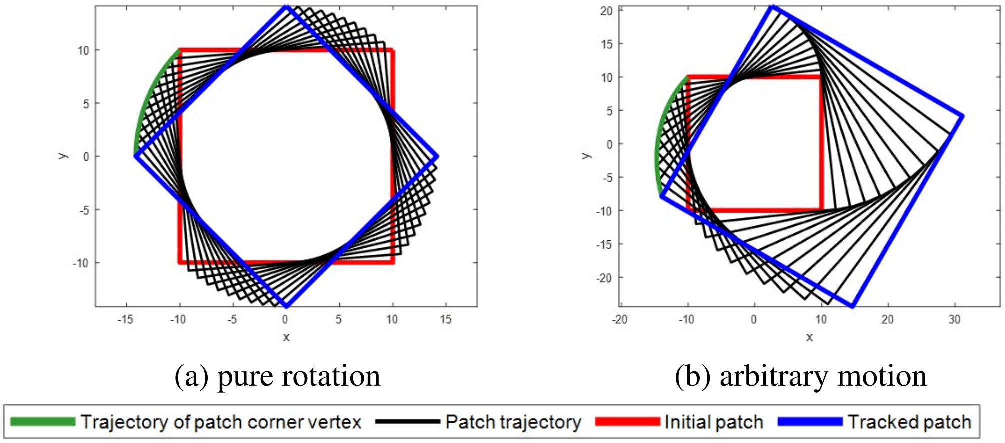
The trajectories of the patch for interpolation models are shown in Section 5.1.1. The trajectory of the vertex of the patch is depicted as a green line. For pure rotational motion, an arc-shaped trajectory should appear for the trajectory of the vertex, but it is expressed as a straight line in the internal division model. Therefore, this naive interpolation method is quite far from the actual behavior and cannot be applied to contrast maximization for feature tracking.

I can define a feature motion model for continuous time by multiplying Eq. (5.5) by time δt_k.

$$\mathbf{T}(t_k) = \exp\begin{pmatrix} 0 & -\theta\delta t_k & u_1\delta t_k \\ \theta\delta t_k & 0 & u_2\delta t_k \\ 0 & 0 & 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R}(t_k) & \mathbf{t}(t_k) \\ 0_{1\times 2} & 1 \end{bmatrix} \tag{5.10}$$

SE(2) does not reflect the scale, thus, tracking may not work well when the camera moves

The proposed motion model



(a) pure rotation

(b) arbitrary motion

Figure 5.3: The trajectories of patch 2D transformation for equal time intervals.

along the principal axis. To supplement this, the following motion model is applied. It is assumed that the scale factors for x-axis and y-axis are the same.

$$\frac{d\mathbf{x}(t)}{dt} = \begin{bmatrix} s & -\theta \\ \theta & s \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \mathbf{A}\mathbf{x}_k(\tau_m) + \mathbf{B} \tag{5.11}$$

The **A** denotes the tangent of the scale and rotation matrix, and the **B** denotes the tangent of the translational motion. For short time intervals, the tangents of scale, rotation and translational motion can be approximated with constant values. Since this system corresponds to the continuous time-invariant system of the state-space model, the solution can be obtained as follows.

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}d\tau \tag{5.12}$$

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \left[e^{\mathbf{A}t} - \mathbf{I}_{2\times 2}\right]\mathbf{A}^{-1}\mathbf{B} \tag{5.13}$$

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = e^{st} \begin{bmatrix} \cos\theta t & -\sin\theta t \\ \sin\theta t & \cos\theta t \end{bmatrix} \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} + \begin{bmatrix} e^{st}\cos\theta t - 1 & -e^{st}\sin\theta t \\ e^{st}\sin\theta t & e^{st}\cos\theta t - 1 \end{bmatrix} \cdot \frac{1}{s^2+\theta^2} \begin{bmatrix} s & \theta \\ -\theta & s \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{5.14}$$

where $\mathbf{A}$ is non-singular matrix s.t. $s^2 + \theta^2 = 0$.

For an event $e_k = (x_k, y_k, p_k, t_k)$, the aligned event at reference time τ_m is compute as follows:

$$\mathbf{x}_k(\tau_m) = e^{-\mathbf{A}\delta t_k}\mathbf{x}_k(t_k) + \left[e^{-\mathbf{A}\delta t_k} - \mathbf{I}_{2\times 2}\right]\mathbf{A}^{-1}\mathbf{B}, \tag{5.15}$$

where $\mathbf{x}_k(t_k) = [x_k, y_k]^\intercal$ is the observed event position.

Since the time difference satisfies that $\delta t_k \ll 1$, the warping function can be approximated using the Taylor series as follows:

$$\mathbf{x}_k(\tau_m) = \mathbf{x}_k(t_k) - (\mathbf{A}\mathbf{x}_k(t_k) + \mathbf{B})\delta t_k + \frac{1}{2}\mathbf{A}(\mathbf{A}\mathbf{x}_k(t_k) + \mathbf{B})\delta {t_k}^2. \tag{5.16}$$

$$\mathbf{I}(\zeta) = \sum_{k=1}^{N_m} \delta_d(\mathbf{x} - \mathbf{x}(\tau_m)), \tag{5.17}$$

$$\text{where} \quad \zeta = \begin{bmatrix} s & \theta & u_1 & u_2 \end{bmatrix}, \tag{5.18}$$

Consequently in Fig. 5.3, the trajectories of patch show correct warping results for equal time intervals. The trajectory of patch corner vertex(green line) shows reasonable results, even when scale change exists. While, Lie group 2D transform cannot reflect the scale changes.

5.1.2 Contrast Maximization Approach

I track a feature by maximizing the contrast of feature patch $\mathbf{P}(\mathbf{f}, \mathbf{I}(\zeta))$ with following cost:

$$\underset{\zeta}{\text{maximize}}\, J(\zeta) \tag{5.19}$$

$$J(\zeta) = \|\mathbf{P}(\mathbf{f}, \mathbf{I}(\zeta))\|^2 \tag{5.20}$$

The Jacobian of the cost function (Eq. (5.20)) is derived as follows:

$$\frac{\partial J}{\partial \zeta} = \frac{\partial J}{\partial \mathbf{I}} \frac{\partial \mathbf{I}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \zeta}, \tag{5.21}$$

where

$$\frac{\partial J}{\partial \mathbf{I}} = \sum_{k=1}^{N} 2I(\mathbf{x}_k, \zeta) \tag{5.22}$$

$$\frac{\partial \mathbf{I}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial I(\mathbf{x}_k, \zeta)}{\partial u} & \frac{\partial I(\mathbf{x}_k, \zeta)}{\partial v} \end{bmatrix} \tag{5.23}$$

$$\frac{\partial \mathbf{x}}{\partial \zeta} = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial s} & \frac{\partial \mathbf{x}}{\partial \theta} & \frac{\partial \mathbf{x}}{\partial u_1} & \frac{\partial \mathbf{x}}{\partial u_2} \end{bmatrix} \tag{5.24}$$

Note that commutative property holds for identity matrix and skew symmetric matrix. Then, derivative of matrix exponential can be computed as follows:

$$e^{\mathbf{A}\delta t_k} = \exp\left[\left(s\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \theta \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\right)\delta t_k\right] \tag{5.25}$$

$$\frac{\partial e^{\mathbf{A}\delta t_k}}{\partial s} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \delta t_k \cdot e^{\mathbf{A}\delta t_k} = \delta t_k e^{\mathbf{A}\delta t_k} \tag{5.26}$$

$$\frac{\partial e^{\mathbf{A}\delta t_k}}{\partial \theta} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \delta t_k \cdot e^{\mathbf{A}\delta t_k} \tag{5.27}$$

$$\frac{\partial \mathbf{A}^{-1}}{\partial s} = \frac{1}{(s^2+\theta^2)^2} \begin{bmatrix} \theta^2 - s^2 & -2s\theta \\ 2s\theta & \theta^2 - s^2 \end{bmatrix} \tag{5.28}$$

$$\frac{\partial \mathbf{A}^{-1}}{\partial \theta} = \frac{1}{(s^2+\theta^2)^2} \begin{bmatrix} -2s\theta & s^2 - \theta^2 \\ \theta^2 - s^2 & -2s\theta \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \frac{\partial \mathbf{A}^{-1}}{\partial s} \tag{5.29}$$

Then,

$$\begin{aligned}\frac{\partial \mathbf{x}}{\partial s} &= \frac{\partial}{\partial s}\left[e^{-\mathbf{A}\delta t_k}\mathbf{x}_k(t_k) + \left(e^{-\mathbf{A}\delta t_k} - \mathbf{I}_{2\times 2}\right)\mathbf{A}^{-1}\mathbf{B}\right] \qquad (5.30)\\ &= -\delta t_k e^{-\mathbf{A}\delta t_k}\mathbf{x}_k(t_k) + \left[-\delta t_k e^{-\mathbf{A}\delta t_k} A^{-1} + \left(e^{-\mathbf{A}\delta t_k} - \mathbf{I}_{2\times 2}\right)\frac{\partial \mathbf{A}^{-1}}{\partial s}\right]\mathbf{B}, \qquad (5.31)\end{aligned}$$

$$\frac{\partial \mathbf{x}}{\partial \theta} = -\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\delta t_k e^{-\mathbf{A}\delta t_k}\mathbf{x}_k(t_k) + \left[-\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\delta t_k e^{-\mathbf{A}\delta t_k}\mathbf{A}^{-1} + \left(e^{-\mathbf{A}\delta t_k} - \mathbf{I}_{2\times 2}\right)\frac{\partial \mathbf{A}^{-1}}{\partial \theta}\right]\mathbf{B}, \tag{5.32}$$

$$= -\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\delta t_k e^{-\mathbf{A}\delta t_k}\mathbf{x}_k(t_k) + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\left[-\delta t_k e^{-\mathbf{A}\delta t_k}\mathbf{A}^{-1} + \left(e^{-\mathbf{A}\delta t_k} - \mathbf{I}_{2\times 2}\right)\frac{\partial \mathbf{A}^{-1}}{\partial s}\right]\mathbf{B}, \tag{5.33}$$

$$= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\frac{\partial \mathbf{x}}{\partial s} \tag{5.34}$$

$$\frac{\partial \mathbf{x}}{\partial u_1} = \left(e^{-\mathbf{A}\delta t_k} - \mathbf{I}_{2\times 2}\right)\mathbf{A}^{-1}\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{5.35}$$

$$\frac{\partial \mathbf{x}}{\partial u_2} = \left(e^{-\mathbf{A}\delta t_k} - \mathbf{I}_{2\times 2}\right)\mathbf{A}^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{5.36}$$

The Jacobian can be simplified by deviating approximation form Eq. (5.16) as follows:

$$\frac{\partial \mathbf{x}}{\partial s} = -\mathbf{x}_k(t_k)\delta t_k + \mathbf{A}\mathbf{x}_k(t_k)\delta {t_k}^2, \tag{5.37}$$

$$\frac{\partial \mathbf{x}}{\partial \theta} = -\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\mathbf{x}_k(t_k)\delta t_k + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\mathbf{A}\mathbf{x}_k(t_k)\delta {t_k}^2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\frac{\partial \mathbf{x}}{\partial s}, \tag{5.38}$$

$$\frac{\partial \mathbf{x}}{\partial u_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}\delta t_k + \frac{1}{2}\mathbf{A}\begin{bmatrix} 1 \\ 0 \end{bmatrix}\delta {t_k}^2, \tag{5.39}$$

$$\frac{\partial \mathbf{x}}{\partial u_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\delta t_k + \frac{1}{2}\mathbf{A}\begin{bmatrix} 0 \\ 1 \end{bmatrix}\delta {t_k}^2. \tag{5.40}$$

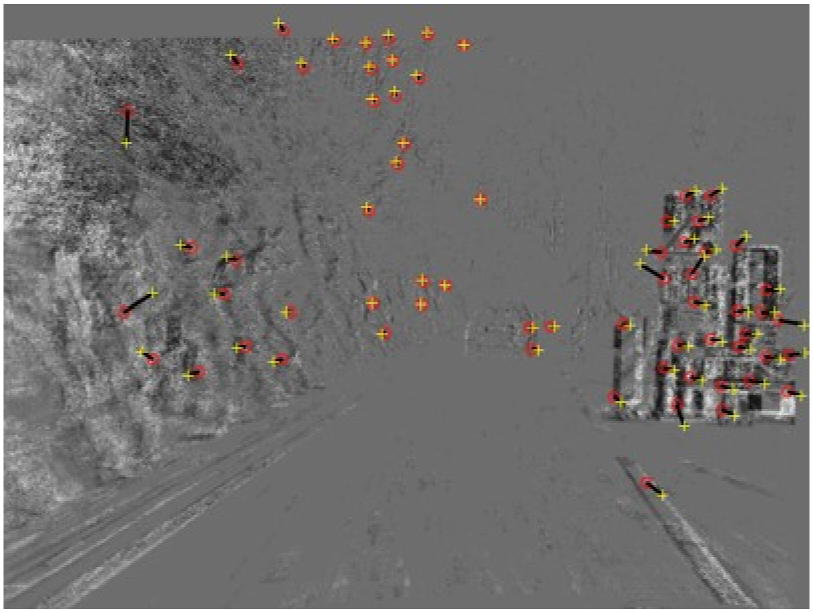


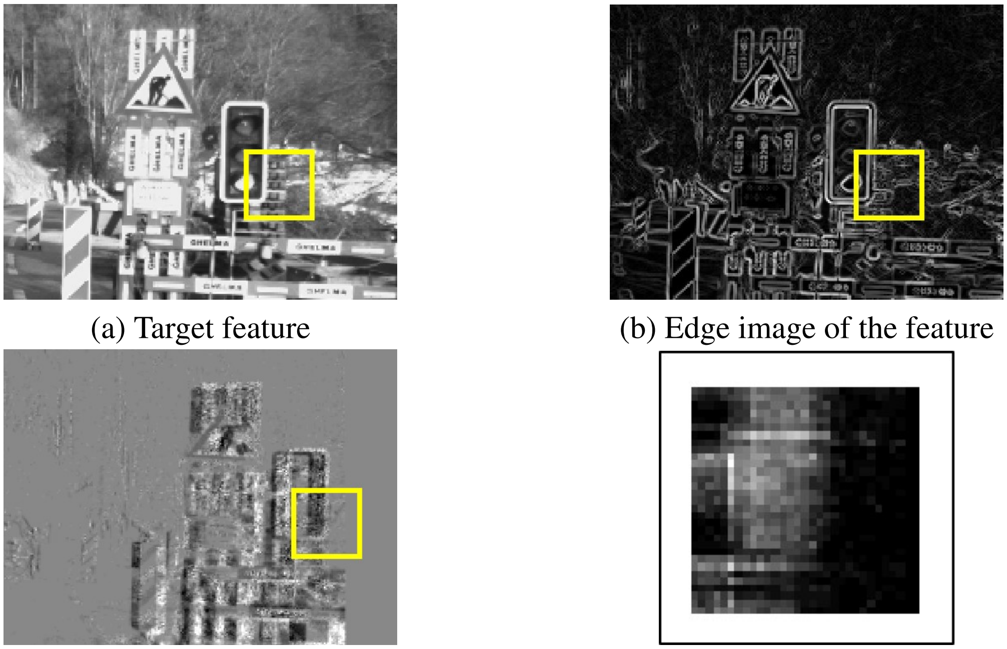
Figure 5.4: Feature tracking fail case

I use RMS-prop optimizer for the optimization problem of Eq. (5.19). The result of the proposed feature tracking is depicted Fig. 5.4. Many features are inaccurately tracked. Unlike other contrast maximization tasks, it easily falls into the local minimum, because target of the image, the patch, is too small. In order to solve the local minimum problem, I apply truncation and zero padding.

I use truncation method as a regularization which is also applied in Chapter 3 to avoid maximizing contrast only for event concentrated areas.

The Fig. 5.5 shows the example of falling into a local minimum.

In contrast maximization, the gradient gets stronger near the optimal solution (in Fig. 5.6, step 4 of the first row). and updated value passes the optimal solution (in Fig. 5.6, step 5 of the first row). When truncation is applied to suppress the gradient near the optimal solution, the convergence value reached optimal value (in Fig. 5.6 from step 1 to 5 of the second row). Fig. 5.6. The converged event patch (in Fig. 5.6, step 5 of the second row) matched the patch of the frame edge image (in Fig. 5.5.(b)).



(a) Target feature

(b) Edge image of the feature

(c) Event image of the feature with polarity

(d) Event image of the feature without polarity

Figure 5.5: Target feature for tracking.

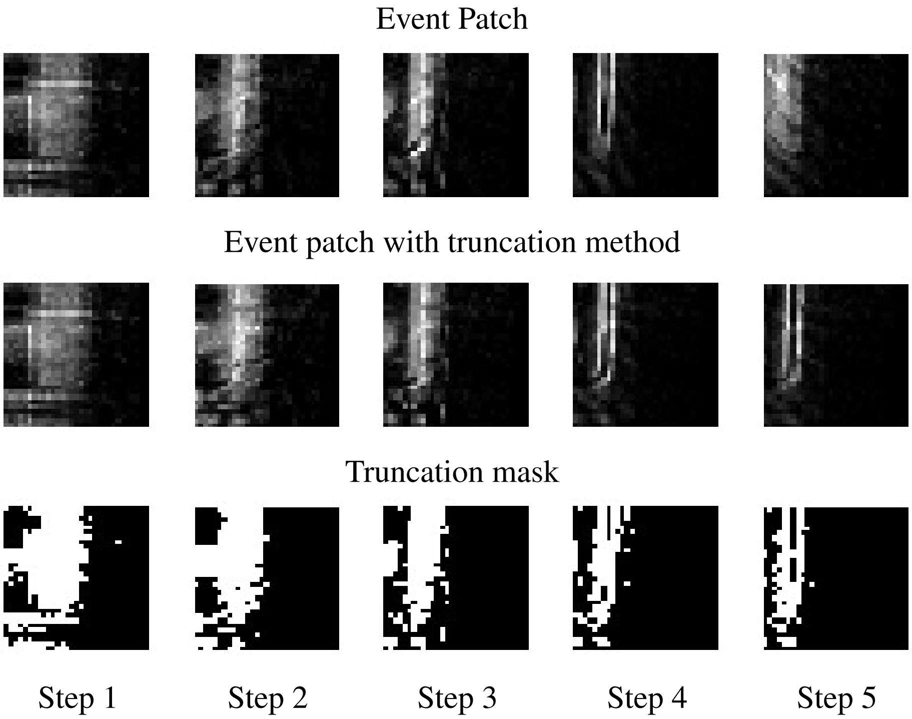


Figure 5.6: Event images during contrast maximization with varying optimization step.

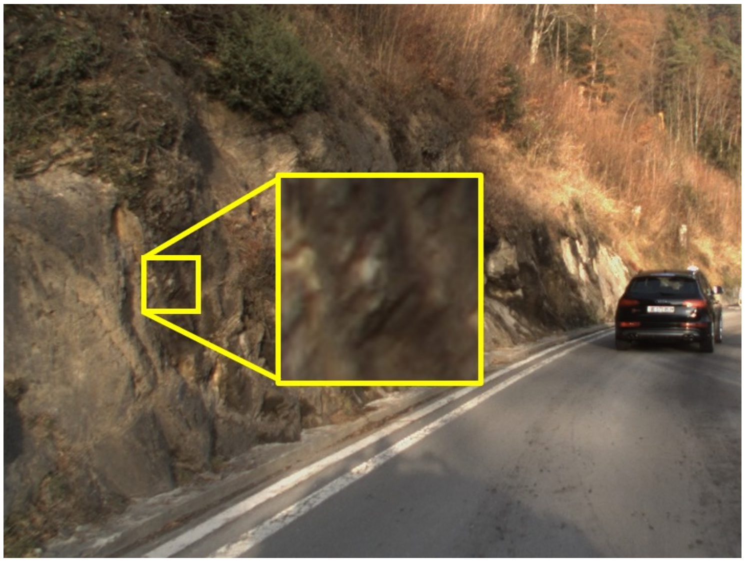


Figure 5.7: Patch of tracking failed feature due to gradient sliding.

Even after truncation is applied, some features are not tracked correctly, because the patch size is still small. Once the feature motion is updated in wrong direction (The red arrows in the first row of Fig. 5.7), the feature motion diverges in one direction and does not converge later. I will refer to this situation as “gradient sliding” in this dissertation. Fig. 5.7 depicts the gradient sliding feature of Fig. 5.8.

If the intensity of the event patch boundary is set to 0 (zero padding), the gradients near the boundaries are oriented to the center of the patch (The blue arrows in the first row of Fig. 5.7). Through this, it is possible to prevent contrast maximization from divergence, and it is confirmed that feature tracking works well as in Fig. 5.8.

I evaluate the proposed feature tracking method in ECDS data set [54] as in Fig. 5.9, Fig. 5.10 and Fig. 5.11.

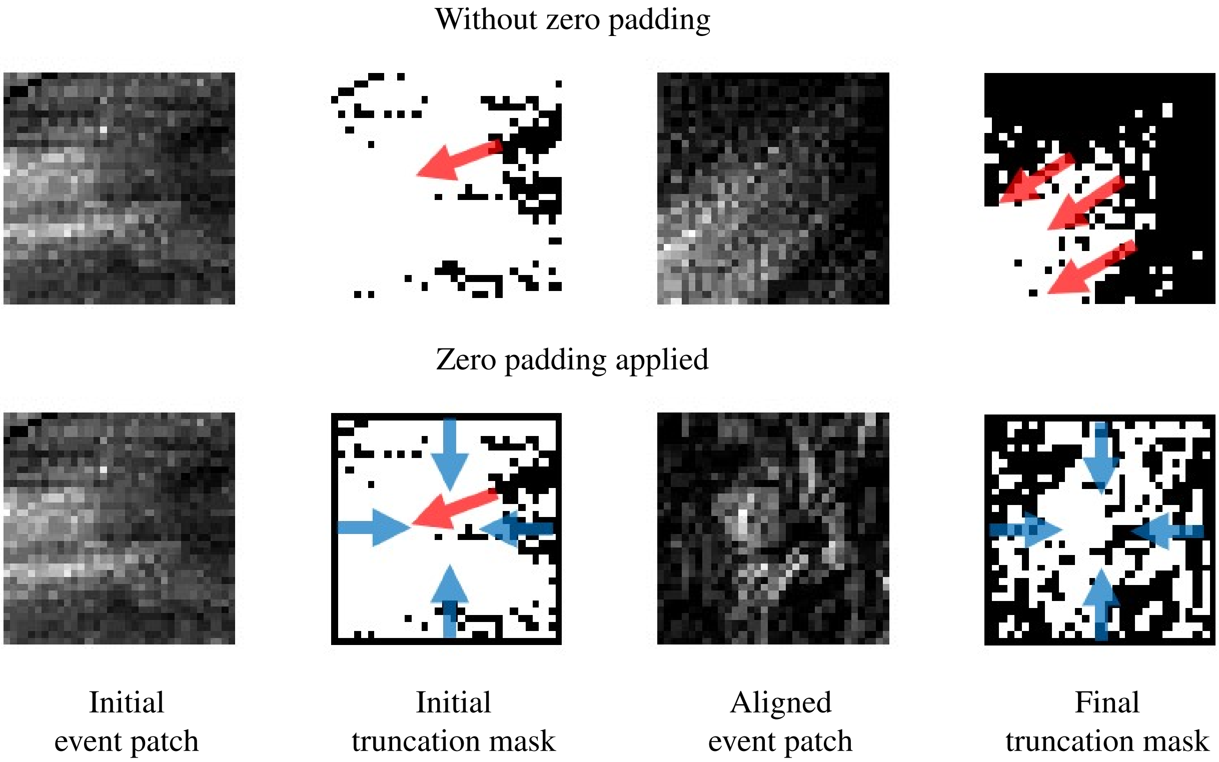


Figure 5.8: Feature tracking with zero padding. Aligned event patch with zero padding shows highly similarity to target patch of Fig. 5.7, while aligned event patch without zero padding shows diverged result due to the gradient sliding. The red arrows are gradients which can induce gradient sliding, and the blue arrows are center oriented gradients from zero padded boundaries that prevent gradient sliding.

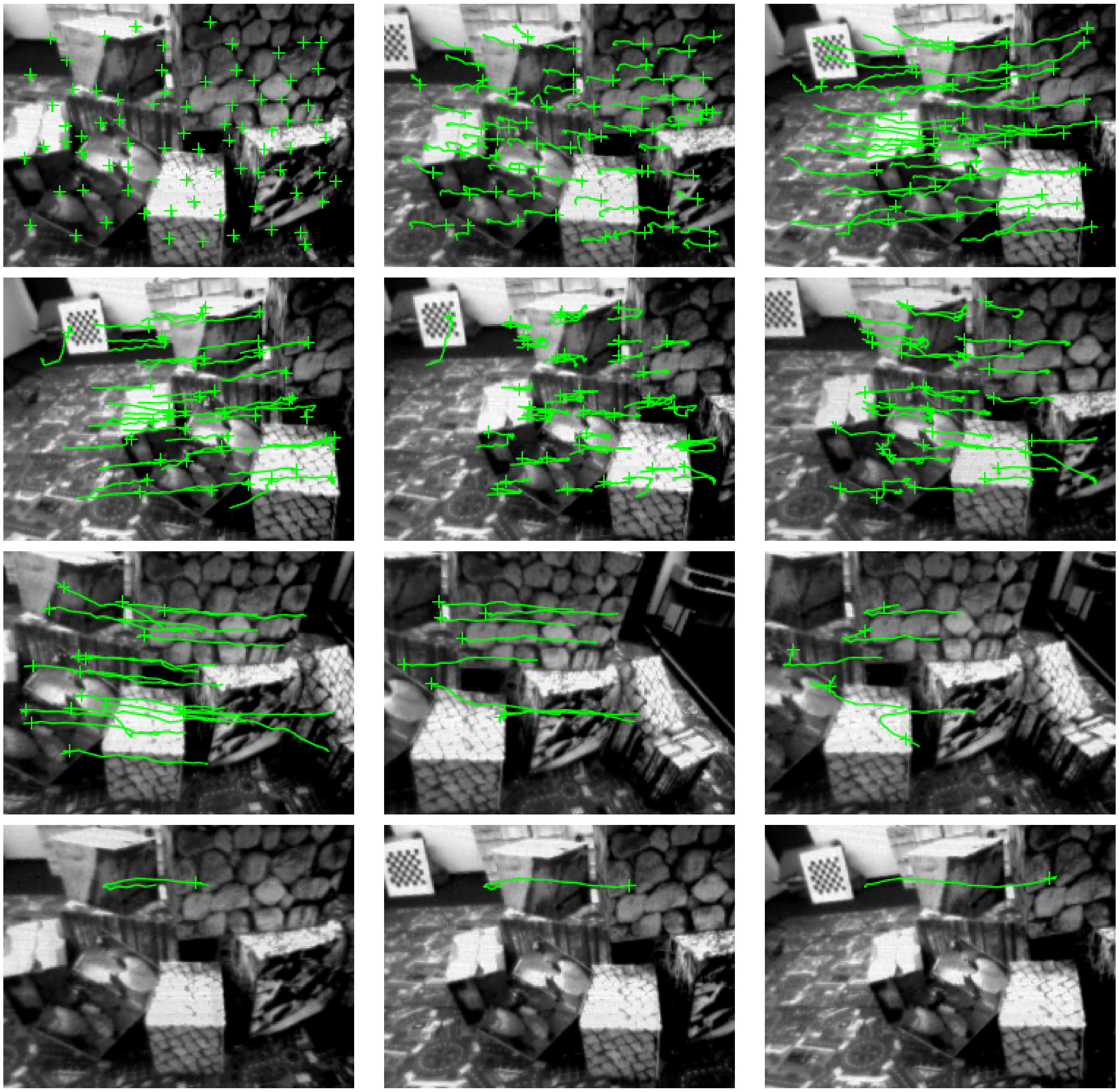


Figure 5.9: Illustration of feature tracking results in ECDS:*boxes_rotation*.

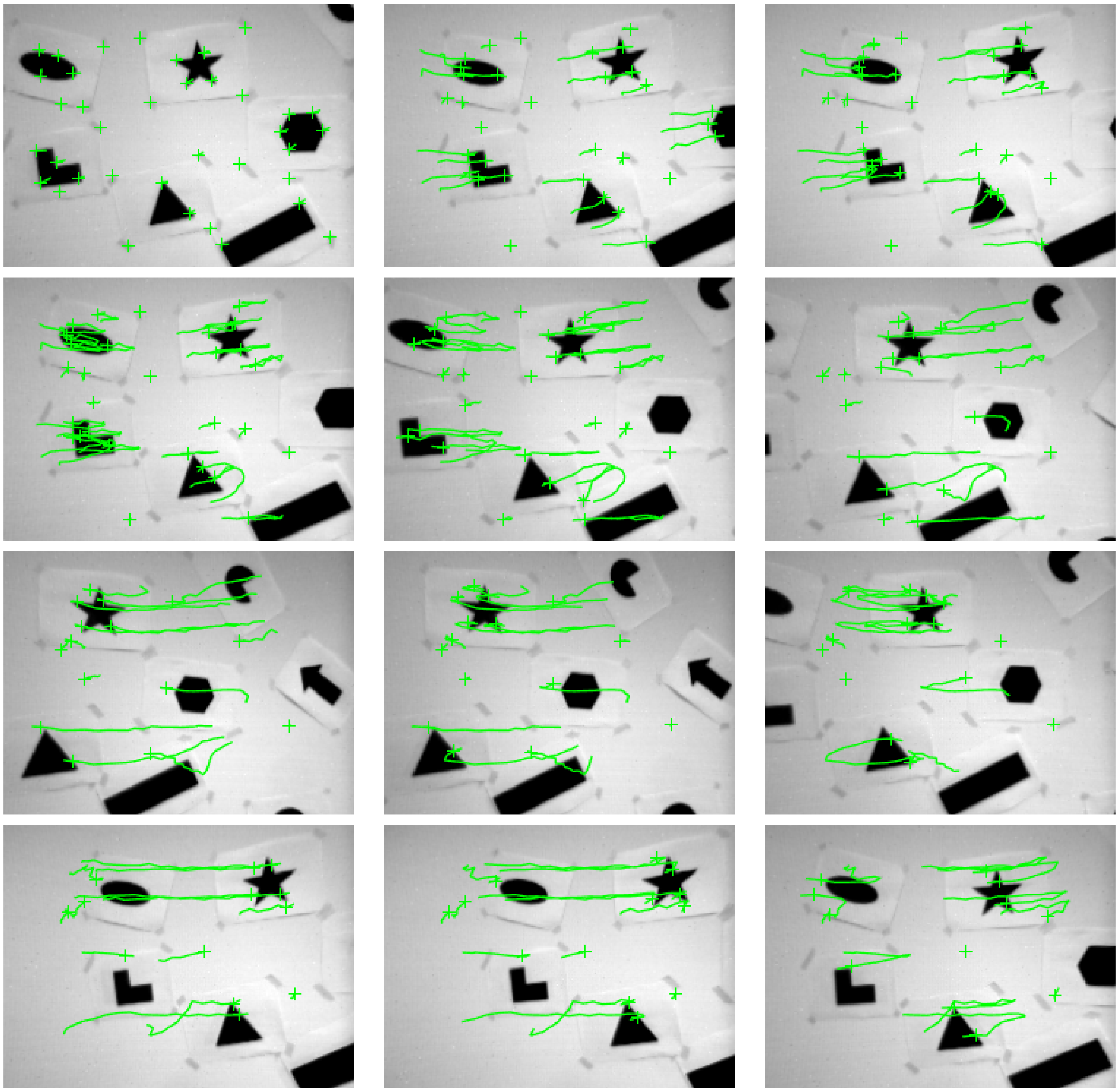


Figure 5.10: Illustration of feature tracking results in ECDS:*shapes_rotation*.

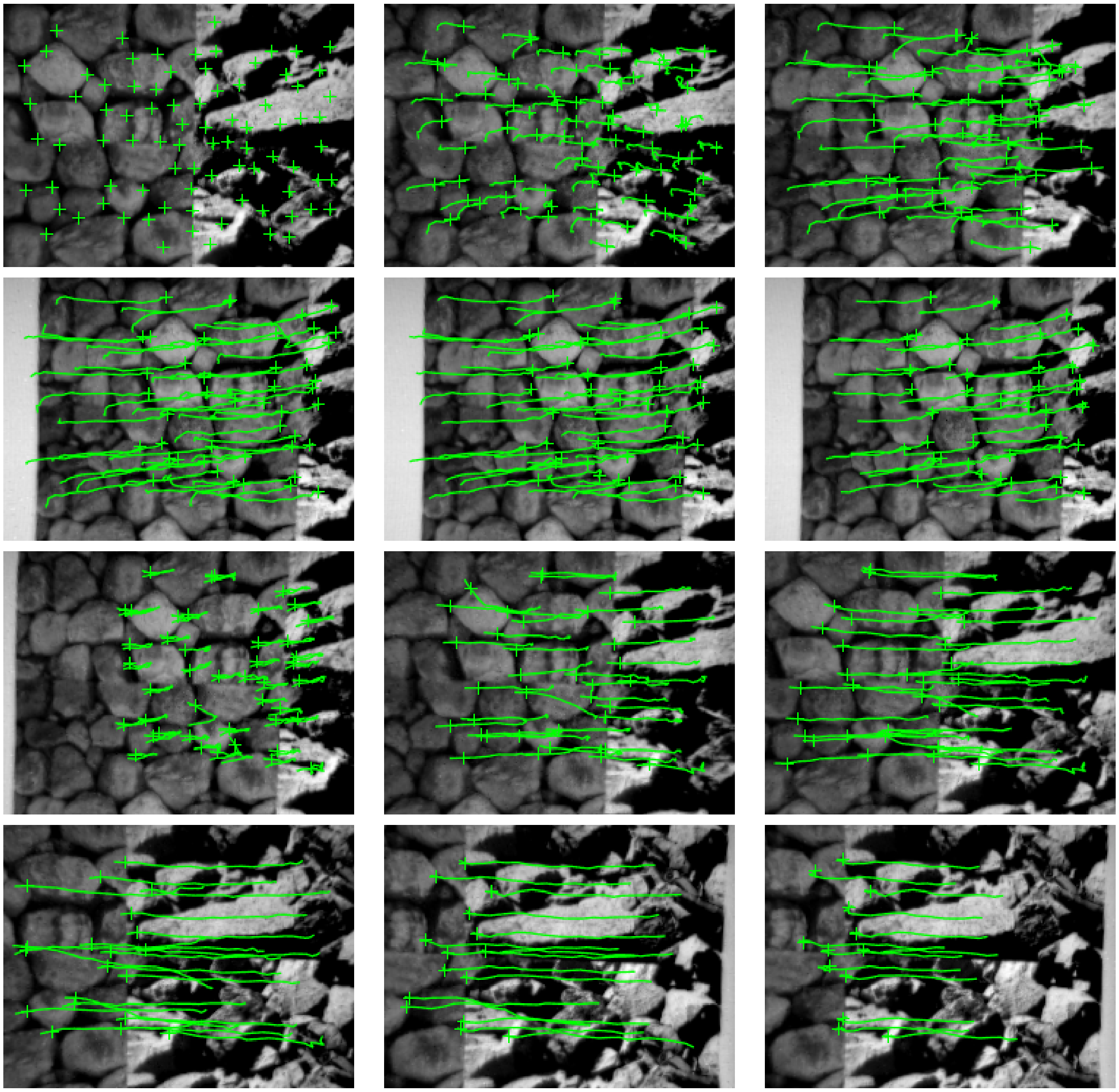


Figure 5.11: Illustration of feature tracking results in ECDS:*poster_rotation*.

5.2 Pose Estimation

5.2.1 ORB-SLAM with Hetero-Stereo Camera

Through the depth estimation method introduced in Chapter 4, the camera motion can be estimated in the hetero-stereo setup. When only monocular frame camera is used, camera motion suffers from the scale ambiguity problem. By attaching an event camera that can be used in a challenging environment, it is possible to estimate the absolute scale preserved depth. Here, I implement the camera pose estimation algorithm by fusion of ORB-SLAM3 [61] and the proposed method in Chapter 4.

I quantitatively and qualitatively compare the localization performance of ORB-SLAM3 [61] with monocular camera and ORB-SLAM3 with hetero-stereo camera. The evaluation is performed in DSEC [51] and TUM-VIE [49]. Since DSEC dataset does not provide ground truth poses, I qualitatively evaluate the algorithm by comparing road of the satellite map and path of the camera. I used RVIZ-satellite [62] to draw satellite map. Then, I qualitatively and quantitatively evaluate the proposed method in TUM-VIE dataset which provides ground truth poses. For fair comparison, I calibrate the global scale of each method to the ground truth and then evaluate the methods.

Monocular setup

ATE	1d-trans	3d-trans	6dof
RMSE	0.028967	0.082613	0.048711
MEAN	0.023494	0.073844	0.046150
STD	0.016945	0.037040	0.015587

(unit: m)

The proposed event and frame setup

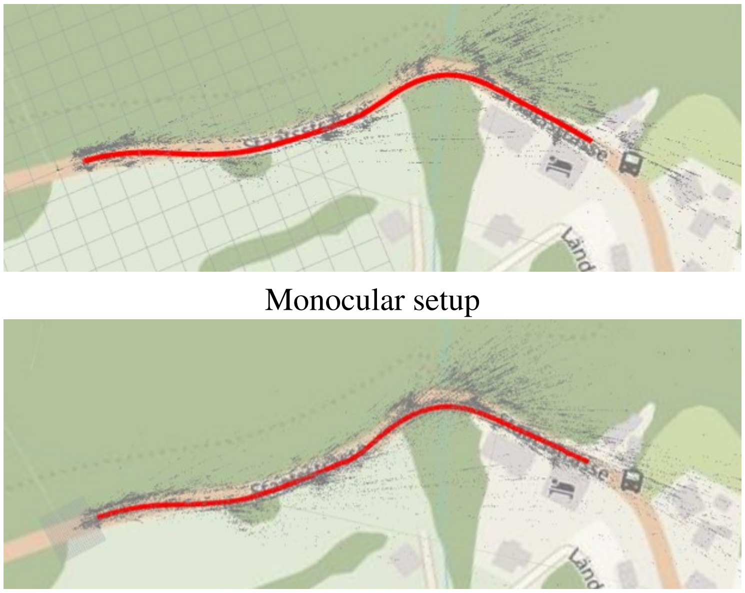
ATE	1d-trans	3d-trans	6dof
RMSE	**0.023498**	**0.024156**	**0.019971**
MEAN	**0.021383**	**0.021908**	**0.018307**
STD	**0.009744**	**0.010176**	**0.007980**

(unit: m)

Table 5.1: The results of absolute trajectory error(ATE).

The quantitative evaluation results for the TUM-VIE dataset are shown in Table 5.1. I evaluate the methods using absolute trajectory error(ATE) metric with the code provided by ORB-SLAM3 [61]. Because the proposed approach can estimate depth with absolute scale, the proposed setup show better performance for all data sequences in ATE than the monocular setup.

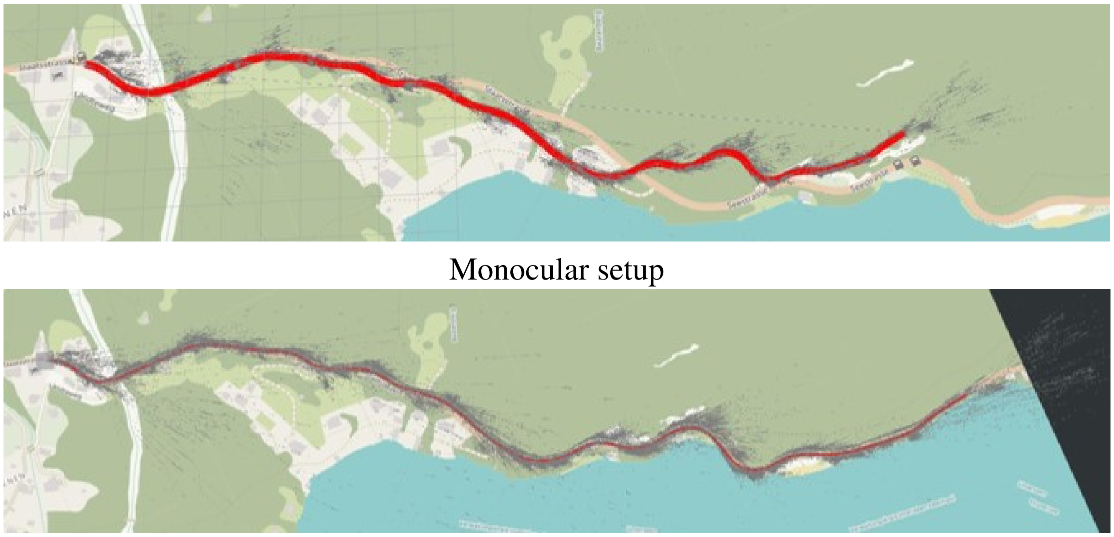
The quantitative results are illustrated in the figures below. In the case of the monocular version of ORB-SLAM3, scale consistency is not maintained when the rotational motion is large. In the *interlaken_c* and *interlaken_f* data sequence, there are no large rotational motions, so the pose estimation performance is not far behind the hetero stereo version. However, since there are several large rotational motions in *interlaken_d*, *interlaken_e*, *interlaken_g* sequences, the scale consistency is not maintained. Although the performance of monocular ORB-SLAM3 is degraded due to scale ambiguity, the proposed method is not affected by scale issue.



Monocular setup

The proposed event and frame setup

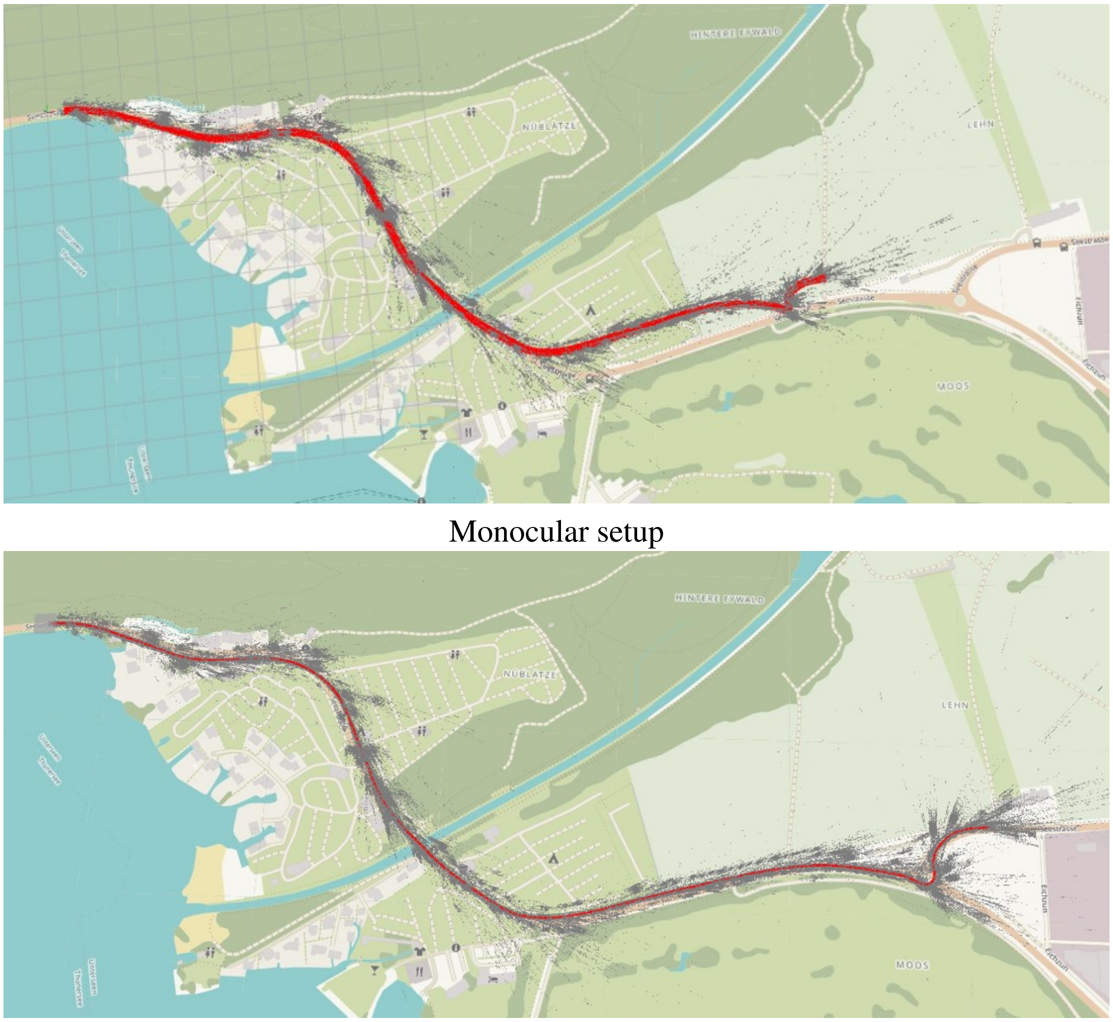
Figure 5.12: Pose estimation results in DSEC:*interlaken_c* data sequence.



Monocular setup

The proposed event and frame setup

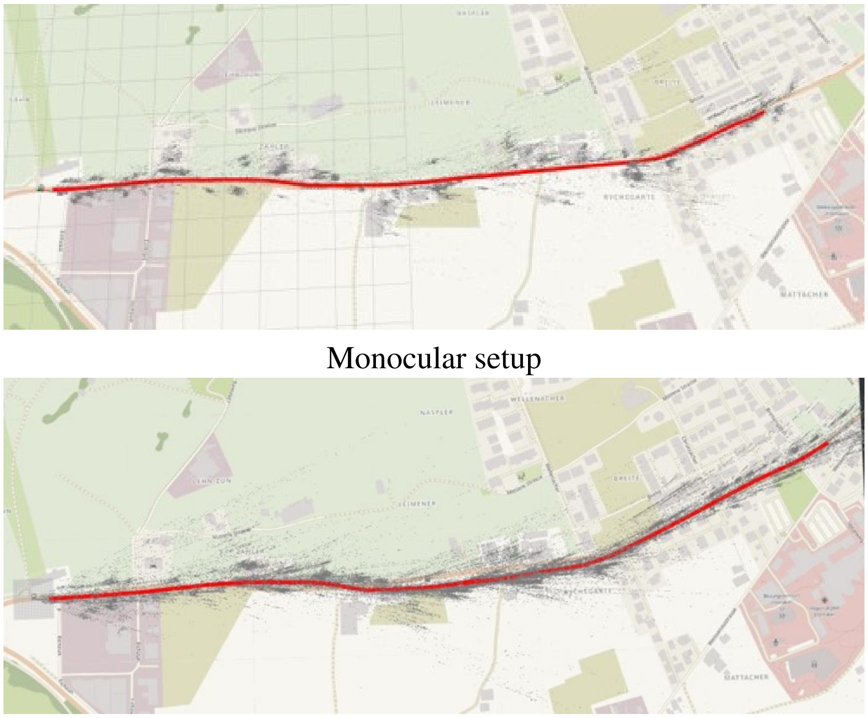
Figure 5.13: Pose estimation results in DSEC:*interlaken_d* data sequence.



Monocular setup

The proposed event and frame setup

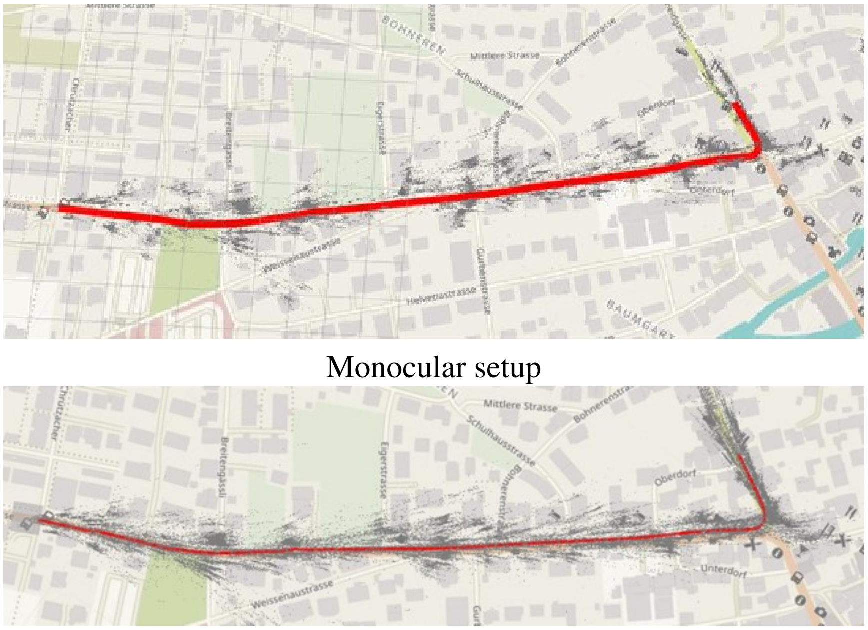
Figure 5.14: Pose estimation results in DSEC:*interlaken_e* data sequence.



Monocular setup

The proposed event and frame setup

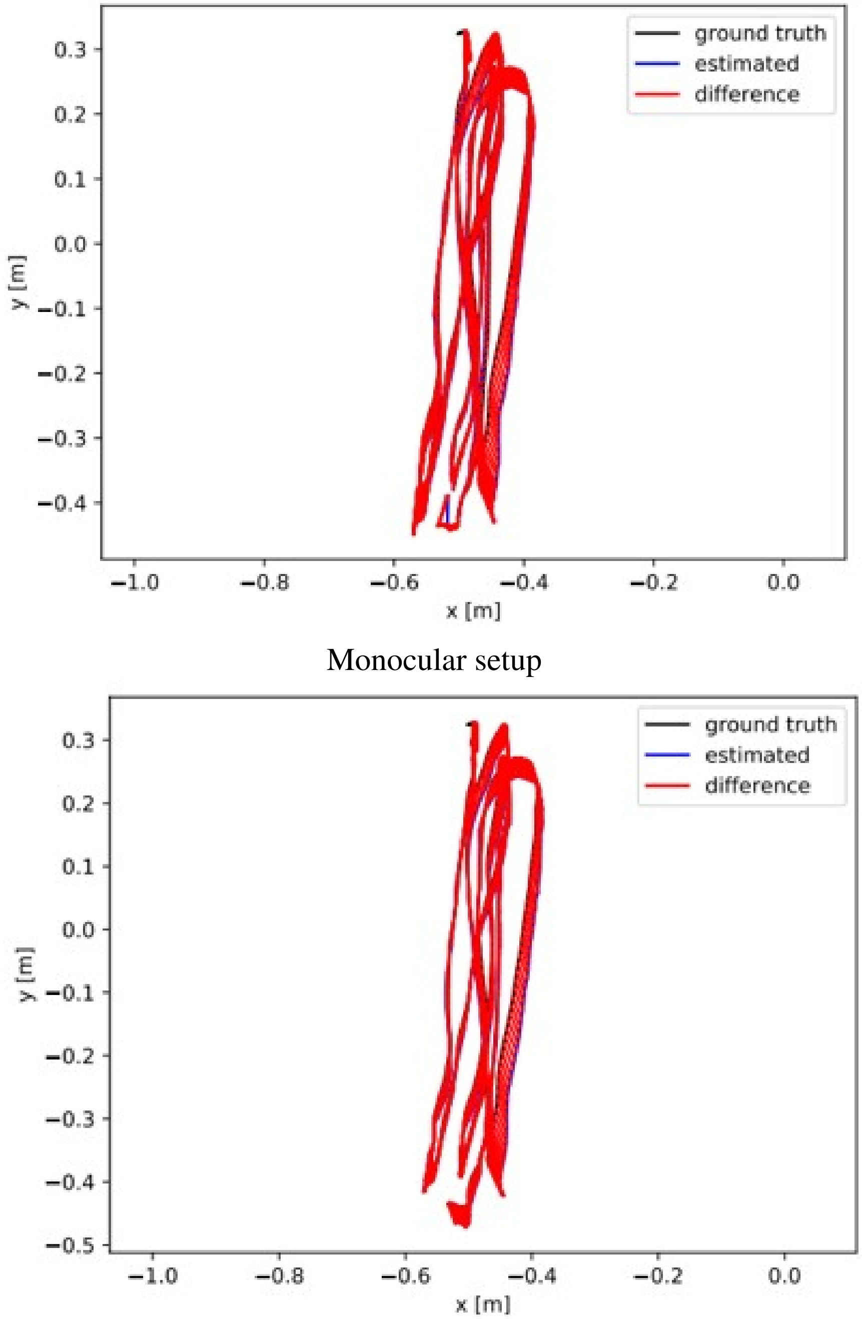
Figure 5.15: Pose estimation results in DSEC:*interlaken_f* data sequence.



Monocular setup

The proposed event and frame setup

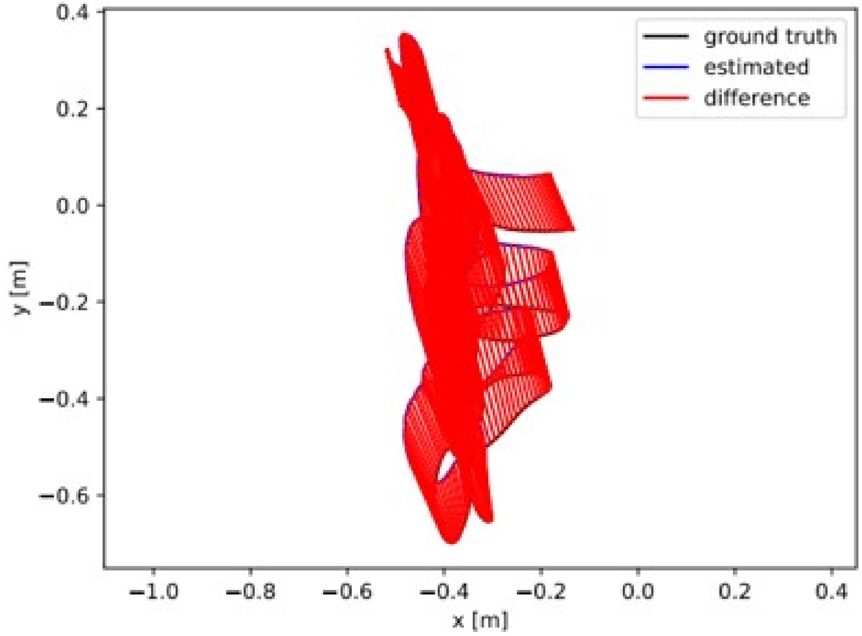
Figure 5.16: Pose estimation results in DSEC:*interlaken_g* data sequence.



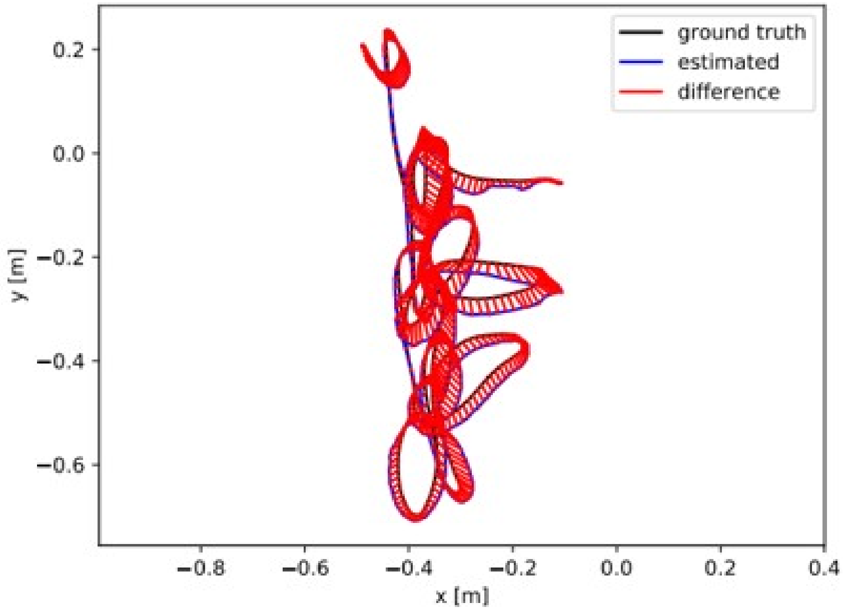
Monocular setup

The proposed event and frame setup

Figure 5.17: Pose estimation results in TUM-VIE:*mocap_1d* data sequence.

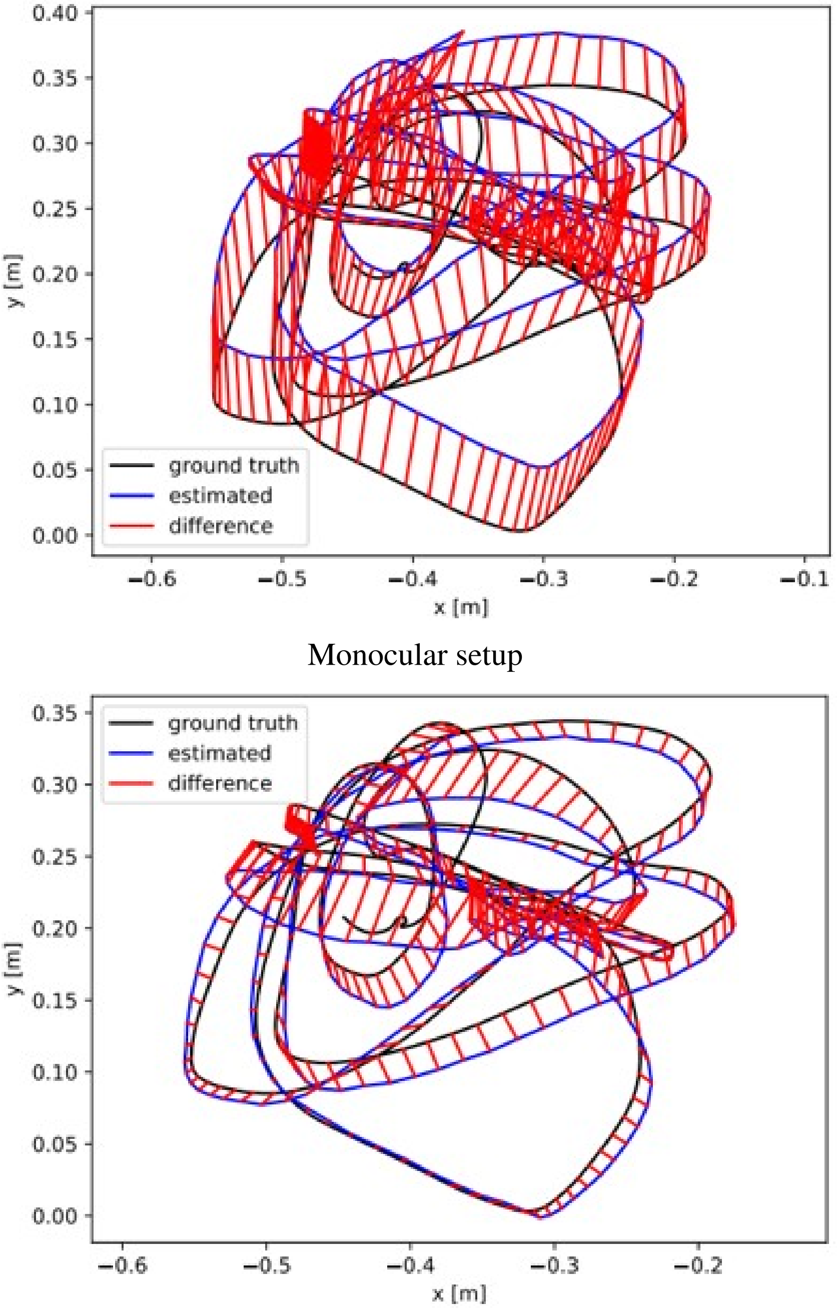


Monocular setup



The proposed event and frame setup

Figure 5.18: Pose estimation results in TUM-VIE:*mocap_3d* data sequence.



Monocular setup

The proposed event and frame setup

Figure 5.19: Pose estimation results in TUM-VIE:*mocap_6dof* data sequence.

5.3 Future Work

Pose estimation fails in some cases when the camera quickly passes through areas where the map has not been reconstructed. If translational motion exists, the depth can be estimated with an event camera, and pose estimation is possible. If the newly discovered area is observed with fast rotational motion, it becomes difficult to estimate the depth, and it is difficult to estimate the translational motion. In noisy visual data, without depth, a panning motion cannot be distinguished from an x-axis translational motion.

As a future work, I will propose a method for depth estimation in a challenging environment with a hetero-stereo setup with one more event camera added. It is expected that not only depth estimation and existing vision algorithms can be used with the frame camera, but also depth can be stably estimated even for fast pure rotational motion by using the event camera as a stereo. The initial stereo matching and the proposed stereo matching with aligned events in Chapter 4 will be applied to the stereo events, and contrast maximization with globally aligned events will be adapted to estimate 6-DOF camera pose.

6

Conclusion

I propose accurate, intuitive and efficient methods to align events with geometric approach. I apply the aligning methods to motion estimation, depth estimation, and feature tracking. The proposed event aligning method assumes that a motion is constant for a short time interval. By utilizing Lie algebra, the proposed method aligns events at a reference time considering the time of every individual event. The event images obtained by the proposed aligned events can identify the shape of an object by depicting the edge of the frame images. In addition, the proposed methods of motion estimation and depth estimation can run in real-time, since the proposed methods consider the computational efficiency. The proposed event alignment method achieves high accuracy for the angular motion estimation, and the depth estimation performance is equivalent to the stereo matching method with frame images. In addition, through the estimated depth from hetero stereo matching, the pose estimation performance is improved than the monocular frame camera. In order to track features even when frame cameras fail, I also propose the feature tracking method for event cameras.

In this dissertation, I show that the event camera can accurately depict edge images with a geometric approach. I expect that the proposed study will help understand the geometrical characteristics of event data.

A

Detailed Derivation of Contrast for Rotational Motion Estimation

This section shows a detailed derivation of the contrast for rotational motion estimation. The cost function $J(\omega_m)$ is the contrast of the event image, and formulated as follows:

$$J(\omega) = \|\mathbf{I}(\omega)\|^2 . \tag{A.1}$$

By expressing all without omission, the event image at pixel x_k, $I(\mathbf{x}_k, \omega)$, can be formulated as follows:

$$\mathbf{I}(\omega) = \sum_{k=1}^{N} I(\mathbf{x}_k, \omega) \tag{A.2}$$

$$I(\mathbf{x}_k, \omega) = \delta_d(\mathbf{x} - \pi(\mathrm{w}(\mathbf{x}_k, \omega_m, \delta t_k))) \tag{A.3}$$

Here, $\pi([x,y,z]^\intercal)$ is the projection function, which satisfying:

$$\pi\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} f_x x/z + c_x \\ f_y y/z + c_y \end{bmatrix}. \tag{A.4}$$

Using the chain rule, Jacobian is derived as follows:

$$\frac{\partial J(\omega)}{\partial \omega} = \frac{\partial J}{\partial \mathbf{I}} \frac{\partial \mathbf{I}}{\partial \pi} \frac{\partial \pi}{\partial \mathrm{w}} \frac{\partial \mathrm{w}}{\partial \omega}, \tag{A.5}$$

$$\frac{\partial J}{\partial \mathbf{I}} = \sum_{k=1}^{N} 2I(\mathbf{x}_k, \omega) \tag{A.6}$$

$$\frac{\partial \mathbf{I}}{\partial \pi} = \nabla I(\mathbf{x}_k, \omega) = \begin{bmatrix} \frac{\partial I(\mathbf{x}_k, \omega)}{\partial u} & \frac{\partial I(\mathbf{x}_k, \omega)}{\partial v} \end{bmatrix} \tag{A.7}$$

$$\begin{aligned} \frac{\partial \pi}{\partial \mathrm{w}} &= \begin{bmatrix} \frac{\partial (f_x x/z + c_x)}{\partial x} & \frac{\partial (f_x x/z + c_x)}{\partial y} & \frac{\partial (f_x x/z + c_x)}{\partial z} \\ \frac{\partial (f_y y/z + c_y)}{\partial x} & \frac{\partial (f_y y/z + c_y)}{\partial y} & \frac{\partial (f_y y/z + c_y)}{\partial z} \end{bmatrix} \\ &= \begin{bmatrix} f_x/z & 0 & -\frac{\partial f_x x}{z^2} \\ 0 & f_y/z & -\frac{\partial f_y y}{z^2} \end{bmatrix} \end{aligned} \tag{A.8}$$

$$\frac{\partial \mathrm{w}}{\partial \omega} = \frac{\partial e^{\omega \delta t_k} x'}{\partial e^{\omega \delta t_k}} \cdot \frac{\partial e^{\omega \delta t_k}}{\partial \omega} \tag{A.9}$$

Let $e^{\omega \delta t_k}$ be:

$$e^{\omega \delta t_k} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{A.10}$$

Then,

$$e^{\omega\delta t_k}x' = \begin{bmatrix} r_{11}x + r_{12}y + r_{13}z \\ r_{21}x + r_{22}y + r_{23}z \\ r_{31}x + r_{32}y + r_{33}z \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \tag{A.11}$$

$$\begin{aligned} \frac{\partial e^{\omega\delta t_k}x'}{\partial e^{\omega\delta t_k}} &= \begin{bmatrix} \frac{\partial\alpha}{\partial r_{11}} & \frac{\partial\alpha}{\partial r_{21}} & \frac{\partial\alpha}{\partial r_{31}} & \frac{\partial\alpha}{\partial r_{12}} & \frac{\partial\alpha}{\partial r_{22}} & \frac{\partial\alpha}{\partial r_{32}} & \frac{\partial\alpha}{\partial r_{13}} & \frac{\partial\alpha}{\partial r_{23}} & \frac{\partial\alpha}{\partial r_{33}} \\ \frac{\partial\beta}{\partial r_{11}} & \frac{\partial\beta}{\partial r_{21}} & \frac{\partial\beta}{\partial r_{31}} & \frac{\partial\beta}{\partial r_{12}} & \frac{\partial\beta}{\partial r_{22}} & \frac{\partial\beta}{\partial r_{32}} & \frac{\partial\beta}{\partial r_{13}} & \frac{\partial\beta}{\partial r_{23}} & \frac{\partial\beta}{\partial r_{33}} \\ \frac{\partial\gamma}{\partial r_{11}} & \frac{\partial\gamma}{\partial r_{21}} & \frac{\partial\gamma}{\partial r_{31}} & \frac{\partial\gamma}{\partial r_{12}} & \frac{\partial\gamma}{\partial r_{22}} & \frac{\partial\gamma}{\partial r_{32}} & \frac{\partial\gamma}{\partial r_{13}} & \frac{\partial\gamma}{\partial r_{23}} & \frac{\partial\gamma}{\partial r_{33}} \end{bmatrix} \\ &= \begin{bmatrix} x & 0 & 0 & y & 0 & 0 & z & 0 & 0 \\ 0 & x & 0 & 0 & y & 0 & 0 & z & 0 \\ 0 & 0 & x & 0 & 0 & y & 0 & 0 & z \end{bmatrix} \end{aligned} \tag{A.12}$$

$$\begin{aligned} \frac{\partial e^{\omega\delta t_k}}{\partial\omega} &= \frac{\partial e^{(G_1 w_1 + G_2 w_2 + G_3 w_3)\delta t_k}}{\partial\omega} \\ &= \begin{bmatrix} \frac{\partial e^{(G_1 w_1 + G_2 w_2 + G_3 w_3)\delta t_k}}{\partial\omega_1} & \frac{\partial e^{(G_1 w_1 + G_2 w_2 + G_3 w_3)\delta t_k}}{\partial\omega_2} & \frac{\partial e^{(G_1 w_1 + G_2 w_2 + G_3 w_3)\delta t_k}}{\partial\omega_3} \end{bmatrix} \end{aligned} \tag{A.13}$$

The derivative of exponential matrix of linear combination can be easily derived, if commutative property holds on matrices. If $\omega\delta t_k \ll 1$, the approximate form of Eq. (A.13) can be computed as follows. The exact solution can be found at [63].

$$\frac{\partial e^{\omega \delta t_k}}{\partial \omega} \approx \begin{bmatrix} \mathrm{vec}(G_1) & \mathrm{vec}(G_2) & \mathrm{vec}(G_3) \end{bmatrix} \delta t_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \delta t_k \qquad \text{(A.14)}$$

By chaining derivatives, Jacobian can be summarized as follows:

$$\frac{\partial J(\omega)}{\partial \omega} = \sum_{k=1}^{N_m} 2I(\mathbf{x}_k, \omega) \begin{bmatrix} \frac{\partial I(\mathbf{x}_k,\omega)}{\partial u} & \frac{\partial I(\mathbf{x}_k,\omega)}{\partial v} \end{bmatrix} \begin{bmatrix} -\bar{x}_k \bar{y}_k f_x & (1+\bar{x}_k^2) f_x & -\bar{y}_k f_x \\ -(1+\bar{y}_k^2) f_y & \bar{x}_k \bar{y}_k f_y & \bar{x}_k f_y \end{bmatrix} \delta t_k, \qquad \text{(A.15)}$$

References

[1] G. Gallego and D. Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 632–639, 2017.

[2] C. Reinbacher, G. Munda, and T. Pock, "Real-time panoramic tracking for event cameras," in *2017 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2017, pp. 1–9.

[3] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[4] Z. Wang, F. C. Ojeda, A. Bisulco, D. Lee, C. J. Taylor, K. Daniilidis, M. A. Hsieh, D. D. Lee, and V. Isler, "Ev-catcher: High-speed object catching using low-latency event-based neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8737–8744, 2022.

[5] D. Falanga, S. Kim, and D. Scaramuzza, "How fast is too fast? the role of perception latency in high-speed sense and avoid," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1884–1891, 2019.

[6] N. Messikommer, D. Gehrig, A. Loquercio, and D. Scaramuzza, "Event-based asynchronous sparse convolutional networks," in *European Conference on Computer Vision*. Springer, 2020, pp. 415–431.

[7] S. Lin, F. Xu, X. Wang, W. Yang, and L. Yu, "Efficient spatial-temporal normalization of sae representation for event camera," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4265–4272, 2020.

[8] Y. Li, H. Zhou, B. Yang, Y. Zhang, Z. Cui, H. Bao, and G. Zhang, “Graph-based asynchronous event processing for rapid object recognition,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 934–943.

[9] L. Wang, Y. Chae, S.-H. Yoon, T.-K. Kim, and K.-J. Yoon, “Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 608–619.

[10] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, “High speed and high dynamic range video with an event camera,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[11] L. Wang, Y.-S. Ho, K.-J. Yoon *et al.*, “Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 081–10 090.

[12] S. Tulyakov, A. Bochicchio, D. Gehrig, S. Georgoulis, Y. Li, and D. Scaramuzza, “Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 755–17 764.

[13] H. Kim, S. Leutenegger, and A. J. Davison, “Real-time 3d reconstruction and 6-dof tracking with an event camera,” in *European Conference on Computer Vision*. Springer, 2016, pp. 349–364.

[14] G. Gallego, J. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza, “Event-based, 6-dof camera tracking from photometric depth maps.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 2402–2412, 2017.

[15] H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza, “Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 593–600, 2016.

[16] Y. Zhou, G. Gallego, and S. Shen, “Event-based stereo visual odometry,” *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1433–1450, 2021.

[17] A. Zihao Zhu, N. Atanasov, and K. Daniilidis, “Event-based visual inertial odometry,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5391–5399.

[18] 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 Robotics and Automation Letters*, vol. 3, no. 2, pp. 994–1001, 2018.

[19] D. Liu, A. Parra, and T.-J. Chin, “Spatiotemporal registration for event-based visual odometry,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4937–4946.

[20] M. Gehrig, S. B. Shrestha, D. Mouritzen, and D. Scaramuzza, “Event-based angular velocity regression with spiking networks,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4195–4202.

[21] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, “Low-latency visual odometry using event-based feature tracks,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 16–23.

[22] A. Censi and D. Scaramuzza, “Low-latency event-based visual odometry,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 703–710.

[23] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, “Emvs: Event-based multi-view stereo—3d reconstruction with an event camera in real-time,” *International Journal of Computer Vision*, vol. 126, no. 12, pp. 1394–1414, 2018.

[24] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. Davison, “Simultaneous mosaicing and tracking with an event camera,” in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014, pp. 66.1–66.12.

[25] D. Liu, A. Parra, and T.-J. Chin, “Globally optimal contrast maximisation for event-based motion estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6349–6358.

[26] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3867–3876.

[27] A. Z. Zhu, Y. Chen, and K. Daniilidis, “Realtime time synchronized event-based stereo,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 433–447.

[28] Y. Zhou, G. Gallego, H. Rebecq, L. Kneip, H. Li, and D. Scaramuzza, “Semi-dense 3d reconstruction with a stereo event camera,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 235–251.

[29] A. Hadviger, I. Marković, and I. Petrović, “Stereo event lifetime and disparity estimation for dynamic vision sensors,” in *2019 European Conference on Mobile Robots (ECMR)*. IEEE, 2019, pp. 1–6.

[30] A. Andreopoulos, H. J. Kashyap, T. K. Nayak, A. Amir, and M. D. Flickner, “A low power, high throughput, fully event-based stereo system,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7532–7542.

[31] D. Zou, P. Guo, Q. Wang, X. Wang, G. Shao, F. Shi, J. Li, and P.-K. Park, “Context-aware event-driven stereo matching,” in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1076–1080.

[32] Z. Xie, S. Chen, and G. Orchard, “Event-based stereo depth estimation using belief propagation,” *Frontiers in neuroscience*, vol. 11, p. 535, 2017.

[33] S.-H. Ieng, J. Carneiro, M. Osswald, and R. Benosman, “Neuromorphic event-based generalized time-based stereovision,” *Frontiers in neuroscience*, vol. 12, p. 442, 2018.

[34] D. Zou, F. Shi, W. Liu, J. Li, Q. Wang, P.-K. Park, C.-W. Shi, Y. J. Roh, and H. E. Ryu, “Robust dense depth map estimation from sparse dvs stereos,” in *British Mach. Vis. Conf. (BMVC)*, vol. 1, 2017.

[35] L. A. Camunas-Mesa, T. Serrano-Gotarredona, S. H. Ieng, R. B. Benosman, and B. Linares-Barranco, “On the use of orientation filters for 3d reconstruction in event-driven stereo vision,” *Frontiers in neuroscience*, vol. 8, p. 48, 2014.

[36] S. Ghosh and G. Gallego, “Multi-event-camera depth estimation and outlier rejection by refocused events fusion,” *arXiv preprint arXiv:2207.10494*, 2022.

[37] S. H. Ahmed, H. W. Jang, S. N. Uddin, and Y. J. Jung, “Deep event stereo leveraged by event-to-image translation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 882–890.

[38] S. Tulyakov, F. Fleuret, M. Kiefel, P. Gehler, and M. Hirsch, “Learning an event sequence embedding for dense event-based deep stereo,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1527–1537.

[39] A. Hadviger, I. Marković, and I. Petrović, “Stereo dense depth tracking based on optical flow using frames and events,” *Advanced Robotics*, vol. 35, no. 3-4, pp. 141–152, 2021.

[40] D. Gehrig, M. Rüegg, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, “Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2822–2829, 2021.

[41] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, “Eklt: Asynchronous photometric feature tracking using events and frames,” *International Journal of Computer Vision*, vol. 128, no. 3, pp. 601–618, 2020.

[42] Z. Wang, Y. Ng, C. Scheerlinck, and R. Mahony, “An asynchronous kalman filter for hybrid event cameras,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 448–457.

[43] S. Tulyakov, D. Gehrig, S. Georgoulis, J. Erbach, M. Gehrig, Y. Li, and D. Scaramuzza, “Time lens: Event-based video frame interpolation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 155–16 164.

[44] Z. Wang, L. Pan, Y. Ng, Z. Zhuang, and R. Mahony, “Stereo hybrid event-frame (shef) cameras for 3d perception,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9758–9764.

[45] Y.-F. Zuo, L. Cui, X. Peng, Y. Xu, S. Gao, X. Wang, and L. Kneip, “Accurate depth estimation from a hybrid event-rgb stereo setup,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6833–6840.

[46] C. Scheerlinck, N. Barnes, and R. Mahony, “Continuous-time intensity estimation using event cameras,” in *Asian Conference on Computer Vision*. Springer, 2018, pp. 308–324.

[47] H. Kim and H. J. Kim, “Real-time rotational motion estimation with contrast maximization over globally aligned events,” *IEEE Robotics and Automation Letters*, 2021.

[48] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, “The multivehicle stereo event camera dataset: An event camera dataset for 3d perception,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2032–2039, 2018.

[49] S. Klenk, J. Chui, N. Demmel, and D. Cremers, “Tum-vie: The tum stereo visual-inertial event dataset,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8601–8608.

[50] L. Gao, Y. Liang, J. Yang, S. Wu, C. Wang, J. Chen, and L. Kneip, "Vector: A versatile event-centric benchmark for multi-sensor slam," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8217–8224, 2022.

[51] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, "Dsec: A stereo event camera dataset for driving scenarios," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4947–4954, 2021.

[52] T. Stoffregen and L. Kleeman, "Event cameras, contrast maximization and reward functions: an analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 300–12 308.

[53] H. Rebecq, D. Gehrig, and D. Scaramuzza, "ESIM: an open event camera simulator," in *Conference on Robotics Learning (CoRL)*, vol. 87. PMLR, 2018, pp. 969–982.

[54] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 142–149, 2017.

[55] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 573–580.

[56] X. Zhang, W. Liao, L. Yu, W. Yang, and G.-S. Xia, "Event-based synthetic aperture imaging with a hybrid network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 235–14 244.

[57] Y. Zou, Y. Zheng, T. Takatani, and Y. Fu, "Learning to reconstruct high speed and high dynamic range videos from events," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2024–2033.

[58] F. Paredes-Vallés and G. C. de Croon, “Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3446–3455.

[59] T. Ke and S. I. Roumeliotis, “An efficient algebraic solution to the perspective-three-point problem,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7225–7233.

[60] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, “Events-to-video: Bringing modern computer vision to event cameras,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3857–3866.

[61] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[62] T. Clephas, “rviz_satellite,” https://github.com/nobleo/rviz_satellite, accessed: 2021-01-19.

[63] J. L. Blanco-Claraco, “A tutorial on **SE**(3) transformation parameterizations and on-manifold optimization,” *arXiv preprint arXiv:2103.15980*, 2021.

국문초록

이벤트 카메라는 기존 카메라가 동작하기 어려운 환경에서 시각 데이터를 안정적으로 얻을 수 있다. 대표적으로 빛 밝기 범위가 넓거나 (High Dynamic Range: HDR) 빠르게 움직이는 환경에서 이벤트 카메라의 장점이 두드러진다. 그러나 이벤트 데이터는 기존의 컴퓨터 비전 알고리즘을 바로 적용할 수가 없다. 이벤트는 프레임 단위가 없으며 비동기적이기 때문에 새로운 접근 방법이 요구된다. 최근 몇 년 간, 동작 깊이 추정, 초고속 이미지 복원, 물체 추정 연구 등 다양한 활용을 보여주는 이벤트 연구가 활발하게 진행되었다. 본 논문에서는 이벤트 카메라를 활용하여 고속 환경에서 운용 가능한 각운동 추정 연구를 다루었다. 제안하는 방법은 대비 최대화 기법을 통해 각속도, 각위치를 추정하였고 실시간으로 동작하며 기존 대비 최대화 기법에서 다루지 않았던 드리프트 에러 누적 문제를 해결하여 뛰어난 성능을 보여주었다.

그러나 여전히 일반적인 사용환경에서는 이벤트 카메라가 기존 카메라를 대체하기에 어려움이 있다. 이벤트와 프레임 카메라의 장점을 모두 활용하기 위해, 본 논문에서는 헤테로 스테레오 카메라 시스템을 제안하였다. 헤테로 스테레오 카메라 시스템은 이벤트와 프레임 카메라를 동시에 활용한다. 제안하는 방법은 두 카메라를 활용하여 실시간으로 이벤트와 프레임 데이터를 매칭하여 준-조밀한(semi-dense) 깊이 영상을 계산하였다. 이 과정에서 이벤트 데이터를 정확하고, 효율적이며, 직관적으로 정렬하는 방법을 제안하였다. 최대 픽셀 이동 거리(maximum shift distance)를 제안하여 실시간 이벤트 정렬을 가능하게 하였으며, 정렬된 이벤트로 획득한 이미지는 프레임 카메라의 모서리 이미지와 매우 유사한 형태를 띄는 것을 보여주었다. 제안하는 깊이 추정 방법은 카메라 위치 및 자세를 추정할 수 있으며 매우 짧은 시간 안에 시스템 초기화 구동(initialization)이 가능하다. 추가적으로, 헤테로 스테레오 카메라에서 프레임 카메라 동작이 불가능한 경우 이벤트 카메라가 대체하여 동작할 수 있도록, 이벤트 카메라 기반 특징점 추적 방법과 자세 추정 연구를 진행하였다.

이벤트 카메라 연구에 기여하기 위해, 본 학위 논문의 코드를 모두 오픈 소스로 공개하

여 개인 프로젝트 페이지에 배포하였다. https://haram-kim.github.io

주요어: 이벤트 카메라, 자세 추정, 깊이 추정, 헤테로 스테레오 이벤트 프레임 카메라, 대비 최대화

학 번: 2019-30458