LCE-Calib: Automatic LiDAR-Frame/Event Camera Extrinsic Calibration With a Globally Optimal Solution

Jianhao Jiao, Feiyi Chen, Hexiang Wei, *Graduate Student Member, IEEE*, Jin Wu, and Ming Liu, *Senior Member, IEEE*

***Abstract*—The combination of light detection and rangings (LiDARs) and cameras enables a mobile robot to perceive environments with multimodal data, becoming a key factor in achieving robust perception. Traditional frame cameras are sensitive to changing illumination conditions, motivating us to introduce novel event cameras to make LiDAR-camera fusion more complete and robust. However, to jointly exploit these sensors, the challenging extrinsic calibration problem should be addressed. This article proposes an automatic checkerboard-based approach to calibrate extrinsics between a LiDAR and a frame/event camera, where the following four contributions are presented: 1) we present an automatic feature extraction and checkerboard tracking method from LiDAR's point clouds; 2) we reconstruct realistic frame images from event streams, applying traditional corner detectors to event cameras; 3) we propose an initialization-refinement procedure to estimate extrinsics using point-to-plane and point-to-line constraints in a coarse-to-fine manner; 4) we introduce a unified and globally optimal solution to address two optimization problems in calibration. Our approach has been validated with extensive experiments on 19 simulated and real-world datasets and outperforms the state-of-the-art.**

Manuscript received 14 October 2022; revised 14 December 2022 and 7 February 2023; accepted 14 March 2023. Recommended by Technical Editor X. Zhang and Senior Editor H. Gao. This work was supported in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2021B1515120032, in part by Foshan-HKUST under Grant FSUST20-SHCIRI06 C, and in part by Project of Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone under Grant HZQB-KCZYB-2020083, awarded to Prof. Ming Liu. *(Corresponding author: Ming Liu.)*

Jianhao Jiao is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China, and also with the CWB Institute of Autonomous Driving, Nanshan, Shenzhen 518000, China (e-mail: jjiao@connect.ust.hk).

Feiyi Chen, Hexiang Wei, and Jin Wu are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China (e-mail: fchenak@connect.ust.hk; hweiak@connect.ust.hk; jin_wu_uestc@hotmail.com).

Ming Liu is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China, and with the HKUST Shenzhen- Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen, China, and also with the Robotics and Autonomous Sysstems, The Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzho, Guangdong 511400, China (e-mail: eelium@ust.hk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMECH.2023.3259444.

Digital Object Identifier 10.1109/TMECH.2023.3259444

***Index Terms*—Machine and computer vision, unmanned autonomous systems.**

I. Introduction

A. Motivation

ONE of the challenges to realizing autonomous mobile robots should be the perception problem. As the front-end module of a robotic system, perception offers essential information to high-level navigation. Nowadays, the fusion of a light detection and ranging (LiDAR) and a camera, called lidar-(frame) camera fusion *LC-Fusion*, provides a promising solution for robust perception. LC-Fusion is able to overcome individual limitations of both LiDARs and cameras, producing confident results to boost various tasks such as simultaneous localization and mapping [1], [2] and pattern recognition [3]. Overall, LC-Fusion enjoys the following advantages.

1) *Accessibility:* Benefiting from the rapid development of sensory technologies, both cameras and LiDARs are portable and accessible for various commercial mobile robots, including drones [4], quadrupedal robots [5], and self-driving cars [6].
2) *Information sufficiency:* LC-Fusion provides multimodal information that is typically sufficient for robot navigation. Specifically, LiDARs directly offer sparse but accurate 3-D measurements of surrounding objects. In contrast, cameras capture dense and high-resolution 2-D images, which is beneficial to object recognition.
3) *Wide usage:* Research on LiDARs, cameras, and their fusion has attracted much attention from the research community. Algorithms targeting at different perception tasks have immediate solutions and are applicable to many robotic navigation tasks.

However, traditional frame cameras are commonly sensitive to changing illumination (e.g., darkness and glare). Cameras cannot capture scene information completely and may fail several vision-based algorithms. LC-Fusion may degenerate into a LiDAR-only configuration since most images are noisy. This issue motivates us to explore a novel type of sensor, event cameras [7], to complement the traditional LiDAR-camera fusion in challenging environments. The event camera augments the original sensor setup, and we can fuse three types of sensor input for perception. This new sensor fusion mode is called *LCE-Fusion*.

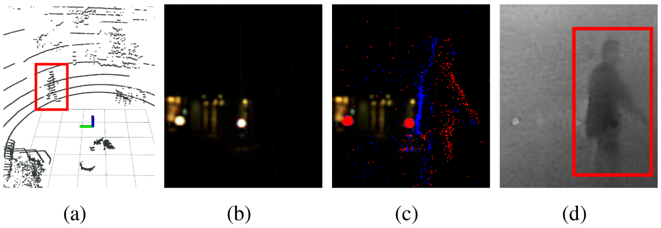


Fig. 1. Sensor measurements in a garden at night. (a) LiDAR point cloud, (b) a frame image under low exposure, (c) events that are printed on a frame image, and (d) reconstructed images using events. We can distinguish a human from the point cloud and event image except for the frame image. This indicates that the frame image is sensitive to the weak light. Please refer to the color version for the better visualization.

Event cameras are bioinspired sensors. Different from frame cameras that capture images at a fixed rate, event cameras asynchronously capture the per-pixel *intensity changes* and output a stream of *2-D events*. Each event is encoded with information, including the triggered time, pixel localization, and the sign of the intensity change. Event cameras have high temporal resolution (μs-level), high dynamic range (140 dB versus 60 dB of frame cameras), and low power consumption. They have great potential for several computer vision and robotic tasks, e.g., high-speed motion estimation [8] and high dynamic range perception [9]. Fig. 1 visualizes the enhancement brought by an event camera in a dark garden.

B. Challenges

The extrinsic calibration, estimating the relative rotational and translational offset from the reference frame to the target frame, is an indispensable step in using the LCE-Fusion. Our goal is to design a general and automatic extrinsic calibration approach for the LCE-Fusion. However, as emphasized in [10], challenges of automatic calibration arise from the the following three aspects: 1) feature extraction given noisy data; 2) data association across multimodal sensors; 3) parameter estimation.

A standard checkerboard is desirable in calibration, offering distinctive features (e.g., corners, boundaries, and a plane) and known geometry for feature matching. Although existing methods [11] have demonstrated the validity of this paradigm, several issues, including feature extraction, data association, event representation, and global optimization, have not been addressed well.

1) *Automatic feature extraction:* It is straightforward to detect the checkerboard from images by the off-the-shelf softwares such as OpenCV. However, this is not applicable to point clouds whose data model is fundamentally different. The detection of the checkerboard from point clouds is a nontrivial problem and is particularly hard if point clouds contain doors, tables, and walls with the planar shape. Several works [11] have to finish the feature extraction manually.
2) *Automatic feature matching:* An automatic approach to match checkerboard features between LiDARs and cameras are needed. However, this raises the issue that the symmetric shape of the checkerboard may lead to ambiguous data association, resulting in suboptimal or unreliable extrinsics.
3) *Event representation:* The asynchronous and sparse nature of event cameras makes the feature selection difficult. We need a method to convert a group of events into an image-type representation since traditional methods can be directly applied.
4) *Globally optimal solution:* The extrinsic optimization problem is generally nonconvex. This implies that the typical Gauss-Newton solution that linearizes the objective is approximate to the globally optimal solution. Thus, introducing a globally optimal solver in calibration given noisy sensor measurements is important.

C. Contributions

To tackle these challenges, we propose *LCE-Calib*, a unified extrinsic calibration method for the LCE sensor configuration. Fig. 2 shows the pipeline of LCE-Calib. In overall, LCE-Calib presents the following *contributions*.

1) We propose an automatic checkerboard extraction and tracking method that is robust to external noisy objects from point clouds (see Section IV-A). This method is also noise-aware, since we reduce the bias of LiDAR points by projecting points onto a reference plane and model the uncertainty of normal vectors.
2) We introduce the learning-based approach to reconstruct frame images from event streams for the downstream extrinsic calibration task. The resulting images allow the usage of traditional corner detectors (see Section IV-C).
3) We design the calibration process with an initialization-refinement philosophy to utilize point-to-plane and point-to-line constraints in a coarse-to-fine manner. This avoids the ambiguity issue in data association caused by the board's symmetric shape (see Section V).
4) We introduce a general solver to globally solve the following two optimization problems in calibration: 1) the perspective-n-point (PnP); 2) point-to-plane registration (PtPL) problems. The benefits from it is that the resulting extrinsics are accurate even if measurements are noisy.

The proposed method is evaluated extensively on different sensor devices in various calibration scenes from indoor offices to outdoor grounds. The proposed LCE-Calib outperforms the state-of-the-art (SOTA) calibration method, achieving an extrinsic accuracy of centimeters in translation and deci-degrees in rotation. To benefit the community, we publicly release the experimental data and code.[1]

II. Related Work

Sensor calibration is a fundamental problem in robotics. This section briefly reviews calibration results on the LiDAR-camera sensor suite and event cameras.

A. LiDAR-Camera Calibration

Related approaches are mainly divided into marker-based and marker-less approaches, depending on whether artificial targets are used or not during the process.

[1] https://github.com/HKUSTGZ-IADC/LCECalib

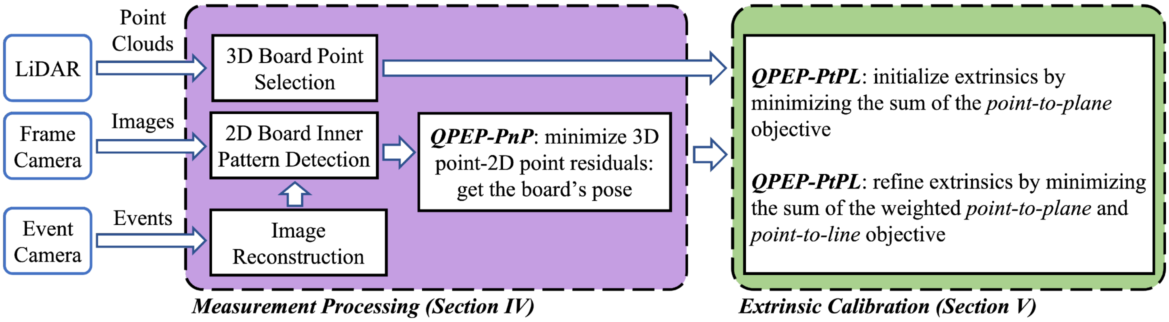

LiDAR
Point Clouds
Frame Camera
Images
Event Camera
Events
3D Board Point Selection
2D Board Inner Pattern Detection
Image Reconstruction
QPEP-PnP: minimize 3D point-2D point residuals: get the board's pose
QPEP-PtPL: initialize extrinsics by minimizing the sum of the *point-to-plane* objective
QPEP-PtPL: refine extrinsics by minimizing the sum of the weighted *point-to-plane* and *point-to-line* objective
Measurement Processing (Section IV)
Extrinsic Calibration (Section V)

Fig. 2. Block diagram illustrating the full pipeline of the proposed LCE-Calib method. The method first processes raw sensor measurements (see Section IV) to extract features and then initializes as well as refines sensors' extrinsics (see Section V).

1) Marker-Based Methods: Although marker-based calibration requires artificial targets to be observable in the scene and sometimes needs manual intervention, such methods are still prevalent due to the higher accuracy and robustness compared with nontarget-based calibration [12]. They benefit from the known geometry information of the target, imposing sufficient and reliable geometric constraints to estimate extrinsics.

Geometric solids [13], checkerboards [14], [15], and polygons [16] are the widely used targets. But among them, checkerboards are the most common targets since they also help the cameras' intrinsic calibration [17]. Zhang and Pless [18] first introduced the checkerboard into the extrinsic calibration of a perspective camera and a 2-D laser rangefinder. They placed the checkerboard at multiple poses to gather sufficient constraints induced by plane-line correspondences. Unnikrishnan and Hebert [19] extended this method to the extrinsic calibration with a 3-D LiDAR. They estimated the planar parameters of the board from the LiDAR and camera. They used plane–plane correspondences to initialize rotation and translation decoupled while minimizing the point-plane distance to refine the LiDAR-camera transformation. Our method also employs this way to initialize extrinsics. Pandey et al. [20] addressed the omnidirectional camera-LiDAR calibration problem, Mirzaei et al. [21] additionally investigated the LiDAR intrinsic model, and Koo et al. [22] analytically derived the uncertainty of plane parameters and explored its effect in calibration.

Besides plane information, the boundaries of objects and corner features also offer strong constraints. Sim et al. [23] determined lines from two noncoplanar surfaces of a V-shaped target and exploited linear constraints to estimate extrinsics. Moghadam et al. [24] extracted natural linear features in the scene, while Zhou et al. [15] jointly exploited 3-D line-to-line and point-to-plane correspondences to establish constraints. Finally, Huang and Grizzle [25] extracted corner features that are the intersections of two boundaries from an Apriltag [26] and then solved the PnP problem [27].

Our checkerboard-based approach presents several new features from following six aspects.

1) Features from images and point clouds are automatically extracted.
2) Board points are continuously tracked to improve the detection success rate and reduce processing time at the next frame.
3) Both planar and line features are fully utilized in different stages of calibration.
4) An event camera is also extrinsically calibrated with the LiDAR.
5) The uncertainty information is represented with the Lie group-based formulation [28].
6) A new globally optimal solver is introduced.

2) Marker-Less Methods: Marker-less calibration methods search for correspondences between geometric features found in natural environments, such as lines, edges, and planar regions. They do not rely on explicit shapes from known targets, having great flexibility for field robots.

Levinson and Thrun [29] put forward the first online calibration approach for a LiDAR-camera system by optimally aligning 3-D edge points with image contours. Different metrics based on the planar information [30], edges [31], and semantic constraints [32] were also proposed. Meanwhile, Shi et al. [33] proposed an end-to-end neural network to calibrate the extrinsics. However, the accuracy of markerless methods is commonly inferior to marker-based methods if calibration scenes are nonideal. In our approach, we show that the checkerboard benefits the intrinsic calibration of cameras. We also investigate the checkerboard-based calibration with an extension to event cameras.

B. Event Camera Calibration

The unique characteristics of event cameras raise the demand for novel solutions to address the calibration problem. Preliminary works on event cameras have designed blinking LED patterns [34], or screens [35] as the calibration target. The rapid illumination change triggers events and the patterns can be detected even if the cameras are static. Once features are extracted, traditional optimization-based calibration backends can be used. Nevertheless, these approaches may have large motion blur and noisy events [36]. Recent works are able to replace the custom-built calibration checkerboards with the standard ones by implementing one of the following two techniques: 1) several event cameras output synchronous frame images [7]; 2) high-quality frame images are reconstructed from event streams by utilizing the sensing characteristics of event cameras [36]. The latter approaches should be more applicable to more types of event cameras that only have an event output. They also benefit from recent advancements in image reconstruction from events [9].

Our method follows the second technique in event-based calibration. We further propose a unified method to calibrate extrinsics between a LiDAR and an event camera based on these works. The method is shown with high accuracy under detailed evaluation.

III. Problem Statement

The extrinsic calibration problem is divided into the following three subproblems: 1) PnP; 2) PtPL; 3) point-to-line (or edge) (PtL) registration problems. All of them are generally formulated as quadratic pose estimation problems (QPEPs) [37]. Before delving into the details of LCE-Calib, we first introduce basic concepts here. Section III-A presents notations. Section III-B introduces the definition of QPEP and general idea to solve QPEPs based on our previous work.

A. Notations and Definitions

We consider a sensor suite that consists of a LiDAR, a frame camera, and an event camera. Frames of these sensors are defined as $()^l$, $()^c$, and $()^e$ respectively. We also define $()^b$ as the frame of the checkerboard, where the origin stays at the board's center, and the z-axis is perpendicular to the board's plane. We use $\boldsymbol{t} \in \mathbb{R}^3$ and $\boldsymbol{R} \in SO(3)$ to represent the 3-D translation and rotation. Especially, the rotation matrix is from the Lie group $SO(3)$ where $\boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}$, $\det \boldsymbol{R} = 1$. For any real 3-D vector $\boldsymbol{\phi} \in \mathbb{R}^3$, its skew-symmetric matrix is

$$\boldsymbol{\phi}^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in \mathfrak{so}(3) \tag{1}$$

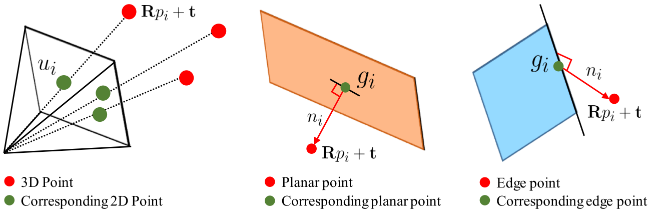
which is an element from the Lie algebra $\mathfrak{so}(3)$. We use the exponential operator to associate an element from $SO(3)$ with an element from $\mathfrak{so}(3)$: $\boldsymbol{R} = \exp(\boldsymbol{\phi}^\wedge)$. From the derivation in [28], we can model the Gaussian uncertainty of rotation using the right perturbation as $\boldsymbol{R} = \bar{\boldsymbol{R}} \exp(\delta\boldsymbol{\phi}^\wedge) \approx \bar{\boldsymbol{R}}(\boldsymbol{I} + \delta\boldsymbol{\phi}^\wedge)$, where $\bar{\boldsymbol{R}}$ is the noise-free rotation and $\delta\boldsymbol{\phi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\delta\boldsymbol{\phi}})$. If $\boldsymbol{R}$ and $\boldsymbol{t}$ are considered simultaneously, we also use the 3-D transformation matrix $\boldsymbol{T} \in SE(3)$ from the Lie group: $\boldsymbol{T} = SE_3(\boldsymbol{R}, \boldsymbol{t}) = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$ to represent poses.

B. Quadratic Pose Estimation Problem

As first proposed in our previous work [37], QPEPs define a series of $SE(3)$-differentiable optimization problems as

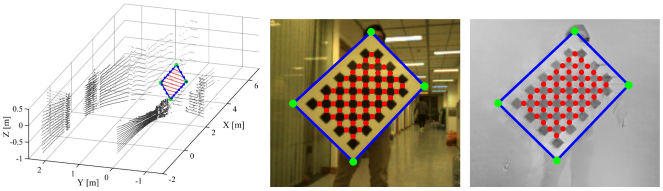
$$\underset{\boldsymbol{R} \in SO(3),\ \boldsymbol{t} \in \mathbb{R}^3}{\arg\min} \quad \mathcal{L}(\boldsymbol{R}, \boldsymbol{t}) \tag{2}$$

where $\mathcal{L}(\cdot)$ is represented as a *quadratic objective* in terms of quadratic products of elements in $\boldsymbol{R}$ and $\boldsymbol{t}$. The PnP, PtPL, and PtL problems are three QPEPs since their objective are quadratic. Fig. 3 visualizes the geometry of these problems. Regarding the PnP problem, our goal is to estimate the transformation between the world frame and the camera frame given a set of 3-D–2-D point correspondences $< \boldsymbol{p}_i, \boldsymbol{u}_i >$



u_i
$\mathbf{R}p_i + \mathbf{t}$
● 3D Point
● Corresponding 2D Point
g_i
n_i
$\mathbf{R}p_i + \mathbf{t}$
● Planar point
● Corresponding planar point
g_i
n_i
$\mathbf{R}p_i + \mathbf{t}$
● Edge point
● Corresponding edge point

Fig. 3. Visualization of geometric constraints of the PnP, PtPL, and PtL problems. The red dot is the reference point, while the green dot is the corresponding corner or edge feature.



(a) (b) (c)

Fig. 4. Features of the checkerboard are extracted from (a) the point cloud, (b) frame image, and (c) reconstructed image from events.

$$\mathcal{L}_{\text{pnp}}(\boldsymbol{R}, \boldsymbol{t}) = \sum ||\boldsymbol{u}_i - \pi(\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t})||^2 \tag{3}$$

where $\pi(\cdot)$ projects a 3-D point onto the image plane. The PtPL problem obtains the optimal transformation between two 3-D point clouds by minimizing the point-to-plane residuals

$$\mathcal{L}_{ptpl}(\boldsymbol{R}, \boldsymbol{t}) = \sum [\boldsymbol{n}_i^\top (\boldsymbol{g}_i - \boldsymbol{R}\boldsymbol{p}_i - \boldsymbol{t})]^2 \tag{4}$$

where $\boldsymbol{g}_i$ and $\boldsymbol{p}_i$ are ith corresponding points from two point clouds and $\boldsymbol{n}^i$ is the ith normal vector of a plane on which $\boldsymbol{g}_i$ stay. For the PtL problem [38], we define the edge residual as a planar residual using (4), where $\boldsymbol{n}_i$ coincides with the projection direction from the line to $\boldsymbol{p}_i$ (see Fig. 3). This formulation allows us to address both the PtPL and PtL problems with one solution. Our previous work has proposed a unified quaternion-based globally optimal solution to solve general QPEPs [37]. The mathematical derivation, solution strategies, and analysis of uncertainty are already detailed in this article [37]. In subsequent sections, we introduce how to extract features from raw sensor measurements and apply the *QPEP-PnP* and *QPEP-PtPL* algorithms to address the extrinsic calibration.

IV. Measurement Processing

This section explains how raw sensor data from LiDARs, frame cameras, and event cameras are preprocessed before estimating their extrinsics. We can take advantage of the structural prior of the checkerboard for reliable feature selection and matching. Fig. 4 shows the feature extraction results.

A. Automatic Feature Extraction From LiDARs

We are interested in extracting the following three types of features from the checkerboard point cloud: 1) edge points; 2) planar points; 3) planar coefficients. These features are useful in

LiDAR-camera data association. Since we work on mechanical scanning LiDARs, the following sections use *ring* to denote the points set from the same emitter. We propose a two-stage method for automatic feature extraction.

1) Checkerboard Point Selection: We need to select board points from the raw point cloud. We have the following observations to design the point selection method: 1) the width and height of the board are known as a prior; 2) points from the same ring form a straight line segment if they lie on the board; 3) points from different rings form a planar patch if they lie on the board and stay near.

Firstly, we compute the pitch angle $\phi = \arctan\left(\frac{z}{\sqrt{x^2+y^2}}\right)$ for each point and split out points into different rings. Some LiDARs directly provide the "ring" as a property of each point. We then cluster each ring into multiple line segments based on the distance of two consecutive points. The principle component of each line segment is computed by principal components analysis (PCA): $[\sigma_1, \sigma_2, \sigma_3]$ where $\sigma_1 \geq \sigma_2 \geq \sigma_3$. The curvature is measured as σ_1/σ_2. We keep line segments if their curvature is bigger than the predefined threshold μ_1 and the length of the line segment smaller than an empirical threshold μ_2. Secondly, we utilize the DBScan algorithm [39] to segment the remaining points as several clusters. Small clusters are then removed. Thirdly, we verify each cluster by registering it with a board point template. We compute the registration error for each cluster and keep the only one with the lowest error. The selected cluster is regarded as the candidate board points.

2) Checkerboard Feature Extraction: After obtaining board points, we use the random sample consensus (RANSAC)-based plane fitting method [40] to compute the normal vector $\boldsymbol{n}^l$, filter out outliers, and obtain a set of planar points $\mathcal{P}^l_{pl}$ (inliers). Due to uncorrected bias and noise of LiDARs' measurements, we observe that board points form a $3-cm$ thick plane approximately. To reduce this effect in the subsequent calibration process, we project each planar point $\boldsymbol{p}$ onto the fitted plane as

$$\boldsymbol{p}^l_{\text{proj}} = \boldsymbol{p}^l - a\boldsymbol{n}^l, \quad a = \boldsymbol{n}^l \cdot (\boldsymbol{p}^l - \boldsymbol{g}^l) \tag{5}$$

where $\boldsymbol{g}^l$ is a point on the plane. The edge points $\mathcal{P}^l_{\text{edge}}$ are then extracted as the starting and ending points of each ring.

3) Checkerboard Tracking: Estimating the board's position allows us to track the board at the next frame. The time-consuming point selection stage (costs 50–60 ms) is skipped. We create a virtual bounding box around the checkerboard with a slightly larger size. In the next frame, only points inside the box are kept. This tracking stage also improves the success rate of the board point selection.

B. Automatic Feature Extraction From Frame Cameras

The coefficients of the board observed by the frame camera, denoted by $[\boldsymbol{n}^c, d^c]^\top$, can be easily obtained. We use the off-the-shelf software to detect the inner patterns of the checkerboard automatically. The board's pose $SE_3(\boldsymbol{R}^c_b, \boldsymbol{t}^c_b)$ is estimated by the *QPEP-PnP* algorithm, where we minimize the point-wise distance between 3-D inner patterns and their corresponding 2-D points. Along the transformed boundaries of the board, we can generate many "fake" edge points $\mathcal{P}^c_{\text{edge}}$. We can also obtain the covariance matrix $\boldsymbol{\Sigma_{\delta\phi}}$ of the checkerboard's rotation from the *QPEP-PnP* algorithm. Based on the uncertainty representation of 6-DoF poses in Section III-A, we can propagate the uncertainty of the rotated normal vector $\boldsymbol{n}^c \approx \bar{\boldsymbol{R}}^c_b(\boldsymbol{I} + \delta\phi^\wedge)\boldsymbol{n}^b$ as

$$\boldsymbol{\Sigma}_{\boldsymbol{n}^c} \approx \left[\bar{\boldsymbol{R}}^c_b(\boldsymbol{n}^b)^\wedge\right] \boldsymbol{\Sigma_{\delta\phi}} \left[\bar{\boldsymbol{R}}^c_b(\boldsymbol{n}^b)^\wedge\right]^\top \tag{6}$$

where the derived covariance will be used in the optimization objective in Section V.

Algorithm 1: LiDAR-Camera Extinsic Calibration.

```
Input: Number of iteration: $I$
       Number of LiDAR-camera data pairs: $N$
       Features: $\{\boldsymbol{n}^c\}$, $\{\mathcal{P}^c_{edge}\}$, $\{\mathcal{P}^l_{pl}\}$, $\{\mathcal{P}^l_{edge}\}$
Output: Estimated extrinsics $\boldsymbol{T}^c_l$;
Initialize extrinsics $\boldsymbol{T}^c_{l,ini} = \mathit{QPEP\text{-}PtPL}(\{\boldsymbol{n}^c\}, \{\mathcal{P}^l_{pl}\})$
foreach $\underline{M \in \{1, 2, \ldots, N\}}$ do
    The set of all candidate extrinsics: $\mathcal{T} = \emptyset$
    while count of iteration $< I$ do
        Randomly select $M$ data pairs: $\{\boldsymbol{n}^c\}^\#$, $\{\mathcal{P}^c_{edge}\}^\#$, $\{\mathcal{P}^l_{pl}\}^\#$, $\{\mathcal{P}^l_{edge}\}^\#$
        Find corresponding edge points of $\{\mathcal{P}^l_{edge}\}^\#$: $\{\mathcal{E}^c\} \subseteq \{\mathcal{P}^c_{edge}\}^\#$ given $\boldsymbol{T}^c_{l,ini}$
        Optimize extrinsics: $\boldsymbol{T} = \mathit{QPEP\text{-}PtPL}(\{\boldsymbol{n}^c\}^\#, \{\mathcal{P}^l_{pl}\}^\#, \{\mathcal{E}^c\}, \{\mathcal{P}^l_{edge}\}^\#)$
        $\mathcal{T} = \mathcal{T} \cup \boldsymbol{T}$
    Compute the mean of $\mathcal{T}$: $\bar{\boldsymbol{T}}$
    $\boldsymbol{T}^c_l = \bar{\boldsymbol{T}}$ if the mean geometric error is smaller
```

C. Image Reconstruction From Events

1) Event Data: An event camera has independent pixels that respond to logarithmic intensity change L. In a noise-free scenario, an event $\boldsymbol{e}_k = [\boldsymbol{u}_k^\top, t_k, p_k]^\top$ is triggered at pixel $\boldsymbol{u}_k = [x_k, y_k]^\top$ and time t_k as soon as the logarithmic intensity increment reaches a contrast threshold $\pm C$ since the last event at the pixel, i.e.,

$$\Delta L(\boldsymbol{u}_k, t_k) \doteq L(\boldsymbol{u}_k, t_k) - L(\boldsymbol{u}_k, t_k - \Delta t_k) \geqslant p_k C \tag{7}$$

where $C > 0$, Δt_k is the time elapsed since the last event at the same pixel, and the polarity $p_k \in \{+1, -1\}$ is the sign of the intensity change [7].

2) Image Reconstruction: One of the fundamental building blocks for camera calibration is the detection of checkerboard corners. However, these corner detectors originally designed for frame images are not directly applicable to events due to their intrinsically asynchronous and sparse nature. Inspired by the work done by Muglikar et al. [36], we resort to a learning-based method called E2VID [9] to reconstruct high-quality frame images from the asynchronous and sparse event stream. This method encodes events in a spatio-temporal voxel grid and uses a recurrent convolutional neural network based on the UNet architecture [41] to process events. The network is already pretrained using a large number of simulated event sequences.

After the reconstruction, we can directly apply the approach in Section IV-B on these images for calibration.

The reconstruction procedure is summarized in the following three steps.

1) Move the checkerboard before the event camera to trigger events.
2) Divide events into chunks of constant time duration (50 ms in our experiments). But the time duration of these chunks does not have to be constant. One could choose to define the chunks by the number of events.
3) Reconstruct images from events using E2VID. The corresponding LiDAR frame specifies the reconstruction timestamp.

Besides the above approach, we also have two possible solutions to detect corners from events, but they present several limitations. The first solution is to aggregate events within a local spatial-temporal window to create an event map that is also an image-type representation [42]. But event maps do not record intensity and contain several noisy pixels, making the traditional corner detectors inaccurate. Another approach is to detect corners from pure events. However, existing event-based corner detectors [43] are not specifically designed for checkerboard corners, and most of them are not publicly released, inducing difficulties in calibration. In contrast, the proposed reconstruction-based calibration scheme is easy to implement and does not require much parameter tuning. Experimental results have demonstrated the effectiveness of the proposed reconstruction-based calibration method.

V. Extrinsic Calibration With Multiframe Measurements

Section IV has explained how checkerboard's normal vectors: $\boldsymbol{n}^c, \boldsymbol{n}^e$, planar points: $\mathcal{P}^l_{\text{pl}}$, and edge points: $\mathcal{P}^l_{\text{edge}}, \mathcal{P}^c_{\text{edge}}, \mathcal{P}^e_{\text{edge}}$ are extracted from each frame of LiDARs, frame cameras, and event cameras. This section introduces how to use these features to constrain extrinsics. We take the LiDAR-frame camera extrinsic calibration as an example without loss of generality. At first, we can easily construct a set of point-plane pairs. But the point-edge association is ambiguous without the initial extrinsics. In other words, we cannot determine which boundaries a point belongs to since the board's shape is symmetric. Thus, we follow the *initialization-refinement* philosophy [38] to estimate extrinsics in a coarse–fine manner, as summarized in Algorithm 1.

The first phase initializes the extrinsics $\boldsymbol{T}^c_{l,\text{ini}}$ (i.e., transformation from $()^c$ to $()^l$). We formulate the point-to-plane optimization objective (4) by utilizing planar features: $\boldsymbol{n}^c$ and $\mathcal{P}^l_{\text{pl}}$ from several frames of measurements. The *QPEP-PtPL* algorithm is applied to solve the problem.

The second phase further refines the initial solution by jointly utilizing the planar and edge points that offer stronger geometric constraints. This phase is executed with multiple computations. Specifically, the following three steps are done at line 6 and 7.

1) With $\boldsymbol{T}^c_{l,\text{ini}}$, we first transform $\mathcal{P}^l_{\text{edge}}$ into the camera frame. For each point from transformed point set, we find the nearest point in $\mathcal{P}^c_{\text{edge}}$ as its corresponding edge point, and then construct the point-to-line objective.

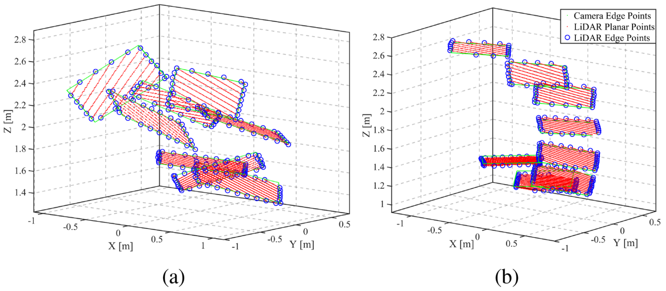


Fig. 5. LiDAR's planar points and edge points are aligned with the board plane of images with the estimated extrinsics.

2) From (4), we propagate the covariance of each point-to-plane distance as $\sigma^2 = (\boldsymbol{R}^c_l \boldsymbol{p}^l_i + \boldsymbol{t}^c_l)^\top \boldsymbol{\Sigma}_{n^c} (\boldsymbol{R}^c_l \boldsymbol{p}^l_i + \boldsymbol{t}^c_l)$. We filter out 10% error terms if their variances are large.
3) The overall optimization objective for the refinement is defined as the sum of all weighted point-to-plane and point-to-line residuals as

$$\mathcal{L}(\boldsymbol{R}^c_l, \boldsymbol{t}^c_l) = w_{\text{ptpl}} \mathcal{L}_{\text{ptpl}}(\boldsymbol{R}^c_l, \boldsymbol{t}^c_l) + w_{\text{ptl}} \mathcal{L}_{\text{ptl}}(\boldsymbol{R}^c_l, \boldsymbol{t}^c_l) \quad (8)$$

where w_{ptpl} and w_{ptl} are tunned parameters which balance these two constraints. Similar to initialization, we can also use the *QPEP-PtPL* solver.

After getting a set of candidate extrinsics, we select one with the minimum geometric error as our resulting extrinsics. Two examples are shown in Fig. 5, where LiDAR's planar points and edge points are aligned with the board plane of images with the estimated extrinsiscs.

VI. Experiment

A. Implementation Details

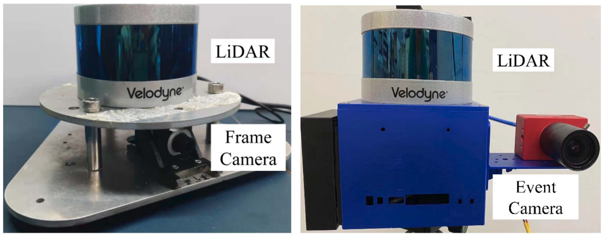
We implement the proposed LCE-Calib in MATLAB. We empirically set $\mu_1 = 0.9$ and $\mu_2 = 1$ in the checkboard point selection, and $I = 100$ in Algorithm 1. Hyperparameters of the objective (8): w_{ptpl} and w_{ptl}, should be set according to a specific sensor configuration, affected by the LiDAR's density and camera's resolution.

1) Sensor Suites: The proposed method is tested on both the simulated and real-world sensor suite. The LiDAR and camera on real platforms are hardware-synchronized. Hyperparameters: The number of checkerboard's inner patterns and board size should be set for a specific calibration target.

1) *The Simulated Sensor Suite* is built on the Gazebo [44]. It consists of a simulated 16-beam LiDAR and frame camera with the same specification to real-world LiDAR-frame camera sensor (RLFS). The ground-truth intrinsics and extrinsics are provided.
2) *The* RLFScontains a VLP-16 LiDAR[2] and a high-resolution camera (GSML-AR0143-H090, resolution: 1280H ×720 V)[3], as shown in Fig. 6(a).

[2] https://velodynelidar.com/products/puck

[3] https://docs.miivii.com/product/apex/xavier/manual/en/common/05.EN_gmsl.html



(a) (b)

Fig. 6. RLFS and RLES devices for calibration tests. (a) LiDAR-frame camera sensor. (b) LiDAR-event camera sensor.

3) *The real-world LiDAR-event camera sensor* (RLES) is made for the mapping application and shown in Fig. 6(b). It is installed with a VLP-16 LiDAR and an event camera (DAVIS346, resolution: 360H ×240 V)[4].

Both frame and event cameras use the pinhole projection model. And the DAVIS346 event camera outputs red, green, blue (RGB) frames that are used to verify the frame camera-based calibration.

2) Evaluation Metrics: We introduce the following two metrics to assess the LiDAR-camera calibration results from different aspects.

1) *Error compared with ground truth (EGT)* computes the distance between the ground truth and the estimated values in terms of rotation and translation as

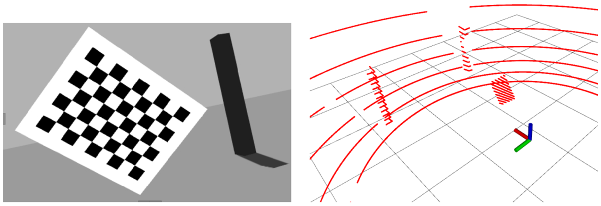
$$\begin{aligned} EGT_{\boldsymbol{R}} &= ||\ln(\boldsymbol{R}_{\text{gt}}\boldsymbol{R}_{\text{est}}^{-1})^{\vee}|| \\ EGT_{\boldsymbol{t}} &= ||\boldsymbol{t}_{\text{gt}} - \boldsymbol{t}_{\text{est}}||. \end{aligned} \tag{9}$$

2) *Mean geometric error (MGE)* is the sum of the mean planar error (MPE) and mean edge error (MEE). MPE and MEE compute the mean point-to-plane and point-to-line distance between the 3-D checkerboard plane in the camera frame and checkerboard points in the LiDAR frame, respectively.

The perfect ground truth is unknown in real-world experiments. We first estimate 3-D corner points from both images and point clouds. We then manually check and modify them. With 3-D–3-D point correspondences that are found, we can compute the "fake" ground truth with the iterative closest point (ICP) algorithm. Our proposed method is compared with a baseline method (denoted by *Zhou-MATLAB*) that has been released in MATLAB[5]. This method implements the SOTA LiDAR-camera calibration method [15] but formulates a different objective function.

B. Calibration Results on Simulated Data

We first verify the LiDAR-frame camera calibration with synthetic data. The sampled image and point cloud are shown in Fig. 7. The LiDAR and camera are given sufficient rotational offset: $[-70, -70, 150]^{\circ}$ at roll, pitch, and yaw respectively, as well as translational offset: $[0.4185, -0.3050, -0.1476]$ m at $x-$, $y-$, and $z-$ axes respectively.



(a) (b)

Fig. 7. Simulated sensor data. (a) Image. (b) Point cloud.

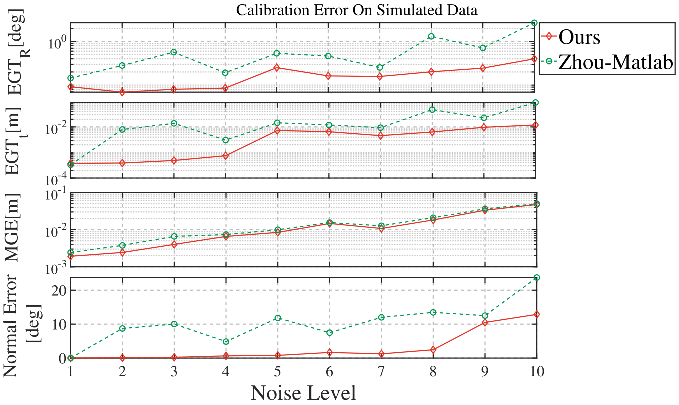


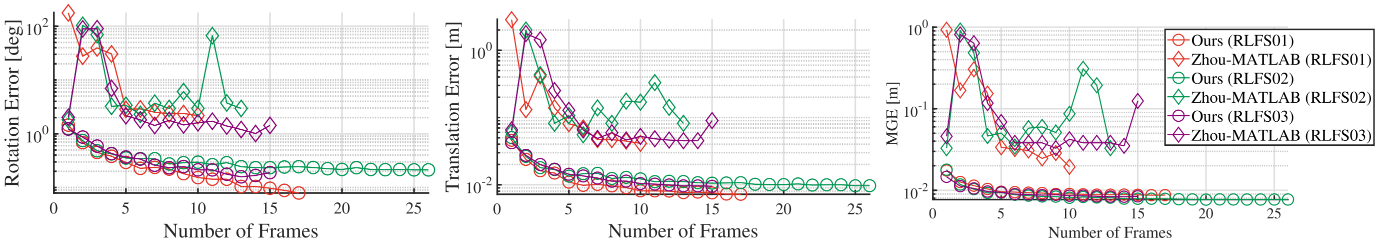
Fig. 8. Calibration error of the proposed method and *Zhou-MATLAB* on calibration data at different levels of noise. The proposed method can achieve calibration accuracy (the top three rows) ranging from $[0.068, 0.400]^{\circ}$, $[0.0004, 0.012]$ m, $[0.0003, 0.024]$ m, $[0.002, 0.023]$ m at $EGT_{\boldsymbol{R}}$, $EGT_{\boldsymbol{t}}$, and MGE respectively. Our method achieves the plane fitting accuracy (the fourth row) that is computed as the relative angle with the ground-truth normal ranging from $[0.0104, 12.8701]^{\circ}$.

We place the checkerboard in front of the sensor with sufficient rotation and translation and obtain 30 pairs of noise-free LiDAR-camera data. To test the robustness and limitations of our method, we further generate another nine sets of data. First, we add zero-mean Gaussian noise on depth measurements with an increasing standard deviation of $\sigma \in \{0.8, 1.6, 2.4, 3.2, 4.0, 4.8, 6.0, 8.0, 10.0\}$ cm to the noise-free LiDAR data. Second, we additionally process points which are next to the checkerboard's boundaries by adding another Gaussian noise. This is because we want to mimic real-world data since the practical emmitted laser has a divergence angle rather than a strict line [31].

In Fig. 8, we plot the calibration error. Benefiting from the proposed feature extraction and global optimization, our method overall has more accurate results than *Zhou-MATLAB* even under large noise. A key step in feature extraction is estimating the normal vector of the checkerboard plane from LiDARs. Both noisy/outlier points and improper plane fitting may negatively affect the estimates. This figure also shows the mean angular error between and ground-truth normals and estimated normals. *Zhou-MATLAB* cannot estimate accurate planar coefficients. Regarding the optimization part, our method has rich planar and edge constraints to enforce the accuracy. Also, the globally optimal solution in our method is essential to these results. Moreover, we find that the unsupervised metric: *MGE* is also suitable to evaluate calibration results, since its curve is consistent with the curves of *EGT*.

[4] https://inivation.com/wp-content/uploads/2019/08/DAVIS346.pdf

[5] https://www.mathworks.com/help/lidar/ug/lidar-and-camera-calibration.html



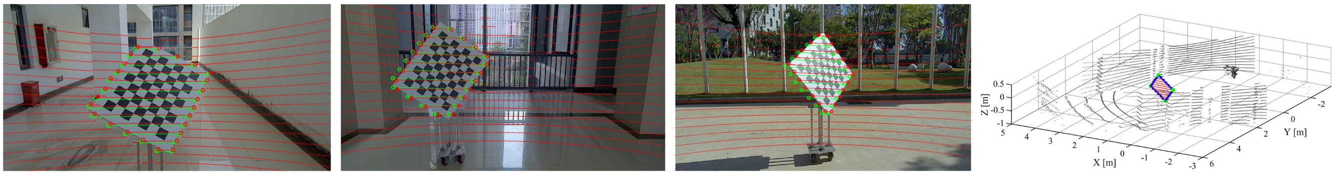
(a) (b) (c)

Fig. 9. Median rotation error, translation error, and MGE of ours and *Zhou-MATLAB* on calibration data: *RLFS01-RLFS03*. The median errors are computed using the set of candidate extrinsics at line 8 of Algorithm 1. (a) Median rotation error. (b) Median translation error. (c) Median MGE.

TABLE I
CALIBRATION RESULTS ON *RLFS*

Dataset	Method	Quaternion				Translation $[m]$			$EGT_{\mathbf{R}}$ $[deg, \downarrow]$	$EGT_{\mathbf{t}}$ $[m, \downarrow]$	MGE $[m, \downarrow]$	N_{detect}/N $[\uparrow]$
		w	x	y	z	x	y	z				
	CAD	0.664	0.664	−0.242	0.242	0.000	−0.059	−0.070	–	–	–	–
	Fake GT	0.666	0.667	−0.233	0.238	−0.008	−0.069	−0.084	–	–	–	–
RLFS01	*Ours*	0.667	0.666	−0.233	0.239	−0.003	−0.074	−0.091	**0.130**	**0.009**	**0.009**	**17**/18
	Zhou-MATLAB	0.669	0.662	−0.238	0.240	0.016	−0.071	−0.093	0.850	0.026	0.011	10/18
RLFS02	*Ours*	0.665	0.668	−0.234	0.238	−0.000	−0.060	−0.090	**0.293**	**0.014**	**0.007**	**26**/29
	Zhou-MATLAB	0.670	0.663	−0.227	0.244	−0.008	−0.110	−0.095	1.223	0.042	0.013	21/29
RLFS03	*Ours*	0.667	0.667	−0.233	0.237	−0.010	−0.068	−0.092	**0.136**	**0.008**	**0.008**	**15**/15
	Zhou-MATLAB	0.669	0.661	−0.238	0.244	0.029	−0.077	−0.100	1.120	0.041	0.013	13/15

↓/↑ indicates that the lower/higher the value, the better the score.



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

Fig. 10. Back-projection of the LiDAR points on images using the extrinsics from Table I from our algorithm (red) and *Zhou-MATLAB* (green). (a) RLFS01. (b) RLFS02. (c) RLFS03. (d) RLFS02 (cloud).

C. Calibration Results on Real-World Data

This section tests the proposed method with two real-world sensor suites. Similarly, we move a checkerboard before sensors slowly to collect several groups of calibration data. Experimental results are shown in the following sections.

1) Extrinsic Calibration of RLFS: We first verify the calibration method with the LiDAR-frame camera setup. We collect three groups of data called *RLFS01*, *RLFS02*, and *RLFS03* to test our approach. *RLFS01* and *RLFS02* were collected in an indoor corridor. Besides the checkerboard, other objects such as grounds, walls, and square pillars appear as planar surfaces. *RLF03* were collected in an outdoor garden. Several noisy objects such as buildings, poles, and lawns exist, affecting the checkerboard detection from point clouds.

We firstly employ the MATLAB camera calibration toolbox[6] to calibrate the camera's intrinsics with these collected data and then calibrate extrinsics. Our method successfully recovers the extrinsics. Fig. 9 plots errors with increasing frames of measurements and Table I gives quantitative sample results. N_{detect}/N shows the portion of successfully detecting the checkerboard from images or point clouds. *Zhou-MATLAB* only uses a portion of measurements since it fails to detect checkerboards from several LiDAR frames and correctly detect features, thus obtaining inaccurate extrinsics. These results are consistent with those in Section VI-B, where our method outperforms the baseline and reaches up to $<0.3°$ and <1.5 cm calibration error as well as <0.9 cm *MGE* in both indoor and outdoor environments.

Fig. 10 shows the back-projection results using extrinsics in Table I. Circular points indicate detected edge points. Our method can provide better results since the projections of edge points nearly stay on the the checkerboard's boundary. We additionally show the calibration error along with the number of frames used in calibration. The error gradually decreases since more constraints help to enforce the accuracy. One example on *RLFS01* is shown in Fig. 12.

2) Extrinsic Calibration of RLES: We further verify the complete LCE-Calib method with RLES. Three groups of calibration data are collected in an indoor room under different lighting

[6] https://www.mathworks.com/help/vision/camera-calibration.html

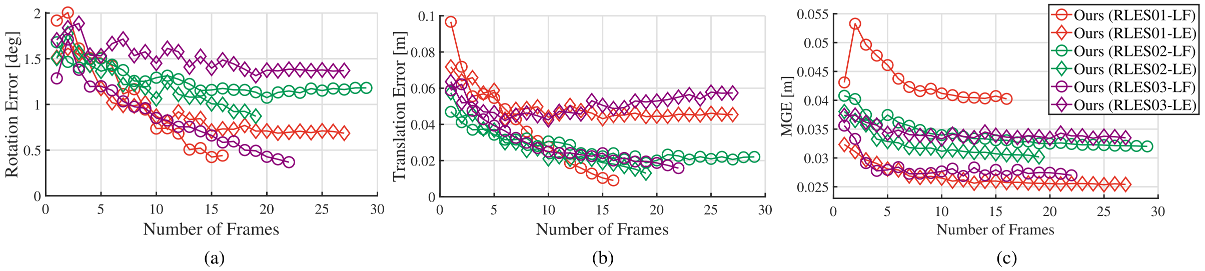


Fig. 11. Median rotation error, translation error, and MGE at multiple runs of our method on calibration data: *RLES01-LF*, *RLES01-LE*, *RLES02-LF*, *RLES02-LE*, *RLES03-LF*, and *RLES03-LE*. The median errors are computed using the set of candidate extrinsics at line 8 of Algorithm 1. (a) Median rotation error. (b) Median translation error. (c) Median MGE.

TABLE II
CALIBRATION RESULTS ON *RLES*. ↓/↑ INDICATES THAT THE LOWER/HIGHER THE VALUE, THE BETTER THE SCORE

Dataset	Method	Quaternion				Translation $[m]$			$EGT_{\mathbf{R}}$ $[deg, \downarrow]$	$EGT_{\mathbf{t}}$ $[m, \downarrow]$	MGE $[m, \downarrow]$	N_{detect}/N $[\uparrow]$
		w	x	y	z	x	y	z				
	CAD	0.500	0.500	−0.500	0.500	0.126	−0.065	−0.080	–	–	–	–
	Fake GT	0.502	0.504	−0.498	0.495	0.112	−0.094	−0.108	–	–	–	–
RLES01-LF	*Ours*	0.499	0.505	−0.498	0.498	0.126	−0.095	−0.106	**0.629**	**0.032**	0.037	**20**/20
	Zhou-MATLAB	0.502	0.498	−0.502	0.498	0.132	−0.094	−0.104	1.151	0.038	**0.036**	**20**/20
RLES01-LE	*Ours*	0.497	0.501	−0.501	0.500	0.150	−0.092	−0.096	**0.839**	**0.041**	0.024	**33**/36
	Zhou-MATLAB	0.496	0.500	−0.500	0.503	0.161	−0.097	−0.100	1.258	0.050	**0.023**	**33**/36
	Fake GT	0.500	0.505	−0.498	0.497	0.129	−0.080	−0.109	–	–	–	
RLES02-LF	*Ours*	0.497	0.506	−0.502	0.495	0.131	−0.064	−0.102	**0.677**	**0.018**	**0.031**	**34**/36
	Zhou-MATLAB	0.495	0.504	−0.504	0.497	0.160	−0.060	−0.090	0.927	0.041	0.037	32/36
RLES02-LE	*Ours*	0.498	0.507	−0.499	0.496	0.130	−0.071	−0.107	**0.331**	**0.009**	**0.028**	**34**/36
	Zhou-MATLAB	0.491	0.505	−0.505	0.498	0.152	−0.049	−0.096	1.270	0.041	0.034	31/36
RLES03-LF	*Ours*	0.497	0.511	−0.496	0.496	0.116	−0.055	−0.110	**0.781**	**0.028**	0.027	**29**/36
	Zhou-MATLAB	0.508	0.504	−0.494	0.495	0.100	−0.104	−0.114	1.034	0.039	**0.024**	26/36
RLES03-LE	*Ours*	0.504	0.503	−0.498	0.495	0.119	−0.090	−0.108	0.554	**0.014**	0.030	**29**/36
	Zhou-MATLAB	0.497	0.507	−0.499	0.497	0.124	−0.066	−0.101	**0.401**	0.016	**0.028**	27/36

The entities with better scores are marked as the bold entities.

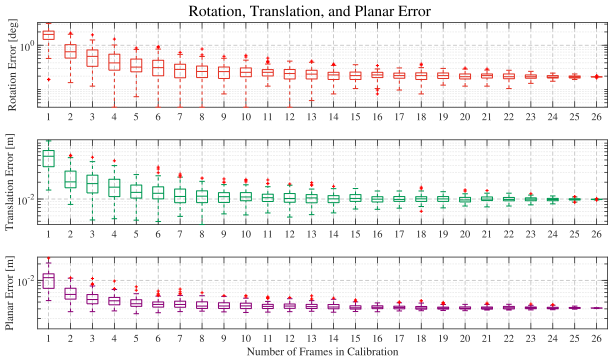
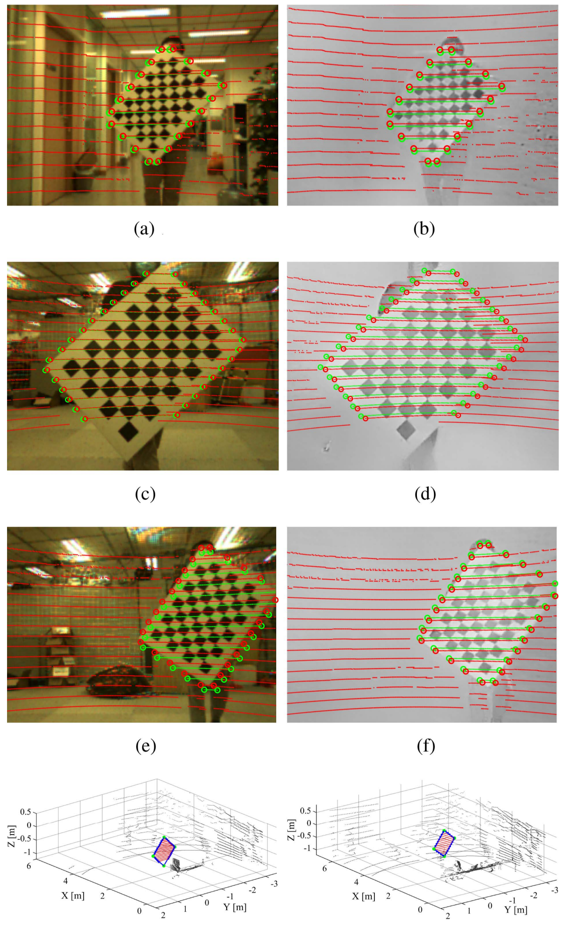


Fig. 12. Calibration error along with frame number on *RLFS01*.

conditions, which are called *RLES01*, *RLES02*, and *RLES03*. In contrast to the last section, we additionally compare calibration results of using frame images (*X-LF*) and reconstructed images (*X-LE*) from event streams. Fig. 11 plots calibration errors of the proposed method and Table II reports quantitative results. The calibration errors consistently decrease with more frames. Though, compared with Table I, the calibration errors are larger. We consider that this is mainly caused by the low resolution of the event camera, affecting the inner pattern detection accuracy. Cameras' low resolution also makes *MGE* values not consistent with *EGT* in some cases since boards' positions are not detected very precisely. But we can still use this metric for selecting good extrinsics. From Fig. 11, we observe that extrinsics converge if around 15 frames are given. More data can further enforce accuracy. For event cameras, inner patterns and boundaries of the board are reconstructed clearly. Thus, the calibration accuracy does not have much degradation. A sufficient number of frames are used in calibration, whether using frame images or images reconstructed from event streams, and the extrinsics are successfully recovered. Fig. 13 shows the back-projection results of LiDAR points using extrinsics in Table II. The checkerboard detection results from LiDAR points are also shown. In *RLES03-LF*, extrinsics from our method can help LiDAR points better align on the board's boundaries than *Zhou-MATLAB*.

D. Ablation Study

Our method proposes several steps to handle the uncertainty of sensor data and calibration results, such as the point projection (Section IV-A2), uncertainty modeling of normal vectors (Section IV-B), and calibration with multiple computations (Section V). To show the impact of these individual modules, we conduct the following ablation study. We disable one of these modules and repeatedly test the proposed method with *RLFS*. Table III reports the calibration accuracy.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 13. Back-projection of the LiDAR points on images using the extrinsics in Table II: Our algorithm (red) and *Zhou-MATLAB* (green). (a) RLES01-LF. (b) RLES01-LE. (c) RLES02-LF. (d) RLES02-LE. (e) RLES03-LF. (f) RLES03-LE. (g) RLES02 (cloud). (h) RLES03 (cloud).

TABLE III
CALIBRATION RESULTS ON *RLFS* WITH OUR PROPOSED METHOD BY DISABLING SOME MODULES

Dataset	Method	$EGT_{\mathbf{R}}$ $[deg, \downarrow]$	$EGT_{\mathbf{t}}$ $[m, \downarrow]$	MGE $[m, \downarrow]$
RLFS01	*WO. PP*	0.517	0.023	0.015
	WO. UM	**0.119**	0.010	**0.008**
	WO. PtPL	0.292	0.011	**0.008**
	WO. PtL	0.556	0.018	**0.008**
	$I = 1$	0.195	**0.007**	0.009
	$I = 5$	0.262	0.012	**0.008**
	Ours	**0.130**	**0.009**	0.009
RLFS02	*WO. PP*	0.369	0.021	0.015
	WO. UM	0.338	0.015	0.008
	WO. PtPL	0.314	**0.012**	0.008
	WO. PtL	0.653	0.021	**0.007**
	$I = 1$	**0.277**	**0.014**	0.008
	$I = 5$	0.342	0.015	0.008
	Ours	**0.293**	**0.014**	**0.007**
RLFS03	*WO. PP*	0.378	0.018	0.016
	WO. UM	**0.207**	**0.007**	**0.008**
	WO. PtPL	0.338	0.016	0.009
	WO. PtL	0.289	0.019	**0.007**
	$I = 1$	0.475	0.024	**0.008**
	$I = 5$	0.324	0.017	**0.008**
	Ours	**0.136**	**0.008**	**0.008**

The first two results are marked as bold.
WO.: without. PP: point projection. UM: uncertainty modelling of normal vectors.
PtPL: point-to-plane constriants. PtL: point-to-line constriants.
I: the computation number in Algorithm 1.
↓/↑ indicates that the lower/higher the value, the better the score.

Disabling the point projection *(WO. PP)* or PtL constraints *(WO. PtL)* significantly degrades the calibration performance. This is because the point projection reduces noise of LiDAR points with the fitted planar parameters and the checkerboard's edges (the source of PtL constraints) offer strong constraints to extrinsics. We also observe that setting $I = 1$ or $I = 5$ may sometimes induce unreliable results (see *RLFS03*). To minimize the error bound of the results, the value of I cannot be small in practice.

VII. CONCLUSION

This article presents a novel automatic checkerboard-based approach to calibrate extrinsics between a LiDAR and a frame/event camera, which offers several desirable features such as automatic checkerboard detection and tracking, image reconstruction from event streams, and globally optimal extrinsics estimation. The proposed method has been extensively evaluated on both simulated sensors and real-world devices, demonstrating its superior performance in terms of accuracy in translation and rotation compared to a SOTA checkerboard-based method. Moving forward, future research could explore the effect of time undistortion on camera and LiDAR sensors, as well as further advancing the LCE configuration to address various robotics challenges, such as ego-motion estimation or object detection in complex scenarios.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Prof. L. Wang and S. Zhang for their insightful comments on this article. They also wish to extend their heartfelt appreciation to the editorial team of Transactions on Mechatronics and all the reviewers for their prompt and constructive feedback, which significantly enhanced this article.

REFERENCES

[1] J. Lin and F. Zhang, "R3LIVE: A robust, real-time, RGB-colored, LiDAR-inertial-visual tightly-coupled state estimation and mapping package," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 10672–10678.

[2] Y. Yu, P. Yun, B. Xue, J. Jiao, R. Fan, and M. Liu, "Accurate and robust visual localization system in large-scale appearance-changing environments," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 6, pp. 5222–5232, Dec. 2022.

[3] C. R. Qiet al., "Frustum pointnets for 3D object detection from RGB-D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 918–927.

[4] J. Zhang, C. Hu, R. G. Chadha, and S. Singh, "Maximum likelihood path planning for fast aerial maneuvers and collision avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 2805–2812.

[5] J. Jiaoet al., "Fusionportable: A multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 3851–3856.

[6] M. Cao, R. Wang, N. Chen, and J. Wang, "A learning-based vehicle trajectory-tracking approach for autonomous vehicles with lidar failure under various lighting conditions," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 2, pp. 1011–1022, Apr. 2022.

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

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

[9] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, “High speed and high dynamic range video with an event camera,” *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 43, no. 6, pp. 1964–1980, Jun. 2021.

[10] D.-G. Choi, Y. Bok, J.-S. Kim, and I. S. Kweon, “Extrinsic calibration of 2-D Lidars using two orthogonal planes,” *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 83–98, Feb. 2016.

[11] S. Verma, J. S. Berrio, S. Worrall, and E. Nebot, “Automatic extrinsic calibration between a camera and a 3D Lidar using 3D point and plane correspondences,” in *Proc. IEEE Intell. Transp. Syst. Conf.*. IEEE, 2019, pp. 3906–3912.

[12] K. Huang and C. Stachniss, “On geometric models and their accuracy for extrinsic sensor calibration,” in *Proc. IEEE IICRAnt. Conf. Robot. Autom.*, 2018, pp. 1–9.

[13] G. Yan, F. He, C. Shi, X. Cai, and Y. Li, “Joint camera intrinsic and lidar-camera extrinsic calibration,” 2022, *arXiv:2202.13708*.

[14] J. Wu, M. Wang, Y. Jiang, B. Yi, R. Fan, and M. Liu, “Simultaneous hand–eye/robot–world/camera–IMU calibration,” *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 4, pp. 2278–2289, Aug. 2022.

[15] L. Zhou, Z. Li, and M. Kaess, “Automatic extrinsic calibration of a camera and a 3D Lidar using line and plane correspondences,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 5562–5569.

[16] Q. Liao, Z. Chen, Y. Liu, Z. Wang, and M. Liu, “Extrinsic calibration of lidar and camera with polygon,” in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2018, pp. 200–205.

[17] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[18] Q. Zhang and R. Pless, “Extrinsic calibration of a camera and laser range finder (improves camera calibration),” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, vol. 3, pp. 2301–2306.

[19] R. Unnikrishnan and M. Hebert, “Fast extrinsic calibration of a laser rangefinder to a camera,” Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-09, 2005.

[20] G. Pandey, J. McBride, S. Savarese, and R. Eustice, “Extrinsic calibration of a 3 d laser scanner and an omnidirectional camera,” *IFAC Proc. Vol.*, vol. 43, no. 16, pp. 336–341, 2010.

[21] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis, “3D Lidar–camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization,” *Int. J. Robot. Res.*, vol. 31, no. 4, pp. 452–467, 2012.

[22] G. Koo, J. Kang, B. Jang, and N. Doh, “Analytic plane covariances construction for precise planarity-based extrinsic calibration of camera and lidar,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6042–6048.

[23] S. Sim, J. Sock, and K. Kwak, “Indirect correspondence-based robust extrinsic calibration of lidar and camera,” *Sensors*, vol. 16, no. 6, 2016, Art. no. 933.

[24] P. Moghadam, M. Bosse, and R. Zlot, “Line-based extrinsic calibration of range and image sensors,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 3685–3691.

[25] J.-K. Huang and J. W. Grizzle, “Improvements to target-based 3D Lidar to camera calibration,” *IEEE Access*, vol. 8, pp. 134101–134110, 2020.

[26] E. Olson, “Apriltag: A robust and flexible visual fiducial system,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3400–3407.

[27] V. Lepetit, F. Moreno-Noguer, and P. Fua, “EPnP: An accurate O (n) solution to the pnp problem,” *Int. J. Comput. Vis.*, vol. 81, no. 2, 2009, Art. no. 155.

[28] T. D. Barfoot and P. T. Furgale, “Associating uncertainty with three-dimensional poses for use in estimation problems,” *IEEE Trans. Robot.*, vol. 30, no. 3, pp. 679–693, Jun. 2014.

[29] J. Levinson and S. Thrun, “Automatic Online Calibration of Cameras and Lasers.,” in *Proc. Robot., Sci. and Syst.*, 2013, vol. 2, Art. no. 7.

[30] F. Chen, L. Li, S. Zhang, J. Wu, and L. Wang, “PBACalib: Targetless extrinsic calibration for high-resolution LiDAR-camera system based on plane-constrained bundle adjustment,” *IEEE Robot. Automat. Lett.*, vol. 8, no. 1, pp. 304–311, Jan. 2022.

[31] C. Yuan, X. Liu, X. Hong, and F. Zhang, “Pixel-level extrinsic self calibration of high resolution LiDAR and camera in targetless environments,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7517–7524, Oct. 2021.

[32] B.-H. Yoon, H.-W. Jeong, and K.-S. Choi, “Targetless multiple camera-lidar extrinsic calibration using object pose estimation,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13377–13383.

[33] J. Shiet al., “Calibrcnn: Calibrating camera and lidar by recurrent convolutional neural network and geometric constraints,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 10197–10202.

[34] M. J. Dominguez-Morales, A. Jimenez-Fernandez, G. Jiménez-Moreno, C. Conde, E. Cabello, and A. Linares-Barranco, “Bio-inspired stereo vision calibration for dynamic vision sensors,” *IEEE Access*, vol. 7, pp. 138415–138425, 2019.

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

[36] M. Muglikar, M. Gehrig, D. Gehrig, and D. Scaramuzza, “How to calibrate your event camera,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1403–1409.

[37] J. Wuet al., “Quadratic pose estimation problems: Globally optimal solutions, solvability/observability analysis, and uncertainty description,” *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 3314–3335, Oct. 2022.

[38] J. Jiao, H. Ye, Y. Zhu, and M. Liu, “Robust odometry and mapping for multi-LiDAR systems with online extrinsic calibration,” *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 351–371, Feb. 2022.

[39] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, “Density-based clustering,” *Wiley Interdiscipl. Reviews, Data Mining Knowl. Discov.*, vol. 1, no. 3, pp. 231–240, 2011.

[40] R. B. Rusu and S. Cousins, “3D is here: Point cloud library (PCL),” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 1–4.

[41] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.

[42] J. Jiao, H. Huang, L. Li, Z. He, Y. Zhu, and M. Liu, “Comparing representations in tracking for event camera-based slam,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1369–1376.

[43] J. Manderscheid, A. Sironi, N. Bourdis, D. Migliore, and V. Lepetit, “Speed invariant time surface for learning to detect corner points with event-based cameras,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10245–10254.

[44] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, vol. 3, pp. 2149–2154.

Jianhao Jiao received the B.Eng. degree in instrument science from Zhejiang University, Hangzhou, China, and the Ph.D. degree, supervised by Prof. Ming Liu, from the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China, in 2017 and 2021, respectively.

He is currently a Research Associate with the Hong Kong University of Science and Technology. His research interests include state estimation, SLAM, dense mapping, sensor fusion, and computer vision.

Feiyi Chen received the B.Eng. degree from the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China, and the master’s degree from the Department of Electronic and Computer Engineering, Hong Kong Unversity of Science and Technology, Hong Kong, China, in 2019 and 2022, respectively.

His research interests include multisensor fusion, calibration, and computer vision.

Hexiang Wei (Graduate Student Member, IEEE) received B.Eng. degree in mechanical engineering from Northeastern University, Shenyang, China, in 2020. He is currently working toward the Ph.D. degree with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China, led by Prof. Ming Liu.

His research interests include SLAM, sensor fusion, and robotics.

Jin Wu received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China. He is currently working toward the Ph.D. degree with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China.

From 2012 to 2018, he was in the UAV industry and has brought up two companies. Since 2018, he has been a Research Assistant with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology. From 2019 to 2020, he was with Tencent Robotics X, Shenzhen, China.

Ming Liu (Senior Member, IEEE) received the Ph.D. degree from the Department of Mechanical and Process Engineering, ETH Zurich, Zurich, Switzerland, in 2013, supervised by Prof. Roland Siegwart.

He is currently an Associate Professor with Robotics and Autonomous Systems, Hong Kong University of Science and Technology, Guangzhou, China, where he is also the Director of Intelligent Autonomous Driving Center. He has published several papers in top journals including IEEE TRANSACTIONS ON ROBOTICS AND INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH. His research interests include dynamic environment modeling, deep-learning for robotics, 3-D mapping, machine learning, and visual control.