Target-free Extrinsic Calibration of Event-LiDAR Dyad using Edge Correspondences

Wanli Xing[1,2], Shijie Lin[1,2,3], Lei Yang[2], and Jia Pan[1,2†]

Abstract—Calibrating the extrinsic parameters of sensory devices is crucial for fusing multi-modal data. Recently, event cameras have emerged as a promising type of neuromorphic sensors, with many potential applications in fields such as mobile robotics and autonomous driving. When combined with LiDAR, they can provide more comprehensive information about the surrounding environment. Nonetheless, due to the distinctive representation of event cameras compared to traditional frame-based cameras, calibrating them with LiDAR presents a significant challenge. In this paper, we propose a novel method to calibrate the extrinsic parameters between a dyad of an event camera and a LiDAR without the need for a calibration board or other equipment. Our approach takes advantage of the fact that when an event camera is in motion, changes in reflectivity and geometric edges in the environment trigger numerous events, which can also be captured by LiDAR. Our proposed method leverages the edges extracted from events and point clouds and correlates them to estimate extrinsic parameters. Experimental results demonstrate that our proposed method is highly robust and effective in various scenes.

Index Terms—Calibration and Identification, Sensor Fusion, Range Sensing.

I. INTRODUCTION

THE estimation of extrinsic parameters is a crucial problem of robotic perception and forms the basis for integrating various sensory inputs such as depth, RGB images, and other data modalities.

Light Detection and Ranging (LiDAR) is widely utilized in robotics to enable high-precision mapping and localization [1], [2] and collision detection [3]. However, LiDAR data solely provides geometric information, and therefore vision sensors are frequently employed in conjunction with LiDAR sensors to address issues such as SLAM degeneracy [4], [5], perception [6], and dynamic object tracking [7].

Event cameras [8], a recent novel vision sensing technology, present unique advantages over conventional frame-based cameras, including high dynamic range, resistance to motion blur, and low power consumption. This novel sensor's efficacy has been demonstrated in a diverse range of applications such as feature extraction [9], optical flow estimation [10], auto focusing [11], classification [12], motion deblurring [13], and visual odometry [14].

(a) Extrinsic calibration on-site without a specialized target

(b) Events under changing light

(c) LiDAR point cloud

(d) Flawed projection
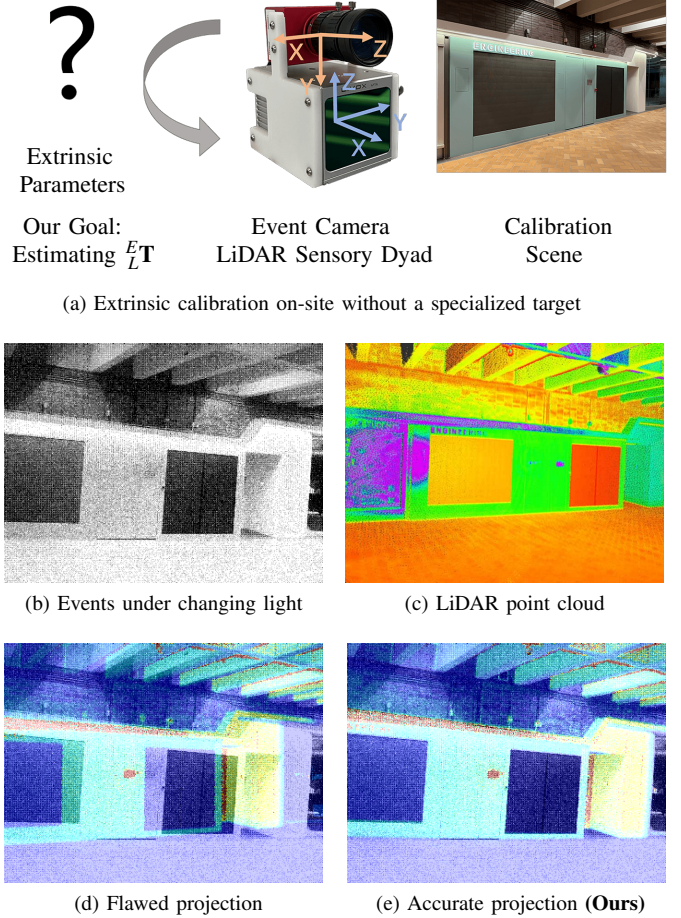

(e) Accurate projection **(Ours)**

Fig. 1. (a) In this paper, our goal is to develop a target-free method to calibrate the extrinsic transform of an event-LiDAR dyad. (b) shows an accumulated event image with a varying luminance. (c) shows a point cloud acquired by LiDAR. (d) Flawed extrinsic parameters would lead to large misalignments in projection. (e) Our estimated extrinsic parameters lead to a high-quality projection result.

Hence, integrating LiDAR and event cameras has become an appealing approach to generate synergy for advanced sensing in robotic applications. A recent work [15] has demonstrated promising results by fusing events and LiDAR data to generate dense depth measurements. However, integrating the two devices, i.e., the event camera and the LiDAR, presents a challenge for calibration. A custom-tailored 3D marker with a flickering display is proposed in [16] for calibrating the two devices, while an earlier work relies on the mechanical parameters for LiDAR to IMU calibration [17]. Despite the use of sophisticated devices and controlled lighting conditions,

[1]Wanli Xing, Shijie Lin, and Jia Pan are with the Department of Computer Science, The University of Hong Kong, Hong Kong SAR, China `wlxing@connect.hku.hk`

[2]Wanli Xing, Shijie Lin, Lei Yang, and Jia Pan are also with the Centre for Transformative Garment Production, Hong Kong SAR, China

[3]Shijie Lin is also with Peng Cheng Laboratory, Shenzhen, Guangdong, China

[†]Corresponding author

these methods struggle to produce reliable extrinsic parameter estimations for the event-LiDAR dyad, as large reprojection errors between the projected point clouds and events are observed, as illustrated in Fig. 1d.

To promote future research on this paired system and explore its potential in robotic applications, a reliable approach for calibrating the extrinsic parameters of the paired devices must be developed. Moreover, it is desirable that the event-LiDAR dyad can be calibrated in a general environment without controlled lighting conditions or specialized equipment.

However, two major challenges hinder the calibration process of the event-LiDAR dyad. The first challenge is the inherent difference between the data modalities. The LiDAR utilizes built-in laser emitters to sense depth, while the event camera detects brightness variations of natural light. This results in a radical difference in data format, with the LiDAR producing 3D points and the event camera generating spatiotemporal events. The second challenge is the sparsity of LiDAR data, requiring a stationary pose to obtain a dense point cloud. Meanwhile, event cameras detect brightness changes due to relative motion or lighting changes. These challenges pose significant hurdles to achieving an accurate calibration process for the event-LiDAR dyad.

In this paper, our primary research goal is to propose a target-free approach to calibrate the extrinsic parameters of an event-LiDAR dyad using reflectivity information in LiDAR. We take inspiration from the use of reflectivity information to calibrate a LiDAR and a frame-based camera in a prior work [18] and observe that the reflectivity changes in the LiDAR data match well with the patterns perceived in the event cameras (cf. Fig. 7). This property allows us to establish a reliable correspondence between the two data streams and bridge the modality gap between events and LiDAR data to estimate the extrinsic parameters between the devices. We detect edges in the point clouds with reflectivity variations and associate them with the events captured by the event camera. The extrinsic transform is then found as the minimizer of the re-projection error between the point-cloud edges and their associated event edges. Due to the sparsity of both sensory modalities, we develop a pipeline (Fig. 2) comprising a static stage to collect dense point cloud data followed by a moving stage to establish correspondence between the events and the point clouds across multiple poses.

In summary, our contributions are:

1) We propose an automatic extrinsic calibration pipeline for an event-LiDAR dyad, which can be used in general scenes without requiring controlled lighting or specialized targets;

2) We observe that geometric edges and reflectivity-changing edges present in the environment well match the events observed by the event camera, and exploit both types of edges to establish correspondence between the two data modalities;

3) We evaluate the consistency and accuracy of our method in multiple scenes, including indoor and outdoor scenes with complex geometry and random surface textures in various lighting conditions. We demonstrate accurate and robust extrinsic estimations in comparison with state-of-the-art methods.

II. RELATED WORKS

In this section, we briefly outline existing methods for calibrating both traditional frame-based cameras with LiDAR sensors and event cameras with LiDAR sensors, while discussing their primary characteristics and limitations.

A. Extrinsic Calibration of Conventional Camera-LiDAR System

Extensive research has been conducted on extrinsic calibration methods for traditional frame-based cameras and LiDAR. These methods can be classified into two categories based on whether they require a specific target. The first category comprises methods that use pre-prepared markers with known geometric parameters, such as standard planar checkerboards [19], [20] or other custom targets [21], [22]. These methods detect the markers from camera images and LiDAR point clouds and establish the correspondence between pixels and 3D points to obtain the extrinsic parameters. The second category comprises methods that do not require specific targets. Certain methods within this category use geometric features in the environment, such as planes and edges, to solve for extrinsic parameters through mutual information or feature matching [23], [24]. However, these methods are designed for traditional frame-based cameras and LiDAR and require cameras and LiDAR to remain stationary during the calibration process. Unfortunately, the imaging principle of event cameras makes these methods unsuitable for use with such cameras.

B. Extrinsic Calibration of Event Camera-LiDAR System

Theoretically, an event camera does not generate events in a stationary state with constant illumination. To address this conflict, some event camera intrinsic calibration methods [25], [26] have been designed with markers featuring blinking LEDs or screens. This allows event cameras to observe these markers in a static pose. This same concept has been applied to the extrinsic calibration of event cameras and LiDAR. Song et al. [16] proposed a calibration marker with four circular holes and placed a blinking screen behind it. The calibration marker and the background can be distinguished in the point cloud according to the depth, and the event camera generates events corresponding to the four circular holes in the calibration marker. However, this method requires a pre-prepared calibration marker and a display, which isn't always feasible or practical. Ta [27] *et al.* noticed that certain types of event cameras are sensitive to infrared light, enabling them to detect the brightness changes caused by LiDAR with the right wavelength. This property was then used to associate LiDAR 3D points with 2D events, and the sensors' extrinsic parameters were optimized by maximizing the mutual information between different sensory measurements. However, this method requires that the LiDAR's wavelength be within the visible spectrum of the event camera and that the LiDAR be capable of producing brightness changes that are strong enough to trigger events, which is not feasible in a general setup.

Most event-based datasets rely on indirect calibration to calibrate their LiDARs and event cameras. For instance, Gehrig
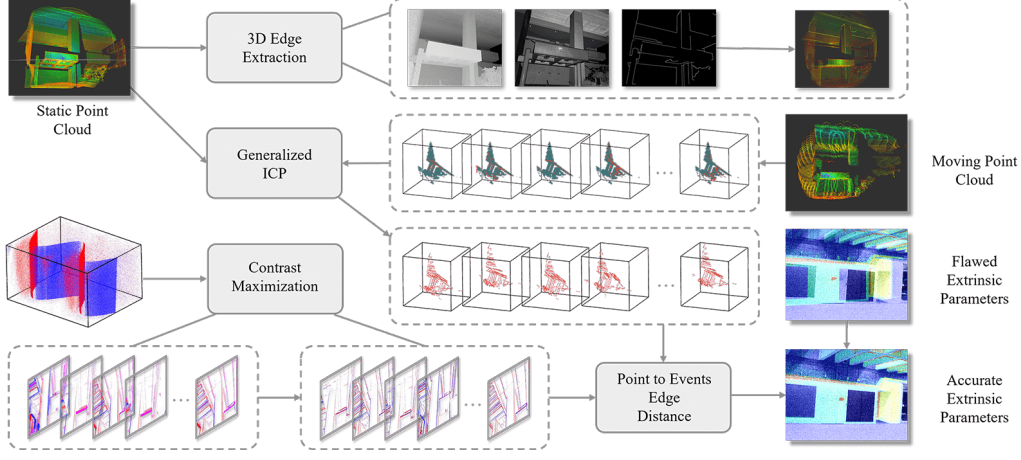
Fig. 2. **Calibration pipeline.** First, we maintain the event-LiDAR dyad still to capture a dense point cloud $\mathcal{P}_s$ (top left). The edges from $\mathcal{P}_s$ are extracted using both geometry and reflectivity information from the LiDAR data. Then, we rotate the sensing system to capture a sequence of motion data. We estimate the angular velocity of the rotational motion of the sensing system and derive a set of sharp events by solving a Contrast Maximization problem based on the acquired events. Subsequently, we use the estimated angular velocity to undistort the (sparse) point cloud data captured in motion and register them to $\mathcal{P}_s$ via Generalized-ICP. Finally, we associate the edges extracted from the point cloud with the sharp events and solve extrinsic parameters based on the points-to-events association.
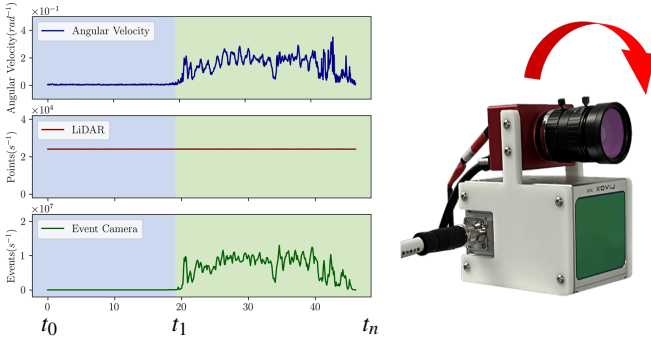

Fig. 3. The event-LiDAR dyad keeps static from t_0 to t_1 to accumulate a dense point cloud and then move from t_1 to t_n to collect events. Due to the imaging property of event cameras, the event rate is low when the sensor is stationary.
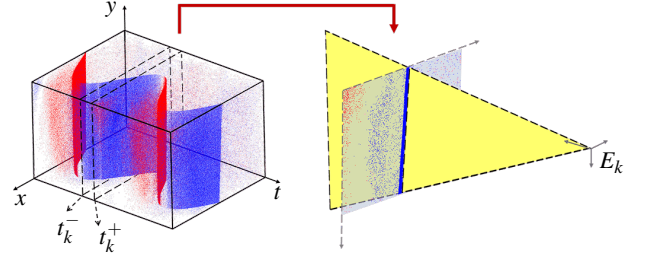

Fig. 4. Upon movement of the event camera, a significant amount of events are triggered by the 3D edge in space. The resulting spatial-temporal distribution of these events is displayed on the left. Notably, the projection relationship between the triggered events and the 3D edge is upheld, as demonstrated on the right.

et al. [28] collect the DSEC dataset using two Prophesee GEN3.1 event cameras and a Velodyne VLP-16 lidar. They first leverage the reconstructed intensity from events to calibrate the stereo event cameras and then peform point-to-plane ICP algorithm [29] to align the 3D points generated by the stereo cameras with the LiDAR points to obtain extrinsic parameters. Zhu *et al.* [17] collect the MVSEC dataset using two DAVIS346 event cameras and a Velodyne VLP-16 LiDAR. They use the standard image output of DAVIS 346 and conduct standard calibration with the Camera and Range Calibration Toolbox [30] and then fine-tune the result manually.

III. METHODOLOGY

A. Overview

In this paper, our goal is to determine the extrinsic transform, denoted as $_E^L\mathbf{T}$, between an event camera and a LiDAR in a target-free manner, without the need for specialized equipment, calibration boards, or controlled lighting conditions. To achieve this goal, we propose a pipeline, as depicted in Fig. 2, consisting of two stages: a stationary stage (from t_0 to t_1 in Fig. 3) and a motion stage (from t_1 to t_n in Fig. 3). During the stationary stage, the Lidar-Event Camera pair remains stationary in the environment, capturing a dense point cloud. In contrast, during the motion stage, the pair is rotated to trigger a significant number of events that correspond to the edges of the scene, as shown in Fig. 4. By formulating an optimization problem that matches the point cloud edges to the events, we are able to recover the extrinsic parameters.

Based on the observation that both geometric and reflectivity-changing edges present in the environment can well match the events triggered when the event camera moves, we first describe how to detect these edges from the static point cloud acquired by the LiDAR in Sec. III-B.

In Sec. III-C we introduce a method based on Contrast Maximization to estimate the angular velocity of the event-LiDAR dyad when it moves. With this, we can warp the accumulated event images to obtain sharp event edges and undistort the (sparse) point cloud acquired during motion. The undistorted point clouds can then be registered with the static

Fig. 5. Typical examples of two types of edges. (a) A geometric edge due to different reflecting angle; (b) A reflectivity edge due to material difference.
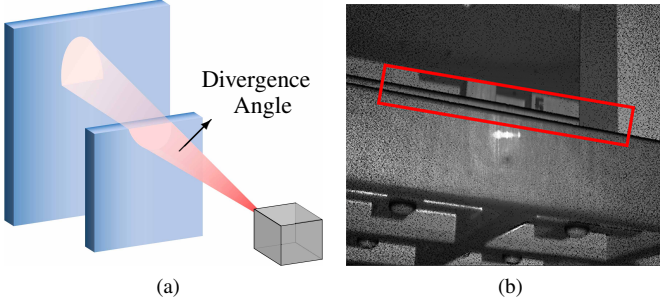

Fig. 6. Incorrect reflectivity due to occlusion. (a) The divergence angle of LiDAR laser; (b) Low reflectivity area.

point cloud to obtain dense geometric and reflectivity-changing edges.

In Sec. III-E, we present how to associate the point-cloud edges to the sharp event edges at a particular frame, as well as the formulation based on the reprojection error that solves the extrinsic transform $_E^L\mathbf{T}$.

B. Edges Extraction from Static Point Cloud

We observe that event-triggering edges can be classified into two categories, namely the geometric edges and the reflectivity edges, as labeled in Fig. 5 and shown in Fig. 7 As the brightness on either side of these edges differs due to the angle between them, geometric edges can trigger a large number of events as the event camera moves. The reflectivity edges similarly arouse many events because the reflectance values of the materials on both sides of the edges are different. Since geometric edges alone are too sparse for establishing correspondences with the dense events, we aim to use the reflectivity edges as well, which are also the major trigger of events.

In order to extract both types of edges from a point cloud, it is necessary to use the reflectivity information of the point cloud obtained by LiDAR. However, the reflectivity information of the LiDAR data can be unreliable when the laser beam emitted from the LiDAR hit two objects with one partially occluded by the other; see Fig. 6a as an example. This is because the laser beam has a divergence angle, and when an object in the environment is occluded by another, the laser beam is partially reflected by both of the objects. In such cases, the reflectivity and depth of the point are determined by calculating the energy and time of flight of the first echo,

respectively. However, since the first echo accounts for only part of the echo energy, the calculated reflectivity will be smaller than the case if the reflectivity is computed based on the full energy in the general case. This decrease in reflectivity can result in rapid reflectance changes around the objects as depicted in Fig. 6b and lead to spurious edges detected by the Canny edge detector. Note that, however, the depth estimation is not influenced by the occlusion present in the environment and thus is reliable.

Drawing from this observation, we first project the 3D point cloud $\mathcal{P}_s$, which was acquired during the stationary stage, onto a virtual imaging plane of the LiDAR as illustrated in Fig. 7a and Fig. 7b. This is followed by employing a median filter to fill in any empty pixels due to the sparsity nature of the LiDAR device we use (i.e., Livox Avia). Next, we apply the Canny edge detector [31] to the reflectivity image to detect reflectivity edges. Owing to the aforementioned problem, spurious edges could exit. Since the depth estimation is reliable in these occluded cases, we make use of the depth information to filter these spurious edges in the reflectivity image. To this end, we also apply the Canny edge detector to the depth map and prioritize the depth edges over those detected on the reflectivity image. Specifically, if there are edges detected on the depth map in the surrounding area (5×5 pixel patch) of an edge detected on the reflectivity image, we neglect the latter from the final detection result as shown in Fig. 7c. Finally, we lift the 2D edges detected to 3D using the inverse projection to find the corresponding 3D points in the point cloud. When a 2D edge point corresponds to multiple 3D edge points, we calculate the center of these 3D points and consider it as the 3D edge point corresponding to this 2D edge point. We denote by $\mathcal{P}_e$ the set of 3D edge points found from the point cloud $\mathcal{P}_s$.

C. Recovery of Distinct Events under Motion

The event camera outputs an event in an asynchronous manner whenever the logarithmic value of a pixel's brightness change exceeds a specified constant. A single event is described by a tuple, $e_i = (\mathbf{x}_i, t_i, p_i)$, where $\mathbf{x}_i$ is the 2D coordinate of the pixel, t_i is the timestamp when an event is triggered, and p_i is a binary value that represents the sign of the brightness change, colored with red and blue in Fig. 7f.

We use the notation $[t_k^-, t_k^+]$ to refer to a time interval centered at t_k with a short duration of Δt. The events that occur within this time interval are denoted by the set $\mathcal{E}_k = \{e_i\}_{i=0}^{N-1}$. We accumulate the events in $\mathcal{E}_k$ to obtain the event image $I_k(\mathbf{x})$ in Fig. 7f, which is given by

$$I_k(\mathbf{x}) = \sum_{i=0}^{N-1} p_i \delta(\mathbf{x} - \mathbf{x}_i), \tag{1}$$

where δ is the Dirac delta function.

Event image $I_k(\mathbf{x})$ is usually blurred due to the continuous motion of the event camera in time interval $[t_k^-, t_k^+]$. During this period, the motion of the camera can be approximated by a pure 3D rotation and the angular velocity $\omega_k \in \mathbb{R}^3$ can be estimated. We employ the Contrast Maximization framework [32], [33] to recover distinct events triggered by

(a) Depth image   (b) Reflectivity image   (c) Extracted 2D edges   (d) Extracted 3D edges   (e) Sharp event image   (f) Blurred event image
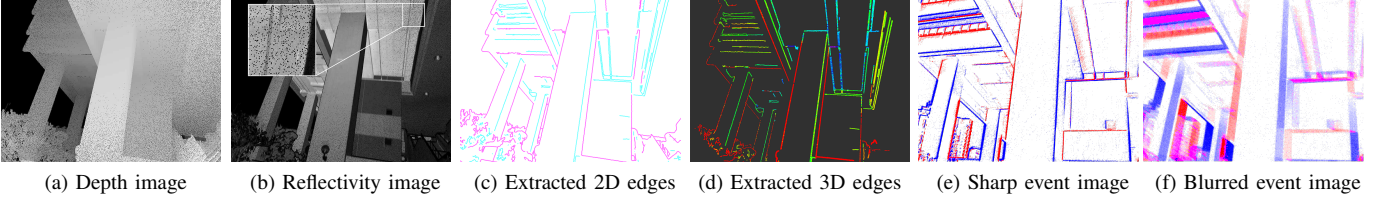
Fig. 7. The 2D edges are first extracted from (a) and (b), respectively, and then filtered to obtain edges in (c), where purple and blue represent geometric and reflectivity edges, respectively. (d) The corresponding 3D points that form the edges in the point cloud are identified through projection. With the Contrast Maximization framework, we can estimate motion parameters ω_k and warp events to a sharp event image (e) from the blurred event image (f) directly accumulated during movement.

edges from the obtained event image. An event e_i triggered at time t_i can be warped to time t_k, given by

$$\mathbf{x}'_i = \exp(\hat{\omega}_k(t_i - t_k))\mathbf{x}_i, \tag{2}$$

where $\hat{\omega}_k$ represents the cross-product matrix of ω_k. By applying this transformation, we obtain the warped event image:

$$I(\mathbf{x}, \omega_k) = \sum_{i=0}^{N-1} p_i \delta(\mathbf{x} - \mathbf{x}'_i). \tag{3}$$

To estimate the angular velocity ω_k, an objective function $f(\omega_k)$ is built from the warped events and defined as the variance of the warped event image:

$$f(\omega_k) = \mathrm{Var}(I(\mathbf{x}, \omega_k)) = \frac{1}{|\Omega|} \int_\Omega (I(\mathbf{x}, \omega_k) - \mu_I)^2 d\mathbf{x}, \tag{4}$$

where $\mu_I = \frac{1}{|\Omega|} \int_\Omega I(\mathbf{x}, \omega_k) d\mathbf{x}$ is the mean of the warped event image, computed as the average intensity over the image plane domain Ω. The functional measures the contrast of the event image with a different angular velocity ω_k. The maximizer ω_k^* of this functional can lead to a warped event image $I_k = I(\mathbf{x}, \omega_k^*)$ that contains distinct edges, as shown in Fig. 7e.

D. LiDAR Self-motion Undistortion and Localization

We aim to obtain the LiDAR's pose at time t_k so as to exploit the extracted 3D edge points $\mathcal{P}_e$ to establish the correspondence with the sharpened event image at t_k. However, the acquired point cloud $\mathcal{P}_k$ at t_k is subject to motion distortion as the LiDAR moves during this time interval $[t_k^-, t_k^+]$. The LiDAR's motion, denoted as ${}^L\mathbf{T}_{t_k^-}^{t_k^+}$, can be represented by the event camera's motion ${}^E\mathbf{T}_{t_k^-}^{t_k^+}$ in the same time interval and the extrinsic transform ${}^L_E\mathbf{T}$,

$$ {}^L\mathbf{T}_{t_k^-}^{t_k^+} = {}^L_E\mathbf{T} \cdot {}^E\mathbf{T}_{t_k^-}^{t_k^+} = {}^L_E\mathbf{T} \begin{bmatrix} \exp(\hat{\omega}_k \Delta t) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{5}$$

To compensate for the motion of a point $P_j \in \mathcal{P}_k$ with the timestamp $t_j \in [t_k^-, t_k^+]$, we compute the transformation at t_j with ${}^L\mathbf{T}_{t_k^-}^{t_k^+}$ and the time difference $t_j - t_k$ by linear interpolation. This way, the point cloud collected during the time interval $[t_k^-, t_k^+]$ can be undistorted and accumulated at t_k. We denote these undistorted point clouds as $\mathcal{P}'_k$. Finally, we solve the transformation between $\mathcal{P}'_k$ and $\mathcal{P}_s$ by the Generalized-ICP algorithm [34] and the obtained transformation at t_k is denoted as ${}^L\mathbf{T}_{t_0}^{t_k}$.
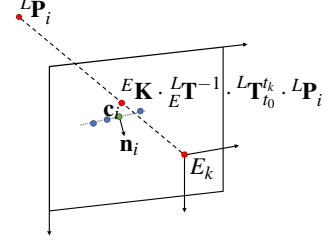

Fig. 8. After projecting the point ${}^L\mathbf{P}_i$ on the 3D edge to the event camera imaging plane at time t_k, we select the nearest m event pixels (blue dots) on the I_k image and calculate the center $\mathbf{c}_i$ and normal vector $\mathbf{n}_i$.

E. Point Cloud to Events Optimization

In our approach, we aim to associate edges extracted from the static point cloud with events. To accomplish this, we warp the events $\mathcal{E}_k$ into the event image I_k with the parameter ω_k^* estimated in Sec. III-C. As illustrated in Fig. 8, each point P_i in the edge point cloud $\mathcal{P}_e$ can be projected onto the event camera's imaging plane to obtain the corresponding projection point

$$ {}^E\mathbf{p}_i = {}^E\mathbf{K} \cdot {}^L_E\mathbf{T}^{-1} \cdot {}^L\mathbf{T}_{t_0}^{t_k} \cdot {}^L\mathbf{P}_i. \tag{6}$$

This projection process involves several parameters, including the intrinsic parameters of the event camera ${}^E\mathbf{K}$, the current extrinsic parameters ${}^L_E\mathbf{T}$, the transformation of LiDAR from t_0 to t_k denoted by ${}^L\mathbf{T}_{t_0}^{t_k}$, and the homogeneous coordinates of the point ${}^L\mathbf{P}_i$.

To ensure that the projection point ${}^E\mathbf{p}_i$ is accurately placed on the edge formed by the neighboring event pixels, we first search for the m closest event pixels of this projection in I_k. From these pixels, we calculate their mass center $\mathbf{c}_i$ and normal vector $\mathbf{n}_i$, respectively. We can then compute the distance of ${}^E\mathbf{p}_i$ to the edge as the inner product of the normal vector and the difference between the projection point and the mass center, as expressed in Eq. 7.

$$d = \langle \mathbf{n}_i, {}^E\mathbf{p}_i - \mathbf{c}_i \rangle. \tag{7}$$

We can sample N points on the edge of the point cloud at each sampled time interval, and they should satisfy the above point-to-edge constraint. For K sampled time interval, or K poses, we can define the cost function as

$$f({}^E_L\mathbf{T}) = \sum_{k=1}^{K} \sum_{i=1}^{N} \langle \mathbf{n}_i, {}^E\mathbf{K} \cdot {}^L_E\mathbf{T}^{-1} \cdot {}^L\mathbf{T}_{t_0}^{t_k} \cdot {}^L\mathbf{P}_i - \mathbf{c}_i \rangle. \tag{8}$$

We employed a gradient-based non-linear optimization algorithm to minimize the error described in Eq. 8 and determine
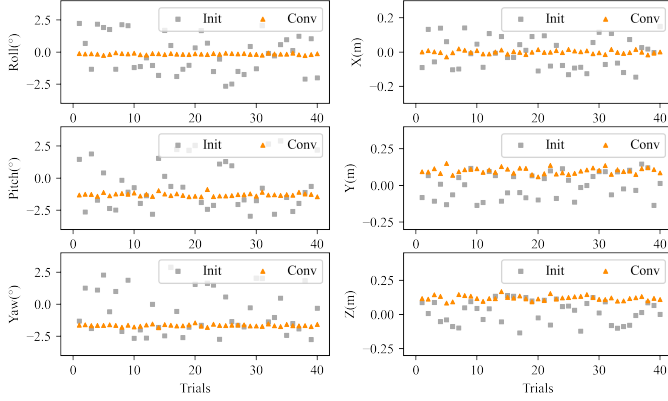
Fig. 9. Random initializations (in grey) and converged solutions (in orange) for extrinsic parameters on scene 4 are presented. It is noteworthy that the demonstrated extrinsic parameters are devoid of their nominal parts.

the extrinsic transform $_E^L\mathbf{T}$. We iteratively solved for $_E^L\mathbf{T}$ until convergence, which yielded the calibrated extrinsic parameters. To implement this approach, we chose to use the Ceres Solver[1].

IV. Experiments

A. Experimental Setup

Our experimental sensor setup comprises an iniVation DAVIS346 event camera and a Livox Avia solid-state LiDAR, as illustrated in Fig. 1a. In our method for extrinsic calibration, we only use the event sensor in DAVIS346. The frame-based sensor in DAVIS346 is used for providing reference results as will be described later. In all experiments, we report the deviation of the estimated extrinsic transform, $_E^L\mathbf{T}$, from the nominal extrinsic transform which is $(0,0,0)$ for translation and $(0, \pi/2, \pi/2)$ for rotation (ZYX Euler Angle).

Prior to data collection, we calibrated the event camera's intrinsic parameters and lens distortion parameters. We conducted data collection in six scenes, where both the event camera and LiDAR were stationary for ∼20 seconds before engaging in motion for ∼15 seconds. The movement was a handheld 3D rotational motion centered on the event camera. Throughout the six scenes, the event camera and LiDAR maintained consistent extrinsic parameters.

B. Robustness Analysis

To assess the robustness of our method against different initializations, we evaluated the extrinsic calibration process with various initial values randomly sampled around the nominal extrinsic parameters. In particular, we introduced a 3° rotation about a random axis to the nominal rotation matrix and added a random translation vector with a length of 0.2m to the nominal translation vector. We conducted 40 trials of the experiment in scene 1, as shown in Fig. 9, with the gray dots representing the initial values and the colored dots representing the final convergence values. Our results demonstrate that our method is highly resilient to different initial values, with the
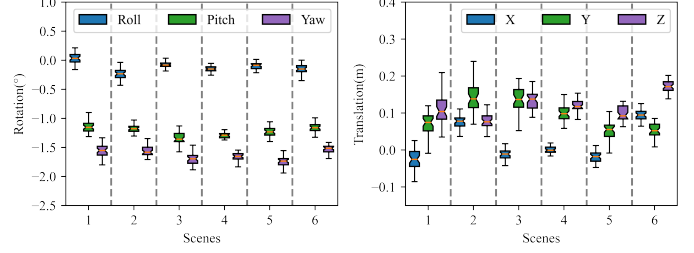
Fig. 10. The box plots encapsulate the statistical details, comprising maximum, third quartile, median, first quartile, and minimum, of extrinsic parameters over six distinct scenes. It is noteworthy that the demonstrated extrinsic parameters are devoid of their nominal parts.
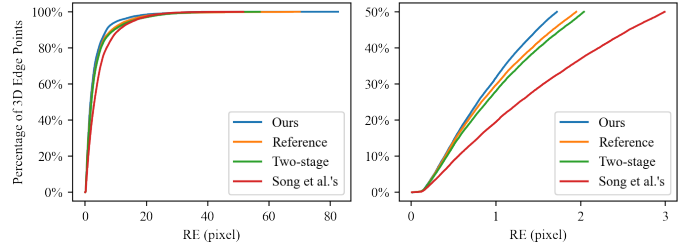

Fig. 11. All 3D edge points' RE and the top 50% points' RE under the extrinsic parameters.

final calibration results remaining consistent across multiple trials.

We also conducted the above calibration experiment (with 40 trials) on all six sets of experimental data using random initial values according to the randomization process described above to evaluate the robustness in different scenes. The statistical details of the convergence values of the algorithm across the different scenarios are presented in Fig. 10. Our method produced consistent calibration results across the different scenarios. We observed that the richness of edges within different scenes influenced the calibration results. In particular, scenes with clearer edges led to better outcomes. This can be attributed to the fact that our method relies on edge correspondences, and clearer edges provide more accurate information for the calibration process.

C. Performance Comparison

We conducted a comparative analysis of our method with two existing methods, namely Song *et al.*'s method [16] and the two-stage method. Song *et al.*'s method involves the use of a calibration board with circular holes and a continuously blinking display. The method detects circles from the accumulated events image and the point cloud and then calibrates the translation and rotation parameters in stages. In the two-stage approach, an additional frame-based camera is introduced, and the event-to-video reconstruction [35], [36] is employed to obtain the frame image of the event camera. The extrinsic parameters of the event camera and introduced frame-based camera are calibrated using the standard stereo camera calibration method. Subsequently, the frame-based camera and LiDAR extrinsic parameter calibration method [19] is used to calibrate the extrinsic parameters between the introduced

(a) Accumulated events image with varying luminance

(b) Point cloud reflectivity image

(c) Our method

(d) Reference method

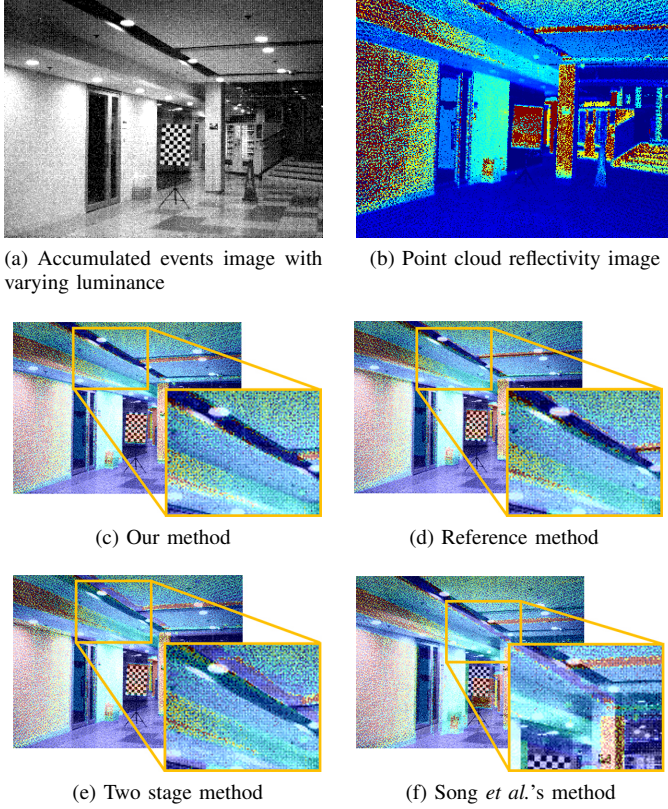(e) Two stage method

(f) Song *et al.*'s method

Fig. 12. Visual comparison among reprojection results by our method (c), the reference method using DAVIS346 frame-based camera (d), the two-stage method (e) and Song *et al.*'s method (f). Our method performs *on par* with the reference method.

(a) Static environment point cloud and segmented moving object point cloud (in white)

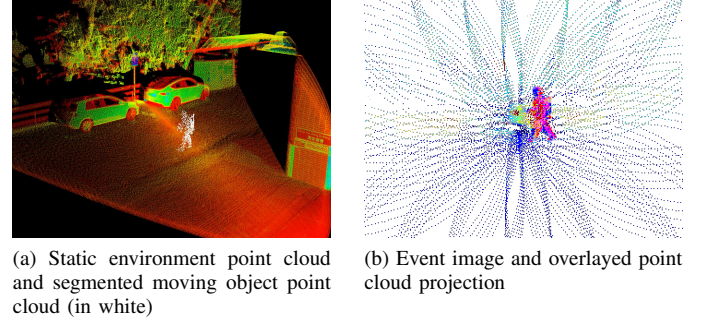(b) Event image and overlayed point cloud projection

Fig. 13. The event camera and LiDAR are fixed within the environment. Leveraging calibrated extrinsic parameters, we can segment the event-related point cloud from the overall point cloud.

frame-based camera and LiDAR, and finally, the two extrinsic parameters are combined to obtain the extrinsic parameters between the event camera and LiDAR. Finally, as a reference, we leverage the frame-based camera in the DAVIS346 event camera for calibrating the event sensor and the LiDAR. As the optical path of the DAVIS346 event camera is shared by both its frame and event sensors, the extrinsic parameters between the frame-based sensor and the LiDAR are equivalent to that of the event sensor and the LiDAR. We use this result as a reference for evaluation. To calibrate the extrinsic parameters between the frame-based sensor and the LiDAR, we utilized the standard calibration board under 20 different poses.

TABLE I
PPRE COMPARISON AMONG OUR CALIBRATION, REFERENCE, THE
TWO-STAGE METHOD AND SONG *et al.*'S METHOD

	Ours	Reference	Two-stage	Song *et al.*'s
PPRE	**3.161**	3.726	3.922	4.691

To evaluate the effectiveness of our calibration method, we conducted experiments in a static environment with varying luminance. We directly accumulated events to obtain event image and utilized the extrinsic parameters obtained by different methods to project the point cloud onto the image plane of the event camera to obtain the reflectivity image of the projected point cloud. Then, we overlaid the reflectivity image with the event image, as shown in Fig. 12, to visually check the reprojection quality. Our method exhibited the best alignment among the evaluated methods, and the calibration quality closely approximated the reference approach that uses the frame-based sensor of DAVIS346 for calibration.

We collected a total of 60 poses, uniformly distributed among 6 scenes, and utilized different extrinsic parameters to project the point cloud edges onto the imaging plane of the event camera. Using Eq. 7, we determined the reprojection error (RE) by calculating the distance between the edge points' reprojections and the adjacent event pixels. While appropriate extrinsic parameters ensure that most edge points are situated near adjacent events, imperfect correspondences between the point cloud edges and events may result in large RE values for certain points, even with ideal extrinsic parameters. Fig. 11 presents the distribution of RE values for all points and the top 50% of points under different extrinsic parameters. Our method's extrinsic parameters led to a significant reduction in RE values for most points. Additionally, we computed the per-point reprojection error (PPRE) for all points, as demonstrated in Tab. I. The results reveal that our method produces the smallest reprojection error compared to other methods.

D. Application

In Fig. 13, we demonstrate the practical utility and efficacy of our event camera and LiDAR extrinsic parameter calibration. With the devices fixed in the environment and calibrated extrinsic parameters applied, we can effectively segregate the event-related point cloud from the overall point cloud.

E. Limitations

Our method may exhibit reduced calibration accuracy in situations with sparse or indistinct edges, resulting in less precise extrinsic parameters and higher reprojection errors. Moreover, when the sensors are significantly separated [37], establishing correspondence between event and LiDAR data becomes challenging, impacting calibration precision.

V. Conclusions

Our paper proposes an innovative method that utilizes edge correspondences to achieve the extrinsic calibration of a dyad of a LiDAR and an event camera. This approach eliminates the requirement for specialized calibration board or equipment to change the luminance, allowing us to complete the calibration process in a general environment. As event camera technology evolves, our method has the potential to be applied in a variety of applications involving event cameras and LiDAR fusion.

References

[1] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[2] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[3] Z. Zhang, Y. Zhang, R. Han, L. Zhang, and J. Pan, "A generalized continuous collision detection framework of polynomial trajectory for mobile robots in cluttered environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9810–9817, 2022.

[4] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R2live: A robust, real-time, lidar-inertial-visual tightly-coupled state estimator and mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7469–7476, 2021.

[5] J. Lin and F. Zhang, "R3live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 672–10 678.

[6] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[7] B. Li, H. Meng, Y. Zhu, R. Song, M. Cui, G. Chen, and K. Huang, "Enhancing 3-d lidar point clouds with event-based camera," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

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

[9] R. Li, D. Shi, Y. Zhang, K. Li, and R. Li, "Fa-harris: A fast and asynchronous corner detector for event cameras," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6223–6229.

[10] P. Bardow, A. J. Davison, and S. Leutenegger, "Simultaneous optical flow and intensity estimation from an event camera," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 884–892.

[11] S. Lin, Y. Zhang, L. Yu, B. Zhou, X. Luo, and J. Pan, "Autofocus for event cameras," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 344–16 353.

[12] 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.

[13] S. Lin, Y. Zhang, D. Huang, B. Zhou, X. Luo, and J. Pan, "Fast event-based double integral for real-time robotics," *International Conference on Robotics and Automation (ICRA)*, 2023.

[14] 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.

[15] M. Cui, Y. Zhu, Y. Liu, Y. Liu, G. Chen, and K. Huang, "Dense depth-map estimation based on fusion of event camera and sparse lidar," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[16] R. Song, Z. Jiang, Y. Li, Y. Shan, and K. Huang, "Calibration of event-based camera and 3d lidar," in *2018 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*. IEEE, 2018, pp. 289–295.

[17] 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.

[18] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic extrinsic calibration of vision and lidar by maximizing mutual information," *Journal of Field Robotics*, vol. 32, no. 5, pp. 696–722, 2015.

[19] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5562–5569.

[20] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2301–2306.

[21] J. Domhof, J. F. Kooij, and D. M. Gavrila, "A joint extrinsic calibration tool for radar, camera and lidar," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 571–582, 2021.

[22] Y. Park, S. Yun, C. S. Won, K. Cho, K. Um, and S. Sim, "Calibration between color camera and 3d lidar instruments with a polygonal planar board," *Sensors*, vol. 14, no. 3, pp. 5333–5353, 2014.

[23] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7517–7524, 2021.

[24] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[25] 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. 138 415–138 425, 2019.

[26] E. Mueggler, B. Huber, and D. Scaramuzza, "Event-based, 6-dof pose tracking for high-speed maneuvers," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2761–2768.

[27] K. Ta, D. Bruggemann, T. Brödermann, C. Sakaridis, and L. Van Gool, "L2e: Lasers to events for 6-dof extrinsic calibration of lidars and event cameras," 2022.

[28] 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.

[29] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.

[30] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 3936–3943.

[31] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[32] 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.

[33] 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.

[34] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[35] 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*, vol. 43, no. 6, pp. 1964–1980, 2019.

[36] M. Muglikar, M. Gehrig, D. Gehrig, and D. Scaramuzza, "How to calibrate your event camera," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, June 2021.

[37] J. S. Berrio, M. Shan, S. Worrall, and E. Nebot, "Camera-lidar integration: Probabilistic sensor fusion for semantic mapping," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7637–7652, 2021.