Targetless Extrinsic Calibration of Multiple Small FoV LiDARs and Cameras using Adaptive Voxelization

Xiyuan Liu, Chongjian Yuan, and Fu Zhang

***Abstract*—Determining the extrinsic parameter between multiple LiDARs and cameras is essential for autonomous robots, especially for solid-state LiDARs, where each LiDAR unit has a very small Field-of-View (FoV), and multiple units are often used collectively. The majority of extrinsic calibration methods are proposed for 360° mechanical spinning LiDARs where the FoV overlap with other LiDAR or camera sensors is assumed. Few research works have been focused on the calibration of small FoV LiDARs and cameras nor on the improvement of the calibration speed. In this work, we consider the problem of extrinsic calibration among small FoV LiDARs and cameras, with the aim to shorten the total calibration time and further improve the calibration precision. We first implement an adaptive voxelization technique in the extraction and matching of LiDAR feature points. Such a process could avoid the redundant creation of k-d trees in LiDAR extrinsic calibration and extract LiDAR feature points in a more reliable and fast manner than existing methods. We then formulate the multiple LiDAR extrinsic calibration into a LiDAR Bundle Adjustment (BA) problem. By deriving the cost function up to second-order, the solving time and precision of the non-linear least square problem are further boosted. Our proposed method has been verified on data collected in four targetless scenes and under two types of solid-state LiDARs with a completely different scanning pattern, density, and FoV. The robustness of our work has also been validated under eight initial setups, with each setup containing 100 independent trials. Compared with the state-of-the-art methods, our work has increased the calibration speed 15 times for LiDAR-LiDAR extrinsic calibration (averaged result from 100 independent trials) and 1.5 times for LiDAR-Camera extrinsic calibration (averaged result from 50 independent trials) while remaining accurate. To benefit the robotics community, we have also open-sourced our implementation code on GitHub.**

***Index Terms*—Multiple LiDAR-Camera Extrinsic Calibration, Small FoV LiDAR, High-Resolution Mapping.**

I. Introduction

LiDAR and camera sensors, due to their superior characteristics in direct spatial ranging and rich color information conveying, have been increasingly used in autonomous driving [1, 2], navigation [3, 4] and high-resolution mapping [5] applications. One drawback of the current 360° mechanical spinning LiDAR is their dramatic high cost, preventing their massive application in industry. Solid-state LiDAR [6] has a much lower cost while achieving a denser point cloud within its FoV. However, solid-state LiDARs are of small FoV that multiple solid-state LiDARs need to be combined to achieve a similar FoV coverage as the mechanical spinning LiDAR. This setup necessitates precise extrinsic calibration among the LiDARs and cameras.

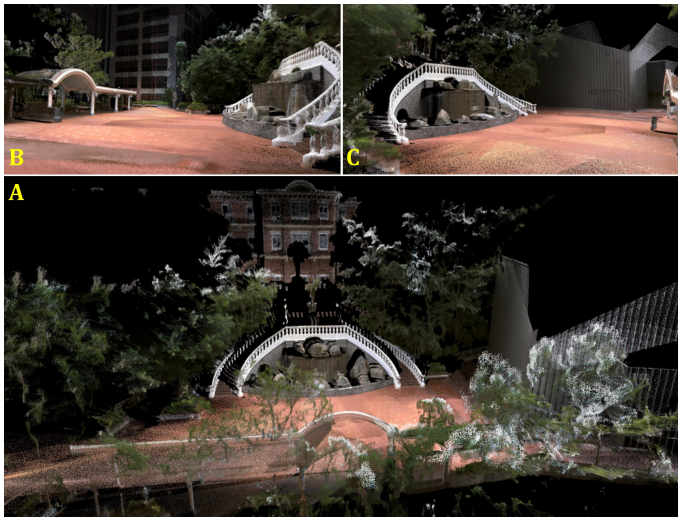


Fig. 1: A) The dense colorized point cloud with the LiDAR poses and extrinsic parameters optimized by our proposed method. The views from other perspectives are exhibited in B) left side and C) right side. Our experiment video is available at https://youtu.be/PaiYgAXl9iY.

Several challenges reside in the extrinsic calibration involving small FoV LiDARs: (1) Limited FoV overlap among the sensors and the precision requirement. Current methods usually require the existence of a common FoV between each pair of sensors [7]–[10], such that each feature is viewed by all sensors. In real-world applications, this FoV overlap might be minimal or not even exist due to the small FoVs of solid-state LiDARs and their numerous sensor mounting positions. The accuracy requirement of the calibration results, e.g., the consistency and colorization of the point cloud (see Fig. 1), is thus more challenging. (2) Computation time demands. For general ICP-based LiDAR extrinsic calibration approaches [5, 11], the extrinsic is optimized by aligning the point cloud from all LiDARs and maximizing the point cloud's consistency. The increase in the number of LiDARs implies that the feature point correspondence searching will be more time-consuming. This is due to the reason that each feature point needs to search for and match with nearby feature points using a k-d tree which

Xiyuan Liu, Chongjian Yuan and Fu Zhang are with the Department of Mechanical Engineering, The University of Hong Kong, Pokfulam, Hong Kong Special Administrative Region, People's Republic of China. (Corresponding author: Fu Zhang) (email: {xliuaa,ycjl}@connect.hku.hk, fuzhang@hku.hk).

contains the whole point cloud. In the LiDAR-camera extrinsic calibration, a larger amount of LiDAR points will also lead to more computation time in the LiDAR feature extraction.

To address the above challenges, we propose a fast and targetless approach for extrinsic calibration of multiple small FoV LiDARs and cameras. To create enough co-visible features among the small FoV sensors, we introduce motions to the sensor platform such that each sensor will scan the same area (hence features) at different times. We first calibrate the extrinsic among LiDARs (and simultaneously estimate the LiDAR poses) by registering their point cloud using an efficient Bundle Adjustment (BA) method we recently proposed [4]. To reduce time consumption in feature correspondence matching among LiDARs, we implement an adaptive voxelization to dynamically segment the point cloud into multiple voxels so that only one plane feature resides in each voxel (see Sec. III-B). We then calibrate the extrinsic between the cameras and LiDARs by matching the co-visible features between the images and the above-reconstructed point cloud. To further accelerate the feature correspondence matching, we inherit the above adaptive voxel map to extract LiDAR edge features. In summary, our contributions are listed as follows:

- We propose a targetless extrinsic calibration pipeline for multiple small FoV LiDARs and cameras that share very few or even no FoV overlap. We formulate LiDAR extrinsic calibration into a Bundle Adjustment problem and implement an adaptive voxelization technique into the LiDAR feature extraction and matching process. The overall pipeline enjoys higher calibration precision and computation efficiency.
- We verify our proposed work on data collected in various test scenes by LiDARs of different scanning patterns, FoVs, and point densities. When compared to various state-of-the-art methods, our proposed work could boost the speed by 15 times for multiple LiDAR calibration and 1.5 times for multiple LiDAR-Camera calibration. Meanwhile, our proposed work maintains high calibration precision, with the average translation and rotation errors down to 6mm and 0.09 degrees for LiDAR-camera and 8mm and 0.2 degrees for LiDAR-LiDAR.
- We open-source our implementation in ROS on GitHub[1] to benefit the robotics community.

II. Related Works

A. LiDAR-LiDAR Extrinsic Calibration

The extrinsic calibration methods between multiple LiDARs could be divided into motion-based and motionless approaches. Motion-based approaches assume each sensor undergoes the same rigid motion in each time interval [2, 12, 13] and transform the extrinsic calibration into a Hand-Eye problem [14]. Authors in [15]–[17] also introduce external inertial navigation sensors to facilitate the motion estimation of LiDARs. The calibration precision of these approaches is easily affected by the accuracy of the LiDAR odometry results, which might be unreliable. Motionless methods have been discussed in [7, 8] where the authors attach retro-reflective tapes to the surface of calibration targets to create and facilitate the feature extraction among multiple LiDARs. These approaches require prior preparation work and FoV overlap between LiDARs, which is unpractical in real-world applications.

In our previous work [5], a simple rotational movement is introduced to eliminate the requirement of FoV overlap, as each onboard sensor could percept the same region of interest. Then the extrinsic parameter is calibrated, along with the estimation of LiDAR poses, by optimizing the consistency of the point cloud map with iterative closest point (ICP) registration. The main problem within [5] is that the ICP registration always registers one scan to the other, leading to an iterative process where only one optimization variable (e.g., extrinsic or LiDAR poses) can be optimized (by registering the point cloud affected by the variable under optimization to the rest). Such an iterative procedure is prolonged to converge. Moreover, at each iteration, the ICP-based feature correspondence matching process might be very time-consuming. As for each point-to-plane correspondence, ICP needs to either search inside a k-d tree containing the entire point cloud or create a k-d tree containing the local point cloud every time before searching.

In this work, we formulate the extrinsic calibration into a bundle adjustment (BA) problem [4], where all the optimization variables (both extrinsic and LiDAR poses) are optimized concurrently by registering points into their corresponding plane. When compared to other plane adjustment techniques [18, 19], the BA technique we use does not estimate the plane parameters in the optimization process but solves for them analytically in a closed-form solution prior to the optimization iteration. The removal of plane parameters from the optimization iteration lowers the dimension significantly and leads to very efficient multi-view registration. To match points corresponding to the same plane, we implement an adaptive voxelization technique [4] to replace the k-d tree in [5]. As only one plane feature exists in each voxel, our proposed work significantly saves the computation time in correspondence searching while remaining accurate (see Sec. III-B).

B. LiDAR-Camera Extrinsic Calibration

The extrinsic calibration between LiDAR and camera could be mainly divided into target-based and targetless methods. In target-based approaches, the geometric features, e.g., edges and surfaces, are extracted from artificial geometric solids [20]–[22] or chessboard [23, 24] using intensity and color information. These features are matched either automatically or manually and are solved with non-linear optimization tools. In [25], authors establish the constraints using the crosswalk features on the streets; however, this method is essentially target-based as the parallelism characteristic of the crosswalk is used. Since extra calibration targets and manual work are needed, these methods are less practical compared with targetless solutions.

The targetless methods could be further divided into motion-based and motionless approaches. In motion-based methods, the initial extrinsic parameter is usually estimated by the motion information and refined by the appearance information.

[1]https://github.com/hku-mars/mlcc

In [26], authors reconstruct a point cloud from images using the structure from motion (SfM) to determine the initial extrinsic parameter and refine it by back-projecting LiDAR points onto the image plane. In [13, 27], authors initialize the extrinsic parameter by Hand-Eye calibration and optimize it by minimizing the re-projection error between images and LiDAR scans. In motionless approaches, only the edge features that co-exist in both sensors' FoV are extracted and matched. Then the extrinsic parameter is optimized by minimizing the re-projected edge-to-edge distances [9, 28]–[30] or by maximizing the mutual information between the back-projected LiDAR points and the images [10].

Our proposed work is targetless and creates co-visible features by moving the sensor suite to multiple poses, hence allowing extrinsic calibration between LiDAR and cameras even when they have no overlap, a circumstance that was not solved in prior works [10, 13, 29]. Moreover, compared with our previous work [28] which extracts LiDAR edge features using the RANSAC algorithm, this work extracts edge features using the same adaptive voxelization already computed in the LiDAR extrinsic calibration, which is more competitive in computation time and calibration precision. Compared with [10] which uses LiDAR intensity information as a feature, our work uses more reliable 3D edge information and is more computationally efficient and accurate (see Sec. IV). Moreover, our work does not require the common FoV between sensors.

III. Methodology

A. Overview

Let ${}^B_A\mathbf{T} = ({}^B_A\mathbf{R}, {}^B_A\mathbf{t}) \in SE(3)$ represent the rigid transformation from frame A to frame B, where ${}^B_A\mathbf{R} \in SO(3)$ and ${}^B_A\mathbf{t} \in \mathbb{R}^3$ are the rotation and translation. We denote $\mathcal{L} = \{L_0, L_1, \cdots, L_{n-1}\}$ the set of n LiDARs, where L_0 represents the base LiDAR for reference, $\mathcal{C} = \{C_0, C_1, \cdots, C_h\}$ the set of h cameras, $\mathcal{E}_L = \{{}^{L_0}_{L_1}\mathbf{T}, {}^{L_0}_{L_2}\mathbf{T}, \cdots, {}^{L_0}_{L_{n-1}}\mathbf{T}\}$ the set of LiDAR extrinsic parameters and $\mathcal{E}_C = \{{}^{C_0}_{L_0}\mathbf{T}, {}^{C_1}_{L_0}\mathbf{T}, \cdots, {}^{C_h}_{L_0}\mathbf{T}\}$ the set of LiDAR-camera extrinsic parameters. To create co-visible features between multiple LiDARs and cameras that may share no FoV overlap, we rotate the robot platform to m poses such that the same region of interest is scanned by all sensors (see Fig. 2). Denote $\mathcal{T} = \{t_0, t_1, \cdots, t_{m-1}\}$ the time for each of the m poses and the pose of the base LiDAR at the initial time as the global frame, i.e., ${}^G_{L_0}\mathbf{T}_{t_0} = \mathbf{I}_{4\times4}$. Denote $\mathcal{S} = \{{}^G_{L_0}\mathbf{T}_{t_1}, {}^G_{L_0}\mathbf{T}_{t_2}, \cdots, {}^G_{L_0}\mathbf{T}_{t_{m-1}}\}$ the set of the base LiDAR poses in global frame. The point cloud patch scanned by LiDAR $L_i \in \mathcal{L}$ at time $t_j \in \mathcal{T}$ is denoted by $\mathcal{P}_{L_i,t_j}$, which is in L_i's local frame. This point cloud patch could be transformed to global frame by

$$
\begin{aligned}
{}^G\mathcal{P}_{L_i,t_j} &= {}^G_{L_i}\mathbf{T}_{t_j}\mathcal{P}_{L_i,t_j} \\
&\triangleq \{{}^G_{L_i}\mathbf{R}_{t_j}\mathbf{p}_{L_i,t_j} + {}^G_{L_i}\mathbf{t}_{t_j},\ \forall \mathbf{p}_{L_i,t_j} \in \mathcal{P}_{L_i,t_j}\}.
\end{aligned} \tag{1}
$$

In our proposed approach of multi-sensor calibration, we sequentially calibrate the $\mathcal{E}_L$ and $\mathcal{E}_C$. In the first step, we simultaneously estimate the LiDAR extrinsic $\mathcal{E}_L$ and the base lidar pose trajectory $\mathcal{S}$ based on an efficient multi-view registration (see Sec. III-C). In the second step, we calibrate the $\mathcal{E}_C$ by matching the depth-continuous edges extracted from images and the above-reconstructed point cloud (see Sec. III-D). Lying in the center of both LiDAR and camera extrinsic calibration is an adaptive map, which finds correspondence among LiDAR and camera measurements efficiently (Sec. III-B).

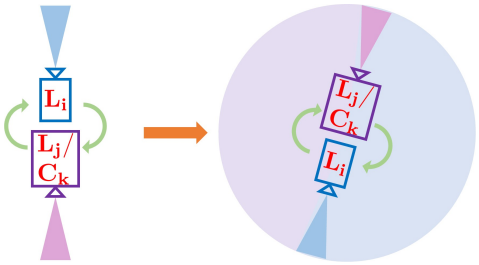


Fig. 2: FoV overlap created by rotation between two opposite pointing sensors. The original setup of two sensors L_i and L_j/C_k share no FoV overlap. With the introduction of rotational motion, the same region is scanned by all sensors across different times.

B. Adaptive Voxelization

To find the correspondences among different LiDAR scans, we assume the initial base LiDAR trajectory $\mathcal{S}$, LiDAR extrinsic $\mathcal{E}_L$, and camera extrinsic $\mathcal{E}_C$ are available. The initial base LiDAR trajectory $\mathcal{S}$ could be obtained by an online LiDAR SLAM (e.g., [3]), and the initial extrinsic could be obtained from the CAD design or a rough Hand-Eye calibration [14]. Our previous work [5] extracts edge and plane feature points from each LiDAR scan and matches them to the nearby edge and plane points in the map by a k-nearest neighbor search (k-NN). This would repeatedly build a k-d tree of the global map at each iteration. In this paper, we use a more efficient voxel map proposed in [4] to create correspondences among all LiDAR scans.

The voxel map is built by cutting the point cloud (registered using the current $\mathcal{S}$ and $\mathcal{E}_L$) into small voxels such that all points in a voxel roughly lie on a plane (with some adjustable tolerance). The main problem of the fixed-resolution voxel map is that if the resolution is high, the segmentation would be too time-consuming, while if the resolution is too low, multiple small planes in the environments falling into the same voxel would not be segmented. To best adapt to the environment, we implement an adaptive voxelization process. More specifically, the entire map is first cut into voxels with a pre-set size (usually large, e.g., 4m). Then for each voxel, if the contained points from all LiDAR scans roughly form a plane (by checking the ratio between eigenvalues), it is treated as a planar voxel; otherwise, they will be divided into eight octants, where each will be examined again until the contained points roughly form a plane or the voxel size reaches the pre-set minimum lower bound. Moreover, the adaptive voxelization is performed directly on the LiDAR raw points, so no prior feature points extraction is needed as in [5].

Fig. 3 shows a typical result of the adaptive voxelization process in a complicated campus environment. As can be seen, this process is able to segment planes of different sizes, including large planes on the ground, medium planes on the building walls, and tiny planes on tree crowns.

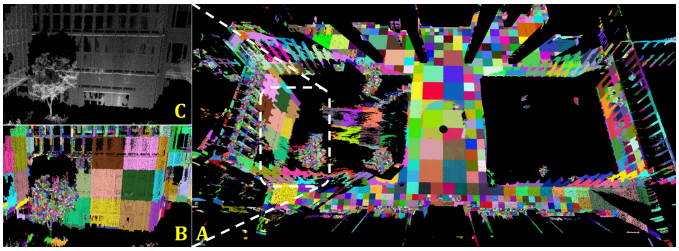


Fig. 3: A) LiDAR point cloud segmented with the adaptive voxelization. Points within the same voxel are colored identically. The detailed adaptive voxelization of points in the dashed white rectangle could be viewed in B) colored points and C) original points. The default size for the initial voxelization is 4m, and the minimum voxel size is 0.25m.

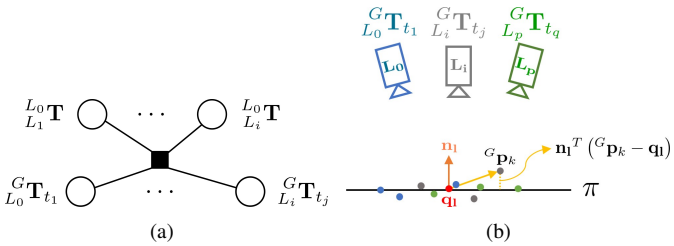


Fig. 4: (a) The l-th factor item relating to $\mathcal{S}$ and $\mathcal{E}_L$ with $L_i \in \mathcal{L}$ and $t_j \in \mathcal{T}$. (b) The distance from the point ${}^G\mathbf{p}_k$ to the plane $\boldsymbol{\pi}$.

C. Multi-LiDAR Extrinsic Calibration

With adaptive voxelization, we can obtain a set of voxels of different sizes. Each voxel contains points that are roughly on a plane and creates a planar constraint for all LiDAR poses that have points in this voxel. More specifically, considering the l-th voxel consisting of a group of points $\mathcal{P}_l = \{{}^G\mathbf{p}_{L_i,t_j}\}$ scanned by $L_i \in \mathcal{L}$ at times $t_j \in \mathcal{T}$. We define a point cloud consistency indicator $c_l\left({}^G_{L_i}\mathbf{T}_{t_j}\right)$ which forms a factor on $\mathcal{S}$ and $\mathcal{E}_L$ as shown in Fig. 4(a). Then, the base LiDAR trajectory and extrinsic are estimated by optimizing the factor graph. A natural choice for the consistency indicator $c_l(\cdot)$ would be the summed Euclidean distance between each ${}^G\mathbf{p}_{L_i,t_j}$ to the plane to be estimated (see Fig. 4(b)). Taking account of all such indicators within the voxel map, we could formulate the problem as

$$\arg\min_{\mathcal{S},\mathcal{E}_L,\mathbf{n}_l,\mathbf{q}_l} \sum_l \underbrace{\left(\frac{1}{N_l}\sum_{k=1}^{N_l}\left(\mathbf{n}_l^T\left({}^G\mathbf{p}_k - \mathbf{q}_l\right)\right)^2\right)}_{l\text{-th factor}}, \tag{2}$$

where ${}^G\mathbf{p}_k \in \mathcal{P}_l$, N_l is the total number of points in $\mathcal{P}_l$, $\mathbf{n}_l$ is the normal vector of the plane and $\mathbf{q}_l$ is a point on this plane.

It is noticed that the optimization variables $(\mathbf{n}_l, \mathbf{q}_l)$ in (2) could be analytically solved (see Appendix A) and the resultant cost function (3) is over the LiDAR pose ${}^G_{L_i}\mathbf{T}_{t_j}$ (hence the base LiDAR trajectory $\mathcal{S}$ and extrinsic $\mathcal{E}_L$) only, as follows

$$\arg\min_{\mathcal{S},\mathcal{E}_L} \sum_l \lambda_3\left(\mathbf{A}_l\right) \tag{3}$$

where $\lambda_3(\mathbf{A}_l)$ denotes the minimal eigenvalue of matrix $\mathbf{A}_l$ defined as

$$\mathbf{A}_l = \frac{1}{N_l}\sum_{k=1}^{N_l} {}^G\mathbf{p}_k\,{}^G\mathbf{p}_k^T - \mathbf{q}_l^*\mathbf{q}_l^{*T}, \mathbf{q}_l^* = \frac{1}{N_l}\sum_{k=1}^{N_l} {}^G\mathbf{p}_k. \tag{4}$$

To allow efficient optimization in (3), we derive the closed-form derivatives w.r.t the optimization variable $\mathbf{x}$ up to second-order (the detailed derivation from (3) to (5) is elaborated in Appendix B):

$$\lambda_3(\mathbf{x} \boxplus \delta\mathbf{x}) \approx \lambda_3(\mathbf{x}) + \bar{\mathbf{J}}\delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^T\bar{\mathbf{H}}\delta\mathbf{x}, \tag{5}$$

where $\bar{\mathbf{J}}$ is the Jacobian matrix, and $\bar{\mathbf{H}}$ is the Hessian matrix. The $\delta\mathbf{x}$ is a small perturbation of the optimization variable $\mathbf{x}$:

$$\mathbf{x} = [\underbrace{\cdots {}^G_{L_0}\mathbf{R}_{t_j}\ {}^G_{L_0}\mathbf{t}_{t_j}\cdots}_{\mathcal{S}}\underbrace{\cdots\cdots {}^{L_0}_{L_i}\mathbf{R}\ {}^{L_0}_{L_i}\mathbf{t}\cdots}_{\mathcal{E}_L}].$$

Then the optimal $\mathbf{x}^*$ could be determined by iteratively solving (6) with the LM method and updating the $\delta\mathbf{x}$ to $\mathbf{x}$.

$$\left(\bar{\mathbf{H}} + \mu\mathbf{I}\right)\delta\mathbf{x} = -\bar{\mathbf{J}}^T \tag{6}$$

D. LiDAR-Camera Extrinsic Calibration

With the LiDAR extrinsic parameter $\mathcal{E}_L$ and pose trajectory $\mathcal{S}$ computed above, we obtain a dense global point cloud by transforming all LiDAR points to the base LiDAR frame. Then, the extrinsic $\mathcal{E}_C$ is optimized by minimizing the summed distance between the back-projected LiDAR edge feature points and the image edge feature points. Two types of LiDAR edge points could be extracted from the point cloud. One is the depth-discontinuous edge between the foreground and background objects, and the other is the depth-continuous edge between two neighboring non-parallel planes. As explained in our previous work [28], depth-discontinuous edges suffer from foreground inflation and bleeding points phenomenon; we hence use depth-continuous edges to match the point cloud and images.

In [28], the LiDAR point cloud is segmented into voxels with uniform sizes, and the planes inside each voxel are estimated by the RANSAC algorithm. In contrast, our method uses the same adaptive voxel map obtained in Sec. III-B. We calculate the angle between their containing plane normals for every two adjacent voxels. If this angle exceeds a threshold, the intersection line of these two planes is extracted as the depth-continuous edge, as shown in Fig. 5. We choose to implement the Canny algorithm for image edge features to detect and extract.

Suppose ${}^G\mathbf{p}_i$ represents the i-th point from a LiDAR edge feature extracted above in global frame. With pin-hole camera and its distortion model, ${}^G\mathbf{p}_i$ is projected onto the image taken by camera C_l at t_j, i.e., $\mathbf{I}_{l,j}$ by

$${}^{\mathbf{I}_{l,j}}\mathbf{p}_i = \mathbf{f}\left(\boldsymbol{\pi}\left({}^{C_l}_{L_0}\mathbf{T}\left({}^G_{L_0}\mathbf{T}_{t_j}\right)^{-1}{}^G\mathbf{p}_i\right)\right), \tag{7}$$

where $\mathbf{f}(\cdot)$ is the camera distortion model and $\boldsymbol{\pi}(\cdot)$ is the projection model. Let $\mathcal{I}_i$ represent the set of images that capture the point ${}^G\mathbf{p}_i$, i.e., $\mathcal{I}_i = \{\mathbf{I}_{l,j}\}$. For each ${}^{\mathbf{I}_{l,j}}\mathbf{p}_i$, the κ

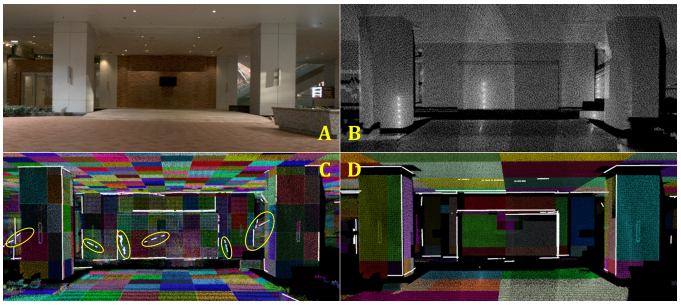


Fig. 5: Depth-continuous LiDAR edge feature extraction comparison. A) Real-world image. B) Raw point cloud of this scene. C) Edges extracted using method in [28] where the yellow circles indicate the false estimations. D) Edges extracted with adaptive voxelization.

nearest image edge feature points $\mathbf{q}_k$ on $\mathbf{I}_{l,j}$ are searched. The normal vector $\mathbf{n}_{i,l,j}$ of the edge formed by these κ points is thus the eigenvector corresponding to the minimum eigenvalue of $\mathbf{A}_{i,l,j}$ that

$$\mathbf{A}_{i,l,j} = \sum_{k=1}^{\kappa} (\mathbf{q}_k - \mathbf{q}_{i,l,j})(\mathbf{q}_k - \mathbf{q}_{i,l,j})^T, \mathbf{q}_{i,l,j} = \frac{1}{\kappa} \sum_{k=1}^{\kappa} \mathbf{q}_k. \quad (8)$$

The residual originated from this LiDAR camera correspondence is defined as

$$\mathbf{r}_{i,l,j} = \mathbf{n}_{i,l,j}^T \left({}^{\mathbf{I}_{l,j}}\mathbf{p}_i - \mathbf{q}_{i,l,j} \right). \quad (9)$$

Collecting all such correspondences, the extrinsic $\mathcal{E}_C$ calibration problem could be formulated as

$$\mathcal{E}_C^* = \arg\min_{\mathcal{E}_C} \sum_i \sum_{\mathbf{I}_{l,j} \in \mathcal{I}_i} \left(\mathbf{n}_{i,l,j}^T \left({}^{\mathbf{I}_{l,j}}\mathbf{p}_i - \mathbf{q}_{i,l,j} \right) \right). \quad (10)$$

Inspecting the residual in (9), we find the ${}^{\mathbf{I}_{l,j}}\mathbf{p}_i$ is dependent on LiDAR poses ${}^{G}_{L_0}\mathbf{T}_{t_j}$. This is due to the reason that LiDARs may have FoV overlap with cameras at different times (as in Fig. 2). Since ${}^{G}_{L_0}\mathbf{T}_{t_j} \in \mathcal{S}$ has been well estimated from Sec. III-C, we keep them fixed in this step. Moreover, the $\mathbf{n}_{i,l,j}$ and $\mathbf{q}_{i,l,j}$ are also implicitly dependent on $\mathcal{E}_C$, since both $\mathbf{n}_{i,l,j}$ and $\mathbf{q}_{i,l,j}$ are related with nearest neighbor search. The complete derivative of (10) to the variable $\mathcal{E}_C$ would be too complicated. In this paper, to simplify the optimization problem, we ignore the influence of camera extrinsic on $\mathbf{n}_{i,l,j}$ and $\mathbf{q}_{i,l,j}$. This strategy works well in practice as detailed in Sec. IV-B.

The non-linear optimization (10) is solved with LM method by approximating the residuals with their first order derivatives (11). The optimal $\mathcal{E}_C^*$ is then obtained by iteratively solving (11) and updating $\delta\mathbf{x}$ to $\mathbf{x}$ using the $\boxplus$ operation [31].

$$\delta\mathbf{x} = -\left(\mathbf{J}^T\mathbf{J} + \mu\mathbf{I}\right)^{-1}\mathbf{J}^T\mathbf{r}, \quad (11)$$

where

$$\begin{aligned} \delta\mathbf{x} &= \begin{bmatrix} \cdots & {}^{C_l}_{L_0}\boldsymbol{\phi}^T & \delta\, {}^{C_l}_{L_0}\mathbf{t}^T & \cdots \end{bmatrix}^T \in \mathbb{R}^{6h} \\ \mathbf{x} &= \begin{bmatrix} \cdots & {}^{C_l}_{L_0}\mathbf{R} & {}^{C_l}_{L_0}\mathbf{t} & \cdots \end{bmatrix} \\ \mathbf{J} &= \begin{bmatrix} \cdots & \mathbf{J}_p^T & \cdots \end{bmatrix}^T, \mathbf{r} = \begin{bmatrix} \cdots & \mathbf{r}_p & \cdots \end{bmatrix}^T, \end{aligned}$$

with $\mathbf{J}_p$ and $\mathbf{r}_p$ being the sum of $\mathbf{J}_{i,l,j}$ and $\mathbf{r}_{i,l,j}$ when $l = p$:

$$\begin{aligned} \mathbf{J}_{i,l,j} &= \mathbf{n}_{i,l,j}^T \frac{\partial \mathbf{f}(\mathbf{p})}{\partial \mathbf{p}} \frac{\partial \boldsymbol{\pi}(\mathbf{P})}{\partial \mathbf{P}} \begin{bmatrix} -{}^{C_l}_{L_0}\mathbf{R}\left({}^{L_0}\mathbf{p}_i\right)^{\wedge} & \mathbf{I} \end{bmatrix} \in \mathbb{R}^{1\times 6} \\ {}^{L_0}\mathbf{p}_i &= \left({}^{G}_{L_0}\mathbf{T}_{t_j}\right)^{-1} {}^{G}\mathbf{p}_i. \end{aligned} \quad (12)$$

E. Calibration Pipeline

The workflow of our proposed multi-sensor calibration is illustrated in Fig. 6. At the beginning of the calibration, the base LiDAR's raw point cloud is processed by a LOAM algorithm [3] to obtain the initial base LiDAR trajectory $\mathcal{S}$. Then, the raw point cloud of all LiDARs are segmented by time into point cloud patches, i.e., $\mathcal{P}_{L_i,t_j}, L_i \in \mathcal{L}, t_j \in \mathcal{T}$ that is collected under the pose ${}^{G}_{L_i}\mathbf{T}_{t_j}$.

In multi-LiDAR extrinsic calibration, the base LiDAR poses $\mathcal{S}$ are first optimized using the base LiDAR's point cloud patches $\mathcal{P}_{L_0,t_j}$. It is noticed that only $\mathcal{S}$ is involved and optimized in (3). Then the extrinsic $\mathcal{E}_L$ are calibrated by aligning the point cloud from the LiDAR to be calibrated with those from the base LiDAR. In this stage's problem formulation (3), $\mathcal{S}$ is fixed at the optimized values from the previous stage, and only $\mathcal{E}_L$ is optimized. Finally, both $\mathcal{S}$ and $\mathcal{E}_L$ are jointly optimized using the entire point cloud patches. In each iteration of the optimization (over $\mathcal{S}$, $\mathcal{E}_L$, or both), the adaptive voxelization (as described in Sec. III-B) is performed with the current value of $\mathcal{S}$ and $\mathcal{E}_L$. Moreover, the Hessian matrix $\mathbf{H}$ has a computation complexity of $O(N^2)$, where N is the number of points. In practice, to reduce this computational complexity, we down-sample the number of points scanned from the same LiDAR to 4 in each voxel. Such a process would lower the time complexity of the proposed algorithm to $O(N_{voxel})$, where N_{voxel} is the total number of adaptive voxels. In Sec. IV-A1 experiment (2), $N_{voxel} \approx 9 \times 10^3$ which is greatly smaller than the total number of raw LiDAR points in this scene, i.e., $N_{points} \approx 4 \times 10^7$.

In multi-LiDAR-camera extrinsic calibration, the adaptive voxel map obtained with the $\mathcal{S}^*$ and $\mathcal{E}_L^*$ in the previous step is used to extract the depth-continuous edges (Sec. III-D). Then those three-dimension edges are back-projected onto each image using the extrinsic parameter $\mathcal{E}_C$ and are matched with two-dimension Canny edges extracted from the image. By minimizing the residuals defined by these two edges, we iteratively solve for the optimal $\mathcal{E}_C^*$ with the Ceres Solver[2].

IV. Experiments and Results

To test the proposed algorithm, we customized a remotely operated vehicle platform[3] (see Fig. 7) with one Livox AVIA LiDAR[4] (with 70.4 degrees of FoV, see L_3 in Fig. 7), one Livox MID-100 LiDAR[5] (which has three internal MID-40 LiDARs, each has 38.4 degrees of FoV with only 8.4 degrees overlap between adjacent MID-40 units, see L_0, L_1, and L_2 in Fig. 7) and two MV-CA013-21UC[6] cameras (with 82.9 degrees

[2]http://ceres-solver.org/
[3]https://www.agilex.ai/product/3?lang=en-us
[4]https://www.livoxtech.com/avia
[5]https://www.livoxtech.com/mid-40-and-mid-100
[6]https://www.rmaelectronics.com/hikrobot-mv-ca013-21uc/

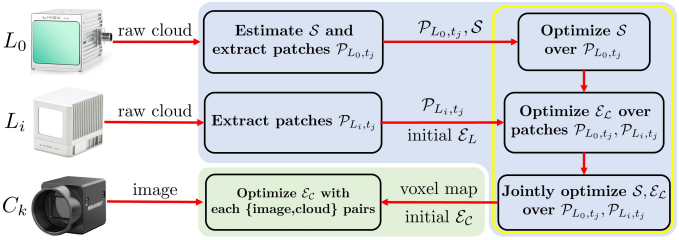
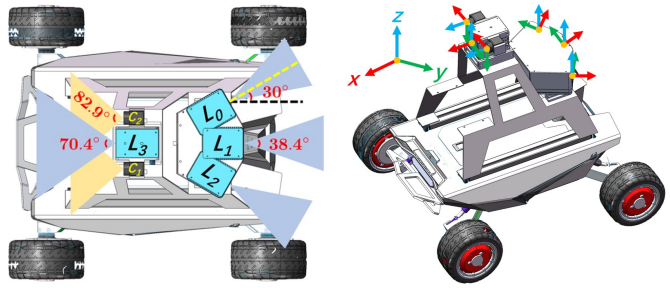


Fig. 6: The workflow of our proposed method: multi-LiDAR extrinsic calibration (light blue region) and LiDAR-camera extrinsic calibration (light green region). The adaptive voxelization takes effect in the steps surrounded by the yellow rectangle.



(a) (b)

Fig. 7: Our customized multi-sensor vehicle platform. Left: the FoV coverage of each sensor with their FoV specs. Right: the orientation of each sensor is denoted in the right-handed coordinate system.

(a) Scene-1 (b) Scene-2

Fig. 8: Our experiment test scenes.

of FoV each, see C_1 and C_2 in Fig. 7). The extrinsic parameters of the three MID-40 inside the MID-100 have been calibrated by the manufacturer and could be used as the ground truth for the calibration evaluation.

We have verified our proposed work with the data collected in two random test scenes on our campus, as shown in Fig. 8. Scene-1 is a square in front of the main library building with moving pedestrians, and scene-2 is an open area near a garden. The calibration data is collected in both scenes by rotating the sensor suite slightly for more than 360° degrees and keeping this platform still every few degrees. Keeping the robot platform still during data collection enables us to acquire a dense enough point cloud from each LiDAR at each pose and also eliminates the problem caused by motion distortion and time synchronization. The timestamps $\mathcal{T}$ are manually selected so that only the point cloud and image data are chosen when the robot platform is still. During sensor suite rotation, a dedicated LiDAR inertial odometry and mapping (LOAM) algorithm loam-livox [3] is called to estimate a rough LiDAR pose trajectory $\mathcal{S}_{\text{init}}$, which serves as the initial pose in our factor graph optimization. Moreover, to obtain an initial estimate of the extrinsic $\mathcal{E}_{L_{\text{init}}}$, we collected another data, the initialization data, which is collected in scene-1 with an '8'-figure path. Similarly, loam-livox is used to estimate each LiDAR's trajectory, based on which the extrinsic is solved by a standard Hand-eye calibration. All experiments are conducted on a desktop computer with an i7-9700K processor and 32GB RAM.

A. Multiple LiDAR Calibration

1) Calibration Precision: In this section, we compare our algorithm with the motion-based method [13] and the ICP-based method [5] using the MID-100 LiDAR and the AVIA LiDAR. Both [5, 13] are targetless, offline, and utilize the motion information to calibrate the extrinsic parameter without requiring significant LiDAR FoV overlap as our method in this work. The method in [13] is essentially a variant of hand-eye calibration, with further consideration of pose uncertainty. To compare the performance of [13] with our method on the calibration data collected above, we run loam-livox [3] to obtain the point cloud of each scan and its poses (odometry). Since the uncertainty information within each LiDAR's motion estimation is also examined in [13] to achieve the optimal performance, we manually calculate the measurement noise in each LiDAR odometry estimation. This is completed by calculating the covariance between the consecutive scans using the above-obtained point cloud and taking the odometry as the initial guess. Then, the odometry and its uncertainty information of each LiDAR are fed to and processed by [13]. The other method under comparison is our previous work [5], which used the same rotation to create an overlap for small FoV LiDARs and an ICP-based factor graph optimization to estimate the extrinsic parameter and LiDAR pose. To make a fair comparison, we feed the same initial pose trajectory $\mathcal{S}_{\text{init}}$ and extrinsic $\mathcal{E}_{L_{\text{init}}}$ obtained above to both [5] and this work before the full calibration on the calibration data collected above.

The experiment is divided into two parts: (1) MID-100 LiDAR self calibration: the middle MID-40 is chosen as the base LiDAR to calibrate the extrinsic $\mathcal{E}_L$ of the other two MID-40s, i.e., ${}^{L_1}_{L_0}\mathbf{T}, {}^{L_1}_{L_2}\mathbf{T}$ (see Fig. 7). To evaluate the calibration precision, we compare the optimized ${}^{L_1}_{L_0}\mathbf{T}^*, {}^{L_1}_{L_2}\mathbf{T}^*$ with the ground-true values obtained from the manufacturer. To further enrich the calibration data collected above, we adopt the calibration data of MID100 in two extra scenes used in [5]. This leads to four test scenes in total (two scenes in this work and two scenes from [5]), each has two LiDAR extrinsic ground-truth (i.e., ${}^{L_1}_{L_0}\mathbf{T}, {}^{L_1}_{L_2}\mathbf{T}$) for evaluation. Consequently, we have eight independent real-world calibration data for MID40 in the evaluation. (2) AVIA and MID-100 LiDAR: the AVIA LiDAR is chosen as the base LiDAR to calibrate the extrinsic $\mathcal{E}_L$ between AVIA and each MID-40s, i.e., ${}^{L_3}_{L_0}\mathbf{T}, {}^{L_3}_{L_1}\mathbf{T}$ and ${}^{L_3}_{L_2}\mathbf{T}$ (see Fig. 7). To evaluate the calibration precision, we calculate the ${}^{L_1}_{L_0}\mathbf{T}^* = ({}^{L_3}_{L_1}\mathbf{T}^*)^{-1}\,{}^{L_3}_{L_0}\mathbf{T}^*$ and ${}^{L_1}_{L_2}\mathbf{T}^* = ({}^{L_3}_{L_1}\mathbf{T}^*)^{-1}\,{}^{L_3}_{L_2}\mathbf{T}^*$

using the above results and compare them with the ground-true values obtained from the manufacturer. This experiment is conducted with calibration data collected in the two scenes from this work only since the previous work [5] did not have an AVIA LiDAR. The two scenes and two LiDAR extrinsic ground-truths lead to four independent real-world calibration data in the evaluation. As a result, we have twelve independent calibration data and two LiDAR types (i.e., Livox MID-40 and AVIA) with completely different scanning patterns, point densities, and FoVs.

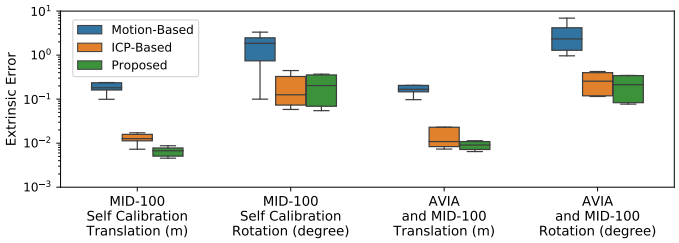


Fig. 9: Extrinsic calibration results of the motion-based [13], ICP-based [5] and our proposed methods in two experiment setups (with few or no FoV overlap between sensors).

The comparison results of our method, the ICP-based previous work [5], and motion-based [13], are shown in Fig. 9. Both the ICP-based previous method [5] and our proposed method in this work outperform the motion-based method [13] in calibration precision. This is due to the reason that the precision of the motion-based method [13] relies heavily on the extent of excitation in the sensor motion. Since the sensors' movements in all test scenes are mainly constrained on the ground, as most ground vehicles do, the excitation in the z-dimension of the extrinsic is much less assured. Besides the excitation, the quality of the LiDAR pose estimation is also a crucial factor affecting the performance of hand-eye calibration in [13]. For the Livox MID-40 and AVIA LiDARs that have very small FoV, the odometry is significantly deteriorated due to the reduction of feature points in a frame [3]. In contrast, our current method and previous one [5] use a rotation motion to create a large FoV overlap, where the same feature points are observed from multiple LiDARs from multiple poses. Exploiting the constraints imposed by these co-visible features considerably increase the calibration accuracy, almost irrelevant to excitation in motion or odometry accuracy. Moreover, when compared to the ICP-based previous method [5], the performance of our method in this work has considerably improved the calibration precision (in terms of average calibration error) and robustness (in terms of the variance in the calibration error), especially in translation. These results are credited to the more accurate feature matching correspondences and solutions brought by the adaptive voxelization and second-order optimization. Moreover, it is shown that our proposed method is less affected by the distinct characteristics (point cloud density, FoV, scan pattern, etc.) introduced by different types of LiDARs.

2) Convergence and Computation Time Comparison: The main benefit of our method in this work, when compared to our previous work [5], is the computation time, which serves as one of the main motivations for this work. In this section, we demonstrate that the proposed algorithm converges much faster than the ICP-based method [5] in terms of both iteration times and computation time while remaining accurate. Since the motion-based method [13] directly generates the extrinsic result, the convergence comparison with this method is not applicable. To ensure the data diversity in the comparison, we use all the calibration data of MID-100 in the previous section collected in four scenes (two scenes collected in this work and two extra scenes from [5]). We choose the middle MID-40 as the base LiDAR to calibrate the adjacent two LiDARs. To further examine the convergence robustness to initial values of the extrinsic, we perform 100 independent trials. In each trial, the initial extrinsic $\mathcal{E}_L$ is randomly perturbed (± 10 degrees for ${}^{L_1}_{L_i}\mathbf{R}$ and ± 0.2m for ${}^{L_1}_{L_i}\mathbf{t}$) from the manufacturer's calibrated values.

The extrinsic rotation and translation errors of both methods versus iteration numbers are plotted in Fig. 10, where the calibration error is calculated from the manufacturer values. Each box in this box-plot contains 800 calibration results from 100 trials and each trail includes the results of two LiDAR extrinsic (i.e., $\{L_0, L_1\}$ and $\{L_1, L_2\}$ see Fig. 7) overall four scenes. As can be seen, our method converges much quicker than the previous method [5], especially in translation. The entire algorithm converges within 5 iterations, even in the worst-case scenario, while that of the previous work converges much slower. After 15 iterations, the convergence in the translation of [5] is slowed down even more. The slow convergence of [5] is attributed to the pairwise ICP registration process, where only one pose or extrinsic can be estimated at a time. In contrast, our method optimizes all the poses and extrinsic concurrently, leading to a more complete point registration in each iteration and hence fewer iterations to converge. The results in Fig. 10 also show how the translation error of the ICP-based method [5] converges to a larger value than that of our method, which is in agreement with the results in the previous section comparing the calibration prevision.

Besides the convergence rate in terms of iteration numbers, our method also achieves a much lower computation time than [5] at each individual iteration. The averaged computation time per trial per iteration of both methods is summarized in Table I. Within each step of the calibration (see Fig. 6), we further dig into and calculate the time cost in feature correspondence matching (Match) and non-linear cost function solving (Solve). It is seen our proposed work significantly saves the computation time in the above two processes due to the implementation of adaptive voxelization (Sec. III-B) and second-order optimization (Sec. III-C). In each iteration, a voxel map is created only once for our proposed work, and for any feature point, its corresponding feature points are simply the points within the same voxel. Whereas in [5], a unique k-d tree data structure needs to be created and searched each time for every feature point during the feature correspondence matching process. In non-linear cost function solving, the Jacobian and Hessian matrix w.r.t. the optimization variables ($\mathcal{S}$ and $\mathcal{E}_L$) are exactly derived in our proposed work, leading to a faster and more accurate solution. In contrast, in [5], only

TABLE I: AVERAGE COMPUTATION TIME PER ITERATION ON MULTI-LIDAR CALIBRATION

	Pose Optimization			Extrinsic Optimization			Global Optimization		
	Match	Solve	Total	Match	Solve	Total	Match	Solve	Total
ICP-Based [5]	4.0220s	1.1057s	5.1613s	4.4635s	1.6041s	6.1045s	11.0557s	3.3616s	14.9829s
Proposed	**0.1040s**	**0.0328s**	**0.2288s**	**0.3419s**	**0.0443s**	**0.5771s**	**2.2940s**	**0.4687s**	**3.0887s**

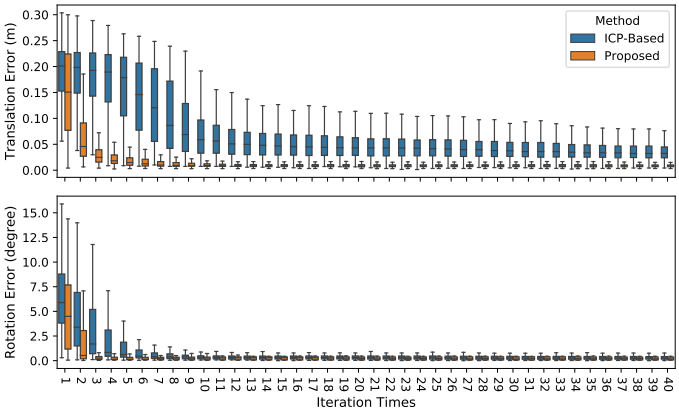


Fig. 10: Convergence comparison of ICP-based [5] method and our proposed work. Each box contains the results from 100 trials. The mean and standard deviation of the initial extrinsic errors are 0.1846m and 0.0562m for translation and 9.4038 degrees and 2.9094 degrees for rotation, respectively.

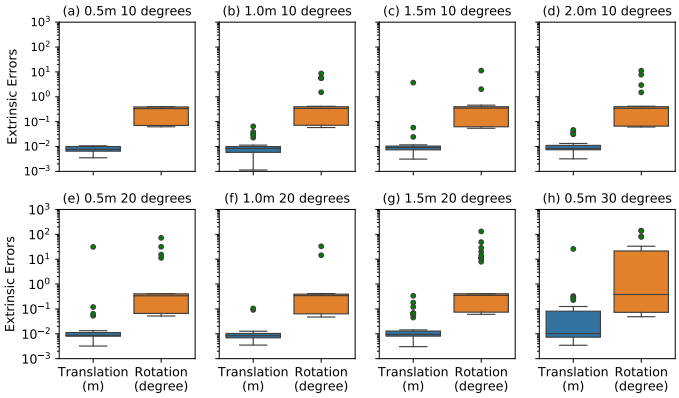


Fig. 11: The distribution of calibration errors of our proposed method under multiple disturbed initial values. Each box contains the results of 100 trials. It is seen under the setups of (a)-(g), most of the initial values could be converged with only a few outliers. The initial disturbance exceeds our convergence tolerance under the setup of (h).

the Jacobian of the residual w.r.t. one LiDAR is considered, causing inaccurate Hessian matrix computation. This analysis is also verified in Fig. 10 that the proposed work makes both the extrinsic translation and rotation errors quickly converge to the appropriate values. The reduction of iteration numbers (more than 3 times) and computation time per iteration (more than 5 times) shorten the calibration time of the previous method [5] by more than 15 times.

3) Robustness Test: To quantify the convergence basin of our proposed method, we test our method when the initial extrinsic $\mathcal{E}_{L_{\text{init}}}$ is perturbed by noises at different levels. We use the MID-100 dataset collected from two test scenes in this work and choose the middle MID-40 as the base LiDAR to calibrate the adjacent two LiDARs. The calibration error is calculated similarly as in Sec. IV-A1. In each configuration, the initial extrinsic is randomly perturbed 100 times (e.g., 0.5m 10 degrees means ± 10 degrees for ${}^{L_1}_{L_i}\mathbf{R}$ and ± 0.5m for ${}^{L_1}_{L_i}\mathbf{t}$) from the manufacturer's calibrated values.

The calibration errors are illustrated in Fig. 11. Each box in this box-plot contains 400 results from 100 trials and each trail contains 4 results from two LiDAR pairs (i.e., $\{L_0, L_1\}$ and $\{L_1, L_2\}$ see Fig. 7) in two test scenes. It is shown that given the rotation noise of 10 degrees, the proposed method could ideally converge when the translation noise is 0.5m and mostly converge when the translation noise is under 2.0m. When the rotation noise is 20 degrees, our proposed method could generally converge when the translation noise is under 1.0m. Such a high noise level is sufficient to cover the faulty scenarios in the real world caused by manufacturing mounting errors or severe vibration during usage.

B. Multiple LiDAR Camera Calibration

1) LiDAR-Camera with FoV Overlap: In this section, we verify the effectiveness of our method in calibrating the extrinsic among LiDARs and cameras when they have FoV overlap. We select the AVIA as the base LiDAR and calibrate its extrinsic w.r.t. two cameras (see Fig. 7). The extrinsic $\mathcal{E}_C$ is initialized by adding disturbance to the values measured from the CAD model. We perform 50 independent trials with the calibration data collected in scene-2, that in each trial the initial extrinsic is randomly perturbed (± 5 degrees for ${}^{C_k}_{L_3}\mathbf{R}$ and ± 0.1m for ${}^{C_k}_{L_3}\mathbf{t}$) from the CAD model's measurements. We calibrate the extrinsic of each camera individually (i.e., ${}^{C_1}_{L_3}\mathbf{T}^*$, ${}^{C_2}_{L_3}\mathbf{T}^*$), then we calculate the ${}^{C_1}_{C_2}\mathbf{T}^* = {}^{C_1}_{L_3}\mathbf{T}^* ({}^{C_2}_{L_3}\mathbf{T}^*)^{-1}$ and compare it with that directly calibrated by the standard chessboard method serving the ground-truth.

We compare our method with three targetless methods that work for LiDAR-cameras with FoV overlaps: RANSAC-based [28], motion-based [13], and mutual information-based [10]. Our previous work [28] is the latest state-of-the-art specifically designed for high-resolution LiDARs, which is most similar to this work. [10, 13] are state-of-the-art methods originally designed for 360° LiDARs. In [13], each point from a LiDAR scan is projected onto and matched with adjacent two images, and the extrinsic is optimized by minimizing the total points' color difference (i.e., the appearance) across adjacent images. In [10], the extrinsic is optimized by maximizing the mutual information between LiDAR intensity images and camera images.

The calibration results are illustrated in Fig. 12 and Fig. 13. It is seen that both [28] and our work are an order of magnitude better than [10, 13] in both rotation and translation. This is due to the reason that the three-dimensional LiDAR edge feature is

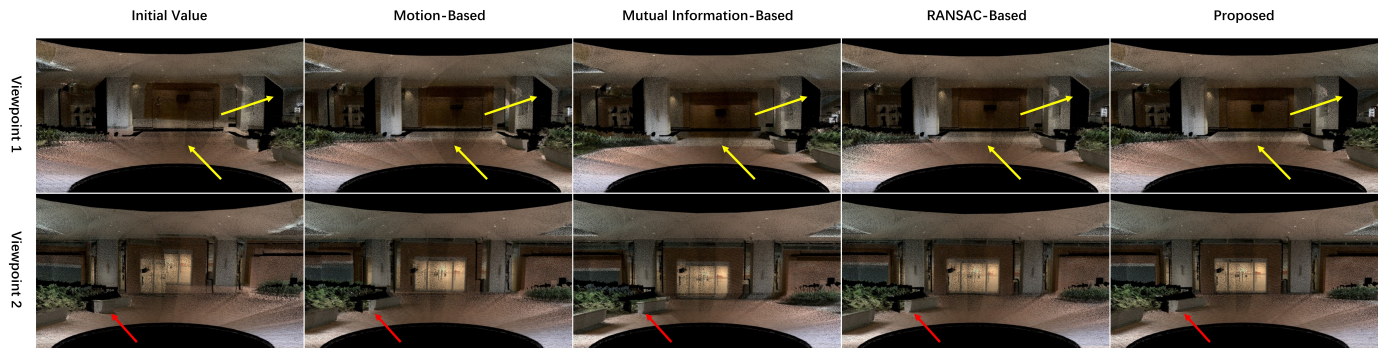


Fig. 12: Point cloud colorized using the extrinsic calibrated by motion-based [13], mutual information-based [10], RANSAC-based [28] and our proposed methods. Each row represents a viewpoint in scene-2. The detailed difference between these methods is pointed out by arrows, e.g., miss-colorization on pillars and benches (zoomed view is recommended).

TABLE II: COMPUTATION TIME ON MULTIPLE LIDAR-CAMERA CALIBRATION

	LiDAR Feature Extraction			Extrinsic Optimization Per Iteration		
	Plane Estimation	Edge Estimation	Total	LiDAR-Camera Feature Matching	Solving Cost Function	Total
Motion-Based [13]	-	-	-	1.7690s	3.5780s	10.8457s
Mutual Information-Based [10]	-	-	-	3.6042s	0.6101s	4.7520s
RANSAC-Based [28]	9.6186s	27.6738s	37.4523s	0.8609s	0.5552s	1.4548s
Proposed	**3.8054s**	**2.4494s**	**6.2892s**	**0.5424s**	**0.2510s**	**0.8278s**

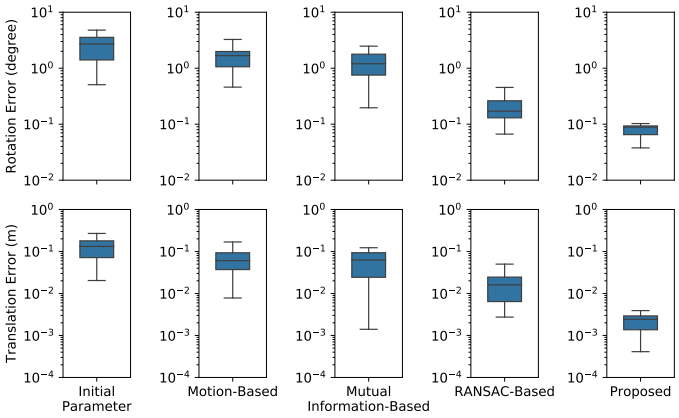


Fig. 13: Extrinsic calibration results of motion-based [13], mutual information-based [10], RANSAC-based [28] and our proposed methods. Each box-plot illustrates the results of 50 trials using the data collected in scene-2. The mean and standard deviation of the initial rotation errors are 2.4768 and 1.2390 degrees. The mean and standard deviation of the initial translation errors are 0.1308m and 0.0682m, respectively.

more reliable than the point cloud intensity information used in [10] and the color appearance in [13], especially in the structured test scene with large planes and long edges. This difference in calibration precision could also be visualized in Fig. 12. It is also interesting to see that our work outperforms the RANSAC-based method [28] quite significantly, although they share many similarities in the overall calibration pipeline. This is due to the reason that our plane estimation method uses adaptive voxels to capture planes (and hence edges) at a finer level with higher quality than that of the fixed-size voxels used in [28]. The more accurate plane and edge estimation in our method (see Fig. 5) eventually leads to higher calibration precision and robustness.

Besides precision, our method also consumes much less computation time in each step and optimization iteration, as summarized in Table II. We first compare our proposed method with [10]. Though no prior feature extraction process is needed in [10], the calculation of the mutual information consumes significant time due to the process of all LiDAR points and image pixels. This phenomenon also appears in the motion-based method [13], as each point from a LiDAR scan is projected onto and matched with two adjacent images. The averaged raw LiDAR points in each LiDAR scan is $N_{raw} \approx 8 \times 10^5$ while the total number of extracted LiDAR edge feature points is $N_{feature} \approx 5 \times 10^4$, and this discovery is in accordance with the recorded time consumption in Table II.

We then compare the detailed time consumption with the RANSAC-based method [28]. In [28], the LiDAR plane feature is extracted by first cutting the point cloud into fixed-size voxels and second analyzing the points distribution in each voxel using RANSAC. In comparison, our proposed work cuts the point cloud into voxels with sizes adapted to the environment and extracts the plane feature by analyzing eigenvalues in each voxel (see Sec. III-B). This difference in operation also leads to distinct total voxel numbers, e.g., $N_{fixed} = 9216$ versus $N_{adaptive} = 1369$ in this scene, which further causes large computation time divergence in LiDAR edge feature estimation as in Table II. Moreover, method in [28] are also prone to false estimations (see Fig. 5) which makes the feature matching and cost function solving processes less reliable (see Fig. 13) and more time consuming.

2) LiDAR-Camera without FoV Overlap: In this section, we demonstrate that the proposed method could also calibrate the extrinsic $\mathcal{E}_C$ between LiDAR and cameras without FoV overlap. We choose the middle MID-40 of the MID-100 as the

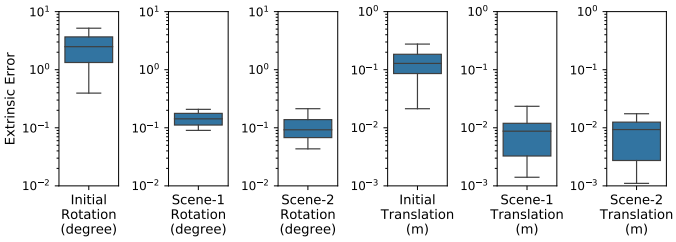


Fig. 14: Extrinsic calibration results of MID-100 and opposite pointing cameras in two test scenes. Each box-plot illustrates the results of 50 trials. The mean and standard deviation of the initial rotation errors are 2.5408 and 1.3645 degrees. The mean and standard deviation of the initial translation errors are 0.1376m and 0.0663m, respectively.

base LiDAR and calibrate the extrinsic of each LiDAR-camera pairs (i.e., ${}^{C_1}_{L_1}\mathbf{T}, {}^{C_2}_{L_1}\mathbf{T}$, see Fig. 7). The initial extrinsic $\mathcal{E}_C$ are calculated by adding disturbance to the values measured from the CAD model. We perform 50 independent trials with the data collected in both scenes from this work, that in each trial we randomly perturb the initial extrinsic value ($\pm$5 degrees for ${}^{C_k}_{L_1}\mathbf{R}$ and $\pm$0.1m for ${}^{C_k}_{L_1}\mathbf{t}$) from the CAD's measurements. Then we calculate the ${}^{C_1}_{C_2}\mathbf{T}^* = {}^{C_1}_{L_1}\mathbf{T}^* ({}^{C_2}_{L_1}\mathbf{T}^*)^{-1}$ and compare it with that obtained by the standard chessboard method. The calibration results and the corresponding colorized point cloud are illustrated in Fig. 14 and Fig. 15.

It is seen that the general extrinsic calibration performance between MID-40 and cameras is less competitive than that between AVIA and cameras. This might be due to the reason that AVIA has larger FoV coverage (70.4 versus 38.4 degrees) and thus larger point cloud density (6 laser beams versus 1 laser beam) than MID-40, which will provide more edge correspondences in all directions. The performance of MID-40 and cameras extrinsic calibration in scene-2 is also slightly better than scene-1. This is probably due to the reason that the extracted LiDAR edges mismatch with and are trapped into the image edges that largely existed on the ground of scene-1.

V. Conclusion

In this paper, we proposed a targetless extrinsic calibration method for multiple small FoV LiDARs and cameras. Unlike existing ICP-based methods, which rely on the k-d tree in LiDAR feature correspondences matching, our proposed work implemented an adaptive voxel map to store and search for the feature points to save the calibration time. We also formulated the multiple LiDAR extrinsic calibration into a Bundle Adjustment problem and derived the cost function up to second order to boost the solving process. In LiDAR-camera extrinsic calibration, we reused the above constructed adaptive voxel map to shorten LiDAR plane feature extraction and edge feature estimation time. Compared with the RANSAC-based methods, our work improved both computation efficiency and accuracy. It is believed that this open-sourced work will benefit the community of autonomous navigation robots and high-resolution mapping, especially when the sensor setups include small FoV LiDARs with few or even no FoV overlap.

Though no external calibration target is required, it is noted that our proposed work relies on the existence of natural

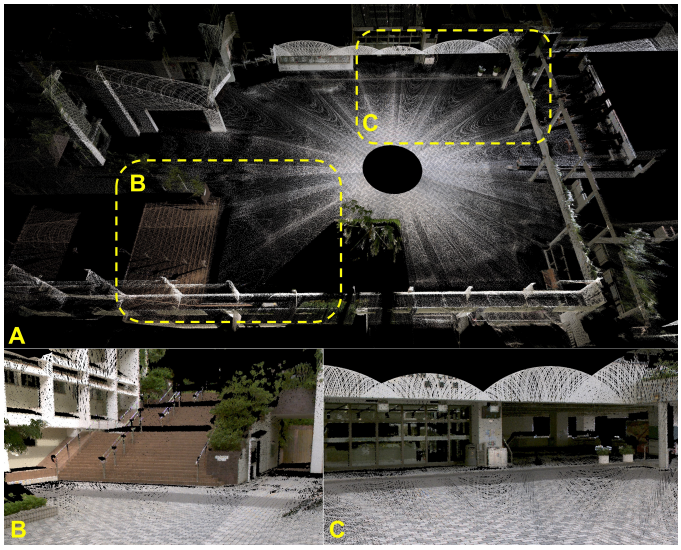


Fig. 15: Colorized point cloud of MID-100 LiDAR and the opposite pointing camera in scene-1. The left camera's images are used to color the point cloud. The brightness of the building wall is due to the reflection of the sunlight. A) Bird-eye's view. B) Details of the stairs, fence, and ground tiles. C) Entrance of the library. The details of flowerpots are clearly shown.

plane features (structured building walls, ground, etc.) in the calibration environment. The precision and robustness of the extrinsic calibration among LiDARs and cameras are based on the correct extraction of LiDAR plane features. Thus, our proposed work is less reliable in unstructured scenes (e.g., country field, mountain valley, or forest). Given appropriate calibration scenes with sufficient plane features, it is believed our proposed work could produce both fast and accurate calibration results. In our future work, we wish to take the sensor measurement's noise model and camera intrinsic parameters into consideration.

Acknowledgment

The authors gratefully acknowledge Livox Technology and AgileX Robotics for their product support. The authors would like to appreciate Zheng Liu for the insightful discussions.

Appendix A

A. Elimination of Feature Parameters From Cost Function

The original optimization dimension in (13)) is too high due to the dependence on the planar parameters $\boldsymbol{\pi} = (\mathbf{n}_l, \mathbf{q}_l)$.

$$\arg \min_{\mathcal{S}, \mathcal{E}_L, \mathbf{n}_l, \mathbf{q}_l} \sum_l \underbrace{\left(\frac{1}{N_l} \sum_{k=1}^{N_l} \left(\mathbf{n}_l^T \left({}^G\mathbf{p}_k - \mathbf{q}_l \right) \right)^2 \right)}_{l\text{-th factor}}. \tag{13}$$

It is noted that the planar parameters $(\mathbf{n}_l, \mathbf{q}_l)$ are independent for different planes and we can optimize over them first, i.e.,

$$\arg \min_{\mathcal{S}, \mathcal{E}_L} \sum_l \left(\min_{\mathbf{n}_l, \mathbf{q}_l} \frac{1}{N_l} \sum_{k=1}^{N_l} \left(\mathbf{n}_l^T \left({}^G\mathbf{p}_k - \mathbf{q}_l \right) \right)^2 \right). \tag{14}$$

The inner optimization over $(\mathbf{n}_l, \mathbf{q}_l)$ in (14) could be further performed on $\mathbf{q}_l$ first and on $\mathbf{n}_l$ then, i.e.,

$$\arg\min_{\mathbf{n}_l}\left(\min_{\mathbf{q}_l}\frac{1}{N_l}\sum_{k=1}^{N_l}\left(\mathbf{n}_l^T\left({}^G\mathbf{p}_k-\mathbf{q}_l\right)\right)^2\right). \tag{15}$$

As can be seen, the cost function in (15) is quadratic w.r.t. $\mathbf{q}_l$. Hence the inner optimization can be solved analytically by setting the derivatives to zeros, i.e.,

$$\mathbf{n}_l\mathbf{n}_l^T\left(\frac{1}{N_l}\sum_{k=1}^{N_l}\left({}^G\mathbf{p}_k-\mathbf{q}_l\right)\right)=\mathbf{0}. \tag{16}$$

It is seen that the solution to (16) is not unique as long as $\sum_{k=1}^{N_l}\left({}^G\mathbf{p}_k-\mathbf{q}_l\right)$ is perpendicular to $\mathbf{n}_l$, which allows $\mathbf{q}_l$ to move freely along any direction perpendicular to $\mathbf{n}_l$. Since this free movement of $\mathbf{q}_l$ does not change the plane parameterized by it, nor affect the cost function in (15), any solution of $\mathbf{q}_l$ satisfying (16) would be an optimal solution to the inner optimization problem of (15). One such solution could be

$$\mathbf{q}_l^*=\frac{1}{N_l}\sum_{k=1}^{N_l}{}^G\mathbf{p}_k. \tag{17}$$

Substituting the optimal solution of $\mathbf{q}_l$ in (17) back to (15) leads to

$$\arg\min_{\|\mathbf{n}_l\|=1}\mathbf{n}_l^T\underbrace{\left(\frac{1}{N_l}\sum_{k=1}^{N_l}{}^G\mathbf{p}_k\,{}^G\mathbf{p}_k^T-\mathbf{q}_l^*\mathbf{q}_l^{*T}\right)}_{\mathbf{A}_l}\mathbf{n}_l. \tag{18}$$

Again, this optimization problem has the well-known analytical optimal solution $\mathbf{n}_l^*$, which is the eigenvector corresponding to the smallest eigenvalue λ_3 of the matrix $\mathbf{A}_l$. As a result, substituting the optimal $\mathbf{n}_l^*$ back to (14) leads to

$$\mathcal{S}^*,\mathcal{E}_L^*=\arg\min_{\mathcal{S},\mathcal{E}_L}\sum_l\lambda_3\left(\mathbf{A}_l\right). \tag{19}$$

As can be seen, the optimization variables $(\mathbf{n}_l, \mathbf{q}_l)$ are analytically solved before the optimization, which significantly reduces the optimization dimension.

B. Second-Order Derivation of Cost Function

The optimization in (19) is nonlinear and solved iteratively. In each iteration, the cost function is approximated to the second order. More specifically, we view λ_3 as a function of all the contained points ${}^G\mathbf{p}$ which is the column vector containing each ${}^G\mathbf{p}_k\in\mathcal{P}_l$:

$${}^G\mathbf{p}=[{}^G\mathbf{p}_1^T\,{}^G\mathbf{p}_2^T\cdots{}^G\mathbf{p}_{N_l}^T]^T\in\mathbb{R}^{3N_l}.$$

The $\lambda_3({}^G\mathbf{p})$ in (19) could be approximated by

$$\lambda_3\left({}^G\mathbf{p}+\delta^G\mathbf{p}\right)\approx\lambda_3\left({}^G\mathbf{p}\right)+\mathbf{J}\cdot\delta^G\mathbf{p}+\frac{1}{2}\,\delta^G\mathbf{p}^T\cdot\mathbf{H}\cdot\delta^G\mathbf{p}, \tag{20}$$

where $\mathbf{J}$ and $\mathbf{H}$ are the first and second derivatives of $\lambda_3({}^G\mathbf{p})$ w.r.t. ${}^G\mathbf{p}$. The expression of $\mathbf{J}$ and $\mathbf{H}$ could be found in [4] and is omitted here due to space limit. Suppose the k-th point ${}^G\mathbf{p}_k$ in ${}^G\mathbf{p}$ is scanned by LiDAR L_i at time t_j, then

$$\begin{aligned}{}^G\mathbf{p}_k&={}^G_{L_i}\mathbf{T}_{t_j}\mathbf{p}_k={}^G_{L_0}\mathbf{T}_{t_j}\cdot{}^{L_0}_{L_i}\mathbf{T}\cdot\mathbf{p}_k\\&={}^G_{L_0}\mathbf{R}_{t_j}\left({}^{L_0}_{L_i}\mathbf{R}\cdot\mathbf{p}_k+{}^{L_0}_{L_i}\mathbf{t}\right)+{}^G_{L_0}\mathbf{t}_{t_j},\end{aligned} \tag{21}$$

which implies ${}^G\mathbf{p}_k$ is dependent on $\mathcal{S}$ and $\mathcal{E}_L$. To perturb ${}^G\mathbf{p}_k$, we perturb a pose $\mathbf{T}$ in its tangent plane $\delta\mathbf{T}=[\boldsymbol{\phi}^T\ \delta\mathbf{t}^T]^T\in\mathbb{R}^6$ with the $\boxplus$ as defined in [31], i.e.,

$$\begin{aligned}\mathbf{T}&=(\mathbf{R},\mathbf{t})\\\mathbf{T}\boxplus\delta\mathbf{T}&=\left(\mathbf{R}\exp\left(\boldsymbol{\phi}^\wedge\right),\mathbf{t}+\delta\mathbf{t}\right).\end{aligned} \tag{22}$$

Based on the error parameterization in (22) for both ${}^G_{L_0}\mathbf{T}_{t_j}$ and extrinsic ${}^{L_0}_{L_i}\mathbf{T}$, the perturbed point location in (21) is

$$\begin{aligned}{}^G\mathbf{p}_k+\delta^G\mathbf{p}_k={}^G_{L_0}\mathbf{R}_{t_j}\exp\left({}^G_{L_0}\boldsymbol{\phi}^\wedge_{t_j}\right)\Big({}^{L_0}_{L_i}\mathbf{R}\exp\left({}^{L_0}_{L_i}\boldsymbol{\phi}^\wedge\right)\mathbf{p}_k\\+{}^{L_0}_{L_i}\mathbf{t}+\delta^{L_0}_{L_i}\mathbf{t}\Big)+{}^G_{L_0}\mathbf{t}_{t_j}+\delta^G_{L_0}\mathbf{t}_{t_j}.\end{aligned} \tag{23}$$

Then, subtracting (21) from (23), we obtain

$$\begin{aligned}\delta^G\mathbf{p}_k\approx{}^G_{L_0}\mathbf{R}_{t_j}\left({}^{L_0}_{L_i}\mathbf{R}\mathbf{p}_k+{}^{L_0}_{L_i}\mathbf{t}\right)^\wedge{}^G_{L_0}\boldsymbol{\phi}_{t_j}+\delta^G_{L_0}\mathbf{t}_{t_j}+\\{}^G_{L_i}\mathbf{R}_{t_j}\left(\mathbf{p}_k\right)^\wedge{}^{L_0}_{L_i}\boldsymbol{\phi}+{}^G_{L_0}\mathbf{R}_{t_j}\delta^{L_0}_{L_i}\mathbf{t}\end{aligned} \tag{24}$$

and

$$\delta^G\mathbf{p}=\mathbf{D}\cdot\delta\mathbf{x}, \tag{25}$$

where

$$\delta\mathbf{x}=[\cdots{}^G_{L_0}\boldsymbol{\phi}^T_{t_j}\ \delta^G_{L_0}\mathbf{t}^T_{t_j}\cdots{}^{L_0}_{L_i}\boldsymbol{\phi}^T\ \delta^{L_0}_{L_i}\mathbf{t}^T\cdots]^T\in\mathbb{R}^{6(m+n-2)}$$

is a small perturbation of the optimization variable $\mathbf{x}$

$$\mathbf{x}=[\cdots{}^G_{L_0}\mathbf{R}_{t_j}\ \ {}^G_{L_0}\mathbf{t}_{t_j}\cdots{}^{L_0}_{L_i}\mathbf{R}\ \ {}^{L_0}_{L_i}\mathbf{t}\cdots],$$

and

$$\begin{aligned}\mathbf{D}&=\begin{bmatrix}\vdots&\vdots\\\cdots\ \mathbf{D}^{\mathcal{S}}_{k,p}&\cdots\ \mathbf{D}^{\mathcal{E}_L}_{k,q}\ \cdots\\\vdots&\vdots\end{bmatrix}\in\mathbb{R}^{3N_l\times6(m+n-2)}\\\mathbf{D}^{\mathcal{S}}_{k,p}&=\begin{cases}\Big[-{}^G_{L_0}\mathbf{R}_{t_j}\left({}^{L_0}_{L_i}\mathbf{R}\mathbf{p}_k+{}^{L_0}_{L_i}\mathbf{t}\right)^\wedge\ \mathbf{I}\Big], & \text{if } p=j\\\mathbf{0}_{3\times6}, & \text{else}\end{cases}\\\mathbf{D}^{\mathcal{E}_L}_{k,q}&=\begin{cases}\Big[-{}^G_{L_0}\mathbf{R}_{t_j}\,{}^{L_0}_{L_i}\mathbf{R}\left(\mathbf{p}_k\right)^\wedge\ \ {}^G_{L_0}\mathbf{R}_{t_j}\Big], & \text{if } q=i\\\mathbf{0}_{3\times6}, & \text{else.}\end{cases}\end{aligned} \tag{26}$$

Substituting (25) to (20) leads to

$$\begin{aligned}\lambda_3(\mathbf{x}\boxplus\delta\mathbf{x})&\approx\lambda_3(\mathbf{x})+\mathbf{J}\mathbf{D}\delta\mathbf{x}+\frac{1}{2}\delta\mathbf{x}^T\mathbf{D}^T\mathbf{H}\mathbf{D}\delta\mathbf{x}\\&=\lambda_3(\mathbf{x})+\bar{\mathbf{J}}\delta\mathbf{x}+\frac{1}{2}\delta\mathbf{x}^T\bar{\mathbf{H}}\delta\mathbf{x}.\end{aligned} \tag{27}$$

REFERENCES

[1] F. Kong, W. Xu, Y. Cai, and F. Zhang. Avoiding dynamic small obstacles with onboard sensing and computation on aerial robots. *IEEE Robotics and Automation Letters*, 6(4):7869–7876, 2021.

[2] J. Lin, X. Liu, and F. Zhang. A decentralized framework for simultaneous calibration, localization and mapping with multiple lidars. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4870–4877, 2020.

[3] J. Lin and F. Zhang. Loam-livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov. In *Proc. of The International Conference in Robotics and Automation (ICRA)*, 2020.

[4] Z. Liu and F. Zhang. Balm: Bundle adjustment for lidar mapping. *IEEE Robotics and Automation Letters*, 6(2):3184–3191, 2021.

[5] X. Liu and F. Zhang. Extrinsic calibration of multiple lidars of small fov in targetless environments. *IEEE Robotics and Automation Letters*, 6(2):2036–2043, 2021.

[6] Z. Liu, F. Zhang, and X. Hong. Low-cost retina-like robotic lidars based on incommensurable scanning. *IEEE/ASME Transactions on Mechatronics*, 27(1):58–68, 2022.

[7] C. Gao and J. R. Spletzer. On-line calibration of multiple lidars on a mobile vehicle platform. In *2010 IEEE International Conference on Robotics and Automation*, pages 279–284, 2010.

[8] B. Xue, J. Jiao, Y. Zhu, L. Zhen, D. Han, M. Liu, and R. Fan. Automatic calibration of dual-lidars using two poles stickered with retro-reflective tape. In *2019 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6, 2019.

[9] J. Levinson and S. Thrun. Automatic online calibration of cameras and lasers. In *Robotics: Science and Systems*, volume 2, page 7. Citeseer, 2013.

[10] G. Pandey, J. R. McBride, S. Savarese, and R. Eustice. Automatic extrinsic calibration of vision and lidar by maximizing mutual information. *J. Field Robotics*, 32:696–722, 2015.

[11] J. Jiao, H. Ye, Y. Zhu, and M. Liu. Robust odometry and mapping for multi-lidar systems with online extrinsic calibration. *IEEE Transactions on Robotics*, pages 1–10, 2021.

[12] L. Heng. Automatic targetless extrinsic calibration of multiple 3d lidars and radars. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10669–10675, 2020.

[13] Z. Taylor and J. Nieto. Motion-based calibration of multimodal sensor extrinsics and timing offset estimation. *IEEE Transactions on Robotics*, 32(5):1215–1229, 2016.

[14] H. Radu and D. Fadi. Hand-eye calibration. *The International Journal of Robotics Research*, 14(3):195–210, June 1995.

[15] J. Levinson and S. Thrun. Unsupervised calibration for multi-beam lasers. *Experimental Robotics Springer Tracts in Advanced Robotics*, 79:179–193, 2014.

[16] W. Maddern, A. Harrison, and P. Newman. Lost in translation (and rotation): Rapid extrinsic calibration for 2d and 3d lidars. In *2012 IEEE International Conference on Robotics and Automation*, pages 3096–3102, 2012.

[17] M. Billah and J. A. Farrell. Calibration of multi-lidar systems: Application to bucket wheel reclaimers. *IEEE Transactions on Control Systems Technology*, page 1–12, 2019.

[18] L. Zhou, D. Koppel, and M. Kaess. Lidar slam with plane adjustment for indoor environment. *IEEE Robotics and Automation Letters*, 6(4):7073–7080, 2021.

[19] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang. Lips: Lidar-inertial 3d plane slam. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 123–130, 2018.

[20] J. Kummerle and T. Kuhner. Unified intrinsic and extrinsic camera and lidar calibration under uncertainties. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[21] Sergio A. Rodriguez F., Vincent Fremont, and Philippe Bonnifait. Extrinsic calibration between a multi-layer lidar and a camera. In *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 214–219, 2008.

[22] Y. Park, S. Yun, C. Won, K. Cho, K. Um, and S. Sim. Calibration between color camera and 3d lidar instruments with a polygonal planar board. *Sensors*, 14(3):5333–5353, 2014.

[23] G. Koo, J. Kang, B. Jang, and N. Doh. Analytic plane covariances construction for precise planarity-based extrinsic calibration of camera and lidar. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

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

[25] J. Jeong, Y. Cho, and A. Kim. The road is enough! extrinsic calibration of non-overlapping stereo camera and lidar using road information. *IEEE Robotics and Automation Letters*, 4(3):2831–2838, 2019.

[26] B. Nagy, L. Kovács, and C. Benedek. Online targetless end-to-end camera-lidar self-calibration. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–6, 2019.

[27] C. Park, P. Moghadam, S. Kim, S. Sridharan, and C. Fookes. Spatiotemporal camera-lidar calibration: A targetless and structureless approach. *IEEE Robotics and Automation Letters*, 5(2):1556–1563, 2020.

[28] 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*, 6(4):7517–7524, 2021.

[29] Y. Zhu, C. Zheng, C. Yuan, X. Huang, and X. Hong. Camvox: A low-cost and accurate lidar-assisted visual slam system. *arXiv preprint arXiv:2011.11357*, 2020.

[30] D. Scaramuzza, A. Harati, and R. Siegwart. Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4164–4169. IEEE, 2007.

[31] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77, 2013.

Xiyuan Liu received the B.Eng. degree in electronic and computer engineering from Hong Kong University of Science and Technology, Hong Kong, in 2017, and the M.Phil. degree in electronic and computer engineering from Hong Kong University of Science and Technology, Hong Kong, in 2019.

He is currently a Ph.D. student with the University of Hong Kong, Hong Kong, and his research interests include LiDAR mapping and sensor calibration.

Chongjian Yuan received the B.Eng. degree in automation from the Zhejiang University (ZJU), Hangzhou, Zhejiang, China, in 2020.

He is currently a Ph.D. student with the University of Hong Kong, Hong Kong, and his research interests include LiDAR SLAM and sensor calibration.

Fu Zhang received the B.E. degree in automation from the University of Science and Technology of China (USTC), Hefei, Anhui, China, in 2011, and the Ph.D. degree in Controls from the University of California, Berkeley, CA, USA, in 2015.

He joined the department of mechanical engineering, the University of Hong Kong (HKU), as an Assistant Professor from Aug 2018. His current research interests are on robotics and controls, with focus on UAV design, navigation, control, and LiDAR-based simultaneous localization and mapping.