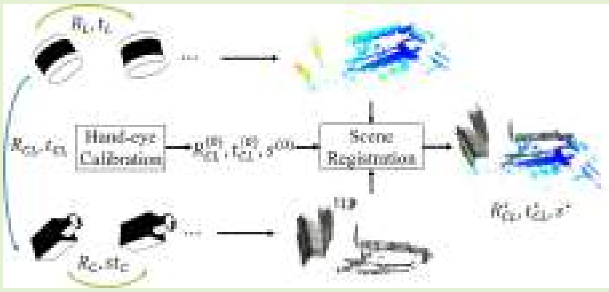
Targetless Extrinsic Calibration of Camera and Low-resolution 3D LiDAR

Ni Ou, Hanyu Cai, Jiawen Yang and Junzheng Wang*

***Abstract*— Autonomous driving heavily relies on LiDAR and camera sensors, which can significantly improve the performance of perception and navigation tasks when fused together. The success of this cross-modality fusion hinges on accurate extrinsic calibration. In recent years, targetless LiDAR-camera calibration methods have gained increasing attention thanks to their independence from external targets. Nevertheless, developing a targetless method for low-resolution LiDARs remains challenging due to the difficulty in extracting reliable features from point clouds with limited LiDAR beams. In this paper, we propose a robust targetless method to solve this struggling problem. It can automatically estimate accurate LiDAR and camera poses and solve the extrinsic matrix through hand-eye calibration. Moreover, we also carefully analyze pose estimation issues existing in the low-resolution LiDAR and present our solution. Real-world experiments are carried out on a UGV-mounted multi-sensor platform containing a CCD camera and a VLP-16 LiDAR. For evaluation, we use a state-of-the-art target-based calibration approach to generate the ground-truth extrinsic parameters. Experimental results demonstrate that our method achieves low calibration error in both translation (3 cm) and rotation (0.59 °).**



***Index Terms*— LiDAR, camera, sensor calibration, pose graph optimization, SLAM**

I. Introduction

NOWADAYS, an increasing number of autonomous driving and robotic systems are equipped with multiple sensors, and Light detection and ranging (LiDAR) sensor and camera are the most popular ones. The LiDAR measures the distance of surrounding points directly irrespective of light conditions, while the camera provides dense texture and color information. The fusion of LiDAR and camera can complement their characteristics to improve performance in perception and navigation tasks such as road detection [1], SLAM [2] and depth estimation [3]. Extrinsic calibration between LiDAR and camera is the prerequisite to sensor fusion, as it estimates the transformation between their coordinate systems. The core of this task is to establish reliable correspondence across a sparse 3D point cloud and a dense RGB image. Recently, the resolution of LiDAR has been increasing with the advancements in manufacturing processes. High-resolution mechanical LiDAR and solid-state LiDAR are advanced representatives but are prohibitively expensive. In some perception and navigation applications [4]–[6], low-resolution LiDAR can be a more attractive alternative due to its cost-performance tradeoff. However, calibrating low-resolution LiDAR with camera is more challenging due to its vertical sparsity of laser scans. The purpose of our work is provide a feasible solution to this problem. In the remaining parts of this section, we will introduce related works on LiDAR-camera calibration and outline our contributions.

A. Related Works

According to whether a specific target is required, existing calibration methods can be divided into two categories: target-based and targetless. Target-based calibration establishes geometric constraints or loss functions based on a target easily detectable to both LiDAR and camera. Planar checkerboards [7]–[9] and other solids with sufficient corners [10], [11] are common targets for these approaches. Target-based methods have dominated the field of LiDAR-camera calibration over several decades owing to their robustness and accuracy, but their limitation is the heavy dependence on external targets. Moreover, the calibration accuracy of these methods is limited to the processing and measurement accuracy of used targets.

Instead of capturing and analyzing specific geometric details from known targets, targetless methods generally extract common features across the laser scan and the RGB image. Early targetless methods design a fusion loss between projected LiDAR points and the image, and edge alignment [12] (EA) is a typical one based on edge correspondence. The

This work is supported by the National Key Research and Development Program of China under Grant 2019YFC1511401, the National Natural Science Foundation of China under Grant 62173038 and 61103157 and the Ensan Frontier Innovation Foundation under Grant 2022024.

N. Ou, H. Cai, J. Yang and J. Wang are all with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing, China. (Corresponding author: Junzheng wang, email: wangjz@bit.edu.cn)

primary principle of this method is a hypothesis that depth-discontinuous LiDAR edges of the point cloud are likely to be associated with 2D edges of the image. Thus, the extrinsic matrix can be optimized by minimizing the reprojection error between LiDAR and camera edges. However, authors of [13] found an inherent drawback of this method called foreground object inflation. It leads to a great loss of accuracy in edge alignment, especially for high-resolution LiDARs. To resolve this problem, they replace depth-discontinuous edges with depth-continuous ones, resulting in higher calibration accuracy. Mutual information (MI) [14] is a similar target-free method proposed for calibration between omnidirectional camera and LiDAR. It maximizes the mutual information between projected LiDAR points' reflectivity and image pixels' gray intensity. Besides these fusion-loss-based approaches, odometry fusion [15] provides another insight into this problem. Rather than find common features in a single LiDAR-camera pair, it utilizes the geometric constraints called hand-eye calibration [16], [17] based on LiDAR and camera motions. Nevertheless, its accuracy is deeply dependent on the performance of motion estimation, which is a challenge to LiDAR and visual odometry. In addition, the development of deep learning technology also promotes the proposal of self-supervision methods [18]–[20]. In these works, a deep neural network is trained from calibrated LiDAR-camera data pairs and used to predict the extrinsic matrix from uncalibrated pairs. They do not require extra operations or external targets, but their generalization ability across unseen scenes and sensors still needs to be demonstrated.

B. Contribution

To the best of our knowledge, although many target-based methods [9], [11], [21] are compatible with low-resolution LiDARs, no targetless ones have succeeded in this task yet. The primary challenge derives from the vertical sparsity of laser scans, which leads to insufficient reliable features. Aiming to alleviate this problem, we consolidate multiple frames of point clouds into one to generate LiDAR features. Specifically, our main contributions are listed below:

- We propose a novel targetless LiDAR-camera calibration approach compatible with low-resolution LiDARs. It leverages hand-eye calibration and scene registration to obtain accurate extrinsic parameters and addresses the scaleless problem of monocular motion estimation.
- We develop a novel algorithm named Cluster Extraction and Integration (CEI) for robust LiDAR pose estimation. With camera poses estimated by Structure from Motion (SfM), CEI can automatically correct erroneously registered LiDAR pairs without any prior knowledge of extrinsic parameters, which is crucial to hand-eye calibration.
- We carried out real-world experiments on a multi-sensor platform containing a VLP-16 LiDAR and a CCD camera. For evaluation, we compare our results to a state-of-the-art target-based method [9] from the perspective of calibration error and edge alignment. Furthermore, ablation study is also conducted to verify the effectiveness of each module in our framework.

II. Preliminaries

A. Overview

As displayed in Fig. 1, our multi-sensor system is fixed on a platform consisting of a VLP-16 LiDAR and a CCD camera. It can rotate in the pitch direction to generate additional orientations, which is advantageous to hand-eye calibration. However, we did not install any sensor to directly measure this angle so as to ensure the generality of our method. The whole platform is fixed on an unmanned ground vehicle (UGV).

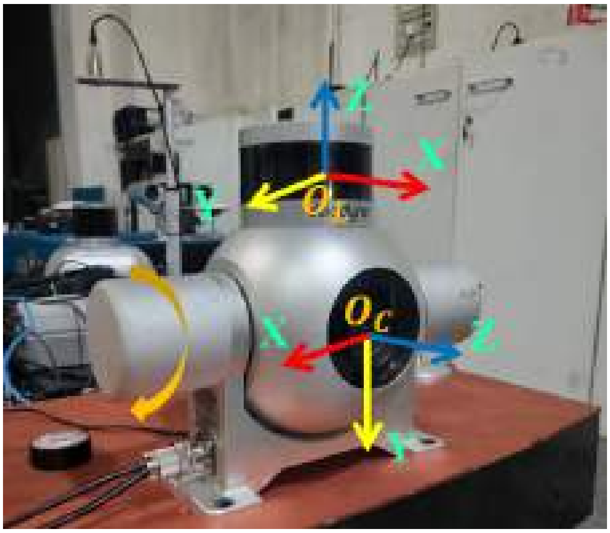


Fig. 1. Coordinate systems. $\boldsymbol{O_L}, \boldsymbol{O_C}$ indicate the coordinate origins of LiDAR and camera, and $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ axes are colored in red, yellow and blue respectively. The orange arrow indicates the capable rotating direction of the platform.

Fig. 2 is the overview flow chart of our method, and primary symbols involved in the following sections are defined in Table I. The main body of our method is hand-eye calibration, which requires the multi-sensor system to move around into a variety of positions and orientations. Different from odometry fusion methods like [15], our approach works on an offline mode and has a back-end optimization—the LiDAR and visual odometry are replaced with Multiway registration [22] and SfM [23], [24] severally. In our framework, SfM and hand-eye calibration are combined together to address the scaleless problem in monocular motion estimation. Furthermore, Cluster Extraction and Integration (CEI) is designed to estimate motions of the low-resolution LiDAR accurately. This algorithm leverages camera poses estimated by SfM to correct failed-registered LiDAR point cloud pairs without any prior knowledge of the extrinsic matrix X_{CL}. With acquired LiDAR and camera poses ($T_{p_i}^L$ & $T_{p_i}^C$), the least-square solution ($X_{CL}^{(0)}$ & $s^{(0)}$) of extrinsic parameters can be solved by a hand-eye (calibration) estimator. Eventually, we fine-tune the solution of hand-eye calibration using scene registration between reconstructed scenes $\boldsymbol{P_L}$ & $\boldsymbol{P_C}$.

The following sections are organized below. For one, the theory of hand-eye calibration and point cloud registration are reviewed in Section II-B and Section II-C respectively, as they are preliminaries to our method. Subsequently, Section III-A and Section III-B jointly introduce our CEI algorithm. Finally,

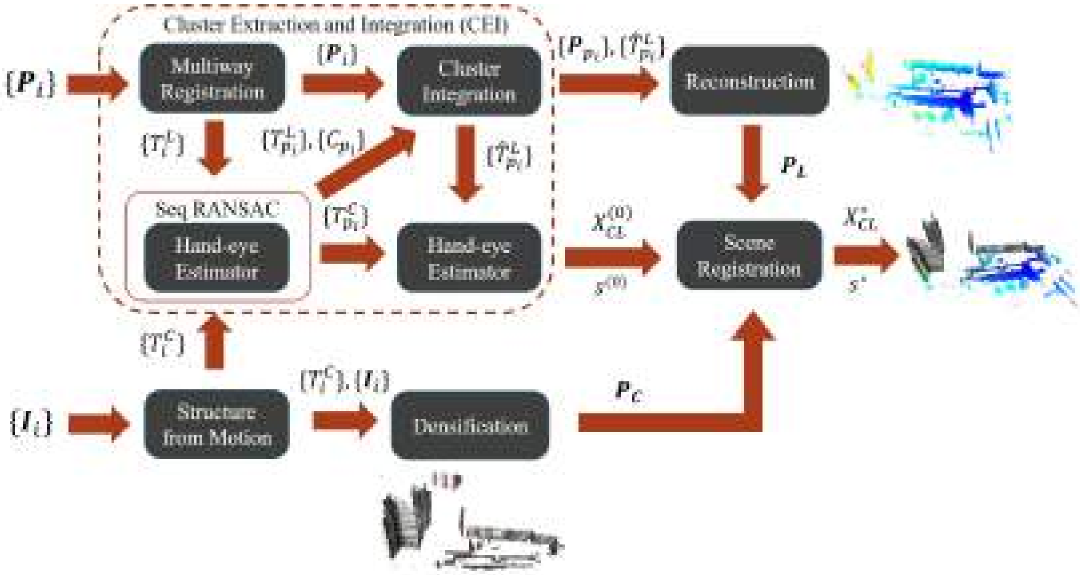


Fig. 2. Flowchart of our method. $\boldsymbol{X}^{(0)}_{\boldsymbol{CL}}$ and $\boldsymbol{s}^{(0)}$ are the extrinsic matrix and scale factor solved by the CEI framework with hand-eye calibration, and $\boldsymbol{X}^*_{\boldsymbol{CL}}$ and $\boldsymbol{s}^*$ are their respective fine-tuned solutions through scene registration.

TABLE I
Definition of Symbols

Symbol	Definition
X_{CL}	the extrinsic calibration matrix from LiDAR to camera
s	the scale factor of monocular motions
$\{\cdot\}$	an ordered list
$\|\cdot\|_F$	the Frobenius norm of a matrix
$\|\cdot\|_2$	the second norm of a vector
$\boldsymbol{P}_i$	the i^{th} frame of laser scan
$\boldsymbol{I}_i$	the i^{th} frame of image
T^L_i	the pose of LiDAR in the i^{th} frame
T^C_i	the pose of camera in the i^{th} frame
$T_{i,j}$	the transformation from node i to node j
p_i	the index of the i^{th} inlier pose in the raw dataset
T^L_i	the i_th inlier LiDAR pose
C_{p_i}	the cluster label of the i^{th} inlier LiDAR pose
$\widehat{T}^L_{p_i}$	the adjustment value of $T^L_{p_i}$ after internal refinement
$\overline{\boldsymbol{P}}_k$	the k^{th} integrated point cloud after internal refinement
$\overline{T}^L_k$	the pose of cluster k in external registration
$\boldsymbol{P_L}$	the scene reconstructed from LiDAR data $\{\boldsymbol{P}_i\}$
$\boldsymbol{P_C}$	the scene reconstructed from camera data$\{\boldsymbol{I}_i\}$
ϵ	the restriction on the change in s (Algorithm 1)

Section III-C describes the solution and tuning of extrinsic parameters.

B. Hand-eye Calibration

Hand-eye calibration aims to estimate X_{CL} and s through LiDAR and camera motions. For convenience, we suppose that their motions have been estimated correctly before. Then, the hand-eye calibration equation can be formulated as (1), which can be further decomposed into (2) and (3).

$$T^C_{i,j}X_{CL} = X_{CL}T^L_{i,j} \ (1 \le i < j \le N)$$
$$s.t. \begin{cases} T^C_{i,j} = T^C_j(T^C_i)^{-1} \\ T^L_{i,j} = T^L_j(T^L_i)^{-1} \end{cases} \tag{1}$$

where N is the number of poses.

$$R^C_{i,j}R_{CL} = R_{CL}R^L_{i,j} \tag{2}$$

$$R^C_{i,j}t_{CL} + s \cdot t^C_{i,j} = R_{CL}t^L_{i,j} + t_{CL} \tag{3}$$

where $R^C_{i,j}, R^L_{i,j}$ are the rotation matrices of $T^C_{i,j}, T^L_{i,j}$ and $t^C_{i,j}, t^L_{i,j}$ are the translation vectors of $T^C_{i,j}, T^L_{i,j}$.

Notably, (1) is a general formulation of hand-eye calibration used in the pose graph. An example pose graph comprised of four nodes is shown in Fig. 3. Each node represents a pose, while each edge signifies the transformation between nodes. Unless specified otherwise, all pose graphs in our paper are fully connected and undirected.

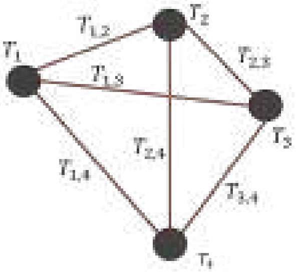


Fig. 3. An example pose graph. Black dots indicate nodes while red lines indicate edges. $\boldsymbol{T_i}$ denotes the pose of node $\boldsymbol{i}$ and $\boldsymbol{T_{i,j}}$ indicates the transformation from node $\boldsymbol{i}$ to node $\boldsymbol{j}$.

As reported in [25], (2) can be simplified to (4), in which R_{CL} can be straightforwardly solved by Singular Value Decomposition (SVD). Once R_{CL} has been determined, t_{CL} and

s can be estimated through the least square method, as shown in (5).

$$\alpha_{i,j} = R_{CL}\beta_{i,j} \tag{4}$$

where $\alpha_{i,j}$ and $\beta_{i,j}$ are rotation vectors of $R^C_{i,j}$ and $R^L_{i,j}$ respectively.

$$\begin{bmatrix} R^C_{i,j} - I & t^C_{i,j} \end{bmatrix} \begin{bmatrix} t_{CL} \\ s \end{bmatrix} = R_{CL} t^L_{i,j} \tag{5}$$

The above are all the steps to solve the extrinsic parameters R_{CL}, t_{CL} and the scale factor s using hand-eye calibration.

C. Point Cloud Registration

Point cloud registration is a technique to compute the transformation (normally rigid) between two point clouds, which is also the foundation of LiDAR pose estimation and scene registration. Suppose that $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ denote two point clouds, the task of registration is to find a rotation matrix $R_{i,j} \in \mathrm{SO}(3)$ and a translation vector $t_{i,j} \in \mathbb{R}^3$ to minimize (6) .

$$\min_{R_{i,j},\, t_{i,j}} \sum_k \|R_{i.j}x_k + t_{i,j} - y_k\|_2^2 \tag{6}$$

where x_k and y_k are one pair of corresponding points.

Note that $\boldsymbol{P}_i$ & $\boldsymbol{P}_j$ are usually partially overlapped, and $\{x_i\}$ & $\{y_i\}$ are only their respective subsets. The Iterative Closest Point (ICP) method [26], [27] is one of the most popular registration methods. It provides a closed-form least-squares solution to this problem. To apply ICP, we need to first compute the centroids of $\{x\}$ and $\{y\}$, along with the cross-covariance matrix H:

$$H = \sum_i (x_i - \overline{x}) \cdot (y_i - \overline{y})^T \tag{7}$$

where $\overline{x}$ and $\overline{y}$ are centroids of $\{x\}$ and $\{y\}$.

Afterwards, we need to decompose H by SVD, mathematically $H = USV$, where U and V are orthogonal while S is diagonal. Finally, the optimal $R_{i,j}$ and $t_{i,j}$ can be solved by (8).

$$R^*_{i,j} = VU^T, \qquad t^*_{i,j} = -R^*_{i,j}\overline{x} + \overline{y} \tag{8}$$

However, it is not easy to find correct correspondences between $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$. ICP searches correspondences based on the closest Euclidean distance and does it iteratively, but it usually requires an initial guess. Point descriptors [28]–[30] are designed to build putative associations. They match points in the feature space independent from the initial transformation using (9).

$$\text{For each } x_k \in \boldsymbol{P}_i,\ y_k = \operatorname*{arg\,min}_{y_m \in \boldsymbol{P}_j} \|F(y_m) - F(x_k)\|_2^2 \tag{9}$$

where $F(x_k)$ and $F(y_k)$ are corresponding feature vectors of x_k and y_k, respectively.

In addition, outlier rejection approaches [22], [31], [32] are developed to improve the matching recall of these descriptors. They are normally robust enough to filter out incorrect correspondences and register point clouds in complex scenarios.

Besides these versatile point cloud registration methods, LiDAR SLAM is also a valuable tool to estimate continuous-time LiDAR poses, including vertically-sparse LiDARs. We test their performance in our ablation study discussed in Section IV.

III. Method

A. LiDAR Pose Estimation: Cluster Extraction

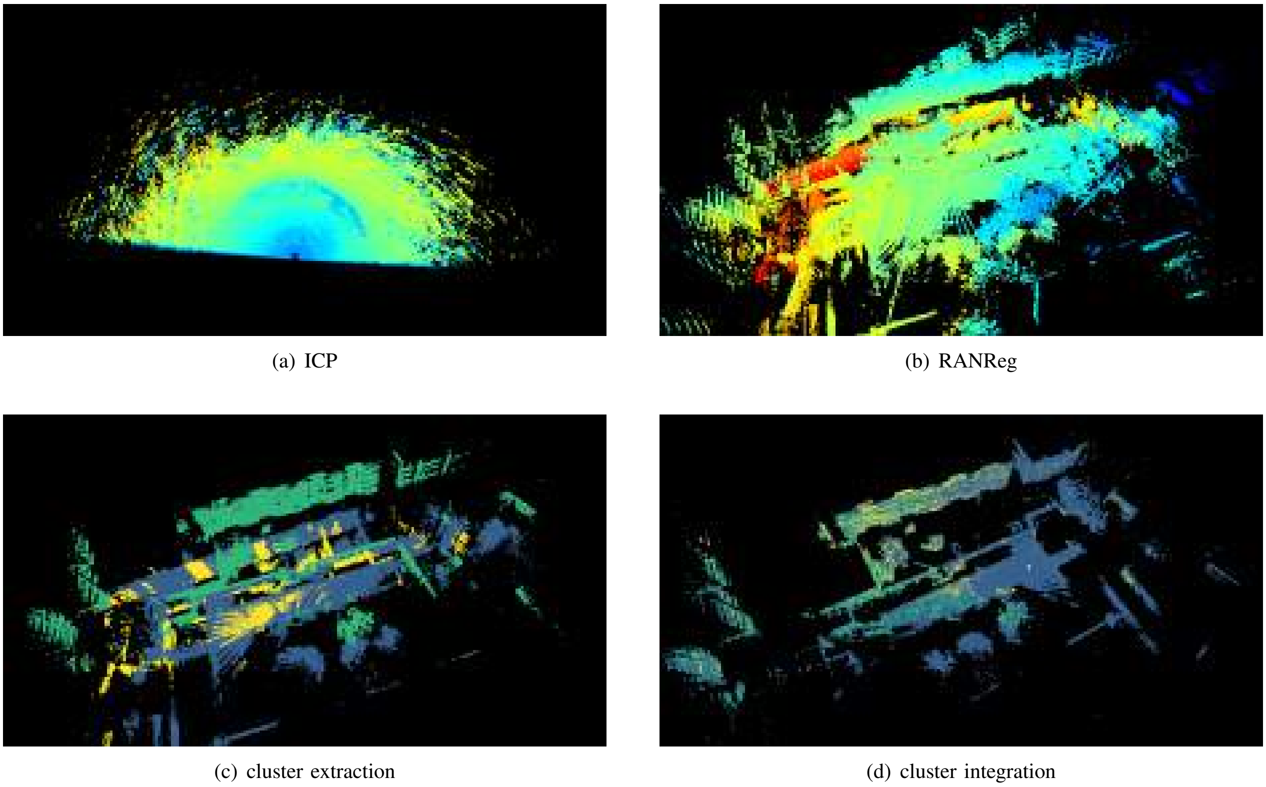
As mentioned in Section II-A, our method relies on 3D reconstruction technologies for pose estimation. We qualitatively evaluate the accuracy of motion estimations (transformations) by visually inspecting the reconstructed point cloud scenes ($\boldsymbol{P}_L$ and $\boldsymbol{P}_C$). We found numerous misalignments in $\boldsymbol{P}_L$ but none in $\boldsymbol{P}_C$. Concerning reconstructing $\boldsymbol{P}_L$, we first tried ICP [26] for pairwise registration, but it encountered a complete failure (Fig. 4(a)), where no patterns can be derived. Then, we replaced ICP with a modified RANSAC registration approach proposed in the supplementary material of [22]. It applies a modified RANSAC criterion and the FPFH [28] descriptor for point cloud registration. We refer this registration approach to **RANReg** to distinguish it from the ordinary RANSAC algorithm. Fig. 4(b) illustrates that a considerable proportion of point clouds are correctly aligned but broken into several clusters with many outliers. Through analysis of its pairwise registration (not shown in pictures), we found that the cause of this phenomenon is the erroneous transformations between adjacent scans, i.e., erroneous odometry edges $T^L_{i,i+1}$. Once $T^L_{i,i+1}$ is incorrect, it will affect all poses after the frame $i+1$. Consequently, nodes can be classified into clusters (inlier nodes with different labels), together with ones that do not belong to any cluster (outlier nodes with no label). Typically, nodes sharing the same label are aligned well to each other, exhibiting a high degree of transformation congruence, which is denoted as cluster consistency in this paper. The process of cluster extraction can be viewed as a classification task. For clarity, we assign the label -1 to outlier nodes. This algorithm aims to capitalize on the cluster consistency to classify nodes into several clusters (extraction) and integrate them into a single graph (integration) through point cloud registration.

Concerning the extraction part, consider a simple case first, where there is only one such cluster in the graph. In this case, all the inlier nodes belong to the same cluster, so edges between inlier nodes should satisfy (1) while other edges do not. We apply the RANSAC [33] algorithm to extract these edges (inlier edges) with a score function presented in (10). As the solution of (1) has been decomposed into (4) and (5), we decompose the RANSAC extraction part into two steps accordingly—first filter out putative inlier edges scored by SO3 loss (11) and then pick out more reliable inliers scored by SE3 loss (10) from the putative ones. Fig. 5(a) displays the result of the first iteration in the form of an adjacency binary matrix, in which ones represent inlier edges while zeros denote other edges.

$$C_1 = \|T^C_{i,j}X_{CL} - X_{CL}T^L_{i,j}\|_F^2 \tag{10}$$

$$C_2 = \|\alpha_{i,j} - R_{CL}\beta_{i,j}\|_2^2 \tag{11}$$

Nonetheless, our final goal is to extract the constituent inlier nodes of each cluster rather than inlier edges, and we employ DBSCAN [34], [35] to achieve this goal. Unlike its conventional usage for clustering, we re-define its adjacency matrix—the distance between every pair of nodes connected by an inlier edge is viewed as zero, while that between any


(a) ICP

(b) RANReg

(c) cluster extraction

(d) cluster integration

Fig. 4. Pose estimation for the VLP-16 LiDAR. Each figure demonstrate a reconstructed scene comprised of multiple point clouds transformed by their relevant poses (a): reconstruction result by ICP (failed); (b): reconstruction result by RANReg [22]; (c): clusters extracted by Algorithm 1 (outliers excluded); (d): integration of clusters. For (c) & (d), point clouds belonging to the same cluster are painted with the same color.

other pair is treated as infinity. In other words, we set the metric distance between P_i and P_j to zero when $T_{i,j}$ is an inlier edge, else one (equivalent to infinity in this algorithm). When there is only one cluster, the labels of nodes predicted by DBSCAN have offered all the necessary information for extracting the cluster and excluding outliers.

Algorithm 1: Cluster Extraction

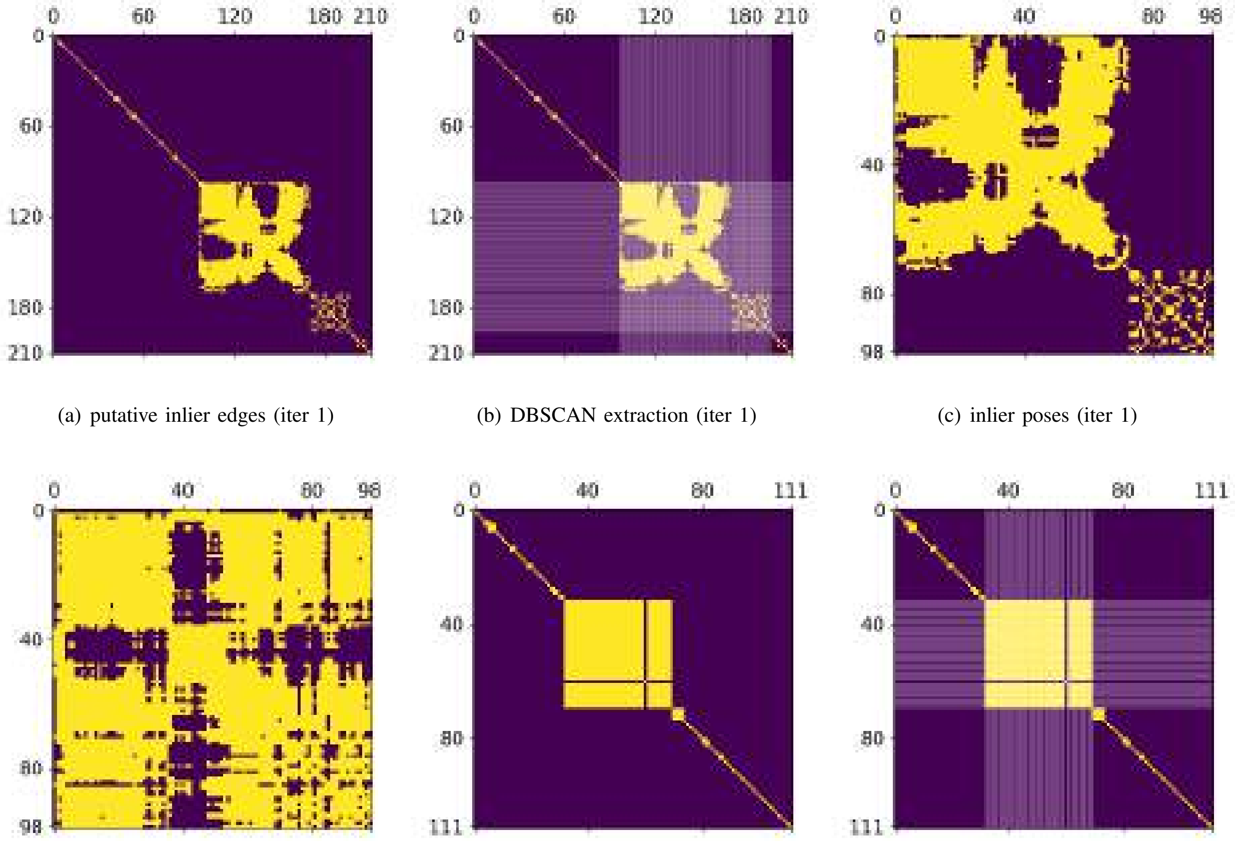
```
Input: $\{T_i^L\}$, $\{T_i^C\}$, $\epsilon$
Output: $\{p_i\}$, $\{C_{p_i}\}$
$\{p_i\} \leftarrow [\,]$, $\{C_{p_i}\} \leftarrow [\,]$, $k \leftarrow 0$;
Solve (4), (5) by RANSAC to obtain the initial scale
  factor $s_0$ and inlier indexes $\{p_{k_i}\}$;
for $p_{k_i}$ in $\{p_{k_i}\}$ do
    Append $p_{k_i}$ to the end of $\{p_i\}$;
    Append $k$ to the end of $\{C_{p_i}\}$;
end
while $|s_0 - s_k|/s_0 < \epsilon$ do
    $k \leftarrow k + 1$;
    $\{T_i^L\} \leftarrow \{T_i^L\} \backslash \{T_{p_i}^L\}$, $\{T_i^C\} \leftarrow \{T_i^C\} \backslash \{T_{p_i}^C\}$
    Solve (4), (5) by RANSAC to update $s_k$ and $\{p_{k_i}\}$
    for $p_{k_i}$ in $\{p_{k_i}\}$ do
        Append $p_{k_i}$ to the end of $\{p_i\}$;
        Append $k$ to the end of $\{C_{p_i}\}$;
    end
end
return $\{p_i\}$, $\{C_{p_i}\}$
```

To generalize this approach to graphs with multiple clusters, we can simply repeat the above procedures iteratively. Specifically, for each iteration, the input is the graph composed of the outlier nodes discarded in the previous iteration, with connected edges unchanged. Meanwhile, we establish a termination condition for this loop. As s is theoretically invariant across clusters, we define a threshold ϵ to restrict the difference in s between the first and the current iterations. The specific steps of this algorithm are summarized in Algorithm 1.

Some intermediate results of Algorithm 1 are visualized in Fig. 5 in the form of adjacency matrices. Fig. 5(a) illustrates the inlier edges extracted by RANSAC in the first iteration. Fig. 5(b) and Fig. 5(c) jointly display the inlier poses clustered by DBSCAN in the first iteration, which constitute the first cluster. We also display cluster consistency in Fig. 5(d) by re-predicting inlier edges within this cluster—most of the edges are classified as inliers, aligning with our theory. Fig. 5(e) and Fig. 5(f) show likewise results in the second iteration. Ultimately, it is demonstrated in Fig. 4(c) that several clusters are extracted by Algorithm 1, while some outlier laser scans have been excluded in the pruned graph.

B. LiDAR Pose Estimation: Cluster Integration

Based on the preceding analysis, the next step is to consolidate these clusters into one. The overall solution is to regard each cluster as an ordinary point cloud and register them in a pose graph. With this cluster-level registration, we can acquire



(a) putative inlier edges (iter 1)

(b) DBSCAN extraction (iter 1)

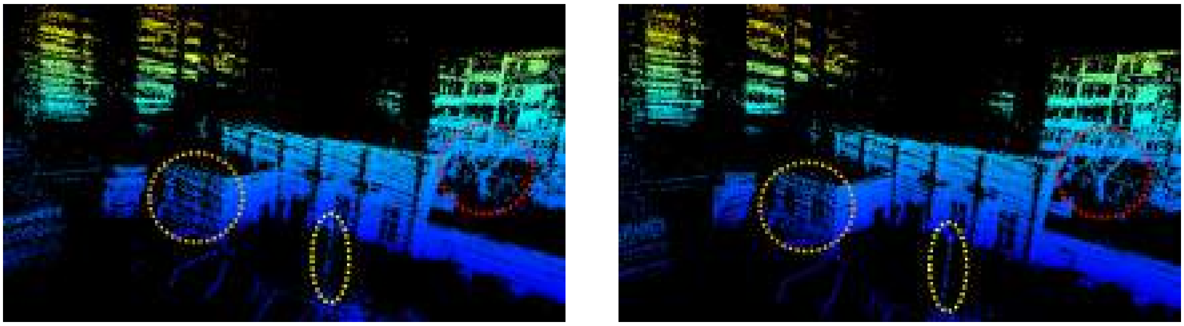
(c) inlier poses (iter 1)

(d) re-predicted inlier edges (iter 1)

(e) putative inlier edges (iter 2)

(f) DBSCAN extraction (iter 2)

Fig. 5. Intermediate results of our cluster extraction algorithm. Each figure demonstrates an adjacency binary matrix (its side length is equal to the number of nodes), in which each element representing an inlier edge is marked yellow. (a): putative inlier edges extracted by the RANSAC filter; (b): inlier poses extracted by DBSCAN (translucent white lines indicate the selected pose indexes); (c): the adjacency matrix consisting of inlier poses; (d): inlier edges re-predicted among the inlier poses. (e) & (f): the results of the second iteration similar to (a) & (b).



(a)

(b)

Fig. 6. Cluster internal refinement. (a): a partial view of one example cluster extracted by our cluster extraction algorithm; (b): the same view of cluster (a) after internal refinement. The colorful circles indicate several noticeable differences between the two visuals.

the pose of each cluster $\overline{T}_k^L$ and update frame-level poses T_i^L accordingly. Considering this process (external registration) can not diminish transformation errors within the cluster, we employ internal refinement within each cluster separately before external registration. In essence, internal refinement and external registration are both multiway registration. The only difference between them is the input.

Regarding the internal refinement, given that point clouds within each cluster are aligned close enough to each other, we choose ICP for local pairwise registration. One example cluster and its refined result are visualized in Fig. 6. It is demonstrated that the integrated point cloud $\overline{\boldsymbol{P}}_k$ becomes denser and sharper after internal refinement.

For external registration, we observed that RANReg [22]

Algorithm 2: Cluster Integration

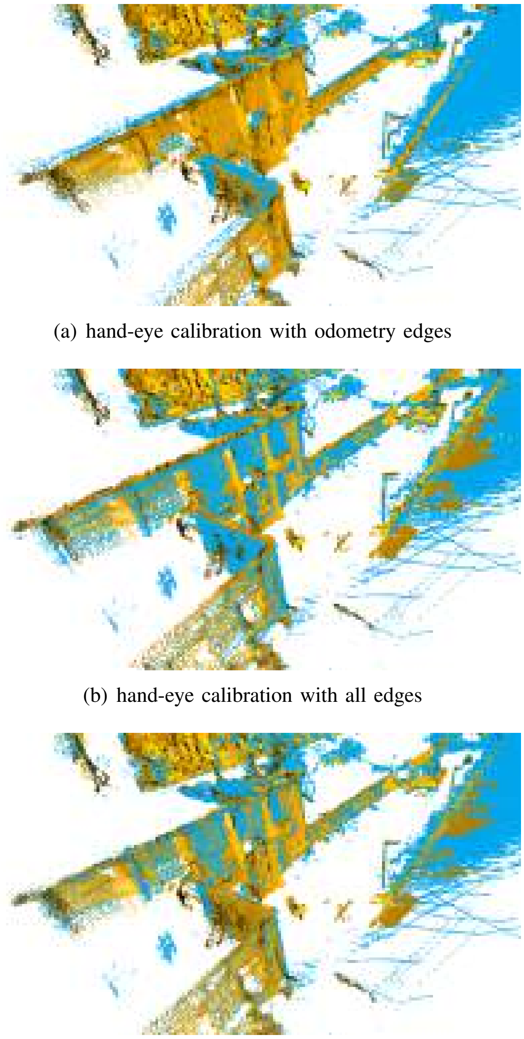
```
Input: $\{T_i^L\}$, $\{\boldsymbol{P}_i\}$, $\{p_i\}$, $\{C_{p_i}\}$
Output: $\{\widehat{T}_{p_i}^L\}$
Split $\{p_i\}$ into clusters $\{\{p_{k_i}\}\}$ using labels $C_{p_i}$;
$\{\{\widehat{T}_{p_{k_i}}^L\}\} \leftarrow [\,]$, $k \leftarrow 0$;
for $\{p_{k_i}\}$ in $\{\{p_{k_i}\}\}$ do
    Extract cluster $\{T_{p_{k_i}}^L\}$ using $\{T_i^L\}$ and $\{p_{k_i}\}$;
    Internal refinement: $\{T_{p_{k_i}}^L\} \rightarrow \{\widehat{T}_{p_{k_i}}^L\}$;
    Append $\{\widehat{T}_{p_{k_i}}^L\}$ to the end of $\{\{\widehat{T}_{p_{k_i}}^L\}\}$;
    Merge point clouds: $\overline{\boldsymbol{P}}_k = \bigcup_{k_i} \widehat{T}_{p_{k_i}}^L \boldsymbol{P}_{p_{k_i}}$;
    Perform voxel downsampling on $\overline{\boldsymbol{P}}_k$;
    $k \leftarrow k + 1$;
end
External registration: $\{\overline{\boldsymbol{P}}_k\} \rightarrow \{\overline{T}_k^L\}$;
for $\{\widehat{T}_{p_{k_i}}^L\}$ in $\{\{\widehat{T}_{p_{k_i}}^L\}\}$, $\overline{T}_k^L$ in $\{\overline{T}_k^L\}$ do
    for $\widehat{T}_{p_{k_i}}^L$ in $\{\widehat{T}_{p_{k_i}}^L\}$ do
        Update pose: $\widehat{T}_{p_{k_i}}^L \leftarrow \overline{T}_k^L \widehat{T}_{p_{k_i}}^L$;
    end
end
Flatten $\{\{\widehat{T}_{p_{k_i}}^L\}\}$ to $\{\widehat{T}_{p_i}^L\}$;
return $\{\widehat{T}_{p_i}^L\}$
```

failed in pairwise registration. Despite no ground-truth correspondences, we empirically attribute the problem to the low feature matching recall. Based on this assumption, we replaced the FPFH point descriptor with FCGF [29], which is a deep learning descriptor pre-trained on KITTI [36] dataset. With this modification, the algorithm eventually has achieved success in external registration. We do not draw figures to expressly show their differences, as it is not relevant to our contributions.

The whole cluster integration algorithm is outlined in Algorithm 2, and the final integrated point cloud is shown in Fig. 4(d), with the same palette as Fig. 4(c). The above are all steps of our cluster extraction and integration (CEI) algorithm, and its role in the whole calibration method is visualized in Fig. 2. With the CEI algorithm, we can obtain accurate LiDAR poses as well as the reconstruction $\boldsymbol{P}_L$.

C. Extrinsic Parameters Solution and Tuning

Considering that some LiDAR poses are classified as outliers and excluded by CEI, we exclude the corresponding camera poses in the following steps accordingly. With the reserved LiDAR and camera poses, we can generate motion pairs ($T_{i,j}^C$ and $T_{i,j}^L$) to solve X_{CL} and s as discussed in Section II-B. The only remaining problem is how to select motion pairs as input. According to [37], the degeneration of motions can affect the uniqueness of solution. In general, the solution accuracy increases with pose diversity. A typical odometry-fusion approach [15] only utilizes odometry edges ($1 \leq i < N,\ j = i + 1$ in (1)) to solve (1), aiming to evade pose drift caused by long-term movement. Nonetheless, a large proportion of edges are not involved in solution, resulting in a significant loss of pose diversity. Instead, we include all edges ($1 \leq i < j \leq N$ in (1)) in solution to enhance pose diversity without concern for long-term drift, as multiway registration and SfM both contain back-end optimization to address this problem.


(a) hand-eye calibration with odometry edges

(b) hand-eye calibration with all edges

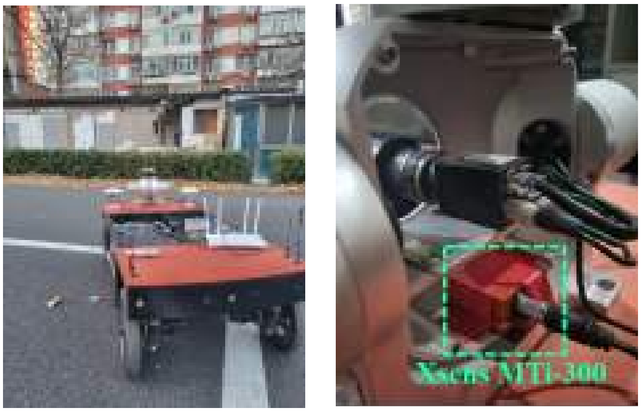
(c) Fine-tuning on (b) by scene registration

Fig. 7. Registration results between $\boldsymbol{P_L}$ and $\boldsymbol{P_C}$. (a),(b): similarity registration using extrinsic parameters estimated by hand-eye calibration. (a) only includes odometry edges; (b) uses all edges; (c): Fine-tuning on (b) by scene registration.

We also qualitatively compare the solution accuracy between selecting all edges and only adjacent edges. With given extrinsic matrix X_{CL} and scale factor s, we can register $\boldsymbol{P}_C$ to $\boldsymbol{P}_L$ through a similarity transformation and inspect their alignment. The formulation of this similarity transformation is given in (12), and registration results are visualized in Fig. 7. Compared with [15] (Fig. 7(a)), using all edges to solve (1) yields better alignment along with a better solution.

$$\boldsymbol{P}_C = \begin{bmatrix} s^{-1}R_{CL} & t_{CL} \\ 0 & 1 \end{bmatrix} \boldsymbol{P}_L \tag{12}$$

Additionally, we use scene registration between $\boldsymbol{P}_L$ and $\boldsymbol{P}_C$



(a) (b)

Fig. 8. Real world experiments. (a): the calibration scene and our multi-sensor platform mounted on the UGV; (b): A Xsens MTi-300 IMU is fixed on our platform for ground-truth LiDAR pose measurement.

to further optimize the solution. It is implemented by a 7-DOF ICP method that can handle point cloud registration with scale tuning. By comparison between Fig. 7(c) and Fig. 7(b), $\boldsymbol{P}_L$ and $\boldsymbol{P}_C$ are aligned closer to each other, indicating higher accuracy of the solution. The above is the whole content of our method.

D. Implementation Details

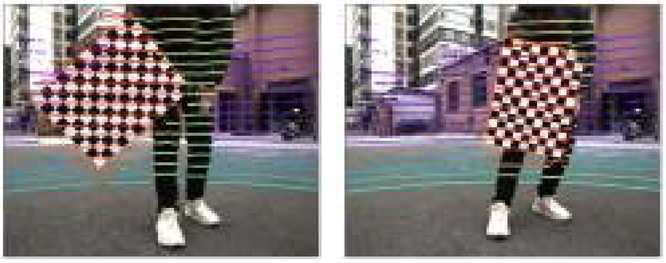
We take advantage in several open-sourced libraries for the code of conduct. Specifically, SfM and multiway registration are implemented based on OpenMVG [38] and Open3D [38] respectively; the densification process in our flowchart (Fig. 2) is conducted based on OpenMVS [39]; the RANSAC and DBSCAN algorithms are implemented based on the scikit-learn library [40]. It should be noted that we have also tried other state-of-the-art pairwise registration methods like TEASER++ [32] and FGR [31] in Section III-A, but they even perform worse than RANReg [22]. To contribute to the community, we provide open-sourced codes at `https://github.com/gitouni/Targetless-LiDAR-camera-calibration`.

IV. Real-world Experiment

A. Calibration Error

Our real-world experiments are carried out on a multi-sensor platform introduced in Section II-A. We collected 211 frames of synchronized LiDAR-camera data, and the pose of our platform varies among these frames. The backward 180° of each LiDAR scan ($x < 0$ in O_L) was discarded during preprocess because the objects on the UGV behind the LiDAR can be a disturbance to LiDAR scanning. One example picture of the calibration scene and our UGV is shown in Fig. 8(a).

To quantitatively evaluate our method, we generate the ground-truth calibration matrix using a state-of-the-art target-based method [9], which has proven to work well for low-resolution 3D LiDARs. Its LiDAR projection figures are displayed in Fig 9. Concerning metrics, we evaluate the root mean square error (RMSE) in rotation and translation respectively.



(a) (b)

Fig. 9. The calibration result of [9] on the checkerboard. (a) and (b) demonstrate the LiDAR projection results on the checkerboard under two different poses. Projected LiDAR points with different ranges are painted with different colors.

Moreover, we add ablation experiments to verify the effectiveness of CEI and scene registration (SR). By comparing the last three rows of Table II, the employment of CEI dramatically improves the accuracy of hand-eye calibration. Scene registration successfully reduces the translation error to a low level, with a slight loss of rotation performance. It is almost certain that CEI and scene registration are both effective in our framework.

Ablation experiments are also conducted on RANReg, with slightly different data settings. The control group consists of three state-of-the-art LiDAR SLAM methods: A-LOAM [41] (LOAM without a IMU), F-LOAM [42] and CT-ICP [43]. In contrast to previous settings, consecutive LiDAR scans were saved in rosbag files as the input of SLAM. $\{\boldsymbol{P}_i\}$ and $\{\boldsymbol{I}_i\}$ were synchronized collected at key timestamps, while $\{T_i^L\}$ were selected from the continuous-time poses evaluated by SLAM.

TABLE II
Rotation and Translation RMSE of SLAM Methods

Method	Rotation (°)	Translation (m)
HE[1]+A-LOAM [41]	7.59	1.240
HE+F-LOAM [42]	5.25	0.333
HE+CT-ICP [43]	4.32	0.588
HE+A-LOAM+CEI	2.74	0.382
HE+F-LOAM+CEI	1.52	1.025
HE+CT-ICP+CEI	**0.42**	0.048
HE+A-LOAM+CEI+SR[2]	1.89	0.979
HE+F-LOAM+CEI+SR	0.48	0.259
HE+CT-ICP+CEI+SR	1.09	0.251
HE+RANReg [22]	1.82	0.896
HE+RANReg+CEI	0.54	0.145
HE+RANReg+CEI+SR	0.59	**0.030**

[1] HE: hand-eye calibration
[2] SR: scene registration

Quantitative calibration errors of aforementioned SLAM methods are presented in Table II. As shown in the first three rows, considerable calibration residual still exists when using LiDAR poses estimated by SLAMs for hand-eye calibration. In contrast, the CEI module noticeably improves the performance of SLAMs, except F-LOAM. When combined with CEI, CT-ICP achieves almost satisfactory calibration performance, which is the best of all SLAM methods. However, scene registration is not effective in all cases. F-LOAM is the only method whose rotation and translation RMSE have been reduced by scene registration, but the translation part of its calibration result still needs improvement.

B. LiDAR Pose Estimation Error

To measure the accuracy of LiDAR poses, we also implemented an advanced LiDAR-inertia SLAM named LIO-SAM [44] to generate ground-truth LiDAR poses. An Xsens MTi-300 IMU is installed at the bottom of the platform (Fig. 8(b)), and its extrinsic matrix with LiDAR is obtained through hand-eye calibration. For trajectory error evaluation, we adopt unit-less Relative Rotation Error (RRE) and Relative Translation Error (RTE) introduced in the *evo* [45] tool. It is illustrated in Table III that RANReg with CEI achieves the lowest RRE and RTE metrics, and CT-ICP is an excellent alternative to RANReg, provided consecutive LiDAR poses and edge computing devices are available. It is also recommended to acquire high-accuracy LiDAR poses if a NIR camera is accessible [46].

TABLE III
RELATIVE TRAJECTORY ERROR OF SLAM METHODS

Method	RRE	RTE
A-LOAM [41]	11.39%	15.57%
F-LOAM [42]	11.78%	16.24%
CT-ICP [43]	0.56%	1.91%
RANReg [22]	1.90%	26.29%
A-LOAM+CEI	1.59%	6.52%
F-LOAM+CEI	1.86%	4.71%
CT-ICP+CEI	0.29%	3.20%
RANReg+CEI	**0.17%**	**1.81%**

C. Edge Alignment Analysis

Inspired by a previous study [12], we conducted an edge alignment analysis to compare our proposed method and the target-based one [9]. To start with, we extracted the LiDAR depth-discontinuous edge points using the technique introduced in [12]. Then, we manually marked the image edges on which the projected LiDAR points may have discontinuous depth. Finally, we projected the LiDAR edge points onto their corresponding images using different groups of extrinsic parameters for comparison. As the manual labeling process can be time-consuming, we selected only 11 representative images for the edge alignment analysis.

We present both qualitative and quantitative results in this section. Regarding the qualitative comparison, we display two cases in Fig. 10, indicating that our method offers more accurate edge alignment results in certain areas, such as edges of the street light, windows, and eaves. As for the quantitative evaluation, we utilized the KD-Tree to compute the shortest distance between the projected LiDAR edge points and the image edges. We define a pair of LiDAR point and image edge as a true correspondence, provided its distance is less than 5 pixels. Based on this concept, we propose two metrics: Correspondence Number and Correspondence Distance. The former denotes the number of such correspondences counted in an image, while the latter indicates the mean of their distances. We present the final resultant figure across 11 samples in Fig. 11. Overall, our method yields more correspondences than the target-based one (mean of 11 cases: 23.27 vs 22.00) while maintaining even lower correspondence distances (mean of 11 cases: 2.01 vs 2.18).

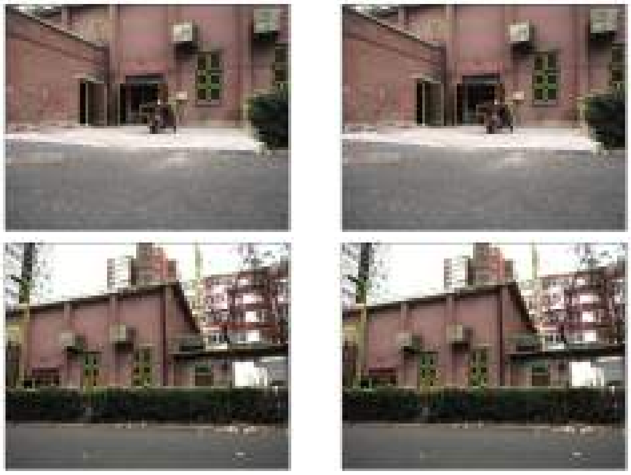
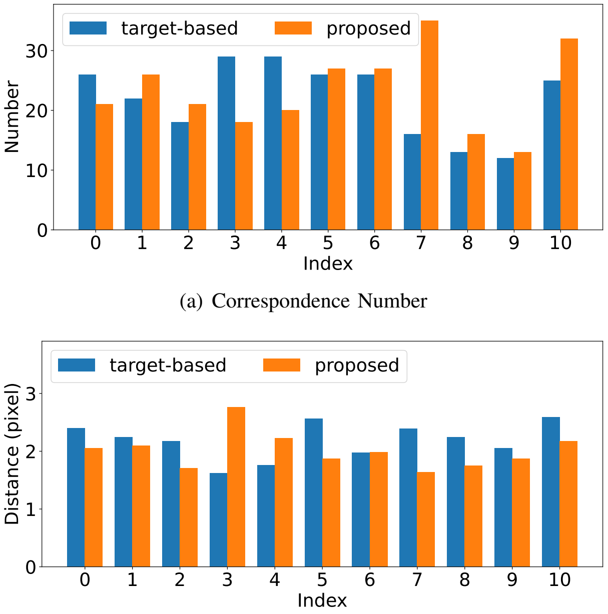

(a) Proposed (b) Target-based [9]

Fig. 10. Qualitative results of edge alignment analysis. We analyze the correlation between the image edges (yellow) and projected LiDAR edge points (red).



(a) Correspondence Number

(b) Correspondence Distance

Fig. 11. Quantitative results of edge alignment analysis. (a): Correspondence Number across 11 samples; (b): Correspondence Distance across 11 samples.

V. CONCLUSION

In this study, a novel targetless method is proposed for extrinsic calibration between camera and low-resolution LiDAR.

It utilizes hand-eye calibration for initialization and scene registration for refinement. The proposed method outperforms other state-of-the-art targetless methods and can be particularly useful in applications where accurate extrinsic parameters are required but external targets are unavailable. Additionally, its CEI module offers a new solution to pose correction in multi-sensor systems.

References

[1] Z. Chen, J. Zhang, and D. Tao, "Progressive lidar adaptation for road detection," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 693–702, 2019.

[2] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: low-drift, robust, and fast," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2174–2181.

[3] Y. Li, X. Liu, W. Dong, H. Zhou, H. Bao, G. Zhang, Y. Zhang, and Z. Cui, "Deltar: Depth estimation from a light-weight tof sensor and rgb image," *arXiv preprint arXiv:2209.13362*, 2022.

[4] S. Dogru and L. Marques, "Drone detection using sparse lidar measurements," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3062–3069, 2022.

[5] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765.

[6] D. Feng, Y. Qi, S. Zhong, Z. Chen, Y. Jiao, Q. Chen, T. Jiang, and H. Chen, "S3e: A large-scale multimodal dataset for collaborative slam," *arXiv preprint arXiv:2210.13723*, 2022.

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

[8] A. Geiger, F. Moosmann, . Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 3936–3943.

[9] 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.

[10] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna, "Lidar-camera calibration using 3d-3d point correspondences," *arXiv preprint arXiv:1705.09785*, 2017.

[11] Z. Pusztai and L. Hajder, "Accurate calibration of lidar-camera systems using ordinary boxes," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[12] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers." in *Robotics: Science and Systems*, vol. 2, no. 7. Citeseer, 2013.

[13] 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.

[14] 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.

[15] R. Ishikawa, T. Oishi, and K. Ikeuchi, "Lidar and camera calibration using motions estimated by sensor fusion odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7342–7349.

[16] Y. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax=xb," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 16–29, 1989.

[17] I. Fassi and G. Legnani, "Hand to sensor calibration: A geometrical interpretation of the matrix equation ax= xb," *Journal of Robotic Systems*, vol. 22, no. 9, pp. 497–506, 2005.

[18] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, "Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1110–1117.

[19] X. Lv, B. Wang, Z. Dou, D. Ye, and S. Wang, "Lccnet: Lidar and camera self-calibration using cost volume network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2894–2901.

[20] K. Yuan, Z. Guo, and Z. J. Wang, "Rggnet: Tolerance aware lidar-camera online calibration with geometric deep learning and generative model," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6956–6963, 2020.

[21] P. An, T. Ma, K. Yu, B. Fang, J. Zhang, W. Fu, and J. Ma, "Geometric calibration for lidar-camera system fusing 3d-2d and 3d-3d point correspondences," *Optics express*, vol. 28, no. 2, pp. 2122–2141, 2020.

[22] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5556–5565.

[23] P. Moulon, P. Monasse, and R. Marlet, "Adaptive structure from motion with a contrario model estimation," in *Proceedings of the Asian Computer Vision Conference (ACCV 2012)*. Springer Berlin Heidelberg, 2012, pp. 257–270.

[24] ——, "Global fusion of relative motions for robust, accurate and scalable structure from motion," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3248–3255.

[25] F. C. Park and B. J. Martin, "Robot sensor calibration: solving ax= xb on the euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.

[26] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[27] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 04, pp. 376–380, 1991.

[28] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 3212–3217.

[29] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *ICCV*, 2019.

[30] J. Li and G. H. Lee, "Usip: Unsupervised stable interest point detection from 3d point clouds," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 361–370.

[31] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European conference on computer vision (ECCV)*. Springer, 2016, pp. 766–782.

[32] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2020.

[33] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[34] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[35] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[36] 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.

[37] R. Y. Tsai, R. K. Lenz, *et al.*, "A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration," *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 345–358, 1989.

[38] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "OpenMVG: Open multiple view geometry," in *International Workshop on Reproducible Research in Pattern Recognition*. Springer, 2016, pp. 60–74.

[39] D. Cernea, "OpenMVS: Multi-view stereo reconstruction library," 2020. [Online]. Available: https://cdcseacave.github.io/openMVS

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[41] 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.

[42] H. Wang, C. Wang, C. Chen, and L. Xie, "F-loam : Fast lidar odometry and mapping," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[43] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "Ct-icp: Real-time elastic lidar odometry with loop closure," 2021.

[44] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, “Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping,” in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.

[45] M. Grupp, “evo: Python package for the evaluation of odometry and slam.” https://github.com/MichaelGrupp/evo, 2017.

[46] H.-S. Song and Y.-K. Kim, “Automatic one-shot lidar alignment inspection system using nir camera,” *IEEE Sensors Journal*, 2023.

Ni Ou Ni Ou received the B.E. degree in Electrical Engineering and Its Automation from China University of Mining Technology, Xuzhou, China, in 2020.

He is pursuing a Ph.D. degree in the School of Automation, Beijing Institute of Technology, Beijing, China, from 2020 to now on. His current research interests include SLAM systems, robotic sensor calibration and point cloud recognition.

Hanyu Cai Hanyu Cai received the Bachelor’s Degree from the ChongQing University, ChongQing, China, in 2021.

He is currently pursuing the masters degree with the Beijing Institute of Technology, Beijing. His research interests include Structure from Motion, Visual SLAM and Visual-Inertial system.

Jiawen Yang Jiawen Yang received the bachelor’s degree from the Nanjing University of Science and Technology, Nanjing, China.

He is currently pursuing the master’s degree with the Beijing Institute of Technology, mainly engaged in laser point cloud data processing.

Junzheng Wang Junzheng Wang received the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 1994.

He is the Deputy Director with the State Key Laboratory of Intelligent Control and Decision of Complex Systems, the director of the Key Laboratory of Servo Motion System Drive and the senior member of the Chinese Mechanical Engineering Society and the Chinese Society for Measurement. His current research interests include motion control, electric hydraulic servo system and object tracking.