2019 IEEE Intelligent Vehicles Symposium (IV)
Paris, France. June 9-12, 2019

A Novel Dual-Lidar Calibration Algorithm Using Planar Surfaces

Jianhao Jiao[1*], Qinghai Liao[1*], Yilong Zhu[1], Tianyu Liu[2], Yang Yu[1],
Rui Fan[1], Lujia Wang[3], Ming Liu[1]

***Abstract*— Multiple lidars are used on mobile vehicles for rendering a broad view to enhance the performance of perception systems. However, precise calibration of multiple lidars is challenging since the feature correspondences in scan points are sparse for providing enough constraints. To address this problem, existing methods require fixed calibration targets in scenes or rely exclusively on additional sensors. In this paper, we present a novel method that enables automatic lidar calibration without these restrictions. Three linearly independent planar surfaces appearing in surroundings is utilized to find correspondences. Two components are developed to ensure the extrinsic parameters to be found: a closed-form solver for initialization and an optimizer for refinement by minimizing a nonlinear cost function. Simulation and experimental results demonstrate the accuracy of our calibration approach with the rotation and translation errors smaller than 0.05rad and 0.1m respectively.**

I. Introduction

Accurate extrinsic calibration has gained importance for vehicles which are equipped with a large number of sensors. Traditional manual calibration techniques, which require fixed calibration targets [1]–[5], suffer limited flexibility and tend not to scale well to multi-sensor configurations.

The automatic calibration methods of 3D lidars will be focused on in this paper. With the development of mobile robots, lidars have become one of the most popular sensors for perceiving the environments. Thanks to their accuracy and stability in measuring distance, they are used in many applications [6]–[8]. However, much recent work prefers the configuration with multiple lidars rather than a single lidar because it can render a richer view of environments and offer denser measurements. Several problems such as occlusion and sparsity can be avoided. On any mobile platform containing multiple lidars, it is of paramount importance that sensors can be calibrated automatically. When the calibration is finished, all measurements will be correctly projected into a unified coordinate system, as shown in Fig. 1(b).

This work was supported by National Natural Science Foundation of China No. U1713211 and 61603376, the Shenzhen Science, Technology and Innovation Commission(SZSTI) JCYJ20170818153518789, the Research Grant Council of Hong Kong SAR Government, China, under Project No. 11210017, and No. 21202816, and was also supported by the Guangdong Innovation and Technology Fund No. 2018B050502009, awarded to Prof. Ming Liu and Dr. Lujia Wang.

[1]J. Jiao, Q. Liao, Y. Zhu, Y. Yu, R. Fan, and M. Liu are with the Robotics and Multi-Perception Laborotary, Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong SAR, China {jjiao, qinghai.liao, Yzhubr, yyubj, eeruifan, eelium}@ust.hk.

[2]T. Liu is with Unity-Drive technology Inc, Shenzhen, China {liutianyu}@unity-drive.com.

[3]L. Wang is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China {lj.wang1}@siat.ac.cn.

*Equal contribution.

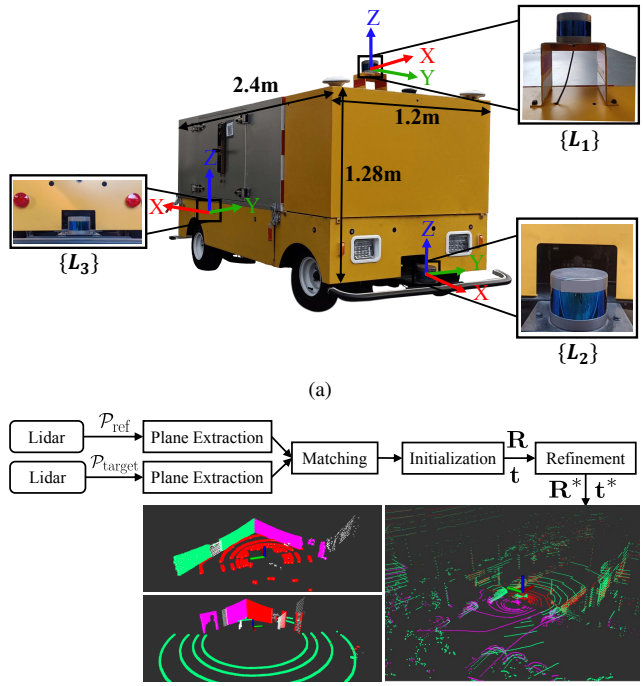


Fig. 1. (a) Our mobile platform. (b) An overview of the proposed method. The bottom-left figure visualizes the extracted planes. The bottom-right figure visualizes the calibration results, where the red, pink and pink dots represent the points perceived by different lidars.

Over the past years, automatic methods for calibrating sensors e.g., lidar to camera [9], multi-camera [10], and camera to IMU [11] have been proposed. However, few efforts have investigated multi-lidar calibration since this problem is challenging; as Choi et al. [12] explained, "searching for correspondences among scan points is difficult". [13] and [14] are two specific multi-lidar calibration methods, but several drawbacks are presented. Firstly, they rely exclusively on an additional sensor. Secondly, their success will depend on the quality of initialization provided by users. Thirdly, both of them assume that mobile platforms should undergo efficient motion.

To tackle these issues, we propose a novel approach for calibrating dual lidars without any additional sensors and artificial markers. This method assumes that three linearly independent planar surfaces forming a wall corner shape are provided as the calibration targets. Through matching these planes, our method can successfully acquire the unknown extrinsic parameters with two steps: a closed-form solution for initialization and an optimizer for refinement by minimizing a defined cost function. This method has been demonstrated

to calibrate three lidars on our mobile platform [see Fig. 1(a)]. An overview of the method is shown in Fig. 1(b). In solving calibration with poor human intervention, we make two significant contributions in this paper:

- We make it possible to use objects with unknown size in the outdoor environment as the calibration target.
- We demonstrate that our method is efficient in applications since the extrinsic parameters can be obtained immediately with one-shot measurement.

The rest of the paper is organized as follows. In Section II, the related work is discussed. The methodology of our approach is introduced in Section III, followed by experimental results in Section IV. Finally, Section V summarizes the paper and discusses possible future work.

II. Related Work

A. Calibration for Multi-Lidar Systems

In recent years, Gao and Spletzer [13] proposed an algorithm to calibrate multiple lidars using point constraints provided by retro-reflective tapes in scenes. He et al. [14] demonstrated a technique to extract geometric features among point clouds, which enables an offline algorithm to calibrate multiple 2D lidars in arbitrary scenes. Shortly after that, their approach was improved in a challenging scenario: an underground parking lot, where GPS is not available [15]. However, such methods rely on an additional localization module, making the calibration process complicated.

Artificial landmarks are prevalently used to find correspondences among sensor data. Xie et al. [16] provided a general solution to jointly calibrate multiple cameras and lidars in the presence of a pre-built environment with apriltags. Steder et al. [17] proposed a tracking-based method to calibrate multiple 2D lidars using a moving object which appears in their overlapping areas. Based on it, Quenzel et al. [18] calibrated the same sensors with an additional verification step. However, these approaches require known markers to be placed in scenes. In this paper, we exploit common planar surfaces as the calibration target inspired by [12], but our approach differs from it by releasing the orthogonal assumption of these surfaces to achieve outdoor calibration.

B. Calibration for Other Sensing Systems

There exist several published papers on lidar to camera, multi-camera and camera to IMU calibration. One of the first work to solve online camera and lidar calibration is [19]. In this method, edge features in images are associated with lidar measurements using depth discontinuities. The extrinsic parameters are optimized by minimizing a cost function. Different metrics based on Gradient Orientation Measure (GOM) [9], Mutual Information (MI) [20], and line-plane constraints [3] were also proposed. However, all of them require initialization provided by users. In our proposed method, we introduce an algorithm to automatically initialize the extrinsic parameters by exploiting the geometric constraints of planar surfaces.

Developed from hand-eye calibration using the structure-from-motion techniques, motion-based approaches have been implemented to solve the calibration. Heng et al. [10] proposed CamOdoCal, an automatic algorithm for four-camera calibration without the assumption of overlapping fields of view. They decouple the calibration process into initialization and refinement. In initialization, a rough estimate of extrinsic parameters is computed by combining visual odometry with the vehicle's egomotion. To refine the estimates, a bundle adjustment is used to optimize all of the cameras' poses and feature data. This pipeline is employed in our method. However, CamOdoCal was explicitly designed for vision sensors, which may not be feasible in various sensor configurations. In contrast, Taylor and Nieto released a system [21], [22] to calibrate multiple heterogeneous sensors and their time offset. Generally, motion-based methods can work for a variety of configurations and can be integrated into several SLAM systems [23]. However, the calibration accuracy of motion-based methods is limited due to the drift of computed odometry, which needs to be refined using the appearance cues in the surroundings.

III. Methodology

Our approach makes use of three linearly independent planar surfaces to calibrate a dual-lidar system. To achieve it, we firstly introduce a robust algorithm to extract planes from scan points. The geometric structure of these planes is then utilized to acquire the extrinsic parameters. We define $\{L_k\}$ as a 3D coordinate system with its origin at the geometric center of a lidar L_k. The $x-$, $y-$ and $z-$ axes are pointing forward, left and upward respectively. In this paper, we consider $\{L_1\}$ as the reference frame, and $\{L_k\}$ as the target frame. The point clouds perceived by a lidar is denoted by $\mathcal{P}$, and the coordinates of a point in $\mathcal{P}$ is represented as $\mathbf{p}_n = [x_n, y_n, z_n]^\top$. Detailed notations are listed in Table I and are visualized in Fig. 2.

A. Plane Extraction

Denoting $\boldsymbol{\beta}_i = \left[\beta_{(i,0)}, \beta_{(i,1)}, \beta_{(i,2)}, \beta_{(i,3)}\right]^\top$ the coefficients of Π_i, the distance between $\mathbf{p}_n$ and Π_i [see Fig. 3] is computed as follows:

$$f_i(\mathbf{p}_n) = \left|\beta_{(i,0)}x_n + \beta_{(i,1)}y_n + \beta_{(i,2)}z_n + \beta_{(i,3)}\right|. \quad (1)$$

To fit a planar model from a series of discrete points, we employ the random sample consensus (RANSAC) algorithm. By randomly selecting N points from $\mathcal{P}$, the planar coeffi-

TABLE I
Annotation table.

Notation	Explanation
$\{L_1\}$ / $\{L_k\}$	Reference / Target coordinate system
Π_i / Π_i'	i^{th} planar surfaces in $\{L_1\}$ / $\{L_k\}$
$\mathbf{o}$ / $\mathbf{o}'$	Intersection point in $\{L_1\}$ / $\{L_k\}$
$\boldsymbol{\beta}_i$ / $\boldsymbol{\beta}_i'$	Coefficients of Π_i / Π_i'
$\mathbf{n}_i$ / $\mathbf{n}_i'$	Unit normal vector of Π_i / Π_i'

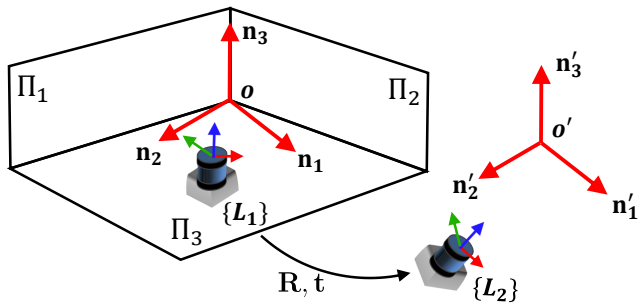


Fig. 2. A diagram of the notations. Red, green, and blue arrows denote the $x-$, $y-$, and $z-$ axes of each lidar coordinate system respectively.

cients are acquired by solving a least-squares problem [24]:

$$\boldsymbol{\beta}_i^* = \arg\min_{\boldsymbol{\beta}_i} \sum_{n=1}^{N} f_i^2(\mathbf{p}_n), \tag{2}$$

where the parameter vector $\boldsymbol{\beta}_i$ will be updated iteratively until an optimal model is acquired with maximum inlier points. To determine whether a point is an inlier, its square distance to a plane is computed. To extract three models, the RANSAC algorithm is executed separately at three times. At each time, points belonging to former extracted models are ignored. Finally, we can obtain three groups of planar coefficients which are denoted by $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$, and $\boldsymbol{\beta}_3$ respectively to describe the planar surfaces. Hence, we can compute $\mathbf{o} = [o_x, o_y, o_z]^\top$ by solving a set of linear systems

$$\begin{bmatrix} \boldsymbol{\beta}_1^\top \\ \boldsymbol{\beta}_2^\top \\ \boldsymbol{\beta}_3^\top \end{bmatrix} \begin{bmatrix} \mathbf{o} \\ 1 \end{bmatrix} = \mathbf{0}. \tag{3}$$

After computing $\boldsymbol{\beta}_i$, the unit normal vectors $\mathbf{n}_1$, $\mathbf{n}_2$ and $\mathbf{n}_3$ can be represented up to scale. According to our assumption of linear independence, there exist three non-zero scalars a, b and c that satisfy the following equation:

$$a\mathbf{n}_1 + b\mathbf{n}_2 + c\mathbf{n}_3 = -\mathbf{o}, \tag{4}$$

where we can fix the directions of normal vectors to make a, b and c positive.

It is impossible to match these planar surfaces directly between lidars using these results. Fig. 1(b) shows an example of the extracted planes, where the color values represent their extraction order. We can observe that the corresponding planes do not have the same order. By utilizing the wall corner shape [see Fig. 2], we find that these orders can be rearranged. Without loss of generality, we set Π_1, Π_2 as the *left* and *right* plane respectively, and Π_3 as the *bottom* plane. Their normal vectors should follow the right-hand rule:

$$(\mathbf{n}_2 \times \mathbf{n}_1) \cdot \mathbf{n}_3 > 0. \tag{5}$$

Following the above steps, we can correctly match the corresponding planes between two lidars.

B. Initialization Using Closed-Form Solution

We can formulate the calibration of dual-lidar as a nonlinear optimization problem by minimizing the distance between corresponding planes. But the defined cost function

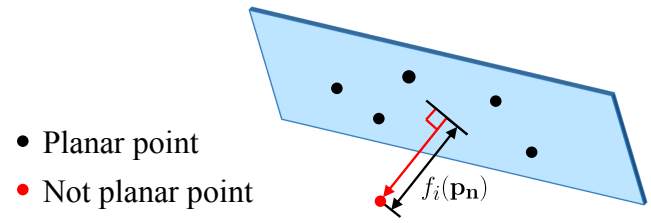


Fig. 3. Point to Plane distance.

(8) is non-convex. To avoid local minima, the parameters should be firstly initialized.

According to Section III-A, we already have two sets of fitted planes Π and Π' with known normal vectors. Consequently, their relative rotation $\mathbf{R}$ can be thus computed for initialization by introducing the Kabsch algorithm [25]. The Kabsch algorithm is an effective approach that provides a least-squares solution to calculate the rotation between a pair of vector sets. We use $\mathbf{P}$, $\mathbf{Q} \in \mathbb{R}^3$ to indicate two 3×3 matrices. The elements at the ith column of $\mathbf{P}$ are defined as $(\mathbf{n}_i - \mathbf{o})$, and those of $\mathbf{Q}$ are defined as $(\mathbf{n}'_i - \mathbf{o}')$. We also denote $\mathbf{H} = \mathbf{P}^\top\mathbf{Q}$ the cross-covariance matrix. By calculating the singular value decomposition (SVD) of $\mathbf{H} = \mathbf{USV}^\top$, $\mathbf{R}$ can be computed as:

$$\mathbf{R} = \mathbf{V} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{pmatrix} \mathbf{U}^\top, \text{ where } d = \det(\mathbf{VU}^\top). \tag{6}$$

On the contrary, the relative translation $\mathbf{t}$ is computed using the plane intersections:

$$\mathbf{t} = \mathbf{o}' - \mathbf{o}. \tag{7}$$

C. Nonlinear Optimization

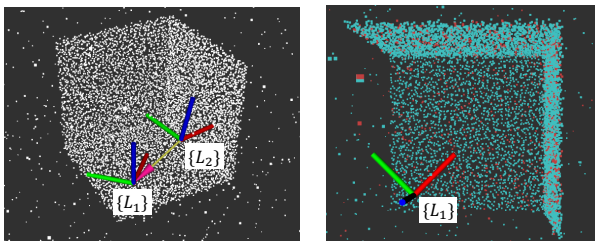
The initial solution is further refined via a nonlinear optimization. By defining a cost function to describe the euclidean distance between Π_i and Π'_i, it can be computed as a sum of the squired distance between a point $\mathbf{p}_n$ and its corresponding plane, i.e., $f_i^2(\mathbf{p}_n)$. Therefore, we can write down the cost function and adopt a Levenberg-Marquardt algorithm for the nonlinear optimization:

$$\begin{aligned} \mathbf{R}^*, \mathbf{t}^* &= \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{3} \mathcal{F}(\mathbf{R}, \mathbf{t}, \Pi'_i, \Pi_i) \\ &= \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{3} \Bigg[\sum_{\mathbf{p}' \in \Pi'_i} f_i^2 (\mathbf{Rp}' + \mathbf{t}) \\ &\qquad + \sum_{\mathbf{p} \in \Pi_i} f_i'^2 (\mathbf{R}'\mathbf{p} + \mathbf{t}') \Bigg]^2, \end{aligned} \tag{8}$$

where $f'_i(\cdot)$ is the counterpart of f_i, and $\mathbf{R}' = \mathbf{R}^{-1}$ as well as $\mathbf{t}' = -\mathbf{R}^{-1}\mathbf{t}$ are a rotation matrix and a translation vector from $\{L_2\}$ to $\{L_1\}$ respectively.

IV. Experiment

To evaluate the proposed extrinsic calibration method, we test it with different configurations of dual lidars. Experiments are presented with synthetic data and real sensor data.



(a) Sythetic data. (b) Calibrated point cloud.

Fig. 4. (a) An example of the synthetic data. (b) The red points and cyan points indicate the point cloud perceived by L_1 and L_2 respectively. We transform the cyan points from $\{L_2\}$ to $\{L_1\}$ using the results provided by our proposed method.

All the resulting values are compared against the ground truth or the values provided by four methods in terms of accuracy.

A. Implementation Details

We adopt `pcl`[1] to preprocess point clouds and implement the RANSAC-based plane fitting. `Eigen`[2] library is applied to implement the Kabsch algorithm, and `Ceres Solver`[3] is used to solve the nonlinear optimization problem. In the optimization, we set the maximum iteration as 1000 and stopping tolerance as $1e^{-3}$.

B. Experiments in Synthetic Data

To verify the performance of the proposed algorithm, we randomly generate 9500 scan points (7500 planar points and 2000 noisy points) in a $10m \times 10m \times 10m$ space. The planar points are generated evenly on three planar surfaces, which are subjected to zero-mean Gaussian noise with a standard deviation of $0.1m$. The rotation angles α on z-axis between Π_1 and Π_2 are set at intervals of $(60°, 120°)$, while Π_3 is set on the bottom, which is orthogonal to Π_1 and Π_2. The noisy points are distributed in the space, which are subjected to zero-mean Gaussian distribution with a standard deviation of 5m. L_1 is set arbitrarily where all the planar surfaces can be observed. Rotations from $\{L_1\}$ to $\{L_2\}$ are randomly generated within $(0°, 20°)$, $(0°, 20°)$, $(0°, 360°)$ on $x-$, $y-$ and $z-$ axis respectively, and translations are generated within $(-1.5, 1.5)m$ respectively. An example of the sensor configuration and the generated points is visualized in Fig. 4(a). In our experiments, we randomly select two configurations with different $\mathbf{R}$ and $\mathbf{t}$ [see Table II (top)] as the ground truth to compare with the resulting values.

The difference in rotation is measured according to the angle difference between the ground truth $\mathbf{R}_{\text{gt}}$ and the resulting rotation $\mathbf{R}_{\text{res}}$, which is calculated as $e_r = \| \log(\mathbf{R}_{\text{gt}}\mathbf{R}_{\text{res}}^{-1})^{\vee} \|_2$[4]. The difference in translation is computed using vector subtraction as $e_t = \|\mathbf{t}_{\text{gt}} - \mathbf{t}_{\text{res}}\|_2$.

For each group of $\mathbf{R}$ and $\mathbf{t}$, we performed 10 trials on the noisy data and computed the mean as well as the standard deviation of the rotation and translation errors. In Fig. 5, blue bars and red lines indicate the mean and standard deviation respectively. Detailed calibration results are shown in Table II (bottom). An example of the calibrated point cloud is shown in Fig. 4(b). In summary, we can see that the rotation and translation errors are tiny on the synthetic data. This proves that the proposed method can successfully calibrate the extrinsic parameters.

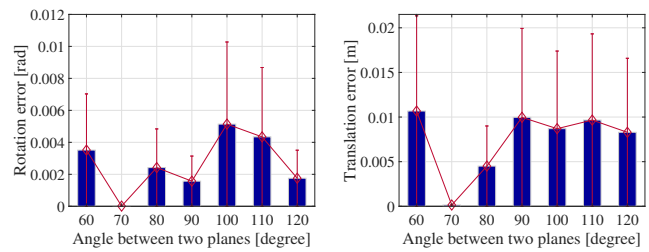
[1] `http://pointclouds.org`
[2] `http://eigen.tuxfamily.org`
[3] `http://ceres-solver.org`
[4] The operator $\boldsymbol{\phi} = \log(\mathbf{R})^{\vee}$ is defined to associate $\mathbf{R}$ in SO(3) to its rotation angle $\varphi \in \mathbb{R}^3$ on the axis.

TABLE II

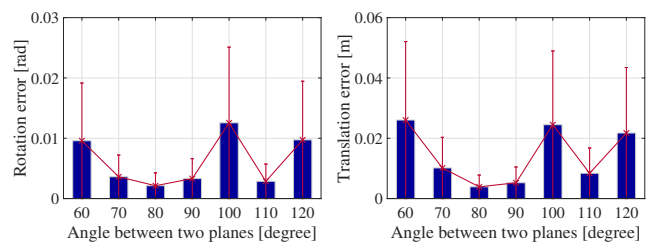
THE GROUND TRUTH OF TWO TESTING CONFIGURATIONS (TOP) AND THE CALIBRATION RESULTS (BOTTOM).

Conf.	Rotation $[rad]$	Translation $[m]$
1	2.7337, -0.3946, -0.1809	0.8766, 0.4672, 1.0474
2	-0.5174, 0.1277, 0.1222	1.3785, -1.3929, 1.3020

Conf.	α $[degree]$	Rotation Error $[rad]$		Translation Error $[m]$	
		mean	std.	mean	std.
1	60	0.0035	0.0035	0.0107	0.0161
	70	0	0	0.0001	0
	80	0.0024	0.0056	0.0045	0.0085
	90	0.0016	0.0040	0.0100	0.0243
	100	0.0051	0.0107	0.0087	0.0178
	110	0.0043	0.0063	0.0097	0.0161
	120	0.0018	0.0051	0.0083	0.0243
2	60	0.0096	0.0104	0.0260	0.0349
	70	0.0036	0.0083	0.0101	0.0274
	80	0.0021	0.0064	0.0039	0.0109
	90	0.0033	0.0071	0.0052	0.0101
	100	0.0126	0.0170	0.0245	0.0339
	110	0.0029	0.0057	0.0084	0.0163
	120	0.0097	0.0139	0.0217	0.0352



(a) Configuration 1



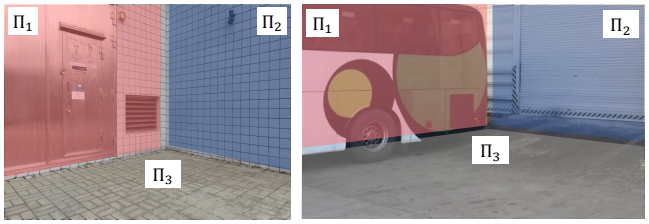
(b) Configuration 2

Fig. 5. Performance analysis using rotation and translation errors on two sensor configurations.

C. Experiments in Real Data

We calibrate a sensor system which consists of three 16-beam RS-Lidars[5] on our vehicle. As presented in Fig. 1(a), these lidars are mounted at the front (L_1), top (L_2), and tail (L_3) position respectively. Especially, L_3 is mounted with approximately 180° rotation offset in yaw. In later sections,

[5] `https://www.robosense.ai/rslidar/rs-lidar-16`



(a) *Easy*: Buildings. (b) *Hard*: Parking lot.

Fig. 6. The calibration environments which can be found in outdoors. They all form a wall corner shape. We extract three linear independent planar surfaces (Π_1, Π_2, and Π_3) for calibration.

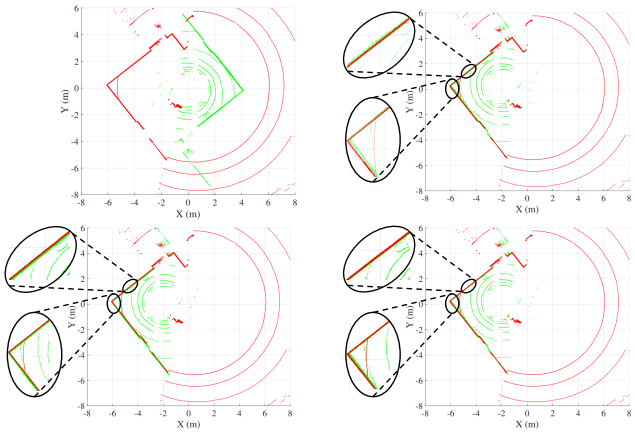


Fig. 7. Top views of the uncalibrated point clouds (top-left) and point clouds calibrated by W/O refinement (top-right), Ground truth (bottom-left) and Proposed (bottom-right) in the *easy* scenario. The point cloud of L_1 and L_3 are denoted by red and green dots respectively.

we denote $L_1 \ominus L_i$ the configuration between L_1 and L_i. The surrounding buildings and ground as the scan planar surfaces are used for calibration. We select two calibration environments in outdoors with two levels (*easy* and *hard*) for calibration, which are shown in Fig. 6.

1) Easy: Calibration is performed in the case of two different configurations: $L_1 \ominus L_2$, as a standard setup, and $L_1 \ominus L_3$, as a challenging setup. We take three methods for comparisons. The former two methods are developed based on the proposed one, but some steps are modified, while the last method is based on the motion-based techniques:

- W/O refinement: The refinement step is removed.
- ICP refinement: The nonlinear optimization refinement is replaced by a point-to-plane ICP [26].
- Motion-based: The motion of lidars are estimated using the LeGO-LOAM [6], and the approach in [22] is implemented to initialize the extrinsic parameters. In refinement, we use the approaches in [27] based on ground surface to obtain the translation on $z-$ axis.

Since the precise extrinsic parameters of the multi-LiDAR system are unknown, we use the values provided by the manufacturer as the ground truth to evaluate these methods.

The results with respect to different configurations are listed in Table III and IV. The estimated extrinsic parameters of the proposed algorithm are quite close to the ground truth. Regarding the relative rotation and translation errors,

TABLE III

ESTIMATED EXTRINSIC PARAMETERS OF $L_1 \ominus L_2$

Method	Rotation $[rad]$	Error $[rad]$
Ground truth	0.0096, 0.0989, 0.0425	—
W/O refinement	-0.0046, -0.1138, -0.0193	0.2221
ICP refinement	-0.0882, -0.4254, 0.0462	0.5333
Motion-based	0.2269, -0.0451, -0.0007	0.2332
Proposed	0.0012, 0.0892, 0.0276	**0.0203**
	Translation $[m]$	Error $[m]$
Ground truth	0.377002, -0.03309009, -1.23236	—
W/O refinement	0.314979, 0.0509327, -1.23395	0.102694
ICP refinement	1.21884, -0.0670357, -1.55964	0.903941
Motion-based	0.2115, 0.4892, -1.1903	0.5474
Proposed	0.336322, 0.00271319, -1.19076	**0.067196**

TABLE IV

ESTIMATED EXTRINSIC PARAMETERS OF $L_1 \ominus L_3$

Method	Rotation $[rad]$	Error $[rad]$
Ground truth	-0.0161, -0.0192, 3.1392	—
W/O refinement	-0.0527, -0.0212, 3.1328	0.0373
ICP refinement	-0.0333, -0.1329, 3.1277	0.1605
Motion-based	-0.0044, 0.0037, -2.6462	0.4986
Proposed	0.0004, -0.0020, 3.1375	**0.0238**
	Translation $[m]$	Error $[m]$
Ground truth	-1.96443, 0.04073154, -1.13756	—
W/O refinement	-1.92115, 0.0968707, -0.80199	0.341959
ICP refinement	-1.95929, 0.0473068, -0.383619	0.753959
Motion-based	-1.7794, -0.4670, -1.11136	0.5472
Proposed	-1.9103, 0.052992, -1.0744	**0.08357**

our method achieved $[0.0203rad, 0.067196m]$ of $L_1 \ominus L_2$ and $[0.0238rad, 0.08357m]$ of $L_1 \ominus L_3$ respectively. We observe that larger errors are caused by ICP refinement and Motion-based methods. Regarding the ICP approach, failure is caused due to the wrong matching of corresponding planes. About the Motion-based approach, the drift and uncertainty of the estimated motion would significantly reduce the accuracy of the calibration results. We can also see that the proposed method performs better than the w/o refinement method since the nonlinear optimization could further reduce the noise. During the calibration, the time for optimization is around $27.3s$ and $35.8s$ of the two configurations.

To evaluate the calibration results qualitatively, we transform the point cloud in $\{L_3\}$ to $\{L_1\}$ using the extrinsic parameters provided by ground truth, W/O refinement, and Proposed respectively. The top view of these fused point cloud can be seen in Fig. 7. We observe that the point clouds calibrated by the Proposed approach have litter uncertainty on the planar surfaces. Therefore, our algorithm can successfully calibrate dual lidars in real data with low error.

2) Hard: In the following, we study the performance of our approach in the *hard* scenario. This scenario is more challenging because there are several objects on these planes that may influence the plane extraction and optimization results. In this experiment, the configuration $L_1 \ominus L_3$ is calibrated. The relative rotation and translation errors compared with the ground truth are $[0.0292rad, 0.0930m]$, while the optimization time is around $106.4s$. The fused point clouds are shown in Fig. 8, where the planar surfaces are registered

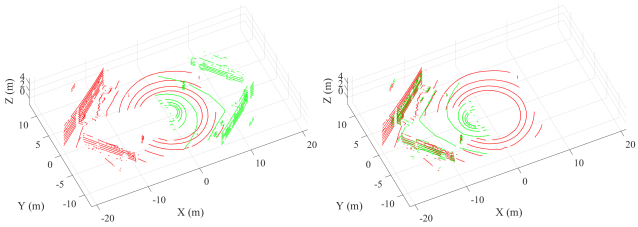


Fig. 8. Top views of the uncalibrated point clouds (left) and point clouds (right) calibrated by the proposed method in the *hard* scenario. The point cloud of L_1 and L_3 are denoted by red and green dots respectively..

well without much offset. We conclude that the extrinsic parameters can be recovered in this scenario.

D. Discussion

We have three certain assumptions in this method: lidars are horizontally mounted; they share large overlapping fields of view with each other and three planar surfaces are provided for calibration. Therefore, the proposed method may fail in several cases. For instance, if lidars are mounted at an arbitrary orientation, a wrong initialization will be caused. Another case is that if planar surfaces are wrongly detected, their correspondences will be mismatched.

V. Conclusion and Future Work

In this paper, we have presented an automatic algorithm for calibrating a dual-lidar system without any additional sensors, artificial landmarks, or information about the motion provided by sensors. The RANSAC-based model fitting approach is used to extract three linearly independent planar surfaces from scan points. Linear constraints for initialization are provided by the geometric structure of these planar surfaces, and a final nonlinear optimization is then used to refine the estimates. Our proposed method has been demonstrated to recover the extrinsic parameters of a dual-lidar system with rotation and translation error smaller than $0.05rad$ and $0.1m$ in different testing conditions.

It would be beneficial to determine the parameters in more general cases, e.g., sensors have non-overlapping fields of view, or they are arbitrarily mounted on a vehicle. Such problems may be solved by developing a simultaneous localization and mapping system, where the extrinsic parameters are jointly optimized with the odometry and map within a unified framework.

References

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

[2] G. Xie, T. Xu, C. Isert, M. Aeberhard, S. Li, and M. Liu, "Online active calibration for a multi-lrf system," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 806–811.

[3] 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.

[4] Q. Liao, M. Liu, L. Tai, and H. Ye, "Extrinsic calibration of 3d range finder and camera without auxiliary object or human intervention," 2017.

[5] Q. Liao, Z. Chen, Y. Liu, Z. Wang, and M. Liu, "Extrinsic calibration of lidar and camera with polygon," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 200–205.

[6] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," 2018.

[7] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.

[8] P. Yun, L. Tai, Y. Wang, C. Liu, and M. Liu, "Focal loss in 3d object detection," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1263–1270, April 2019.

[9] Z. Taylor, J. Nieto, and D. Johnson, "Automatic calibration of multi-modal sensor systems using a gradient orientation measure," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1293–1300.

[10] L. Heng, B. Li, and M. Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1793–1800.

[11] Z. Yang, T. Liu, and S. Shen, "Self-calibrating multi-camera visual-inertial fusion for autonomous mavs," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 4984–4991.

[12] D.-G. Choi, Y. Bok, J.-S. Kim, and I. S. Kweon, "Extrinsic calibration of 2-d lidars using two orthogonal planes," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 83–98, 2016.

[13] C. Gao and J. R. Spletzer, "On-line calibration of multiple lidars on a mobile vehicle platform," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 279–284.

[14] M. He, H. Zhao, F. Davoine, J. Cui, and H. Zha, "Pairwise lidar calibration using multi-type 3d geometric features in natural scene," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1828–1835.

[15] M. He, H. Zhao, J. Cui, and H. Zha, "Calibration method for multiple 2d lidars system," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3034–3041.

[16] Y. Xie, R. Shao, P. Guli, B. Li, and L. Wang, "Infrastructure based calibration of a multi-camera and multi-lidar system using apriltags," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 605–610.

[17] J. Röwekämper, M. Ruhnke, B. Steder, W. Burgard, and G. D. Tipaldi, "Automatic extrinsic calibration of multiple laser range sensors with little overlap," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2072–2077.

[18] J. Quenzel, N. Papenberg, and S. Behnke, "Robust extrinsic calibration of multiple stationary laser range finders," in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1332–1339.

[19] J. Levinson and S. Thrun, "Automatic online calibration of cameras and lasers." in *Robotics: Science and Systems*, vol. 2, 2013.

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

[21] Z. Taylor and J. Nieto, "Motion-based calibration of multimodal sensor arrays," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4843–4850.

[22] Z. Taylor and J. Niet, "Motion-based calibration of multimodal sensor extrinsics and timing offset estimation," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1215–1229, 2016.

[23] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[24] R. Fan, J. Jiao, J. Pan, H. Huang, S. Shen, and M. Liu, "Real-time dense stereo embedded in a uav for road inspection," 2019.

[25] W. Kabsch, "A discussion of the solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 34, no. 5, pp. 827–828, 1978.

[26] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing icp variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[27] G. Pandey, S. Giri, and J. R. Mcbride, "Alignment of 3d point clouds with a dominant ground plane," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 2143–2150.