

# Automatic Multi-camera Multi-LiDAR Extrinsic Calibration using Geometric Corners

Feiyi Chen<sup>1</sup>, Shuyang Zhang<sup>1</sup>, Xupeng Xie<sup>1</sup>, Qingwen Zhang<sup>2</sup>, Yutong Wang<sup>2</sup>, Jianhao Jiao<sup>1</sup> and Ming Liu<sup>1</sup>

**Abstract**—In autonomous driving systems, multiple sensors are installed to provide complementary information. Extrinsic calibration is the prerequisite of multi-sensor fusion, which is critical to the accuracy and performance of the fused system. In this paper, we propose a unified algorithm for extrinsic calibration among multiple cameras and light detection and ranging (LiDAR) sensors using only a standard checkerboard. The geometric information of the checkerboard, containing plane coefficients, boundary points and geometric corners, is extracted automatically with a lightweight approach by leveraging the characteristics of mechanical LiDARs. A coarse-to-fine manner is adopted in the extrinsic calibration by utilizing these features in different stages. Our algorithm has been experimentally validated through simulations and real-world experiments. The experimental results show that the proposed method is capable of providing accurate extrinsic parameters and is robust in different levels of Gaussian noise. All procedures in our approach operate automatically, without any manual intervention, which is beneficial to practical applications.

## I. INTRODUCTION

In autonomous driving systems, multi-sensor fusion provides a wider perception field and richer surrounding information, which enables an autonomous vehicle to operate robustly. Cameras and LiDARs, the most commonly used sensors in robotics, can provide complementary information in such systems. Cameras capture the dense texture of surroundings, while LiDARs actively measure sparse 3D structures. To jointly use these raw data from different sources, extrinsic calibration must be carried out by aligning raw measurements into the same coordinate system.

In most cases, autonomous vehicles are equipped with multiple cameras and LiDARs, not only a single camera-LiDAR pair. Due to the difference in the sensors' modalities and fields of view (FOVs), the calibration approaches for each sensor pair must be designed and adjusted specifically. However, it is tedious to calibrate camera-to-camera, LiDAR-to-LiDAR and LiDAR-to-camera extrinsic parameters using different methods. Conducting calibration under the same framework is thus crucial but challenging. The core problem is to extract and find the association of the common features from the multi-modality sensors. One direction is to directly find and align the common features from the environment. For this, the data intensity and geometric boundary are chosen by some popular methods [1], [2]. But it is non-trivial to extract and find the association of these features

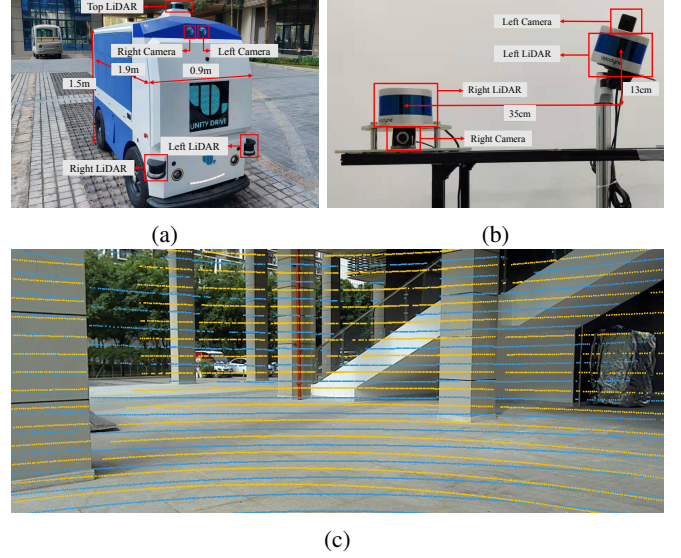


Fig. 1: (a) Multi-sensor configuration in real autonomous vehicle. (b) The custom-built mobile platform. (c) The projection result of a point cloud from two VLP-16 LiDARs in the corresponding image. Two LiDAR points are marked with blue and yellow color respectively.

because of the low data resolution in the LiDAR frame and unavailability of 3D information in the camera frame. Another direction is to firstly find a common target and then calibrate the sensors using the common features on the target, which is more robust and does not rely on the environment. However, not all targets are applicable, like building corner [3], which is hard to be adjusted to the overlapping FOVs of different sensor pairs. On the other hand, the checkerboard is the mostly common used and many algorithms [4], [5] have been proposed based on it. But it still has limitations in 1) automatic feature extraction in various scenarios, 2) fully utilizing the prior information of the checkerboard, and 3) the correct data association between source and target sensors.

In our work, we choose a *checkerboard* to calibrate multiple cameras and multiple LiDARs system for its availability and portability. Other fiducial markers, like AprilTag [6] and ArUco [7], are also applicable. We consider these limitations using the geometric information of the checkerboard extracted from sensor data independently, including the plane coefficients, boundary points and geometric corners. The proposed algorithm is verified through simulations and real-world experiments quantitatively and qualitatively. Our contributions are summarized as follows:

<sup>1</sup>Feiyi Chen, Shuyang Zhang, Xupeng Xie, Jianhao Jiao and Ming Liu are with the Department of Electronic and Computer Engineering, the Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, China (email: fchenak@connect.ust.hk, eelium@ust.hk)

<sup>2</sup>Yutong Wang and Qingwen Zhang are with the System Hub, the HongKong University of Science and Technology (GZ), Guangzhou, China.

- A unified framework that calibrates LiDAR-to-LiDAR, camera-to-camera and LiDAR-to-camera extrinsic parameters using a standard checkerboard under the same optimization equation. The accuracy of translation reaches 1cm and rotation reaches 0.3 degrees.
- A lightweight board extraction algorithm is proposed, which is accurate and robust in various scenarios. All procedures in the calibration process are automatically conducted without manual intervention.
- Our code is publicly released on Github<sup>1</sup> to benefit the community.

## II. RELATED WORK

Extrinsic calibration between cameras and LiDARs has been studied by researchers worldwide for decades, and various approaches have been proposed. We summarize them into three categories: camera-to-camera, LiDAR-to-LiDAR and LiDAR-to-camera calibration.

### A. Camera-to-Camera Calibration

To get the camera's 3D position with respect to another one, classic epipolar geometry [8] is the most popular strategy. Many tasks, like visual SLAM and 3D reconstruction, are developed based on it, while there is a scale uncertainty on translation, and the physical size of the environment needs to be specified to acquire the scale. On the other hand, camera position can be restored with the assistance of a fiducial marker [6], [9] using the Perspective-n-Point (PnP) [10] algorithm. Other off-the-shelf toolkits are available on Matlab and OpenCV. In our proposed method, for the checkerboard can be fully represented by four 3D geometric corners, the extrinsic parameters between two cameras are calculated by aligning the paired corner points based on PnP.

### B. LiDAR-to-LiDAR Calibration

To achieve LiDAR-to-LiDAR calibration, Gao *et al.* [11] pasted several retro-reflective tapes upon vertical poles. Extrinsic parameters were recovered by registering the segmented scan points to a common world frame. He *et al.* [12] utilized multi-type geometric features in the surroundings, like lines, planes and cones, to register the features into the world frame with manual intervention. Other sensors like GPS and IMU were involved in localizing the vehicle, which made this calibration process complicated. Jiao *et al.* [3] leveraged three orthogonal planar surfaces to constrain the parameters by minimizing the plane-plane distance. Maddern *et al.* [13] utilized Renyi entropy to find the extrinsic parameters, and the intuition behind it is to maximize the compactness of the accumulated point cloud in the global coordinate system. The drawback of all these LiDAR-to-LiDAR calibration approaches is that they are hard to extend to the other two types of calibration for different feature representations in multi-modality sensors.

### C. LiDAR-to-Camera Calibration

In the literature, extrinsic parameters between a LiDAR and camera can be calibrated with or without a calibration target.

1) *With a target:* Many calibration targets have been designed, and among them, the checkerboard is the most commonly used. Zhou *et al.* [4] utilized the boundary points and 3D edges of the checkerboard to minimize overall point-to-line and point-to-plane distances. Lee *et al.* [5] proposed a more complete framework to calibrate multiple cameras and multiple LiDARs using a checkerboard, but they only used point-plane constraints.

2) *Without a target:* The geometric boundary [2] and data intensity [1] are the main clues to calibrate the extrinsic parameters. Kang *et al.* [2] constructed a point-pixel loss function based on a Gaussian mixture model to align the geometric boundary in the environment. Pandey *et al.* [1] calculated the mutual information of the intensity on each sensor's data. They proved that the mutual information reaches its maximum when the relative transformation is correct. Other strategies like SFM [14] and semantic information [15] are also applicable to solve the issue of calibration. However, these methods are not robust and relies on the scenario as well, which makes it difficult to be deployed.

Generally, extrinsic calibration is done to correct the misalignment of common features between the target and source sensors. To provide robust common features among multiple cameras and LiDARs, we choose the *checkerboard* as the calibration target in our work and summarize the three categories of calibration into one minimization equation solved by singular value decomposition (SVD).

## III. AUTOMATIC FEATURE EXTRACTION

In this section, we present the automatic feature extraction on the checkerboard, which includes the plane coefficients, boundary points and geometric corners.

### A. Board Extraction

In the LiDAR frame, we first set a maximum search radius  $\mu_1$  to eliminate the influence of irrelevant points. A mechanical LiDAR contains multiple emitters aligned vertically, and it rotates them physically at a high rate to obtain a 360° horizontal field of view. We call the points set from the same emitter a *ring*. If the board is placed in front of the LiDAR, the points that hit the board should form a straight line in each ring. We utilize this feature to filter out most outliers by clustering points into line segments based on the Euclidean distance of two consecutive points. Afterward, the curved and oversize line segments are filtered out. By conducting principal component analysis (PCA) on each line segment, its eigenvalues are computed and represented by  $[\lambda_1, \lambda_2, \lambda_3]$  where  $\lambda_1 \geq \lambda_2 \geq \lambda_3$ . If the ratio  $\lambda_1/\lambda_2$  is less than an empirical threshold  $\mu_2$  or the length of the line segment is larger than  $\mu_3$ , this line segment will be removed.

The DBScan [16] algorithm is adopted to cluster the remaining points, and the match error is computed between each voxel down-sampled cluster and template board points

<sup>1</sup><https://github.com/chenfeiyi/AutoExCali.git>

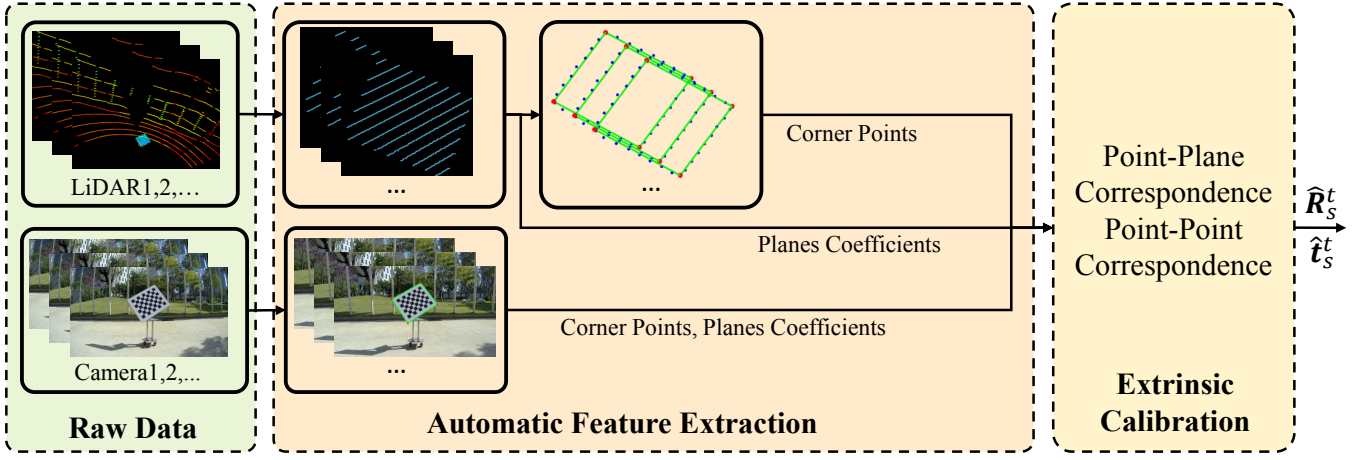


Fig. 2: The system overview of our proposed unified calibration using geometric features of the checkerboard

using the iterative closest point (ICP) [17] algorithm. As the size of the checkerboard is given, template board points are sampled evenly in the checkerboard coordinate system. The cluster with minimum match error should be the correct board points. Additionally, to ensure robustness and enough constraints, we discard the cluster if it contains fewer than four rings or the match error is greater than an empirical threshold  $\mu_4$ . The details of LiDAR board extraction are summarized in Algorithm 1, which is validated in three scenarios, as shown in Fig. 3. We collected 100 frames of point cloud with different poses to evaluate the proposed board extraction algorithm. Compared with the manually extracted board points, the board points extracted by our proposed algorithm is considered positive when the intersection over union (IoU) is greater than 0.95. The final result shows that the accuracy in indoor cases reaches 0.97, and 0.98 is reached for outdoor cases, which indicates that the proposed board extraction is accurate and robust.

In the image frame, the geometric corners and plane coefficients of the checkerboard can be easily extracted by existing tools based on the PnP algorithm.

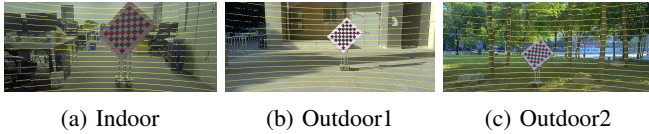


Fig. 3: The validation of the board extraction algorithm in three different scenarios. (a) Indoor case with structured surroundings. (b) Open outdoor cases. (c) Outdoors surrounded with trees and rocks.

### B. Geometric Corners Extraction

After obtaining the board points, random sample consensus (RANSAC) plane-fitting method is applied to extract the plane coefficients, and all board points are vertically projected onto the extracted plane. Afterward, the boundary points are obtained from the starting and ending points of each ring in these projected board points. To utilize the

boundaries of the checkerboard, Zhou *et al.* [4] minimized the point-to-line distance. However, it has several limitations. They applied the RANSAC algorithm to fit the line along

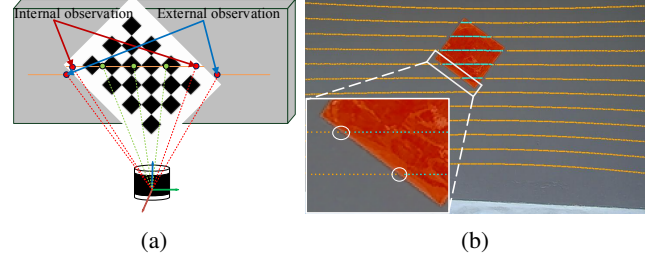


Fig. 4: The internal (foreground) and external (background) observations of the boundaries. (a) The internal observations of the boundary lie on the board and the external observations hit the background. (b) A red reflective tape is pasted on the wall. The orange points are the background and the blue points hit the tape. The marked white circles show that the boundary of the tape should be in the middle of the external and internal observations.

with the boundary points, which implicitly assumes that the boundary points of the checkerboard are distributed symmetrically along the board edges. In real cases, however, the boundary points (internal observations) are only distributed along one side of the edge, as shown in Fig. 4, and the background points (external observations) are difficult to utilize because of the zero-valued and multi-valued mapping problems [18]. Moreover, in real case, there exist many fake points of foreground object near the boundary, as described in [18]. Secondly, at least three points in each boundary are needed to execute the RANSAC algorithm. Therefore, the RANSAC line-fitting approach will cause inaccurate edge extraction to some extent, and it is unstable to deal with the fake boundary points. Instead, we directly use a rectangle to fit all boundary points. As the checkerboard size is given, the ideal rectangle in the checkerboard coordinate system is transformed to the LiDAR coordinate system to fit all internal observations. We first initialize the rigid

transformation  $\mathbf{R}_L^b, \mathbf{t}_L^b$  from the LiDAR coordinate system to the checkerboard coordinate system with the ICP algorithm, where  $\mathbf{R}_L^b \in SO(3)$  and  $\mathbf{t}_L^b \in \mathbb{R}^3$ . Afterward, it is optimized by iteratively minimizing the point-to-line distance using the following equation:

$$\hat{\mathbf{R}}_L^b, \hat{\mathbf{t}}_L^b = \arg \min_{\mathbf{R}_L^b, \mathbf{t}_L^b} \sum_{i=1}^4 \sum_{j=1}^{M_i} \|\mathbf{A}_i(\mathbf{R}_L^b \mathbf{p}_{ij}^l + \mathbf{t}_L^b - \mathbf{c}_i^b)\|^2, \quad (1)$$

where  $\mathbf{A}_i = \mathbf{I} - (\mathbf{c}_{i+1}^b - \mathbf{c}_i^b)(\mathbf{c}_{i+1}^b - \mathbf{c}_i^b)^\top$ ,  $\mathbf{p}_{ij}$  represents the  $j^{th}$  points in the  $i^{th}$  edge and  $\mathbf{c}_i^b$  is the  $i^{th}$  geometric corner in the checkerboard coordinate system, which is sorted clock-wise around the plane normal and  $\mathbf{c}_5^b = \mathbf{c}_1^b$ . The point-line correspondences can be found by nearest neighbor search (NSS) strategy. To avoid local minima, the scatter search algorithm [19] is adopted, with multiple trials generated around the current solution. Among the multiple local minimas found by the scatter algorithm, we select the result with the lowest loss as the global solution. Additionally, the rotation matrix is converted into Lie algebra  $\mathfrak{so}(3)$  and the Gauss-Newton method is adopted. Our final geometric corners are calculated by  $\mathbf{c}_i^L = (\mathbf{R}_L^b)^\top \mathbf{c}_i^b - (\mathbf{R}_L^b)^\top \mathbf{t}_L^b$ .

---

#### Algorithm 1 Automatic Board Points Extraction

---

**Require:** Raw points cloud  $\mathbf{P}$  and template board points  $\mathbf{X}$ . Search radius  $\mu_1$ , eigenvalue threshold  $\mu_2$ , maximum length of line segment  $\mu_3$ , match error threshold  $\mu_4$

**Ensure:** Board points set  $\mathbf{Q}$

- 1:  $\mathbf{Y} \leftarrow \{\|\mathbf{P}_i\| < \mu_1\}$
- 2:  $\mathbf{Z} \leftarrow \text{LineSegment}(\mathbf{Y})$
- 3:  $\mathbf{Y} \leftarrow \{\}$
- 4: **for**  $i_{th}$  line  $\mathbf{Z}_i$  in  $\mathbf{Z}$  **do**
- 5:    $\lambda_1, \lambda_2, \lambda_3 \leftarrow \text{PCA}(\mathbf{Z}_i)$
- 6:   **if**  $\frac{\lambda_1}{\lambda_2} > \mu_2$  and  $\text{Length}(\mathbf{Z}_i) < \mu_3$  **then**
- 7:      $\mathbf{Y} \leftarrow \mathbf{Y} \cup \mathbf{Z}_i$
- 8:   **end if**
- 9: **end for**
- 10:  $\mathbf{C} \leftarrow \text{DBScan}(\mathbf{Y})$
- 11:  $err_{min} \leftarrow \infty$
- 12: **for**  $i_{th}$  cluster  $\mathbf{C}_i$  **do**
- 13:    $\mathbf{C}_i \leftarrow \text{VoxelDownSample}(\mathbf{C}_i)$
- 14:    $err \leftarrow \text{MatchErrUsingICP}(\mathbf{C}_i, \mathbf{X})$
- 15:   **if**  $err < err_{min}$  and  $err < \mu_4$  **then**
- 16:      $err_{min} \leftarrow err$
- 17:      $\mathbf{Q} \leftarrow \mathbf{C}_i$
- 18:   **end if**
- 19: **end for**
- 20: **return**  $\mathbf{Q}$ .

---

#### IV. EXTRINSIC CALIBRATION

In this section, we present the unified calibration algorithm, and the detailed process is shown in Fig. 2. To simplify notation, the extrinsic parameters between the source sensor and target sensor are denoted by the rotation matrix  $\mathbf{R}_s^t$  and

translation vector  $\mathbf{t}_s^t$ . Note that the source and target sensor could be a camera or LiDAR.

We utilize point-plane and point-point correspondence to calibrate the extrinsic parameters. In the point-plane correspondence, the plane coefficients of the  $i^{th}$  checkerboard in each sensor are denoted by  $[\mathbf{n}_i^s, d_i^s], [\mathbf{n}_i^t, d_i^t]$ . We can formulate the following residuals:

$$\varepsilon_1 = \|\mathbf{R}_s^t \mathbf{n}_i^s - \mathbf{n}_i^t\|, \quad (2)$$

$$\varepsilon_2 = \mathbf{n}_i^t \cdot (\mathbf{R}_s^t \mathbf{p}_i^s + \mathbf{t}_s^t) + d_i^t, \quad (3)$$

where  $(\cdot)$  denotes the inner product. In the ideal case,  $\varepsilon_1 = \varepsilon_2 = 0$ , where  $\varepsilon_1$  represents the Euclidean distance of the end points between two corresponding plane normal vectors and  $\varepsilon_2$  denotes the Euclidean distance from the board points  $p_i^s$  to the corresponding plane.

In the point-point correspondence, the board corners are extracted independently in each sensor's frame. The  $j^{th}$  corner in the  $i^{th}$  checkerboard is represented with  $\mathbf{c}_{ij}$ . Once we build the point-point correspondence with NSS, we can simplify the notation to  $\mathbf{c}_i$ . The error function is formulated by aligning the corner points between two sensors:

$$\varepsilon_3 = \mathbf{R}_s^t \mathbf{c}_i^s + \mathbf{t}_s^t - \mathbf{c}_i^t. \quad (4)$$

In the coarse stage, we first initialize the extrinsic parameters using the plane constraints and the rotation is calculated by

$$\tilde{\mathbf{R}}_s^t = \arg \min_{\mathbf{R}_s^t} \sum_{i=1}^N \|\mathbf{R}_s^t \mathbf{n}_i^s - \mathbf{n}_i^t\|^2. \quad (5)$$

This problem has a closed-form solution [20] using SVD. By taking the rotation into Eq. 3, the translation can be estimated by solving a least-squares problem:

$$\begin{bmatrix} \mathbf{n}_1^t & \mathbf{n}_2^t & \dots & \mathbf{n}_N^t \end{bmatrix}^\top \mathbf{t}_s^t = \begin{bmatrix} -d_1^t - \mathbf{n}_1^t \cdot \tilde{\mathbf{R}}_s^t \mathbf{p}_1^s \\ -d_2^t - \mathbf{n}_2^t \cdot \tilde{\mathbf{R}}_s^t \mathbf{p}_2^s \\ \vdots \\ -d_N^t - \mathbf{n}_N^t \cdot \tilde{\mathbf{R}}_s^t \mathbf{p}_N^s \end{bmatrix}. \quad (6)$$

After taking these initial parameters, we can calculate the final transformation by minimizing the point-to-point distance:

$$\hat{\mathbf{R}}_s^t, \hat{\mathbf{t}}_s^t = \arg \min_{\mathbf{R}_s^t, \mathbf{t}_s^t} \sum_{i=1}^{4N} \|\mathbf{R}_s^t \mathbf{c}_i^s + \mathbf{t}_s^t - \mathbf{c}_i^t\|^2. \quad (7)$$

As mentioned before, this problem has close-form solution based on SVD.

#### V. EXPERIMENTS

##### A. Experimental Setup

Our proposed algorithm is evaluated both on the synthetic and real-world datasets. The custom-built mobile platform are equipped with two VLP-16 LiDARs and two SENSING-GSML-0143-H090 cameras with resolution  $1280 \times 720$  and FOV  $90^\circ$ , as shown in Fig. 1b. The checkerboard is a wooden board of 0.77 m in width and 0.63 m in length, and the pattern size is  $0.07\text{m} \times 0.07\text{m}$ .



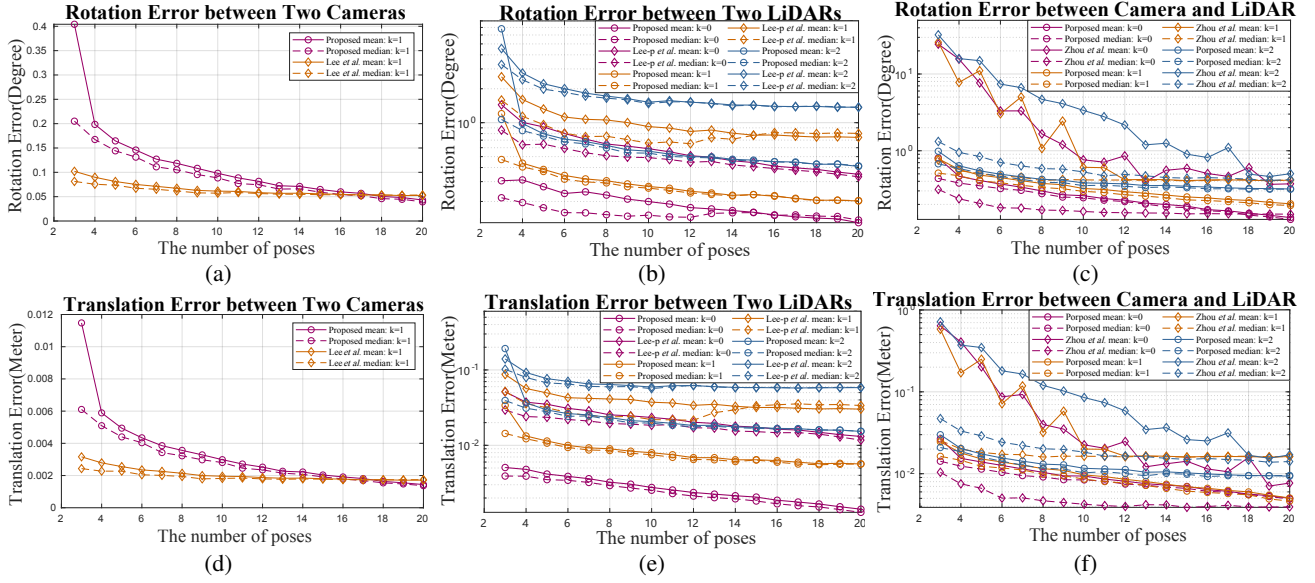


Fig. 5: Rotation and translation error in synthetic dataset. Gaussian noise  $N(0, k\sigma^2)$  is added to the sensors' captured data where  $k \in \{0, 1, 2\}$ . Specifically, noise covariance is set to  $\sigma_C = 0.007$  for monocular camera and  $\sigma_L = 0.015$  for LiDAR.

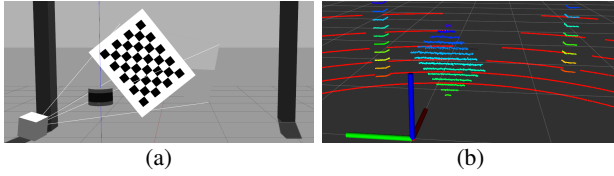


Fig. 6: Simulation environment in gazebo

For the synthetic data, the environment was built on Gazebo [21], which is shown in Fig. 6. We mimic the real world with the same sensor configuration and the board size. Different levels of Gaussian noise  $N(0, k\sigma^2)$  are added to the sensors' captured data, where  $k \in \{0, 1, 2\}$ . Specifically, the noise covariance of the monocular camera is set to  $\sigma_C = 0.007$  and  $\sigma_L = 0.015$  for LiDAR range measurements. In addition, the empirical thresholds in the board extraction are set to  $\mu_1 = 5, \mu_2 = 0.9, \mu_3 = 1$  and  $\mu_4 = 0.06$ . Note that  $\mu_3, \mu_4$  need to be adjusted based on the board size.

### B. Evaluation Metrics

- 1) *Transformation error*  $e_\theta, e_t$  computes the distance between the estimation  $\hat{\mathbf{T}}$  and the ground truth  $\mathbf{T}$ , which contains rotation error  $e_\theta$  and translation error  $e_t$ ,

$$e_\theta = \|\ln(\delta\mathbf{R})^\vee\|_2, \quad (8)$$

$$e_t = \|\delta\mathbf{t}\|_2, \quad (9)$$

$$\delta\mathbf{T} = \hat{\mathbf{T}}\mathbf{T}^{-1} = \begin{bmatrix} \delta\mathbf{R} & \delta\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \quad (10)$$

where  $^\vee$  converts a skew-symmetric matrix to a  $3 \times 1$  vector.

- 2) *Mean projection error*  $e_{PE}$  computes the euclidean distance between the projected points and the detected

pixel geometric corners.

$$e_{PE} = \frac{1}{N} \sum_{i=1}^N \left\| \frac{1}{z_i} \mathbf{K}(\mathbf{R}_s^t \mathbf{p}_i + \mathbf{t}_s^t) - \mathbf{p}_x \right\|_2, \quad (11)$$

$$\begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^\top = \mathbf{K}(\mathbf{R}_s^t \mathbf{p}_i + \mathbf{t}_s^t), \quad (12)$$

where  $\mathbf{K}$  denotes the camera intrinsic matrix,  $\mathbf{p}_i$  is the 3D point from the source sensor and  $\mathbf{p}_x$  is the corresponding pixel in the image plane.

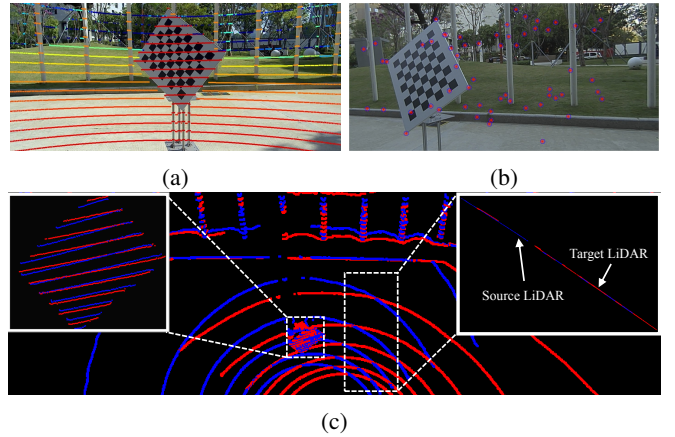


Fig. 7: The calibration results in real world. (a) LiDAR-to-camera: The projection of point cloud in image plane. (b) camera-to-camera: The red star points are geometric corners in target camera and blue circle points are geometric corners in source camera. Multiple frames of sensory data are projected. (c) LiDAR-to-LiDAR: The red points are from the target LiDAR, and the blue points are the points transformed from the source LiDAR.

### C. Evaluation in Synthetic Dataset

For the LiDAR-to-camera calibration, we compared our method with Zhou’s approach [4] implemented by Matlab [22]. For the camera-to-camera calibration, we took Lee’s approach [5] as the baseline. For the LiDAR-to-LiDAR calibration, Lee’s approach was hard to converge for only taking two LiDARs’ observations in the global optimization. So we actually minimize point-to-plane distance where points refer to the board points in source LiDAR, and we use Lee-p to represent this planar-based approach. We randomly collected 27 pairs of cameras and LiDARs data.  $N$  frames of paired data were randomly chosen to evaluate the influence of the number of poses where  $N \in [3, 20]$ . Note that the checkerboard should be visible in both sensors’ FOVs and placed diagonally, just like Fig. 7a. Each algorithm was repeated 1000 times. The result is shown in Fig. 5. They share a similar pattern that the transformation error decreases with the increase of pose number. For the LiDAR-to-camera calibration, it clearly shows that our method has smaller rotation and translation error compared to [4]. The mean and media estimation stay closer than Zhou’s method, which indicates the calibration result is more stable in different levels of Gaussian noise. For the LiDAR-to-LiDAR calibration, it shows that the result of the proposed method with  $k = 2$  reaches similar accuracy compared to Lee’s approach with  $k = 0$ . Additionally, fewer poses are needed using our proposed method to reach the same accuracy in camera-LiDAR and LiDAR-LiDAR pair. However, Lee’s approach outperforms our method in the camera-to-camera calibration for fully using all pattern corners, and Lee’s method only inferred the board 3D pose on one side, which reduced the accumulation of the 3D pose inference error. Specifically, for  $k = 1$ , which noise level is closest to the real world, the rotation error can be reduced to  $0.3^\circ$  in the LiDAR-to-LiDAR and LiDAR-to-camera calibration and  $0.1^\circ$  is reached in the camera-to-camera calibration. The translation error for all calibration is lower than 1cm.

### D. Validation in Real-World Dataset

For the ground truth of rigid transformation among sensors is unavailable in real world, other extrinsic calibration methods were adopted and we took their results as the “pseudo ground truth”. In the camera-LiDAR configuration, we manually selected 2D pixel corners in camera and the corresponding 3D corner points in LiDAR to calibrate the “pseudo ground truth” using the PnP algorithm in Matlab. We also utilized these manually selected 2D-3D pixel-point correspondences to measure  $e_{PE}$ . In the LiDAR-LiDAR configuration, we took the result of Jiao’s approach [3] as the “pseudo ground truth”, and  $e_{PE}$  is inapplicable in this case. For the camera-to-camera calibration, we only took  $e_{PE}$  as the evaluation metric to evaluate the performance. In real world experiment, we fix the number of frame used in calibration. And 10 frames of sensory data were randomly chosen in each sensors’ pair from the collected 27 pairs of data using the custom-built platform (see Fig. 1b). Likewise, the experiments were repeated 1000 times. The results are

shown in Table I. It verifies that the proposed method can provide accurate extrinsics and outperforms Lee’s approach in LiDAR-to-LiDAR calibration and Zhou’s approach in LiDAR-to-camera calibration, which is consistent with the previous results in simulations.

Metric	Cam-to-Cam		LiDAR-to-LiDAR		LiDAR-to-Cam	
	Ours	Lee [5]	Ours	Lee-p [5]	Ours	Zhou [4]
$\widetilde{e}_{PE}(\text{px})$	0.50	<b>0.26</b>	-	-	<b>2.71</b>	3.45
$\widetilde{e}_\theta(\text{degree})$	-	-	<b>0.31</b>	2.33	<b>0.32</b>	0.62
$\widetilde{e}_t(\text{m})$	-	-	<b>0.010</b>	0.074	<b>0.021</b>	0.037

TABLE I: The median rotation, translation and corner reprojection error on real-world dataset

In the real application on autonomous driving, we qualitatively validated our algorithm on the real vehicle equipped with two HEISAI-PandarXT-16 LiDARs on both sides, one VLP-16 LiDAR on the top and two SENSING-GSML-0143-H90 cameras towards front with the resolution  $1280 \times 720$  and FOVs  $90^\circ$ , as shown in Fig. 1a. 10 frames of data were collected to calibrate the extrinsic parameters and the qualitative result is shown in Fig. 7. Especially, we can clearly see that the laser points precisely hit the flagpole in the projection of point cloud in image. In Fig. 7c, after calibration, the data from multiple LiDARs is fused in the same coordinate system. And the shape of the flagpole becomes more clear. Compared with single LiDAR, the multi-LiDAR system increases the perception resolution, which is critical for the autonomous vehicle to operate in complicated environments.

## VI. CONCLUSION

In this paper, we proposed a unified framework to automatically calibrate Camera-to-Camera, LiDAR-to-LiDAR and LiDAR-to-Camera extrinsic parameters using a standard checkerboard by aligning the geometric corners. And conventional SVD is utilized to solve the final ICP-like problem. Additionally, we leveraged the characteristic of mechanical LiDAR to automatically extract the board points in LiDAR frame, which makes the calibration more efficient. The proposed algorithm was verified through simulations and real-world experiments, which demonstrates that it is capable of providing accurate and robust extrinsics. However, it requires the checkerboard to be visible on both the source and target sensors. For the future work, we plan to model the uncertainty of corners and further increase the robustness of the algorithm.

## ACKNOWLEDGMENT

This work was supported by Zhongshan Municipal Science and Technology Bureau Fund, under project ZSST21EG06, Collaborative Research Fund by Research Grants Council Hong Kong, under Project No. C4063-18G, and Department of Science and Technology of Guangdong Province Fund, under Project No. GDST20EG54, awarded to Prof. Ming Liu.

## REFERENCES

- [1] G. Pandey, J. McBride, S. Savarese, and R. Eustice, "Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, vol. 26, no. 1, 2012.
- [2] J. Kang and N. L. Doh, "Automatic targetless camera-lidar calibration by aligning edge with gaussian mixture model," *Journal of Field Robotics*, vol. 37, no. 1, pp. 158–179, 2020.
- [3] J. Jiao, Q. Liao, Y. Zhu, T. Liu, Y. Yu, R. Fan, L. Wang, and M. Liu, "A novel dual-lidar calibration algorithm using planar surfaces," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1499–1504.
- [4] 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)*, 2018, pp. 5562–5569.
- [5] W. Lee, C. Won, and J. Lim, "Unified calibration for multi-camera multi-lidar systems using a single checkerboard," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9033–9039.
- [6] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4193–4198.
- [7] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320314000235>
- [8] R. Hartley and A. Zisserman, *The Background: Projective Geometry, Transformations and Estimation*, 2nd ed. Cambridge University Press, 2004, p. 23–24.
- [9] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [10] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 930–943, 2003.
- [11] 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*, 2010, pp. 279–284.
- [12] M. He, H. Zhao, F. Davoine, J. Cui, and H. Zha, "Pairwise lidar calibration using multi-type 3d geometric features in natural scene," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1828–1835.
- [13] 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*, 2012, pp. 3096–3102.
- [14] 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)*, 2019, pp. 1–6.
- [15] Y. Zhu, C. Li, and Y. Zhang, "Online camera-lidar calibration with sensor semantic information," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4970–4976.
- [16] 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 *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, vol. 96, no. 34. AAAI Press, 1996, pp. 226–231.
- [17] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation (ICRA)*, 1991, pp. 2724–2729 vol.3.
- [18] 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.
- [19] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí, "Scatter search and local nlp solvers: A multistart framework for global optimization," *INFORMS Journal on computing*, vol. 19, no. 3, pp. 328–340, 2007.
- [20] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.
- [21] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.
- [22] I. The MathWorks, *Lidar Toolbox Release 2021b*, Natick, Massachusetts, United State, 2021.