ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446466

MULTI-DIMENSIONAL GEOMETRIC FEATURE-BASED CALIBRATION METHOD FOR LIDAR AND CAMERA FUSION

Yuhan Hao, Xin Jin, Dongyu Du

Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China

ABSTRACT

Extrinsic calibration between LiDAR and camera has become an indispensable task across diverse domains, including autonomous vehicles, robotics, and surveillance systems. However, existing methods suffer from limited precision due to the inaccurate and insufficient detected features caused by the sparsity of point-clouds and the inherent ranging errors of LiDAR. In this paper, we propose a multi-dimensional geometric feature-based calibration method between LiDAR and camera. First, a 3D structured calibration target is proposed with multi-normal surfaces, edges in different directions, and distinctive corner features. Secondly, a point-plane and angle error-based point-cloud feature detection method is designed to establish 3D-2D feature point pairs with image features. Finally, a Perspective-n-Point (PnP) problem is solved to estimate the extrinsic parameters. The experimental results show that the proposed method reduces Mean Reprojection Error (MRE) by 0.05 pixels and achieves a 70% reduction in Normalized Reprojection Error (NRE) compared with state-of-the-art (SOTA) methods under the conditions of smaller training size, larger test size, and more repeat times.

Index Terms— Extrinsic calibration, LiDAR and camera fusion, geometric feature

1. INTRODUCTION

In recent years, there has been a surge of interest in LiDAR-camera fusion. To better integrate the information from both sensors, calibration is fundamentally required to align multi-sensor data to a unified spatial coordinate system, whose precision directly affects the performance of object detection, localization, semantic segmentation, etc. However, the sparsity of point-clouds and the inherent ranging errors of LiDAR pose significant challenges for high-precision extrinsic calibration. To address this problem, efforts have been made in designing new calibration targets and implementing new feature detection methods.

Among the calibration targets, the checkerboard pattern is the most commonly employed, serving as the basis for image feature detection [1]. However, the checkerboard structure is perceived as a separate planar entity in point-clouds, with few distinctive features, resulting in inaccurate point-cloud feature detection. AprilTags [2][3] and LiDARTags [4] change the checkboard pattern to a QR code-like pattern to provide more intensity features for camera and LiDAR [5]. Circular, square or triangular holes were introduced to provide more geometric features [6][7], which still maintaining a planar structure. Although there are some targets with 3D structures [8], the most of them are independent regular shapes such as spheres and cubes, which lack intrinsic geometric constraints, resulting in limited precision of point-cloud feature detection.

Different calibration targets can provide different detectable features, leading to the establishment of different feature correspondences and error constraints between LiDAR and camera. The most common correspondence is matched point pairs. 3D points in point-clouds and 2D points in images can establish 3D-2D point pairs, and the extrinsic parameters are then obtained by minimizing the reprojection error [7][8][9]. 2D points in images can also be transformed into the camera coordinate system, establishing 3D-3D point pairs with 3D points in LiDAR coordinate system, and the extrinsic parameters are then solved by minimizing the Euclidean distance [10]. Additionally, point-to-line distance [11][12] and point-to-plane distance [1][11] are also frequently used as the error functions to solve extrinsic parameters. However, existing methods typically employ one of these constraints only during the final optimization stage while ignoring them in the feature detection stage, leading to limited precision in feature detection. The proposed approach incorporates a combination of various geometric constraints in feature detection, which makes better use of the known geometric information, leading to improved accuracy.

In this paper, a multi-dimensional geometric feature-based calibration method between LiDAR and camera is proposed. First, we introduce a 3D structured calibration target with multi-normal surfaces, multi-directional edges, and distinctive corner features. Secondly, a point-cloud feature detection method is developed based on point-to-sub-plane distance error and angle error to establish 3D-2D feature point pairs with image features. Finally, a PnP problem is solved to estimate the extrinsic parameters. The experimental results show that MRE and NRE reach below 0.6 pixels and 0.1 pixels respectively and decreases by 0.05 pixels and at least 70% compared with SOTA methods under the conditions of smaller training size, larger test size, and more repeat times.

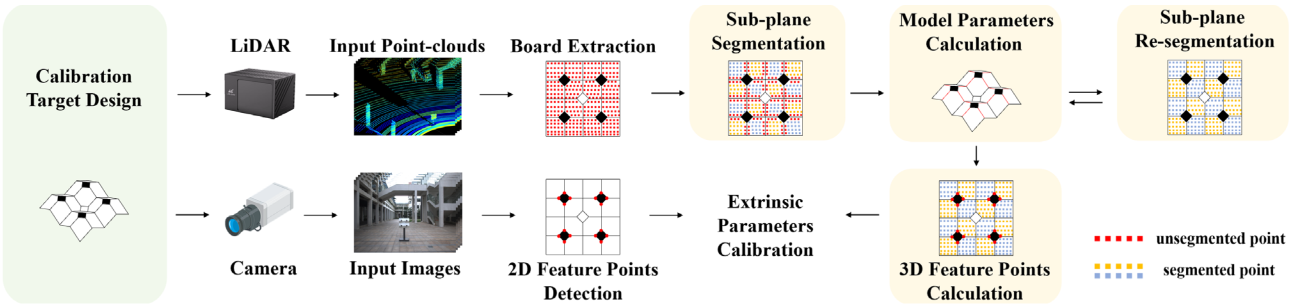
Calibration Target Design
LiDAR
Input Point-clouds
Board Extraction
Sub-plane Segmentation
Model Parameters Calculation
Sub-plane Re-segmentation
Camera
Input Images
2D Feature Points Detection
Extrinsic Parameters Calibration
3D Feature Points Calculation
unsegmented point
segmented point

Fig. 1. The pipeline of the proposed method.

2. PROPOSED METHOD

2.1. Overall Framework

Fig. 1 shows the pipeline of the proposed method. The 3D structured calibration target is designed with multi-normal surfaces, edges in different directions, and distinctive corner features (Sec. 2.2). Time-aligned sequences of images and point-clouds are collected with the target placed at M different positions, denoted as $I = \{I_i\}_{i=1}^{M}$ and $\mathcal{P} = \{\mathcal{P}_i\}_{i=1}^{M}$. Then, they are fed into the image feature detection module and the point-cloud feature detection module respectively.

Image adaptive binarization and polygons fitting are performed to the input images [13], and the pixel coordinates of the image feature points $\mathcal{P}_i^{2d} \in \mathbb{R}^{16\times2}$ are obtained for each I_i after sub-pixel optimization. In point-cloud feature detection, we first adopt the pass-through filter to extract the region of interest where the target is positioned and the noise filter to remove the points that represent the holder of the target, resulting in the target point-cloud, denoted as $\mathcal{P}^{board} = \{\mathcal{P}_i^{board}\}_{i=1}^{M}$. Then we conduct sub-plane segmentation (Sec. 2.3) to assign each point to its corresponding sub-plane, followed by the calculation of all model parameters that uniquely represent the target (Sec. 2.4). Furthermore, sub-plane re-segmentation is an alternative to improving the initial segmentation results, and the feature points calculation module is introduced to obtain the point-cloud feature points $\mathcal{P}_i^{3d} \in \mathbb{R}^{16\times3}$ for each $\mathcal{P}_i$ (Sec. 2.5).

Finally, we obtain 3D-2D feature points denoted as $\mathcal{P}^{2d} = \{\mathcal{P}_i^{2d}\}_{i=1}^{M} \in \mathbb{R}^{16M\times2}$ and $\mathcal{P}^{3d} = \{\mathcal{P}_i^{3d}\}_{i=1}^{M} \in \mathbb{R}^{16M\times3}$, and the PnP problem is established to minimize the reprojection error of 3D-2D point pairs,

$$\boldsymbol{T}^* = \underset{\boldsymbol{T}}{\operatorname{argmin}} \frac{1}{16M} \|\mathcal{P}^{2d} - \boldsymbol{D}(\boldsymbol{KT}\mathcal{P}^{3d})\|_2^2. \tag{1}$$

where $\boldsymbol{T} \in \mathrm{SE}(3)$, $\boldsymbol{K} \in \mathbb{R}^{3\times3}$, and $\boldsymbol{D}(\cdot)$ respectively represent the extrinsic parameters between LiDAR and camera, the camera intrinsic parameters, and the distortion. This problem is initialized by EPnP [14] and solved by the Ceres solver [15].

2.2. Calibration Target Design

Fig. 2(a) illustrates our design of the target, a white 3D symmetrical structure comprising four symmetric square-shaped holes and sixteen sub-planes connected at specific angles. The holes allow the camera to observe through them, facilitating the detection of distinctive corner features. The repetitive arrangement of the sub-planes offers multi-normal surfaces and edges in different directions for the collected point-clouds.

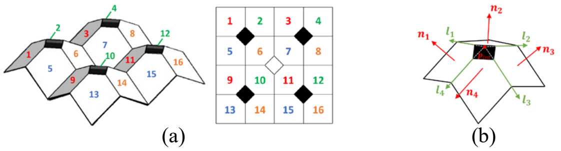

Fig. 2. (a) The calibration target and sub-plane indexes. (b) The details of the calibration target. The red arrows indicate four normal vectors, the green arrows indicate four intersection line directions of each set, and the red dot indicates the intersection point of the four planes.

We index the sub-planes of our target as 1~16, where planes of the same color index are parallel to each other. Specifically, the four planes marked in red together form three planes with equal distances, as plane 3 and plane 9 are co-planar. We formulate plane 1 as

$$a_1x + b_1y + c_1z + d_1 = 0, \tag{2}$$

where $\|(a_1, b_1, c_1)\| = 1$. Then the other two planes can be formulated as

$$a_1x + b_1y + c_1z + d_1 + \Delta t = 0, \tag{3}$$

$$a_1x + b_1y + c_1z + d_1 + 2 * \Delta t = 0, \tag{4}$$

where Δt represents the known distance. By formulating the other three sets of parallel planes in the same way, the whole calibration target can be modeled by 16 parameters $Param = \{a_i, b_i, c_i, d_i\} \in \mathbb{R}^{16}$, $i = 1, 2, 3, 4$.

2.3. Sub-plane segmentation

We designed a point-cloud segmentation pipeline to assign each $\boldsymbol{p}_i \in \mathcal{P}^{board}$ to its corresponding sub-plane π_i. First, $\mathcal{P}^{board}$ is projected onto the plane passing through the farthest point in the direction of $\boldsymbol{n}_b \in \mathbb{R}^3$, where $\boldsymbol{n}_b$ is the general normal vector of $\mathcal{P}^{board}$, resulting in $\mathcal{P}^{proj}$, which contains four square-shaped holes. The hole centers are obtained by searching for the positions within the neighborhood of the initial coordinates where the fewest number of points fall into the corresponding square [16]. Finally, $\mathcal{P}^{proj}$ is segmented based on the four hole centers, enabling the establishment of correspondences between each $\boldsymbol{p}_i$ and π_i.

It is noticeable that $\mathcal{P}^{board}$ is not perfectly symmetrical due to the orientation of the target, resulting in inaccurate $\boldsymbol{n}_b$, which further affects the accuracy of sub-plane segmentation. To enhance the confidence level in the point-to-sub-plane correspondences, we discard points with distances from the junctions between sub-planes less than the threshold δ, which is set to 1 cm in our method.

2.4. Model parameters calculation

To calculate the model parameters, we introduce the point-to-sub-plane distance error e_{dis},

$$e_{dis} = \frac{1}{N}\sum_{\boldsymbol{p}_i \in \mathcal{P}^{board}} \sigma_i \cdot \boldsymbol{dis}(\boldsymbol{p}_i, \pi_i), \tag{5}$$

where σ_i is a binary value indicating whether p_i has its corresponding sub-plane π_i , $\boldsymbol{dis}(\boldsymbol{p}_i, \pi_i)$ represents the distance from $\boldsymbol{p}_i$ to π_i, and N is the number of valid points where $\sigma_i = 1$.

We also introduce the angle error e_{angle} among the four normal vectors $\boldsymbol{n}_i = \{a_i, b_i, c_i\}$, $i = 1, 2, 3, 4$, as shown in Fig. 2(b). We define two types of angles to strengthen the constraints. One type is the angles between adjacent normal vectors, and the other type is the angles between intersecting normal vectors. Using $\angle_{ij}$ to represent the angle between $\boldsymbol{n}_i$ and $\boldsymbol{n}_j$, the constraints for the two types can be expressed as $\angle_{12} = \angle_{23} = \angle_{34} = \angle_{14}$ and $\angle_{13} = \angle_{24}$. Therefore, we formulate e_{angle} as:

$$e_{angle} = \|\angle_{12} - \angle_{23}\|^2 + \|\angle_{23} - \angle_{34}\|^2 + \|\angle_{34} - \angle_{14}\|^2 + \|\angle_{14} - \angle_{12}\|^2 + \|\angle_{13} - \angle_{24}\|^2. \tag{6}$$

Finally, all 16 model parameters are solved by minimizing the total error function,

$$Param^* = \underset{Param}{\operatorname{argmin}}(e_{dis} + e_{angle}), \tag{7}$$

where the initial value is obtained by RANSAC [17].

2.5. Sub-plane re-segmentation and Feature Points Calculation

The initial segmentation results are improved in this operation, resulting in the inclusion of more points in the point-to-sub-plane correspondence, as shown in Fig. 1. First, the plane equations of all sub-planes are derived from $Param$. Then, $\boldsymbol{n}_b$ is updated with the average of all normal vectors, and the updated coordinates of the four hole centers are calculated from the equations. Subsequent segmentation steps are the same as before.

The feature points, i.e., the corners of all the rectangular holes, can be calculated from $Param$. Taking the hole formed by one set of parallel planes as an example, as shown in Fig. 2, we first get the intersection point $\boldsymbol{p}_{int}$ through an optimization problem,

$$\boldsymbol{p}_{int}{}^* = \underset{\boldsymbol{p}_{int}}{\operatorname{argmin}} \sum_{i=1}^{4} \boldsymbol{dis}(\boldsymbol{p}_{int}, \pi_i). \tag{8}$$

In addition, the intersection line direction $\boldsymbol{l}_i$ of the corresponding adjacent planes (arranged clockwise) can be obtained by the cross product of their normal vectors. Therefore, the feature points fall on the lines formed by the intersection point $\boldsymbol{p}_{int}$ and the corresponding intersection line directions, and can be obtained by

$$\boldsymbol{p}_i^{3d} = \boldsymbol{p}_{int} + \theta \boldsymbol{l}_i,\ i = 1, 2, 3, 4 \tag{9}$$

where θ represents the fixed distance calculated from known geometric information.

3. EXPERIMENTS AND RESULTS

3.1. Experimental Setup

The proposed method was validated in a real environment. Table 1 shows the specifications of LiDAR and camera.

Table 1. Sensor Specifications.

Modality	Device	Resolution	FOV
LiDAR	LSLIDAR CH128x1 [18]	128 layers, 0.4°	120° × 25°
Camera	FLIR Blackfly BFS-PGE-16S2 [19]	1440×1080	82.9° × 66.5°

To better verify the effectiveness and robustness of the proposed method, we conducted experiments in both indoor and outdoor scenes, as shown in Fig. 3. For both scenes, we placed the calibration target evenly in front at distances ranging from approximately 3 to 10 meters and collected image-point cloud pairs with $M = 50$ different poses.

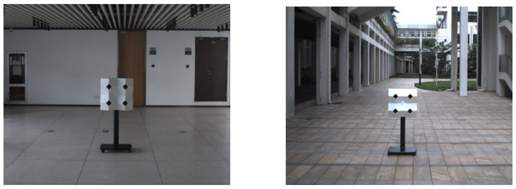

Fig. 3. Experimental scenes. The left is the indoor scene and the right is the outdoor scene.

3.2. Re-projection evaluation

Since it is hard to obtain the ground truth extrinsic parameters between LiDAR and camera in realistic environment, we use MRE and NRE as metrics to evaluate the effectiveness of the proposed method:

$$MRE = \frac{1}{16M}\sum_{i=1}^{16M} \left\| \mathcal{P}_i^{2d} - \boldsymbol{D}(\boldsymbol{KT}\mathcal{P}_i^{3d}) \right\|_2^2 \tag{10}$$

$$NRE = \frac{1}{16M}\sum_{i=1}^{16M} \frac{d_i}{d_{max}} \left\| \mathcal{P}_i^{2d} - \boldsymbol{D}(\boldsymbol{KT}\mathcal{P}_i^{3d}) \right\|_2^2 \tag{11}$$

where d_i denotes the distance from $\mathcal{P}_i^{3d}$ to LiDAR, and d_{max} denotes the maximum of all d_i.

We evaluate MRE and NRE under a different number of placements. N placements are randomly selected from a previously collected dataset as training data, while the remaining placements reserved for test data. For each $N \in [1,15]$, we repeat the random selection 100 times. Fig. 4

shows the MRE and NRE boxplots. The results indicate that MRE and NRE reach stable values when N reaches 4 in both indoor and outdoor scenarios, where MRE is less than 0.6 pixels and NRE is less than 0.1 pixels.

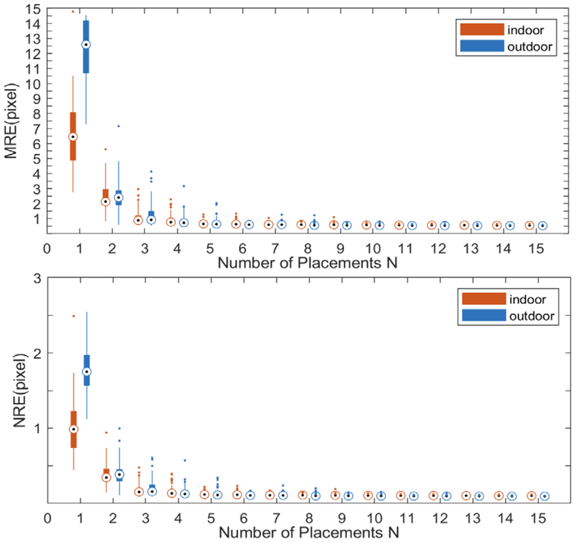

Fig. 4. MRE and NRE under a different number of placements.

3.3. Comparison against other methods

Given the diverse input data of various target-based calibration methods, many of which are not open-source, we selected SOTA methods with similar metrics to ours for comparison. We then evaluated our results against the best-performing one reported in each method. Table 2 and Table 3 compare MRE and NRE with comparison methods, which are Xie's work [9], Fan's work [8], Li's work [20], Cui's work [21] and Lai's work [22].

The results indicate that the proposed method outperforms all other methods under stricter experimental conditions, i.e., smaller training size, larger test size, and more repeat times. Specifically, MRE shows an improvement of 0.05 pixels compared with [20] under the same training size and a much larger test size, and NRE decreases by at least 70% compared with SOTA methods.

3.4. Subjective results

For subjective experiments, we project the point-clouds onto the images using the obtained extrinsic parameters. Fig. 5 shows that the point-clouds and images are well-aligned at columns, leaves, and other positions.

3.5. Complexity analysis

To evaluate the complexity, the execution time of each algorithm module is retrieved using a PC with Intel® Core(TM) i7-9750 CPU @ 2.60 GHz with 13.3GB RAM and 64-bits Ubuntu 18.04 operating system. The average execution time for image feature detection is 0.2075 seconds, for point-cloud feature detection is 2.2248 seconds, and for solving extrinsic parameters is 0.0750 seconds. Consequently, with 4 target placements considered, the entire calibration process can be accomplished within 10 seconds, and the most time-consuming part is the point-cloud feature detection. We also measure the execution time of other open-source methods in the same way. Cui's work [21] and Lai's work [22] take approximately 10 seconds and 8 seconds respectively, which is similar to ours.

Table 2. MRE comparison. "Training Size", "Test Size", "Repeat Times" respectively refer to the number of target placements in training data, test data, and the repeat times for random selection. The symbol "~" indicates that the item is not mentioned in their papers.

Method	Training Size	Test Size	Repeat Times	MRE /pixel
Xie et al. [9]	8	1	5	1.12
Fan et al. [8]	36	~	~	1.01
Li et al. [20]	4	1	~	0.56
Proposed	4	46	100	**0.51**

Table 3. NRE comparison.

Method	Training Size	Test Size	Repeat Times	NRE /pixel
Cui et al. [21]	~	~	~	2.11
Lai et al. [22]	<=20	~	30	> 0.3
Proposed	4	46	100	**0.09**

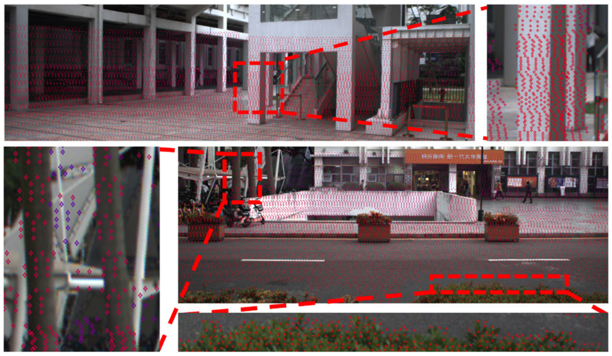

Fig. 5. Re-projection results.

4. CONCLUSIONS

This paper presents a multi-dimensional geometric feature-based calibration method between LiDAR and camera. First, a novel calibration target with a well-defined 3D structure is introduced. Secondly, a point-cloud feature detection method is developed based on the introduced calibration target. Finally, a PnP problem is solved to estimate the extrinsic parameters. The experimental results show MRE and NRE reach below 0.6 pixels and 0.1 pixels respectively and decrease by 0.05 pixels and at least 70% compared with SOTA methods under stricter experimental conditions.

5. ACKNOWLEDGMENT

This work is supported in part by Shenzhen Project under Grant JSGG20220831095602005, NSF of Guangdong Province under Grant 2023A1515012716 and NSFC under Grant 61827804, China.

6. REFERENCES

[1] Geiger, Andreas, et al. "Automatic camera and range sensor calibration using a single shot." 2012 IEEE international conference on robotics and automation. IEEE, 2012.

[2] Olson, Edwin. "AprilTag: A robust and flexible visual fiducial system." 2011 IEEE international conference on robotics and automation. IEEE, 2011.

[3] Wang, John, and Edwin Olson. "AprilTag 2: Efficient and robust fiducial detection." 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016.

[4] Huang, Jiunn-Kai, et al. "LiDARTag: A real-time fiducial tag system for point clouds." IEEE Robotics and Automation Letters 6.3 (2021): 4875-4882.

[5] Wang, Weimin, Ken Sakurada, and Nobuo Kawaguchi. "Reflectance intensity assisted automatic and accurate extrinsic calibration of 3d lidar and panoramic camera using a printed chessboard." Remote Sensing 9.8 (2017): 851.

[6] Beltrán, Jorge, et al. "Automatic extrinsic calibration method for lidar and camera sensor setups." IEEE Transactions on Intelligent Transportation Systems 23.10 (2022): 17677-17689.

[7] Cai, Huaiyu, et al. "A novel calibration board and experiments for 3D LiDAR and camera calibration." Sensors 20.4 (2020): 1130.

[8] Fan, Song, et al. "High-Precision External Parameter Calibration Method for Camera and Lidar Based on a Calibration Device." IEEE Access 11 (2023): 18750-18760.

[9] Xie, Shichao, et al. "Pixels and 3-D points alignment method for the fusion of camera and LiDAR data." IEEE Transactions on Instrumentation and Measurement 68.10 (2018): 3661-3676.

[10] Dhall, Ankit, et al. "LiDAR-camera calibration using 3D-3D point correspondences." arXiv preprint arXiv:1705.09785 (2017).

[11] Zhou, Lipu, Zimo Li, and Michael Kaess. "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.

[12] Liao, Qinghai, et al. "Extrinsic calibration of lidar and camera with polygon." 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2018.

[13] Yan, Guohang, et al. "Opencalib: A multi-sensor calibration toolbox for autonomous driving." Software Impacts 14 (2022): 100393.

[14] Lepetit, Vincent, Francesc Moreno-Noguer, and Pascal Fua. "EP n P: An accurate O (n) solution to the P n P problem." International journal of computer vision 81 (2009): 155-166.

[15] Agarwal, Sameer, and Keir Mierle. "Ceres solver." (2012).

[16] Yan, Guohang, et al. "Joint camera intrinsic and lidar-camera extrinsic calibration." arXiv preprint arXiv:2202.13708 (2022).

[17] Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." Communications of the ACM 24.6 (1981): 381-395.

[18] Leishen | The World's Leading LiDAR and Whole Solutions Provider. https://www.lslidar.com/.

[19] Blackfly S GigE | Teledyne FLIR. https://www.flir.cn/products/blackfly-s-gige.

[20] Li, Xingxing, et al. "Accurate and automatic extrinsic calibration for a monocular camera and heterogenous 3D LiDARs." IEEE Sensors Journal 22.16 (2022): 16472-16480.

[21] Cui, Jiahe, et al. "ACSC: Automatic calibration for non-repetitive scanning solid-state LiDAR and camera systems." arXiv preprint arXiv:2011.08516 (2020).

[22] Lai, Zhengchao, et al. "Laser reflectance feature assisted accurate extrinsic calibration for non-repetitive scanning LiDAR and camera systems." Optics Express 30.10 (2022): 16242-16263.