2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

A Novel, Efficient and Accurate Method for Lidar Camera Calibration

Zhanhong Huang, Xiao Zhang, Antony Garcia and Xinming Huang

***Abstract*— As autonomous systems evolve, the precise calibration of lidar and camera sensors remains a pivotal concern. Among the myriad of available techniques, target-based calibration methods, which employ planar boards with distinct geometry and image patterns, have been a popular choice. These methods simplify the task of extracting corresponding features between the image and lidar point cloud. But many of these approaches also face a significant challenge, which is their sensitivity to lidar resolution and Field of View (FOV), which may degrade the reliability of the calibration results. Therefore, our research introduces a novel calibration method using a uniquely designed acrylic checkerboard which allows the lidar beam to pass through the white grids and reflect back from the black grids. This innovative technique sidesteps the common challenges associated with lidar feature extraction. Our method's distinct advantage lies in its ability to perform accurate calibrations at close distances, owing to the efficient feature extraction from both lidar and camera sensors. This novel, efficient, and accurate method can provide state-of-the-art results for camera lidar calibration in the field. Please also check our Github repository: `https://github.com/WPI-APA-Lab/Acrylic-Board-Lidar-Camera-Calibration`**

I. INTRODUCTION

In the era of artificial intelligence, there is a surge of innovations for real-world autonomous systems. Self-driving cars are now routinely tested in autonomous modes on public roads, and advanced robots have become staples in contemporary manufacturing and warehousing. As tasks grow in complexity and safety requirements become more stringent, it calls more advanced perception systems. Instead of relying on a single type of sensors, modern solutions integrate multiple types of sensors to comprehensively understand the surrounding environment. For instance, a standard self-driving car now employs various sensors: radars for short-range obstacle detection, camera arrays for mid-range dense RGB data, and lidar for long-range depth information resilient to varying light conditions. Given this plethora of sensor data, fusing multi-sensor information accurately and efficiently becomes a new challenge.

Calibration technology enables consistent alignment across various sensor nodes. Specifically, lidar-camera calibration aligns the two primary visual sensors, enabling the system to utilize both lidar and camera inputs for essential autonomous driving tasks like object detection [1] [2], semantic segmentation [3] [4] [5], and panoptic segmentation [6] [7]. Proper calibration not only ensures the safety and reliability of sensor fusion-based autonomous systems but also facilitates spatially synchronized 3D reconstructions using lidar point clouds and camera images. It provides a precise framework and RGB color integration for tasks like map building or object recreation [8] [9].

Fig. 1. Calibration Setup. **Top left**: The calibrated lidar and camera and their coordinate frames. **Top right**: picture of the specially painted acrylic calibration board with a white foam board in the background. **Bottom left**: colorized point cloud for the calibration board. **Bottom right**: projection point cloud on image for the acrylic calibration board.

Although camera calibration is well-established, lidar calibration is still developing as new lidar technologies are introduced. lidar often has a limited Field of View (FoV) vertically, so the calibration board cannot be placed too close in order to capture all four corners. On the contrary, placing the calibration board further would reduce the number of pixels on camera image and compromise camera calibration. Furthermore, the introduction of solid-state lidar, with its narrower FoV and front-view-only capability, calls for better calibration methods. In summary, it is a technical challenge to find a balance between distance, board size, and grid number during the lidar camera calibration.

Addressing the challenges above, we propose a novel lidar camera calibration method in this paper. The contributions are listed as follows:

- A lidar camera calibration method uses a specially painted transparent acrylic checkerboard that allows the laser to pass through the "white" grid areas.

This work was partially supported by US National Science Foundation and by The MathWorks.

The authors are with Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609, USA {zhuang5, xzhang25, agarcia3, xhuang}@wpi.edu

- The approach is deployed by the 13×9 checkerboard with a 4 cm grid size in a short distance, which benefits lidar with a small FoV to detect features as well as the accurate image feature extraction.
- Comprehensive calibration experiments are conducted indoors and outdoors, comparing the results with the existing calibration method.

II. Related Works

Current lidar camera extrinsic calibration methods could be categorized into two classes - target-based [10]–[18] and targetless-based [19]–[26] methods. Both types of methods use the lidar patterns from objects and corresponding features extracted from images to calculate the rigid 3D transformation between sensors.

Planar board with image feature pattern is widely used as the target object for the sensor detection. Benefiting from the image pattern on the board, it is relatively easy to extract corresponding features to match the lidar shape, so most contributions of target-based methods relate to the lidar feature extraction. The current target-based methods can achieve a fine result, but their robustness suffers from the lidar resolution, which may even cause the result hard to converge after a number of iterations. Zhou et al. [10] proposed a lines and plane algorithm to estimate the board's four corners to interpolate the board pose. However, its performance is affected by the density of the lidar data, which makes the translation error hard to reduce. Yan et al. [11] use hollow circles to determine the board pose and then estimate segmentation results from the camera. However, the method needs to calculate the circle center using sampled board cutting circle edges, causing its accuracy will drop significantly when the vertical resolution of the lidar is low. Cui et al. [12] introduce a method that uses the reflectance intensity from lidar data to extract the image patterns to match the detected image features. However, the reflectance intensity could be varied by the reflected materials. Also, the difference between the reflectance intensity could be hard to define while using low-resolution lidar. Xie et al. [13] propose a method that uses a depth-based Fiducial Marker with a special order of cutting holes, which allows the calibration to be performed over a short distance. However, its result is hard to evaluate since the calibration target's feature pattern is hard to measure using the reprojection method. Also, its calibration accuracy depends on the hollow circle radius, which has to be enlarged in the case needs to calibrating sensors at a relatively long distance.

Targetless-based methods burst out recently as the development of AI on image feature extraction. These methods process the edges and surfaces in the image to valid geometric constraints to solve the extrinsic. These methods are also divided into static-based and motion-based methods. Static-based methods are similar to target-based calibration, which finds the common features in sensor data. Yuan et al. [19] detect the depth continues edges in the image to match the 3D edges. Luo et al. [20] use segment outline from SegmentAnything [27] to match the shape in lidar detect objects. Koide et al. [26] further developed a mechanism using SuperGlue [28] 2D-3D matching network to find the matching points. Where in motion-based methods, Pandey et al. [21] proposes a method that calculates the extrinsic by minimizing the trajectory error from the SLAM result of both camera and lidar data. However, while targetless-based methods benefit from their low cost of preparation, their robustness is always being questioned. Since the image feature extraction performance could be heavily affected by camera distortion coefficients, outline detection, and SLAM trajectory can be both affected by the undistorted image result, which introduces errors.

Our proposed method does not heavily depend on the lidar resolution, which solves the robustness issue in the existing current target-based methods. Also, by utilizing this novel acrylic checker board, we are able to reduce the size of the calibration board, allowing the user to calibrate the sensor at a close distance while still extracting enough lidar and image features from the board to achieve accurate calibration results.

III. Methodology

For a lidar-camera system, the extrinsic calibration problem is to estimate the relative rotation and translation between the lidar and the camera. The problem can be represented as using the 2D and 3D coordinates correspondence in the same frame to solve the extrinsic matrix ($\mathbf{E} \in$ SE3). The proposed method uses a painted acrylic checkerboard as the calibration target, where only black grids is painted on the transparent acrylic board so that the laser is desired to only get reflected on these black grids and pass through the transparent "white grids" on the board. The dimension of the checkerboard is represented by its inner corners $N_r \times N_c$ and the board's grid size G_s. To align the OpenCV calibration standard [29], the N_r should always be larger than N_c, which means the board should always put the long edge perpendicular to the ground. The specially designed calibration board is shown in Fig. 1. The main challenge of this method is how to accurately estimate the corners of black grids from the point cloud. During the research, we found the lidar scan for our specially designed board shows several problems:

- Even though the transparent grids on acrylic board are transparent, they still reflects a small amount of the lidar beams back at certain angles. Fig.2 (b)
- The undesirable jitters are shown in the pint cloud data if the calibration target is in a close distance to the lidar. Fig.2 (c)
- The point cloud from the black grid usually does not show in perfect square shape, which may not be able to accurately indicate the boundaries, introducing the pose estimation errors. Fig.2 (a)

The proposed methods consider all aforementioned issues and alleviate their effects on feature extractions, which can provide a robust calibration result. The complete calibration process is shown in Fig3.

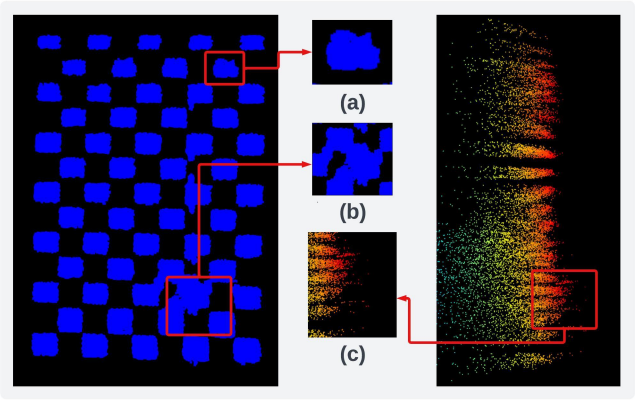

Fig. 2. Challenges when extracting features from the acrylic checkerboard. The right image is the side view of the board's original point cloud. The left image shows the point cloud after ray projection (a) The grid shape is not a perfect square. (b) Transparent grids may still reflect some. (c) Jitters are reflected while the board is placed in a near distance.

A. Calibration Target Pre-processing

1) Raw point cloud stacking and denoise: Instead of using a single scan of lidar data, we integrate the continuous scans to illustrate the calibration target's shape better. There are two reasons to do this: (1) Some solid-state lidar like Livox series lidar can densify the target surface due to their non-repetitive scanning pattern, which gives better shape of the calibration feature. (2) Stacking point clouds could get a better distribution on the depth of the surface, which helps us to get a better result on RANSAC plane estimation. While stacking the point cloud, we also apply statistic outlier removal to reduce the accumulated noise from each frame.

2) Calibration Pattern Refinement: We crop the black grid patterns out from the point cloud to obtain a better result from the RANSAC plane fitting. Due to the property of the acrylic board, there should be no points surrounding the black grid pattern, which allows us to easily cut out the point cloud P_c within the preferred region.

Then we perform the RANSAC-based method to fit the optimal plane. Given the fitting plane parameter, we keep the points within a certain distance of the fitting plane. The remaining point cloud $P_c^{'}$ could be expressed as Eq. 1:

$$P_c^{'} = \{(x_0, y_0, z_0) \in P_c \mid \frac{\mid Ax_0 + By_0 + Cz_0 + D \mid}{\sqrt{A^2 + B^2 + C^2}} < \delta\} \tag{1}$$

where the A, B, C, and D are the parameters for the estimated fitting plane, and δ is the maximum allowed normal distance from the points to the plane. A point (x_0, y_0, z_0) s will be filtered out if its distance to the plane is larger than δ.

To better fit the shape of the black pattern, we perform a ray projection for each point onto the plane. According to the ranging principle of the lidar, we can assume that the points are on a ray passing through the origin. After ray projection, we obtain a point cloud P_r, which can be described as in Eq. 2:

$$P_r = \{(tx_0, ty_0, tz_0) \mid (x_0, y_0, z_0) \in P_c^{'}\} \tag{2}$$

where

$$t = \frac{-D}{Ax_0 + By_0 + Cz_0} \tag{3}$$

Finally, we perform a voxel downsampling to solve the uneven density of the lidar pattern. For each voxel, it can be represented using a single point by calculating the mean values of all points in the voxel.

B. 3D Corner Estimation

1) Black grid clustering and labeling: Once we obtain the pre-processed point cloud P_d, we start to separate each black grid from the pattern using DBSCAN [30]. The cluster and label condition can be conducted as Eq. 4:

$$L(P_d, \varepsilon, N_{\text{min}}) = \begin{cases} L_i & \text{if } \exists p_j \in \text{Nb}(p_i, \varepsilon) \\ & \text{and } |\text{Nb}(p_i, \varepsilon)| \geq N_{\text{min}} \\ \text{Noise} & \text{otherwise} \end{cases} \tag{4}$$

where the L_i are the cluster label output for each point $p_j \in P_d$, N_{min} represents the minimum number of points to form a dense region, and $\text{Nb}(p_i, \varepsilon)$ represents the number of points in the neighborhood within distance ε.

Since there is a possibility that the DBSCAN algorithm combines two grids into one cluster or divides one grid into two clusters, we use a min-max filter to filter out an output cluster that has too many or too few points, the filter condition can be expressed as Eq.5:

$$F(L_i, \text{minPoints}, \text{maxPoints}) = \begin{cases} L_i \text{ if minPoints} \\ \quad \leq |L_i| \leq \text{maxPoints} \\ \emptyset \text{ otherwise} \end{cases} \tag{5}$$

where the max limit maxPoints can be calculated from the ratio between the size of grid G_s and the size of voxel V_s in the downsampling process, and the min limit minPoints could be ratio of the maxPoints but need to be fine-tuned since the calibration target orientation may affect the minimum shot points for each grid.

2) Orientation estimation: Next, we need to find the orientation of the calibration board. At this stage, we only need to find the two vectors along the board's long edge and short edge since we have co-planar points. However, due to the imperfect shape of the projected grids, we cannot directly use the edges of the projected grid to determine the orientations. Here, we first calculate the centroid of each cluster and try to connect them with a vector if their Euclidean distance has a length of 2 times the grid size G_s so that we get two groups of vectors that should follow the horizontal and vertical directions of the calibration board. Given n points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ from the i-th cluster, this operation can be expressed as Eq. 6 and Eq. 7:

$$C_i = \left(\frac{1}{n} \sum_{j=1}^{n} x_j, \frac{1}{n} \sum_{j=1}^{n} y_j, \frac{1}{n} \sum_{j=1}^{n} z_j \right) \tag{6}$$

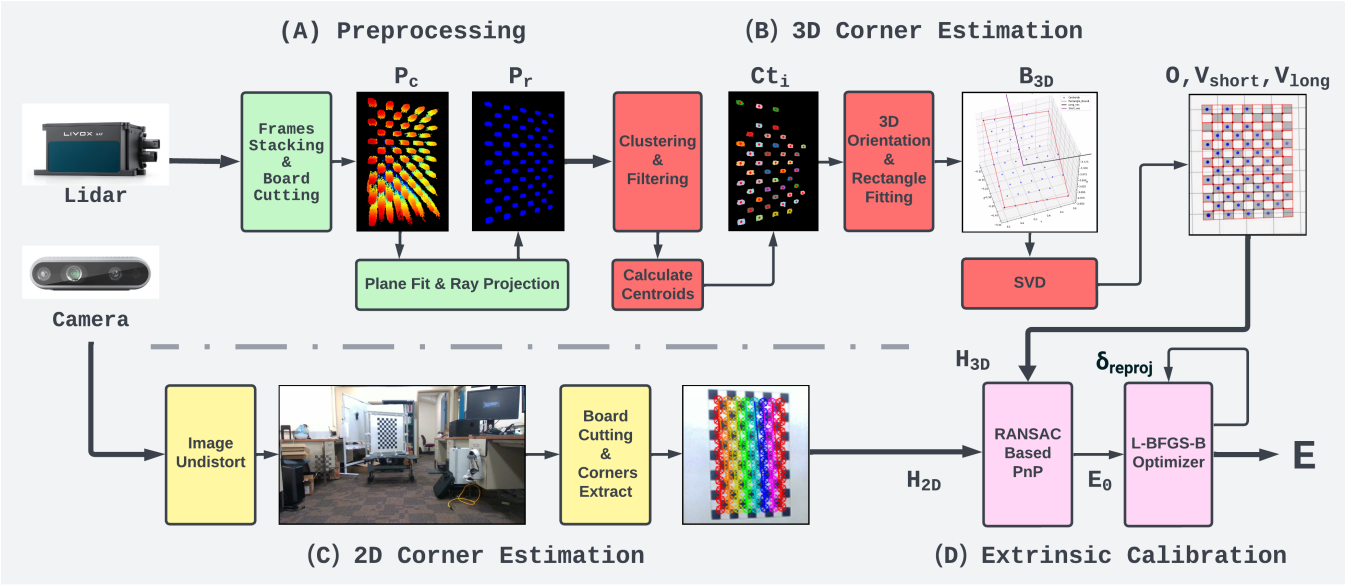

Fig. 3. The proposed calibration method. (A) Calibration target pre-processing from the captured point cloud (B) 3D corner estimation from a cutting plane fitted calibration board. (C) 2D corner estimation from the image. (D) Extrinsic matrix calibration from 3D-2D inner grid corners.

and for each pair of centroids C_i and C_j and tolerance ratio T_e,the paring method is shown in Fig. 4:

$$V_{ij} = \begin{cases} \frac{C_j - C_i}{D_{ji}} & \text{if } \mid D_{ji} - 2G_s \mid \leq T_e G_s \\ \text{undefined} & \text{otherwise} \end{cases}, \tag{7}$$

where

$$D_{ji} = \|C_j - C_i\|_2$$

where the two groups of the vector could be defined as Eq.8:

$$\begin{aligned} G_1 &= \{V_{ij} \mid V_{11} \cdot V_{ij} \leq 0.5\} \\ G_2 &= \{V_{ij} \mid V_{11} \cdot V_{ij} > 0.5\} \end{aligned} \tag{8}$$

Next we remove the outlier vectors from G_1 and G_2, and then calculate an average vector for each group, denoted as V_{avg1} and V_{avg2} for G_1 and G_2.

After that, we perform a 3D rectangle fitting for the centroids C_i. Since all centroids are co-planar, using V_{avg1} and V_{avg2} as the horizontal and vertical directions, we can find four intersection points that form a rectangular box.

Next, we sort the order of these four corners of the rectangle following the OpenCV calibration standard [29]. However, this rectangle is slightly smaller than the actual calibration board. To fix that, We expand the rectangle by moving four corners in the diagonal directions further in a small distance of $\frac{\sqrt{2}}{2}G_s$. Finally, we obtain a rectangular box B matching the actual size of the calibration board. The four corners are denoted as B_0, B_1, B_2 and B_3, respectively.

Afterwards, we perform a shape consistent check with expected edge lengths and the actual estimated edge lengths, where the expected length for the calibration board's short edge $\mathcal{E}_{\text{short}}$ and long edge $\mathcal{E}_{\text{long}}$ are calculated from numbers of columns and rows N_c, N_r, and grid size G_s:

$$\begin{aligned} \mathcal{E}_{\text{short}} &= G_s(N_c + 1) \\ \mathcal{E}_{\text{long}} &= G_s(N_r + 1) \end{aligned} \tag{9}$$

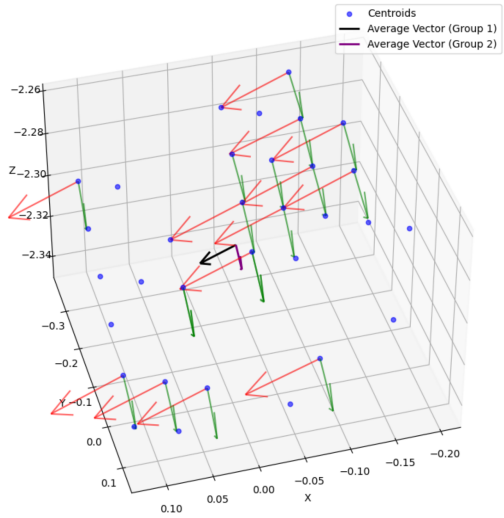

Fig. 4. An Example to find the calibration board rough orientation by connecting centroids. Red and green vectors are the pairing vectors for centroids. All paring vectors are aligned to the positive directions. The black and purple vectors are the average vectors for each group after outlier removal.

And the edge to check can be expressed as Eq. 10:

$$\begin{aligned} \mathcal{A}_{\text{short}} &= B_1 - B_0 \\ \mathcal{A}_{\text{long}} &= B_3 - B_0 \end{aligned} \tag{10}$$

and the checking condition with tolerance ratio T_c would be:

$$\begin{aligned} \Delta_{\text{short}} &= \|\mathcal{A}_{short}\|_2 - \mathcal{E}_{\text{short}}, \Delta_{\text{short}} < T_c G_s \\ \Delta_{\text{long}} &= \|\mathcal{A}_{long}\|_2 - \mathcal{E}_{\text{long}}, \Delta_{\text{long}} < T_c G_s \end{aligned} \tag{11}$$

If the consistent check passes, then we have our initial representation of the calibration board, where the board origin is B_0 and short edge in the direction of $\mathcal{A}_{short}$ and long edge in the direction of $\mathcal{A}_{long}$.

Once we have the representation of the board, we could interpolate the expected black grid centers C_e for every black grid using N_r, N_c, and G_s. Then, we can get the grid index of every observed centroid C. Using the index, we can perform single value decomposition (SVD) between C_e and C to get rotation R_o and translation T_o between the initial pose and optimized pose, where the optimized pose of the calibration board can be represented as:

$$
\begin{aligned}
O &= \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix} B_0 \\
V_{\text{short}} &= \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix} \mathcal{A}_{short} \\
V_{\text{long}} &= \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix} \mathcal{A}_{long}
\end{aligned} \tag{12}
$$

where the O is the optimized origin of the board, V_{short} and V_{long} represent the optimized short and long edge directions.

3) 3D inner grid corner extraction: Given the optimized origin O and the edge vectors V_{short} and V_{long} of the calibration board in 3D space, and also board properties N_r, N_c and G_s, we can interpolate the inner grid corners $H_{3\text{D}}$ in 3D space, where these corners are sorted following the OpenCV calibration order [29], column first, then rows.

C. Inner grid corner extraction from image

With the board properties N_r, N_c, the checkerboard corner detection algorithm [31] is used to find the inner grid corners $H_{2\text{D}}$ from the undistorted images. Note that the corners should also have the same order as the 3D corners.

D. Extrinsic matrix Calculation

The initial extrinsic parameters $E_0 \in$ SE3 are estimated using RANSAC-based PnP solver [29], given the 3D-2D inner grid corner pairs as shown in Fig. 5. Then, an L-BFGS-B optimizer [32] is adopted to refine the estimation. During the optimization, the corner pairs reprojection error δ_{reproj} is minimized from the initial value calculated by E_0. The final result of the optimizer is denoted as $\mathbf{E} \in$ SE3.

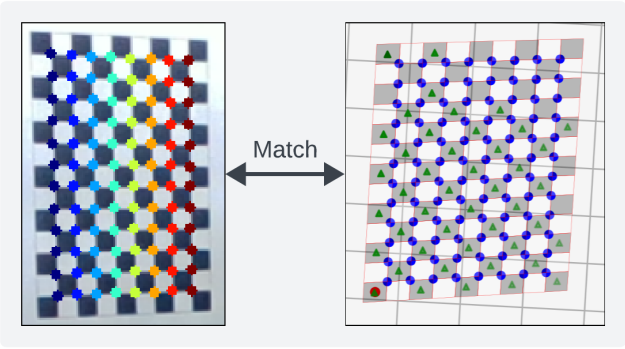

Fig. 5. An Example of pairing 2D and 3D inner grid corners. The left picture shows the 2D extracted corners. The right picture shows the 3D estimated checkerboard. The blue points represent the 3D estimated inner corners. The green triangle represents the extracted centroids for the detected clusters. And, the red point on the bottom left shows the origin of the board.

IV. Experiments and Result

A. Calibration Setup

During the experiment, the lidar and camera are fixed by a 3D printed case, which allows us to measure the relative position between two sensors. The camera intrinsic parameters are assumed to be known or pre-calibrated before the experiments. The acrylic calibration board is fixed by the aluminum frame and placed in different poses while still remaining in both sensors' FoV. A whiteboard is placed behind the acrylic board to improve the image contract for corner extraction. The multi-frame point cloud and corresponding images are collected by the ROS framework and then exported to estimate the extrinsic matrix. The calibration result is computed using four collection sets, each containing 200 frames of point cloud and one image.

Our method is evaluated on Livox HAP with Realsense D435i. The sensors' FoV are described in Table I.

TABLE I
FoV of Sensors used in Expereiments

Sensor	FoV	
	Horizontal	Vertical
Livox HAP lidar	120(°)	25(°)
Intel RealSense D435i camera	86(°)	57 (°)

B. Ground Truth Comparison

TABLE II
Translation and Rotation Quantitative Evaluation

	$t_x(m)$	$t_y(m)$	$t_z(m)$	$roll(°)$	$pitch(°)$	$yaw(°)$
Ground Truth	0.033	0.077	0.008	-177	-90	-90
ACSC [12]	0.0342	0.0452	0.0431	-164.048	-87.768	-106.918
DVLC [26]	0.0407	0.0799	0.0068	-174.668	-87.367	-96.236
Ours	**0.0325**	**0.0758**	**0.0089**	**-177.347**	**-90.156**	**-90.905**

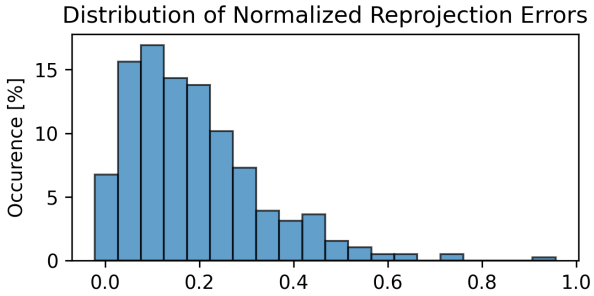

Fig. 6. Distribution of Normalized Reprojection Error

Given the 3D printed case model, we can measure the ground truth between sensors' optical frames. We quantitatively compared the mean of our calibration results with the ground truth in Table II. Our proposed method achieved translation errors at the millimeter level and rotation errors at the sub-degree level. We also estimated the sensors' extrinsic parameters using ACSC [12] and targeless calibration DVLC [26]. In the ACSC pipeline, we found that its board pose optimizer can easily fall into a wrong local minimum caused by

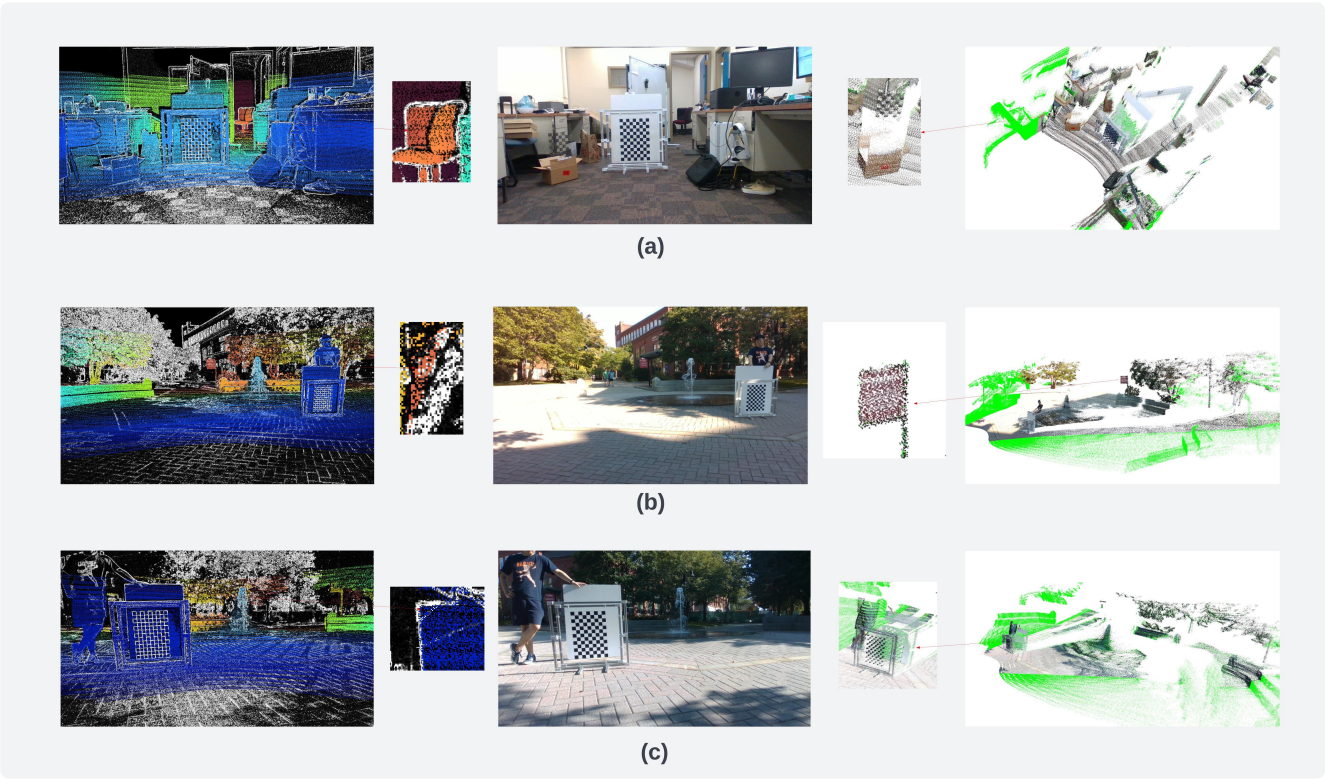

Fig. 7. Verification with 3 Scenes, using 200 stacking point clouds and first shot image. Reminds that the person holding the whiteboard and the whiteboard itself might move a bit during the data collection. (a) Indoor scene with calibration board in 3 meters. The zoom-in picture for the reprojection shows matching points for the chairs in the back room. The zoom-in picture for the colorized point cloud shows the paper box and another small acrylic calibration board aligned with the chair. (b) Outdoor scene with the calibration board in 3 meters on the right. The enlarged picture for the reprojection shows the matching points for the building (orange) behind the tree (yellow). The enlarged picture for the colorized point cloud shows the signage. (c) Outdoor scene with calibration board on the left in 2 meters. The zoom-in picture for the reprojection shows the matching points for the foam board behind the acrylic calibration board. The enlarged picture for the colorized point cloud shows the calibration board.

unclear edges between black and white grid reflectance intensity, which brought errors in 3D corners estimation. In DVLC pipeline, we found the SuperGlue 2D-3D feature matching network makes the optimization hard to converge as the number of features points increases from the data collection. We also evaluate the standard deviation (STD) of our calibration result by running the algorithm 100 times on the same collections of data. The evaluation shows that our algorithm achieves translation STD in $[0.0052, 0.0083, 0.0165]$ meters, rotation STD in $[0.0006, 0.0002, 0.0014]$ degrees.

C. Reprojection Error and Visualization

We also evaluate the reprojection error from the calibration. The occurrence of normalized reprojection error is shown in Fig. 6. We can see that the major distribution of the errors lies between 0 to 0.5 pixels. We also evaluate the result by visualization. Using the estimated extrinsic matrix, we reproject the point cloud of 50 continuous frames onto the image as shown in the left side of Fig. 7, where the color of the projected points is aligned with the depth of the point from the camera frame plane. To clearly see the image features, the images are converted into grayscale with amplified edges. We also colorize the point cloud by projecting the 2D image pixels as shown in the right side of Fig. 7. In the colorized point cloud, points that have no color assigned are rendered into green. All projected images and colorized point cloud has a zoom-in view to verify the edge alignments. We can see the projected points and well align the edges. Also, from the zoom-in colorized point cloud from Fig.7 (c), we can see that points of the black grid have clear color edges, proving our proposed method can be perfodistributermed in a short distance benefiting from the design of the calibration board.

V. Conclusion

This paper proposed a novel, efficient, and accurate extrinsic calibration method for lidar-camera sensors. A specially painted acrylic checkerboard is designed, which allows the lidar beam to pass through the white grids and reflect back from the black grids. More importantly, we propose an efficient and reliable algorithm that can extract the board's 3D geometric features in a short distance, improving the accuracy of image corner extraction. The experimental results demonstrate that the proposed methods can perform accurate calibration for a real lidar camera system. Future works include releasing the algorithm for open source sharing, building a more convenient calibration target instrument, and expanding different shapes of the calibration target.

REFERENCES

[1] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, Y. Lu, D. Zhou, Q. V. Le, A. Yuille, and M. Tan, "DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 182–17 191.

[2] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection With Transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1090–1099.

[3] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss, "Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset," *The International Journal of Robotics Research*, vol. 40, no. 8-9, pp. 959–967, Aug. 2021.

[4] Z. Zhuang, R. Li, K. Jia, Q. Wang, Y. Li, and M. Tan, "Perception-Aware Multi-Sensor Fusion for 3D LiDAR Semantic Segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 280–16 290.

[5] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Gläser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.

[6] J. Behley, A. Milioto, and C. Stachniss, "A Benchmark for LiDAR-based Panoptic Segmentation based on KITTI," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 13 596–13 603.

[7] Y. Zhao, X. Zhang, and X. Huang, "A technical survey and evaluation of traditional point cloud clustering methods for lidar panoptic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2464–2473.

[8] J. S. Berrio, M. Shan, S. Worrall, and E. Nebot, "Camera-LIDAR Integration: Probabilistic Sensor Fusion for Semantic Mapping," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7637–7652, Jul. 2022.

[9] W. Zhen, Y. Hu, J. Liu, and S. Scherer, "A Joint Optimization Approach of LiDAR-Camera Fusion for Accurate Dense 3-D Reconstructions," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3585–3592, Oct. 2019.

[10] 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)*. Madrid: IEEE, Oct. 2018, pp. 5562–5569.

[11] G. Yan, F. He, C. Shi, X. Cai, and Y. Li, "Joint Camera Intrinsic and LiDAR-Camera Extrinsic Calibration," Feb. 2023.

[12] J. Cui, J. Niu, Z. Ouyang, Y. He, and D. Liu, "ACSC: Automatic Calibration for Non-repetitive Scanning Solid-State LiDAR and Camera Systems," Nov. 2020.

[13] Y. Xie, L. Deng, T. Sun, Y. Fu, J. Li, X. Cui, H. Yin, S. Deng, J. Xiao, and B. Chen, "A4LidarTag: Depth-Based Fiducial Marker for Extrinsic Calibration of Solid-State Lidar and Camera," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6487–6494, Jul. 2022.

[14] C. Fang, S. Ding, Z. Dong, H. Li, S. Zhu, and P. Tan, "Single-Shot is Enough: Panoramic Infrastructure Based Calibration of Multiple Cameras and 3D LiDARs," Jul. 2021.

[15] J. Beltrán, C. Guindel, A. de la Escalera, and F. García, "Automatic Extrinsic Calibration Method for LiDAR and Camera Sensor Setups," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 677–17 689, Oct. 2022.

[16] J. Jiao, F. Chen, H. Wei, J. Wu, and M. Liu, "LCE-Calib: Automatic LiDAR-Frame/Event Camera Extrinsic Calibration With A Globally Optimal Solution," Mar. 2023.

[17] J.-K. Huang and J. W. Grizzle, "Improvements to Target-Based 3D LiDAR to Camera Calibration," *IEEE Access*, vol. 8, pp. 134 101–134 110, 2020.

[18] Y. Lyu, L. Bai, M. Elhousni, and X. Huang, "An Interactive LiDAR to Camera Calibration," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, Sep. 2019, pp. 1–6.

[19] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level Extrinsic Self Calibration of High Resolution LiDAR and Camera in Targetless Environments," Jun. 2021.

[20] Z. Luo, G. Yan, and Y. Li, "Calib-Anything: Zero-training LiDAR-Camera Extrinsic Calibration Method Using Segment Anything," Jun. 2023.

[21] 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.

[22] X. Jing, X. Ding, R. Xiong, H. Deng, and Y. Wang, "DXQ-Net: Differentiable LiDAR-Camera Extrinsic Calibration Using Quality-aware Flow," Mar. 2022.

[23] H. Yi, B. Liu, B. Zhao, and E. Liu, "Extrinsic Calibration for LiDAR–Camera Systems Using Direct 3D–2D Correspondences," *Remote Sensing*, vol. 14, no. 23, p. 6082, Nov. 2022.

[24] W. Xing, S. Lin, L. Yang, and J. Pan, "Target-free Extrinsic Calibration of Event-LiDAR Dyad using Edge Correspondences," May 2023.

[25] N. Ou, H. Cai, J. Yang, and J. Wang, "Targetless Extrinsic Calibration of Camera and Low-Resolution 3-D LiDAR," *IEEE Sensors Journal*, vol. 23, no. 10, pp. 10 889–10 899, May 2023.

[26] K. Koide, S. Oishi, M. Yokozuka, and A. Banno, "General, Single-shot, Target-less, and Automatic LiDAR-Camera Extrinsic Calibration Toolbox," Feb. 2023.

[27] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment Anything," Apr. 2023.

[28] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning Feature Matching with Graph Neural Networks," Mar. 2020.

[29] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[30] D. Deng, "DBSCAN Clustering Algorithm Based on Density," in *2020 7th International Forum on Electrical Engineering and Automation (IFEEA)*, Sep. 2020, pp. 949–953.

[31] Z. Wang, W. Wu, X. Xu, and D. Xue, "Recognition and location of the internal corners of planar checkerboard calibration pattern image," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 894–906, Feb. 2007.

[32] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software*, vol. 23, no. 4, pp. 550–560, Dec. 1997.