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

SGCalib: A Two-stage Camera-LiDAR Calibration Method Using Semantic Information and Geometric Features

Zhipeng Lin[1,3], Zhi Gao[2,3*], Xinyi Liu[2], Jialiang Wang[2], Weiwei Song[4], Ben M. Chen[1], Chenyang Li[2], Yue Huang[2], Yuhan Zhu[2]

***Abstract*— Extrinsic calibration is an essential prerequisite for the applications of camera-LiDAR fusion. Existing methods either suffer from the complex offline setting of man-made targets or tend to produce suboptimal and unrobust results. In this paper, we propose an online two-stage calibration method that estimates robust and accurate extrinsic parameters between camera and LiDAR. This is a novel work to use semantic information and geometric features jointly in calibration to promote accuracy and robustness. In the first stage, we detect objects in the image and point cloud and build graphs on the objects using Delaunay triangulation. Then, we design a novel graph matching algorithm to associate the objects in the two data domains and extract pairs of 2D-3D points. Using the PnP solver, we get robust initial extrinsic parameters. Then, in the second stage, we design a new optimization formulation with semantic information and geometric features to generate accurate extrinsic parameters with the initial value from the first stage. Extensive experiments on solid-state LiDAR, conventional spinning LiDAR and KITTI datasets have verified the robustness and accuracy of our method which outperforms existing works. We will share the code publicly to benefit the community (after review stages).**

I. INTRODUCTION

The camera-LiDAR calibration plays a vital role in sensor-fused tasks, such as resilient SLAM [1], autonomous driving [2], robot navigation [3], and 3D reconstruction [4]. Sensor fusion can utilize complementary sensor information and reduce data uncertainty in challenging scenes. The accurate extrinsic parameters can be the prerequisite for camera-LiDAR fusion as they realize the injective mapping between image color and 3D points. As the extrinsic parameters can slightly drift during the running of the vehicle or robot, online calibration is needed as it is fast to execute and does not need artificial targets or tuning environment settings.

Traditional methods for camera-LiDAR calibration usually rely on checkerboards or other man-made objects [5], [6], which makes the calibration work laborious and time-consuming. The methods based on geometric features [7], [8], [9] or motion estimation [10], [11], [12] get rid of the external target and enable online calibration. But geometric feature based approaches rely on a robust initial value which is usually hard to estimate directly. And the motion based methods lack the pixel-level alignment to ensure accuracy. Recently, the methods based on CNN networks [13], [14], [15] are proposed for camera-LiDAR calibration. These methods typically input a large amount of data for training, and their application prospects are affected by the weakness of deep learning models in generalization.

Fig. 1. Point cloud accumulated by a Livox Avia LiDAR. It is colored using an image and extrinsic parameters generated from the proposed method. The color detail of the point cloud has been restored correctly.

In this paper, we propose a coarse-to-fine method to generate accurate and robust results for camera-LiDAR calibration. The main novelty is the data association between the image and point cloud based on graph matching and joint optimization with semantic information and geometric features. In the first stage, we perform object detection and graph matching to associate the objects in the image and point cloud. Then, we extract pairs of 2D-3D points and use a PnP solver to get robust initial extrinsic parameters. In the second stage, we design a new optimization formulation with semantic information and edge features to generate accurate calibration results with the initial parameters from the first stage. Fig.1 visualizes our calibration result. With the image and extrinsic parameters produced by our work, the point cloud of the building is colored accurately with no deviation. Besides, our approach enables online calibration as it does not need any markers or environment settings. Specifically, follows are our main contributions:

- We propose a novel graph matching method to generate robust coarse calibration results utilizing object detection, Delaunay triangulation, and PnP solver.
- We design a joint optimization objective function using geometric features and semantic information to produce accurate extrinsic parameters.
- We have validated the robustness and accuracy of our methods with extensive tests across various environ-

This work was supported in part by Hubei Province Natural Science Foundation under Grant 2021CFA088, and in part by Science and TechnologyMajor Project under Grants 2021AAA010 and 2021AAA010-3. (Corresponding author: Zhi Gao.)

[1]Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong SAR 999077, China

[2]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430072, China

[3]Hubei Luojia Laboratory, Wuhan 430079, China.

[4]Department of Mathematics and Theories, Peng Cheng Laboratory, Shenzhen 518000, China.

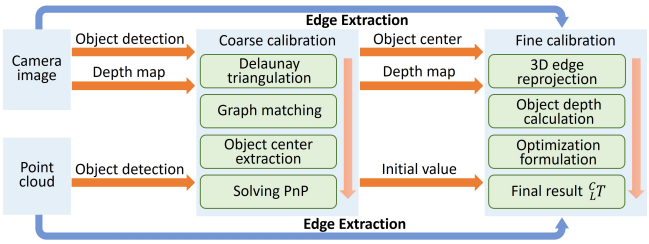

Fig. 2. Overview of the architecture of our proposed method.

Fig. 3. (a) 2D Object detection result. (b) BEV result from monocular 3D object detection. (c) 3D object detection result in point cloud.

ments on the emerging solid-state LiDAR and the conventional spinning LiDAR.

II. RELATED WORK

Camera-LiDAR calibration has experienced significant development in recent years. Classical methods usually rely on external targets, while geometric feature based and motion based approaches try eliminating them. Furthermore, many techniques use a CNN network or semantic information to perform calibration.

A. External target based approaches

The well-known KITTI dataset [5] uses the most representative checkerboard based calibration method. It needs up to 9 checkerboards, which makes it difficult to reproduce. The ICP-based method [12], [16] extracted the checkerboard corners as 3D points to get the rigid body transformation. The polygonal planar board is also widely used to supply corresponding features to align the point cloud and image [17]. Also, the Quick Response (QR) code [18] or ball object [19] is also used as it is easily detected in the image. These approaches rely on complicated setup requirements and are only suitable for offline usage.

B. Geometric feature based and motion based approaches

The utilization of geometric features and sensor motion is researched to get rid of external targets in calibration. The optimization objective can be formulated using the distance of the gradients of the depth map from the image and the projected intensity from point cloud [20]. The correspondences [7] between the edge features from point clouds and images are also widely used. Similarly, some methods [8], [9] utilize local edge features in real environments to formulate 3D-to-2D errors. Their accuracy relies heavily on the initial extrinsic parameters, which are difficult to obtain directly.

The motion of each sensor is used to obtain the calibration results which can be refined with appearance information [10]. Other methods [12], [11], [21] introduce the hand-eye calibration methods into 2D-3D calibration, which can

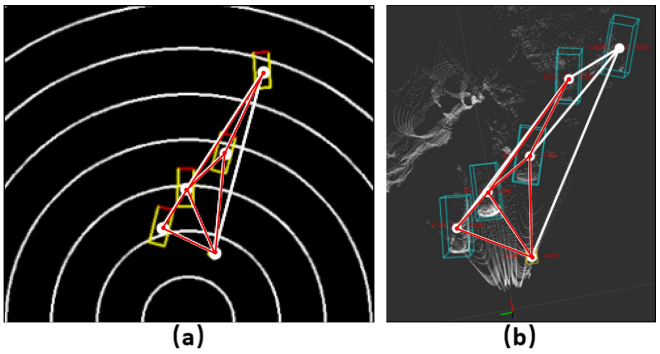

Fig. 4. Delaunay triangulation and graph matching in (a) BEV image from monocular 3D object detection (b) 3D object detection in point cloud.

produce a coarse estimation of the extrinsic parameters. The motion-based calibration approaches need a large quantity of data on different motion patterns. These methods lack the pixel-level alignment to ensure their accuracy.

C. CNN and sematic information based approaches

The supervised CNN network can be trained directly to generate the extrinsic parameters directly on the KITTI dataset [13]. And some works introduce additional constraints to promote the performance of the model, such as geometric and photometric consistency [14]. Some methods [15], [22] also added the cost volume layer of the features extracted from RGB image and the point cloud depth map to refine the result. The effectiveness of these CNN based calibration models binds tightly together with the pattern of the training set and the generalization is limited.

To avoid the generalization limitation of the CNN model, some approaches [23], [24] used CNN based semantic segmentation with specific calibration quality metrics to achieve the object-based alignment method. The semantic [25] centroid is introduced to get the initial extrinsic parameters by solving the PnP problem. The performance of these methods depends largely on the accuracy of semantic segmentation, whose deviation will be transmitted to the calibration result.

Our method associates the objects in the image and point cloud to generate initial calibration result without using external target. And we align the pixel-level geometric features in the fine calibration to ensure accuracy and generalization. Our method ensures the accuracy and robustness of the results and enables online calibration.

III. METHODS

The overall flow of the proposed method is shown in Fig.2. In coarse calibration stage, we perform object detection and graph matching to associate the objects in the image and point cloud. Then, we use a PnP solver to get coarse calibration results. In fine calibration stage, we formulate a joint objective function with extracted geometric features and semantic results to produce accurate extrinsic parameters.

A. Object detection and association in 2D and 3D

We perform 2D object detection using YOLO v8 [26], a state-of-the-art method with high accuracy and speed. To detect 3D objects, we use Livox Detection V2.0, which is

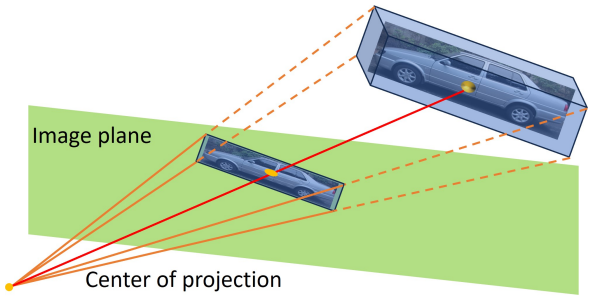

Fig. 5. Projection transformation of an object from 3D to 2D.

designed and trained for point cloud data generated by Livox LiDAR. A typical 2D and 3D detection result in the same scene is shown in Fig.3.

To associate the objects in the image and point cloud, we construct the graph with the targets in the scene as nodes. To facilitate the graph matching process, we expect to keep the geometry structure of the graphs. Thus, we leverage the monocular 3D object detection [27] algorithm to transfer the 2D object detection to the bird's eye view (BEV) as shown in Fig.3 (b). Meanwhile, we project the 3D objects on the ground to get the BEV from the point cloud, as shown in Fig.3 (c). Then, the geometry structure of the relative position of the object is maintained, especially the similarity of geometric shapes.

We perform Delaunay triangulation to establish the graph of the objects in the BEV of the image and point cloud. The Delaunay triangulation has the property of uniqueness, which is fit for our task [28]. The result is shown in Fig.4 (in white).

We perform graph matching to find the correspondence of the object in the BEV of image and point cloud. As the camera and LiDAR usually have different fields of view (FOV), objects detected in one data domain may not be detected in another. Thus, our task becomes the subgraph isomorphism problem. As shown in Fig.4, our task is to find the common subgraph (in red) and match the node. Different from the general subgraph isomorphism problem, we keep the geometry structure and the node category. With these prerequisites, we can design a more direct method to solve it. Firstly, we query triangle similarity to find the similar triangles in the two graphs. Then, for a pair of similar triangles, we perform verification for the Euclidean distance and category similarity of the nearest nodes in the whole graph. If the verification passes, we consider it a good match and output the matching result. In this way, we establish data association between image and point cloud.

As illustrated in Fig.5, for an object that appears in the image and point cloud, the geometric center of the 3D object is approximately still the geometric center on the 2D image after projection transformation. Thus, we extract pairs of 2D-3D points from the center of the bounding box in the image domain and the point cloud when the objects are correctly associated. Now, we can solve the PnP problem to get the rigid body transformation between LiDAR and camera. Note that the coplanar PnP problem can have one unique closed-form solution for configurations of four pairs of points [29], [30], [31]. The center points of vehicles and pedestrians on the ground form coplanar points.

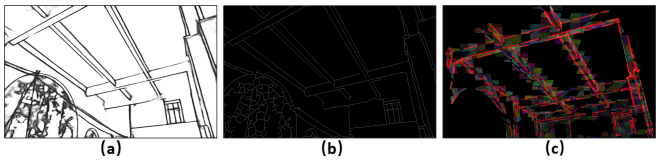

Fig. 6. (a) Edge extraction result in the image (b) skeletonization result (c) edge extraction result in the point cloud.

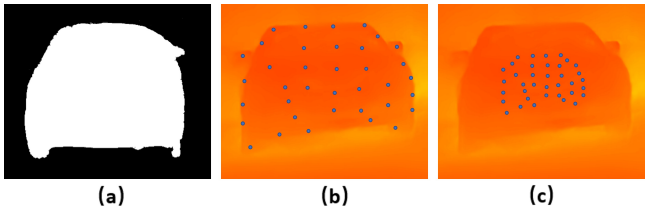

Fig. 7. (a) Semantic segmentation mask for a car object. (b) 3D Object points projected on the depth map. (c) The null space of the optimization.

B. Edge extraction in image and point cloud

In the image domain, we use the DexiNed [32] to perform the edge extraction, which is the state-of-the-art CNN based edge detection network. Compared with traditional methods such as the Canny algorithm [33], DexiNed can use high-level image features to ensure consistency at the edge. As shown in the Fig.6(a), the edges extracted by DexiNed are not precise enough. So we conduct skeletonization [34] to reduce the thick edges to a 1-pixel wide representation, shown in the Fig.6(b). A K-D tree (k = 2) will be built to store the extracted edge features.

We adopt the idea to extract edges in the point cloud with voxels [9]. Note that the density is not essential, but we aim to improve accuracy through it. Firstly, we divide the point cloud into voxels of appropriate grids. In each voxel, we iteratively fit and extract planes using RANSAC, and merge duplicate planes. Then, we can solve for the plane intersection lines utilizing the intersacted plane pairs and form an angle within a certain range. In Fig.6(c), the red line represents the point cloud edges reprojected to image, and the colored areas represent the voxels. Moreover, we achieve the self-adaptive voxel size strategy to extract edges effectively at all distances.

Then, we establish the correspondence between edges extracted from point clouds and images. The edges extracted from point cloud are transformed into camera frame and then projected on the image plane with the intrinsic and initial extrinsic parameters. Then, we search for the nearest image edges for projected LiDAR edges and establish correspondences. In addition, the direction vectors and normal vectors of the edges are also used to verify the matches, as the corresponding edges should be nearly parallel.

C. Calibration Formulation and Optimization

With accurate extrinsic parameters, the edge projected to the image from the point cloud should be collinear with its

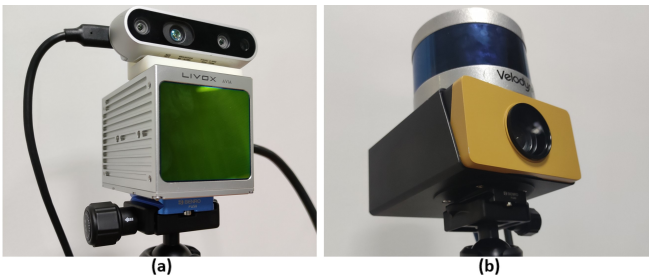

Fig. 8. Our sensor suites. (a) Livox Avia LiDAR and Intel Realsense D435i camera, which are used for most experiments. (b) spinning LiDAR (Velodyne VLP-16 LiDAR) and industry camera (MER2-630-60U3C).

Fig. 9. Point cloud reprojection result of (a) coarse calibration and (b) fine calibration. The details of the car and pedestrian in the boxes are improved a lot after the fine calibration.

corresponding edge $(\mathbf{n}_i, \mathbf{q}_i)$ in the image, where $\mathbf{q}_i$ is a point lying on the line and $\mathbf{n}_i$ represents the normal vector. We investigate and simplify the formulation [9] below.

$$\mathbf{n}_i^T \left(\mathbf{f} \left({}_L^C\mathbf{T}\, {}^L\mathbf{P}_i \right) - \mathbf{q}_i \right) = 0 \tag{1}$$

where ${}^L\mathbf{P}_i \in \mathbb{R}^3$ is an edge point extracted from the point cloud and $\mathbf{q}_i \in \mathbb{R}^2$ is an edge point extracted from the corresponding edge in the image with its normal vector $\mathbf{n}_i$. And $\mathbf{f}(\cdot)$ represents the projection from the camera frame to the image plane. The ${}_L^C\mathbf{T}$ represents the transformation from LiDAR frame to camera frame. Then we define a least square objective function of edge reprojection $\mathbf{E}_{edge}$.

$$\mathbf{E}_{edge} = \sum_i \frac{1}{2} \parallel \mathbf{n}_i^T \left(\mathbf{f} \left({}_L^C\mathbf{T}\, {}^L\mathbf{P}_i \right) - \mathbf{q}_i \right) \parallel_2^2 \tag{2}$$

Calculating the derivative of the transformation ${}_L^C\mathbf{T}$ directly is difficult. Here, we use the left perturbation scheme and apply increment on the Lie Group. Note that using other methods, such as quaternion, is equally effective.

Thus, let $\xi = [\rho, \phi] = (\alpha\, \beta\, \gamma\, x\, y\, z)^T \in \mathfrak{se}(3)$, and the transformation matrix:

$${}_L^C\mathbf{T} = \exp\left(\xi^\wedge\right) \tag{3}$$

According to the chain rule and the derivative of Lie algebra, we present the Jacobian for the Gaussian-Newton method.

$$\mathbf{J}_{edge} = \frac{\partial(\mathbf{n}_i^T \left(\mathbf{f} \left({}_L^C\mathbf{T}\, {}^L\mathbf{P}_i \right) - \mathbf{q}_i \right))}{\partial \xi} = \mathbf{n}_i^T \frac{\partial \mathbf{f}}{\partial \xi} = \mathbf{n}_i^T \mathbf{J}_{\mathbf{f}} \tag{4}$$

$$\mathbf{J}_{\mathbf{f}} = \begin{bmatrix} \frac{-f_x X_i Y_i}{Z_i^2} & f_x + \frac{f_x X_i^2}{Z_i^2} & \frac{-f_x Y_i}{Z_i} & \frac{f_x}{Z_i} & 0 & \frac{-f_x X_i}{Z_i^2} \\ -f_y - \frac{f_y Y_i^2}{Z_i^2} & \frac{f_y X_i Y_i}{Z_i^2} & \frac{f_y X_i}{Z_i} & 0 & \frac{f_y}{Z_i} & \frac{-f_y Y_i}{Z_i^2} \end{bmatrix} \tag{5}$$

Similarly, the center of a 3D object should coincide with the center of the matched 2D object in the image after projection transformation. Thus, we can utilize the semantic result in coarse calibration to formulate the objective function of PnP. In optimization, this item is used as a monitoring item to prevent falling to a local optimal and lose accuracy.

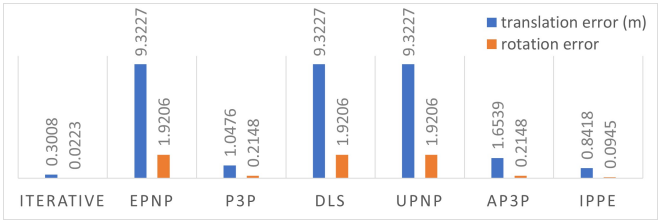

Fig. 10. The MSE error of different methods to solve the PnP problem.

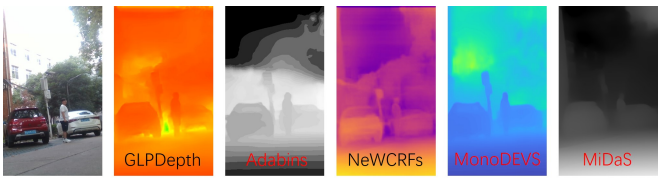

Fig. 11. Results of some monocular depth estimation methods. The GLPDepth [35] outperforms other methods in object depth consistency.

$$\mathbf{E}_{PnP} = \sum_j \frac{1}{2} \parallel \mathbf{f} \left({}_L^C\mathbf{T}\, {}^L\mathbf{P}_j \right) - \mathbf{q}_j \parallel_2^2 \tag{6}$$

$$\mathbf{J}_{PnP} = \frac{\partial \left(\mathbf{f} \left({}_L^C\mathbf{T}\, {}^L\mathbf{P}_j \right) - \mathbf{q}_j \right)}{\partial \xi} = \frac{\partial \mathbf{f}}{\partial \xi} = \mathbf{J}_{\mathbf{f}} \tag{7}$$

where the ${}^L\mathbf{P}_j$ and $\mathbf{q}_j$ are the pairs of points of object center extracted for solving PnP in subsection III. B.

Semantic segmentation result is often used as a reward mask to correspond the laser points and camera frame. However, the semantic mask is usually a binary mask that is not smooth for optimization, as shown in Fig.7(a). We find that the depth map reserves the semantic border and is smooth for optimization. The depth of an object is usually smaller than its surroundings, as shown in Fig.7. With the pixel-wise result generated by GLPDepth [35], we have a depth map $\mathcal{D} : \mathbb{R}^2 \to \mathbb{R}$,

$$\mathbf{d} = \mathcal{D}(\mathbf{p}_I) \tag{8}$$

We first extract the laser points of corresponding 3D objects ${}^L\mathbf{P}_k$. To encourage laser points to fall on the image object area, we define the objective function $\mathbf{E}_{depth}$ in eq.(5) by measuring the overlap between the depth map and the laser points projected onto the camera frame. Minimizing this objective function can align the object's shape borders.

$$\mathbf{E}_{depth} = \sum_k \frac{1}{2} \parallel \mathbf{d} \circ \left(\mathbf{f} \left({}_L^C\mathbf{T}\, {}^L\mathbf{P}_k \right) \right) \parallel_2^2 \tag{9}$$

$$\mathbf{J}_{depth} = \frac{\partial(\mathbf{d} \circ \left(\mathbf{f} \left({}_L^C\mathbf{T}\, {}^L\mathbf{P}_k \right) \right)}{\partial \xi} = \frac{\partial \mathbf{d}}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial \xi} = \frac{\partial \mathbf{d}}{\partial \mathbf{f}} \mathbf{J}_{\mathbf{f}} \tag{10}$$

Note that $\frac{\partial \mathbf{d}}{\partial \mathbf{f}}$ is the image gradient vector in the depth map. Due to objects in the depth map typically having a consistent depth, there exists a null space along the camera observation axis. as shown in Fig.7(c) theoretically. This will not affect

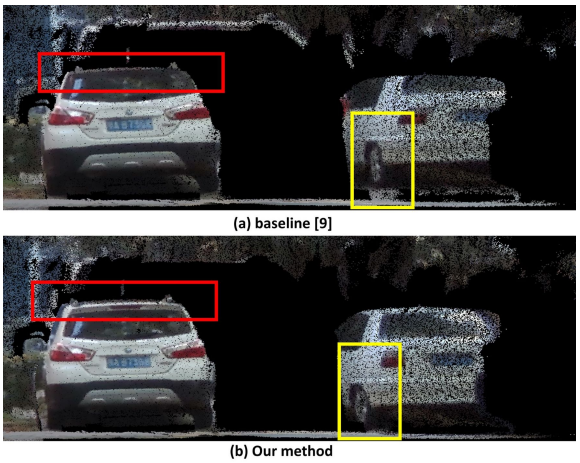

Fig. 12. Point cloud coloring results from: (a) baseline [9]; (b) our method.

the result as other objective terms will generate effective optimization directions.

Now, we have the joint optimization formulation, and it is sufficient to use the Gaussian-Newton method to get the optimal result.

$$\mathbf{E} = \mathbf{E}_{edge} + \mathbf{E}_{PnP} + \mathbf{E}_{depth} \tag{11}$$

$$\mathbf{J} = \mathbf{J}_{edge} + \mathbf{J}_{PnP} + \mathbf{J}_{depth} \tag{12}$$

IV. EXPERIMENTS

We validate the proposed method in extensive real-world experiments. The experiments are mainly performed on Livox Avia, which is a solid-state LiDAR and can generate high-resolution point-cloud at stationary due to its non-repetitive scanning [36]. And we use the camera on Intel Realsense D435i (see Fig.8(a)). We also present results on KITTI datasets. We assume that the camera's intrinsic parameters have been accurately calibrated. During data acquisition, we collect images and point clouds simultaneously with the LiDAR and camera at a fixed position. A desktop PC computing platform is used, which has an AMD Ryzen 9 7900X CPU and a NVIDIA GeForce RTX 3090 GPU.

A. The coarse calibration

In the first stage, we perform 2D object detection, monocular 3D object detection, and 3D object detection in point cloud. The results are shown in Fig.3. For the objects, we perform Delaunay triangulation and graph matching as shown in Fig.4. The white graph is the result of Delaunay triangulation, and the red graph is the matched isomorphic subgraph. Thus, we correspond the object in image and laser points. Then, we extract the object center to get the 2D-3D corresponding points and use a PnP solver to generate the coarse calibration result.

Here we perform reprojection using the initial extrinsic parameters and show the result in Fig.9(a). The point cloud has been correctly colored on the whole, but there are slight deviations in detail, especially in the purple and green rectangles. In the next subsection, we will demonstrate the effectiveness of our initial value for fine calibration.

TABLE I

ERROR OF CALIBRATION METHODS.

Method	$t_x(m)$	$t_y(m)$	$t_z(m)$	r_x	r_y	r_z
baseline [9]	0.0089	0.0145	0.0098	0.0203	0.0273	0.0198
ours	0.0039	0.0013	-0.0016	0.017	0.0073	0.0168

TABLE II

TIME CONSUMPTION IN CALIBRATION.

data collection	coarse calibration	fine calibration	overall
1s	77ms	873ms	1.95s

Many methods can solve the PnP problem using at least 4 pairs of points. Here we present a comparison of some of the PnP solvers. We use the ground truth from the result of Appendix section and calculate the MSE error of the translation and rotation vector. The result is shown in Fig.10. We can see that the iterative method produces minimal translation and rotation errors, which outperforms other approaches. This experiment is performed with the help of the OpenCV toolkit.

B. The fine calibration

The edge extraction result in the image and point cloud is shown in Fig.6. Then we investigate monocular depth estimation methods such as GLPDepth [35], Adabins [37], NeWCrFs [38], MonoDEVS [39], and MiDaS [40]. The results are compared in Fig.11. We can see that only GLPDepth divides the depth of the objects clearly and keeps the depth consistency of the same object. So, we suppose that GLPDepth outperforms other methods for our task, and we adopt it as our approach's depth estimator.

We formulate the joint objective and perform optimization to get the fine calibration results. We present the reprojection result in Fig.9(b) for visualization. The result is improved compared with the initial result, especially for the cars and the pedestrians in the boxes. We use the state-of-the-art geometric feature-based calibration method [9] as the baseline. We compare our method with the baseline, and the result is shown in Fig.12. We present the error of calibration of our method and baseline in TABLE I. The method to acquire ground truth is shown in the Appendix section. Note that without a initial guess, the baseline method will not work. Thus we provide our raw calibration result for the baseline method. As is shown in Fig.12, our method achieves better performance than the baseline, with the point cloud more accurately colored in detail, especially as noted in the red and yellow boxes. Since the baseline method only uses the edge features for optimization, it only optimizes the edge reprojection objective $\mathbf{E}_{edge}$. Thus, it could fall into local minimum and cannot produce optimal result. The results in Fig.12 proved that our introduced PnP error $\mathbf{E}_{depth}$ and depth error $\mathbf{E}_{PnP}$ have improved the performance effectively. We

Fig. 13. The coarse calibration process using our Velodyne VLP-16 LiDAR devices.

Fig. 14. The calibration result of our Velodyne VLP-16 LiDAR devices: (a) coarse calibration (b) fine calibration.

present the time consumption in TABLE II to show the ability of our method to perform online calibration.

C. Experiments on Velodyne LiDAR

The prior result is based on Livox Avia LiDAR. We can also applied our method to conventional mechanical spinning LiDAR, which can only generate low-density point clouds in stationary state. We use the device in Fig.8(b), which consists of a spinning LiDAR (Velodyne VLP-16 LiDAR) and an industrial camera (MER2-630-60U3C). The process of coarse calibration is shown in Fig.13. Specifically, we use Part-A2-Free [41] to perform 3D object detection. Then, with the corresponding points, we solve the PnP problem and get the coarse calibration result. Using a single frame is already sufficient for calculation, but we use LOAM [42] to generate a more dense point cloud. In practice, the phenomenon of detecting duplicate planes occurs. We will merge the same plane based on the distance and angle between the detected planes. The final result is shown in Fig.14. The left part is the point cloud colored with the coarse calibration result and the right is colored with the fine calibration result. Due to space limitations, we present more experimental result in our github repository.

D. Experiments on KITTI dataset

We evaluate our method on KITTI dataset to show the generalization and robustness. The coarse calibration process is presented in Fig.15. With the coarse calibration parameters, we perform fine calibration and projecting the point cloud to the image. There is no dense point cloud in the KITTI dataset, so we also use LOAM [42] to generate a much higher resolution scan for edge extration. The point cloud projection is shown in Fig.16 where the car and the road are accurately projected. Due to space limitations, we present more experimental result in our github repository.

V. CONCLUSIONS

This paper proposes an online two-stage camera-LiDAR calibration method to generate accurate extrinsic parameters with semantic information and geometric features. We design a graph matching algorithm to generate robust raw calibration results based on object detection, Delaunay triangulation, and PnP solver. We formulate a joint optimization objective with natural edge features and semantic results to generate high-accuracy extrinsic parameters. Extensive experiments demonstrate the effectiveness of our approach. And our method ensures the generalization that can be applied to different types of LiDARs with camera.

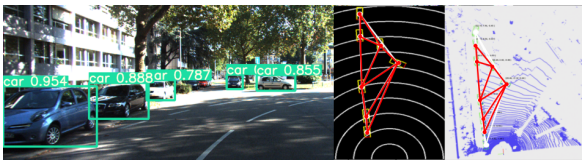

Fig. 15. The raw calibration process on the KITTI dataset.

Fig. 16. The calibration result on the KITTI dataset.

VI. APPENDIX

Here, we present our method to acquire the ground truth of the extrinsic parameters between the camera and LiDAR. We use the equipment called standard target ball, which is widely used in 3D laser scanners such as Faro and Leica. The ball is shown in the upleft of Fig.17 with 145 mm diameter and less than 1mm machining error. We extract the ball in laser points and the corresponded in the image. With the pairs of ball centers in the 2D and 3D, we solve the PnP problem to get the extrinsic parameters between the camera and LiDAR.

The circle detection result in the image is shown in the upright of Fig.17. In our practice, we find that the circle is more like an ellipse in images. So, we perform ellipse detection in practice. The laser points are accumulated for 10s. The result of ball detection, namely sphere fitting, is shown at the bottom of Fig.17. Note that we change the position of the balls and collect the data more than 5 times to ensure the accuracy of the result.

REFERENCES

[1] J. Lin and F. Zhang, "R 3 live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 10672–10678, IEEE, 2022.

[2] K. Banerjee, D. Notz, J. Windelen, S. Gavarraju, and M. He, "Online camera lidar fusion and object detection on hybrid data for autonomous driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1632–1638, IEEE, 2018.

[3] S. Gatesichapakorn, J. Takamatsu, and M. Ruchanurucks, "Ros based autonomous mobile robot navigation using 2d lidar and rgb-d camera," in *2019 First international symposium on instrumentation, control, artificial intelligence, and robotics (ICA-SYMP)*, pp. 151–154, IEEE, 2019.

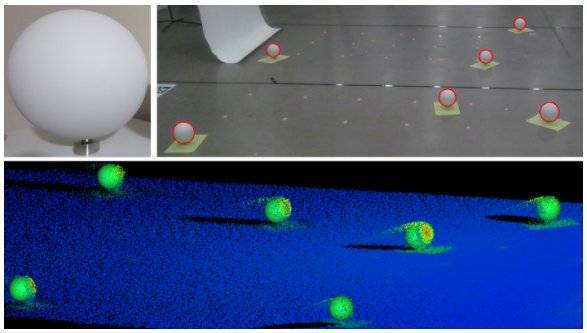

Fig. 17. The standard target ball and the extraction in image and point cloud. The point cloud is colored with laser reflection intensity.

[4] M. Waechter, N. Moehrle, and M. Goesele, "Let there be color! — Large-scale texturing of 3D reconstructions," in *Proceedings of the European Conference on Computer Vision*, Springer, 2014.

[5] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE international conference on robotics and automation*, pp. 3936–3943, IEEE, 2012.

[6] G. Yan, F. He, C. Shi, P. Wei, X. Cai, and Y. Li, "Joint camera intrinsic and lidar-camera extrinsic calibration," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11446–11452, IEEE, 2023.

[7] Z. Bai, G. Jiang, and A. Xu, "Lidar-camera calibration using line correspondences," *Sensors*, vol. 20, no. 21, p. 6319, 2020.

[8] M. Á. Muñoz-Bañón, F. A. Candelas, and F. Torres, "Targetless camera-lidar calibration in unstructured environments," *IEEE Access*, vol. 8, pp. 143692–143705, 2020.

[9] 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.

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

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

[12] Y. Su, Y. Ding, J. Yang, and H. Kong, "A two-step approach to lidar-camera calibration," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 6834–6841, IEEE, 2021.

[13] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "Regnet: Multimodal sensor registration using deep neural networks," in *2017 IEEE intelligent vehicles symposium (IV)*, pp. 1803–1810, IEEE, 2017.

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

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

[16] J. Cui, J. Niu, Z. Ouyang, Y. He, and D. Liu, "Acsc: Automatic calibration for non-repetitive scanning solid-state lidar and camera systems," *arXiv preprint arXiv:2011.08516*, 2020.

[17] Y. Park, S. Yun, C. S. Won, K. Cho, K. Um, and S. Sim, "Calibration between color camera and 3d lidar instruments with a polygonal planar board," *Sensors*, vol. 14, no. 3, pp. 5333–5353, 2014.

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

[19] M. Pereira, D. Silva, V. Santos, and P. Dias, "Self calibration of multiple lidars and cameras on autonomous vehicles," *Robotics and Autonomous Systems*, vol. 83, pp. 326–337, 2016.

[20] J. Castorena, U. S. Kamilov, and P. T. Boufounos, "Autocalibration of lidar and optical cameras via edge alignment," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2862–2866, IEEE, 2016.

[21] X. Liu, C. Yuan, and F. Zhang, "Targetless extrinsic calibration of multiple small fov lidars and cameras using adaptive voxelization," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[22] A. N. Duy and M. Yoo, "Calibration-net: Lidar and camera auto-calibration using cost volume and convolutional neural network," in *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 141–144, IEEE, 2022.

[23] B. Nagy, L. Kovács, and C. Benedek, "Sfm and semantic information based online targetless camera-lidar self-calibration," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 1317–1321, IEEE, 2019.

[24] 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)*, pp. 4970–4976, IEEE, 2020.

[25] W. Wang, S. Nobuhara, R. Nakamura, and K. Sakurada, "Soic: Semantic online initialization and calibration for lidar and camera," *arXiv preprint arXiv:2003.04260*, 2020.

[26] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan. 2023.

[27] Y. Zhang, J. Lu, and J. Zhou, "Objects are different: Flexible monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3289–3298, 2021.

[28] J. Dou, J. Li, *et al.*, "Robust image matching based on sift and delaunay triangulation," *Chinese Optics Letters*, vol. 10, no. s1, pp. 011001–311005, 2012.

[29] D. Oberkampf, D. F. DeMenthon, and L. S. Davis, "Iterative pose estimation using coplanar feature points," *Computer Vision and Image Understanding*, vol. 63, no. 3, pp. 495–511, 1996.

[30] M. L. Liu and K. H. Wong, "Pose estimation using four corresponding points," *Pattern Recognition Letters*, vol. 20, no. 1, pp. 69–74, 1999.

[31] Z. Hu and F. Wu, "A note on the number of solutions of the noncoplanar p4p problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 550–555, 2002.

[32] X. S. Poma, E. Riba, and A. Sappa, "Dense extreme inception network: Towards a robust cnn model for edge detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1923–1932, 2020.

[33] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[34] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[35] D. Kim, W. Ga, P. Ahn, D. Joo, S. Chun, and J. Kim, "Global-local path networks for monocular depth estimation with vertical cutdepth," *arXiv preprint arXiv:2201.07436*, 2022.

[36] Z. Liu, F. Zhang, and X. Hong, "Low-cost retina-like robotic lidars based on incommensurable scanning," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 58–68, 2021.

[37] S. F. Bhat, I. Alhashim, and P. Wonka, "Adabins: Depth estimation using adaptive bins," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4009–4018, 2021.

[38] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "New crfs: Neural window fully-connected crfs for monocular depth estimation," *arXiv preprint arXiv:2203.01502*, 2022.

[39] A. Gurram, A. F. Tuna, F. Shen, O. Urfalioglu, and A. M. López, "Monocular depth estimation through virtual-world supervision and real-world sfm self-supervision," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[40] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[41] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2647–2664, 2020.

[42] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time.," in *Robotics: Science and Systems*, vol. 2, pp. 1–9, Berkeley, CA, 2014.