2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

DEdgeNet: Extrinsic Calibration of Camera and LiDAR with Depth-discontinuous Edges

Yiyang Hu[1], Hui Ma[2], Leiping Jie[3] and Hui Zhang[4]

***Abstract*— This paper addresses the problem of calibrating extrinsic parameter matrix between an RGB camera and a LiDAR. Multimodal sensing systems are essential for fully autonomous navigation platforms. A key pre-requisite for such a system is calibration between different sensors. As the two most widely equipped sensors, calibration between RGB cameras and LiDARs remains challenging. Existing methods address this problem without using explicit geometric priors. In this paper, we propose a novel real-time network that utilizes depth-discontinuous edges extracted from a single image to calibrate cameras and LiDARs. Our network consists of two key components: (1) a self-supervised edge extraction network named DEdgeNet, which detects depth-discontinuous edges from a single image and extracts corresponding features; (2) prediction of the extrinsic parameter matrix between the camera and the LiDAR by matching fixed features in RGB images and updating depth features in a coarse-to-fine frame. Specifically, considering that edges are rich and common in natural scenes, DEdgeNet simplifies RGB image encoding and extracts fixed edges for feature matching. We conducted extensive experiments on the KITTI-odometry dataset. The results show that our method achieves an average rotation error of 0.028° and an average translation error of 0.247 cm, which demonstrates the superiority of our method.**

I. Introduction

Multimodal sensor fusion provides data in different dimensions for environment perception, improving the robustness and accuracy of many robotic applications, such as robot perception [1], autonomous navigation [2], dense mapping [3], and localization [4]. The two sensors, camera and LiDAR, provide the richest scene information for the entire system. Cameras provide rich 2D color appearances in images, and LiDARs provide accurate 3D geometric metrics in point clouds. The fusion of image and point cloud captures the color of an object while obtaining geometric metrics. As a key step in fusion, accurate camera calibration is a critial step and can provide a rigid body transformation with six degrees of freedom (DoF) between cameras and LiDARs [5].

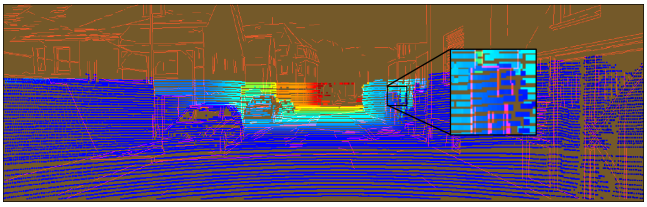

Fig. 1. This figure correlates the projection of edges in the RGB image with the corresponding depths in the point cloud. Different colors indicate depth changes, red represents large depth changes, and blue represents small depth changes. There is a noticeable change in color at the edges of objects, which means that the edges are critical for matching the image to the point cloud.

The estimation of rigid body transformations has been extensively studied in [6][7]. [8] and [9] can accurately assess the positional relationship between camera and LiDAR, but are limited by offline processing requirements and are not feasible in many natural environments. [10] provides a target-free method to accomplish calibration, but their method requires high-quality point clouds and proper initialization. Most optimization-based calibration algorithms [10][11] are offline and time-consuming and require proper optimization. In contrast, learning-based methods [12][13][14] can simultaneously predict the extrinsic parameter matrix of each image and the corresponding point cloud. These methods often repeatedly extract RGB images and depth map of point cloud projections, regardless of the potential correlation of 2D and 3D features. In contrast, we find that edges in images are usually the locations of depth changes (see Fig. 1). Therefore, we associate the RGB image with the depth map to calculate the average depth around the edges of the image, which corresponds to edges where the depth map varies largely. The projection of the point cloud varies greatly with a given initial transformation matrix, while the RGB image does not, suggesting that it is feasible to keep the features of the RGB image unchanged.

In this paper, we propose DEdgeNet, an online self-supervised network that extracts heatmaps of depth-discontinuous edges corresponding to invariant features from a single RGB image. Depth-discontinuous edges [10] correspond to edges with large variations in depth map. Our network filters out most of the intricate information from

*This work was supported by the National Natural Science Foundation of China (62076029), National Key R&D Program of China (2022YFE0201400), Guangdong Science and Technology Department (2022B1212010006, 2017A030313362), Guangdong Key Lab of AI and Multi-modal Data Processing (2020KSYS007), and internal funds of the United International College (R202012, R201802, R5201904, UICR0400025-21).

[1]Yiyang Hu is with the Faculty of Science and Technology, United International College, BNU-HKBU, Zhuhai, China. yiyanghu@uic.edu.cn

[2]Hui Ma is with Guangdong Provincial Key Laboratory of Robotics and Intelligent System, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, and also with the Faculty of Science and Technology, United International College, BNU-HKBU, Zhuhai, China. huima@uic.edu.cn

[3]Leiping Jie is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, China, and also with the Faculty of Science and Technology, United International College, BNU-HKBU, Zhuhai, China. leipingjie@uic.edu.cn

[4]Hui Zhang is the corresponding author and with the Faculty of Science and Technology, United International College, BNU-HKBU, Zhuhai, China. amyzhang@uic.edu.cn

the RGB image, leaving only valid edges that can be used for subsequent line-based calibration, such as [10] [11]. The calibration network encodes the projected depth map from the point cloud and matches it to the extracted features, then predicts translation and rotation matrix using two seperate branches. Our contributions can be summarized as follows:

- We propose a novel method to label depth-discontinuous straight edges in real-time, by matching features from the RGB image to corresponding depth map;
- We design a novel online self-supervised network DEdgeNet to refine the edges, using the depth-discontinuous straight edges as the pseudo ground truth;
- We present a novel calibration network that encodes only invariant features from reprojected depth maps and RGB images.

This paper is organized as follows. Section II reviews previous works in calibration of camera and LiDAR. Section III first introduces our novel methods to extract the depth-discontinous edges and then proposes a network to refine them, which will be input to our network for calibrating extrinsic parameter matrix between the camera and LiDAR. Section IV shows the dataset of KITTI-odometry, followed by the experimental results. Conclusions are given in Section V.

II. Related Works

The calibration between LiDAR and camera can be formulated as a 2D-3D registration problem, represented by a rigid body transformation of 6 DoF. Generally, there are three categories of calibration methods, namely target-based, targetless or learning-based methods.

A. Target-based methods

Most of the early calibration works use target-based methods. Zhang *et al.* [15], Unnikrishnan *et al.* [5] and Pandey *et al.* [16] all use the checkerboard as a calibration object to restore the plane in the coordinate systems of camera and LiDAR. [15][5][16] optimize the point mapping from LiDAR to the plane in the camera system. Recent works attempt to match their features by traditional algorithms and then calculate the extrinsic parameters. Dhall *et al.* [17] extracts the 3D corners of cardboard and ArUco, and matches the corners by the ICP (Iterative Closest Point) algorithm. Wang *et al.* [18] extracts 2D corners from image and 3D corners from point clouds, uses PnP (Perspective-n-Point) to optimize the angle difference. Liao *et al.* [19] extract 2D edges and corners from image, 3D edges and points in planes from point clouds, and optimizes the point-to-line distances. Zhou *et al.* [20] combines the edge and the point-to-line distances, simultaneously extract edges in 2D and 3D, and the plane in image and point cloud. It has better accuracy than single feature matching. Huang *et al.* [21] and Cui *et al.* [22] use PnP to match 2D corners and 3D corners. J Beltrán *et al.* [23] uses circles and ArUco to create a new type of cardboard, recovers 3D structure from the image, and uses ICP to match 3D points from both camera and LiDAR. Fang *et al.* [24] extract the CCTag of camera and corners of LiDAR in panoramic infrastructure, then uses PnP and ICP for feature matching. Target-based methods use calibration objects that can obtain accurate results of extrinsic parameter matrix. However, these target-based methods cannot be used everywhere, such as self-driving vehicles that require real-time calibration of camera and lidar offsets.

B. Targetless methods

The targetless methods can conveniently calibrate the camera and LiDAR anywhere. However, it is more difficult to extract reliable features, and the accuracy is generally lower than that of target-based methods. Most target-based methods require appropriate initialization. Common targetless methods are scene-based method. Pandey *et al.* [25] and Taylor *et al.* [26] extract the grayscale of image and the reflectivity of point cloud, obtaining optimized results using gradient ascent and normalization. Levinson *et al.* [27] extracts Canny edges and depth-discontinuous edges for matching. The latest study [28] attempts to extract straight lines from the image and point cloud, then change to a perspective 3-line problem to optimize matching results. Liu *et al.* [29] propose a new method combining the point-to-plane and point-to-edge distance to design a new algorithm and package it into a toolbox, and has a good performance. In addition, Ishikawa *et al.* [30] and Park *et al.* [31] propose motion-based methods to independently extract the motion of camera and LiDAR. Using the theory of hand-eye calibration [32], the extrinsic parameters matrix can be further optimized.

C. Learning-based methods

Schneider *et al.* [14] leverages convolutional neural networks (CNNs) to predict the extrinsic parameter matrix between the camera and LiDAR. Iyer *et al.* [13] designs a geometrically supervised neural network to estimate the extrinsic parameter matrix in real time. Lv *et al.* [12] proposes an online and end-to-end LiDAR and camera calibration network based on cost volume layers to express the correlation between LiDAR and camera. Wang *et al.* [33] extracts the semantic centroid from camera and LiDAR, and uses PnP to optimize the extrinsic parameter matrix.

Both target-based and targetless methods require known special scenario or appropriate initialization. The learning-based methods extract the underlying features in the natural environment and complete feature matching, with a large error.

In this work, we propose a novel online targetless self-calibrating network to calibrate the camera and LiDAR. The architecture is shown in Fig. 2. The network requires RGB image from a monocular camera and the corresponding depth map from projection of point cloud. There are two steps. DEdgeNet provides a fixed feature matrix, the calibration network encodes and matches the features of depth map, and then predicts translation and rotation. Previous learning-based methods aim to modify the network structure to obtain more accurate features. Our method first combines traditional algorithms and state-of-the-art networks to extract substantial features in 2D and 3D matching, and then the

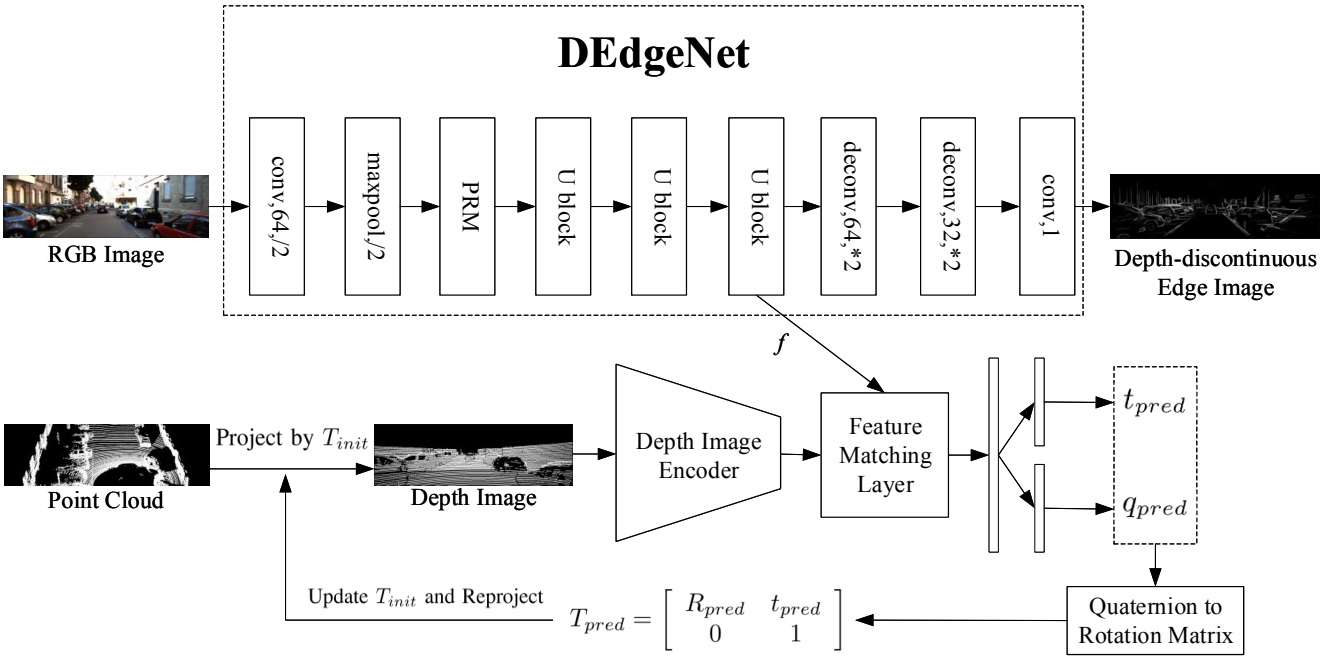

Fig. 2. An overview of our method. DEdgeNet takes an RGB image as input and then outputs a depth-discontinuous edges and an encoded feature matrix f. The calibration network takes f as a feature of RGB image and encodes the uncalibrated depth map projected by the point cloud via the initial extrinsic parameter matrix. The output is the predicted extrinsic parameter matrix T_{pred}, which is used to reproject the point cloud and update the depth map. The calibration network encodes the new depth map and matches it to the feature matrix f of the RGB image, simplifying the workflow of calibration. The depth map need to be updated, which takes the feature matrix f from the RGB image as a fixed input.

feature matching layer can easily and accurately search for correspondences.

III. Methodology

The goal of this work is to develop a more accurate and faster calibration method for camera-LiDAR in robotic system. Our method selects effective edge features of RGB image corresponding to depth maps. First, most of the depth-discontinuous edges in RGB image have good correspondences in depth maps. Furthermore, a large number of edge features can be extracted in both outdoor and indoor scenes. Traditional optimization methods can also solve the feature matching problem, which often require extensive initialization and is difficult to calibrate in real-time. Using neural networks to match these corresponding features can get poor or even no initialization.

A. Projection of Point Cloud to Depth Map

Point clouds from LiDAR can be projected into a depth map corresponding to each image frame from camera via the pinhole camera model. Each point ${}^{L}\mathbf{P}_i \in \mathbb{R}^3$ in LiDAR coordinate can be transformed into camera coordinate as

$$ {}^{C}\mathbf{P}_i = {}^{C}_{L}\mathbf{T} * {}^{L}\mathbf{P}_i, \tag{1} $$

where ${}^{C}_{L}\mathbf{T} = \left({}^{C}_{L}\mathbf{R}, {}^{C}_{L}\mathbf{t}\right) \in SE(3)$ is the extrinsic matrix from LiDAR to camera, ${}^{C}\mathbf{P}_i \in \mathbb{R}^3$ is the point in camera coordinate. ${}^{C}\mathbf{P}_i$ can be projected onto the image plane, by

$$ {}^{C}\mathbf{p}_i = \mathbf{K} * {}^{C}\mathbf{P}_i, \tag{2} $$

where $\mathbf{K}$ is the intrinsic matrix of camera, ${}^{C}\mathbf{p}_i \in \mathbb{R}^2$ is the projected point on the image plane. Each ${}^{C}\mathbf{p}_i$ is labeled in the image to preserve the depth value in depth map D.

B. Extraction of Depth-discontinuous Straight Edges

We propose an online algorithm to extract depth-discontinuous straight edges in Alg. 1. This algorithm requires a single image I and the corresponding depth map D, using the method of [34] to extract the line support area for each line. The line support area is all the pixels of a line and can be approximated by a rectangle that surrounds the entire line, as shown in Fig. 3 (a).

Algorithm 1 shows this procedure. *Rect* is used to represent all line support areas, and *rect* is used to represent the support area of a single line. Let *grad* denote the gradient for a line support area $rect \in Rect$. For each pixel in the line support area, a positive area and a negative area can be calculated, labeled as mask area 1 and mask area 2, respectively (see Fig. 3 (b)). Using these two areas as masks for the depth map, the average depth value of each masked area can be calculated, and comparing the two depth values can help us select straight edges with discontinuous depths and keep them in the heatmap H. The larger the width of the selected region, the more accurate the average depth value can be calculated, but it also increases the time consumption. τ represents the depth difference, which can filter out edges with small depth difference. If τ is small (width=5px and τ=0.1m in this paper), the heatmap will have fewer rows.

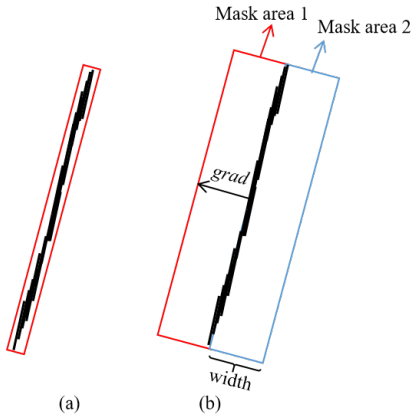

Fig. 3. Line support area and mask area. (a) The support area of a line is all the pixels in the line bounding box (red rectangle). (b) Two mask areas of width=5 pixels adjacent to the line support areas. These two areas mask the depth map to calculate the average depth value within these two areas.

Algorithm 1 Extraction of depth-discontinuous edges

Input: An image I and the corresponding depth map D
Output: A heat map H of depth-discontinuous edges

$Rect \leftarrow LSD(I)$
for all $rect \in Rect$ **do**
  $grad \leftarrow Gradient(rect)$
  for all $pixel\ P \in rect$ **do**
    $Mask1 \leftarrow GenerateMask(P, grad, width)$
    $Mask2 \leftarrow GenerateMask(P, -grad, width)$
  end for
  $DepthUp \leftarrow AverageDepth(D, Mask1)$
  $DepthDown \leftarrow AverageDepth(D, Mask2)$
  if $Abs(DepthUp - DepthDown) > \tau$ **then**
    Draw $rect \rightarrow H$
  end if
end for

C. Network for Extracting Optimal Depth-discontinuous Edges

This section introduces the online self-supervised network DEdgeNet, which requires a single image and outputs the corresponding depth-discontinuous edge heatmap and features. DEdgeNet takes the deatmap of depth-discontinuous straight edges in Sec. III-B as a peudo ground truth and outputs a more accurate heatmap of depth-discontinuous edges, which is not limited to straight edges. And it can fix the feature matrix of RGB image for the calibration network, without encoding the RGB image.

a) Loss function: Our self-supervised network extracts depth-discontinuous straight edges as pseudo ground truth in real-time. Pixels on the heatmap $\mathbf{H}(p)$ can be defined as:

$$\mathbf{H}(p) = \begin{cases} 1 & \text{The pixel } p \text{ is on a line,} \\ 0 & \text{The pixel } p \text{ is not on any line.} \end{cases} \tag{3}$$

We choose the ℓ_2 loss function to optimize the network:

$$L = \sum_{i,j} \left\| \hat{\mathbf{H}}(p_{ij}) - \mathbf{H}(p_{ij}) \right\|_2^2, \tag{4}$$

where $\hat{\mathbf{H}}(p)$ is the predicted heat map from DEdgeNet and $\mathbf{H}(p)$ is that extracted by the Alg. 1 in real-time. p_{ij} is the pixel in the i-row and j-column of the heat map.

b) Implementation detail: In Fig. 2, the input to DEdgeNet is an RGB image of size $256 \times 512 \times 3$, which is extracted into a feature matrix of $64 \times 128 \times 128$ through two convolutional layers, and then passed to Pyramid Residual Module (PRM) [35]. The result is then computed by three U blocks (see 4). U-block is inspired by U-Net [36], which is an autoencoder structure that automatically learns feature matrices. f is the intermediate layer in the last U block, which is the input to the camera calibration network. Finally, two deconvolutional layers and one convolutional layer restore the result to $256 \times 512 \times 1$ and output a heatmap of depth-discontinuous edges.

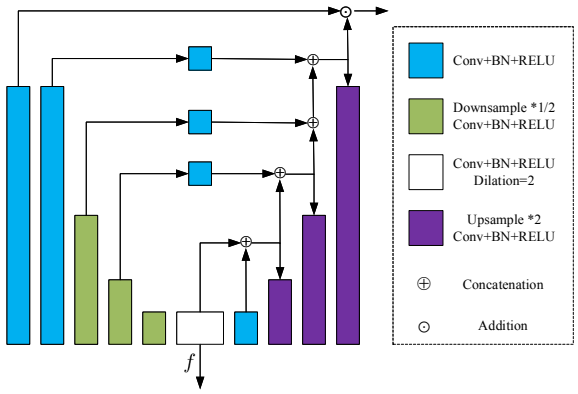

Fig. 4. The structure of U blocks in Fig. 2. U block can extract multi-scale features within stages and more effectively aggregate multi-level features between stages. During upsampling, concatenating the previous one can effectively prevent overfitting. The last output layer will be added to the input layer which is the key point in ResNet [37]. f is the middle layer of the U block, which is a feature matrix of $8 \times 16 \times 256$ for calibration network as a feature of the RGB image.

D. Network for Calibration

Other calibration networks extract features from both RGB image and depth map. Our method uses fixed features from DEdgeNet and only extracts features from depth map. The initial conditions

a) Feature extraction: Projection using the initial extrinsic matrix is poor due to changes of projected depth maps. Therefore, it is inevitable to reproject the point cloud and update the depth map using the predicted extrinsic parameter matrix. Our calibration network removes the step of feature extraction from RGB images and only extracts repeatedly updated features from depth maps. The features of the RGB images are replaced by f in DEdgeNet, which are features that preserve depth-discontinuous edges. The encoder for depth map feature extraction is ResNet-18. It outputs a feature matrix of $8 \times 16 \times 256$, which has the same dimension as the feature f of the RGB image.

b) Feature matching: For two feature matrices of the same-dimension, we construct the cost volume [38] to store the matching cost in order to associating depth-discontinuous

edges on the depth map. That is why we extract the depth difference edges from the RGB image. Depth-discontinuous edges are more sensitive to matching with the depth map than other parts in the RGB image. The matching cost as the correlation between the heatmap of depth-discontinuous edges and the depth map can be defined as:

$$cost\ volumn\left(p_1,p_2\right)=\frac{1}{N}\left(c\left(\mathbf{x}_{edge}\left(p_1\right)\right)\right)^T c\left(\mathbf{x}_{depth}\left(p_2\right)\right), \tag{5}$$

where $\mathbf{x}_{edge}$ and $\mathbf{x}_{depth}$ are feature matrices of the depth-discontinuous edge image and the projected depth map. $c(\mathbf{x})$ is a flatten operation on the feature matrix $\mathbf{x}$, N is the length of the flattened vector of $\mathbf{x}$, T is the transpose operator. This cost volume computes the correlation of each pixel with the surrounding d pixels. Since this feature matrix is only 8×16 in size and has 256 channels, d needs to be set smaller. The dimension of the 3D cost volume is $d^2\times H\times W$, where H and W are the dimensions of the feature matrix ($d=2, H=8, W=16$ in this paper).

c) Global regression: To predict results of the 6-DoF rigid body transformation, the global information has been aggregated through features of the cost volume. This global information is flattened by a full connected layer and split into two branches by two full connected layers. The output of this network is a 1×3 vector $\mathbf{t}_{pred}$ to represent the translation and a 1×4 vector $\mathbf{q}_{pred}$ to represent the quaternion rotation.

d) Loss function: In order to guide the network to converge better, our loss function consists of three modules. Given the feature matrix $\mathbf{f}$ of RGB image from DEdgeNet and a depth map $\mathbf{D}_{init}$, we write the loss function as follows:

$$L=\lambda_T L_T+\lambda_R L_R+\lambda_P L_P, \tag{6}$$

where λ_T, λ_R and λ_P denote the loss weights of translation loss, rotation loss and point cloud distance loss, respectively. The translation loss uses the ℓ_2 loss:

$$L_T=\|t_{gt}-t_{pred}\|_2^2. \tag{7}$$

Since the Euclidean distance cannot accurately describe the difference between two quaternions, the rotation loss is not simply using the ℓ_2 loss, but the quaternion distance [39]:

$$L_R=D_a(q_{gt},q_{pred}), \tag{8}$$

where D_a is a function that computes the angular distance of two quaternions. The point cloud distance loss describes the distance of each corresponding point in the point cloud:

$$L_p=\frac{1}{N}\sum_{i=1}^{N}\left\|\hat{P}_i-P_i\right\|_2^2, \tag{9}$$

where N is the number of points in this point cloud, P_i denotes the point in the initial point cloud, and $\hat{P}_i$ denotes the point in the point cloud reprojected by the predicted extrinsic matrix.

IV. EXPERIMENTS

To evaluate our proposed method, we test DEdgeNet and calibration network on the KITTI odometry dataset. This section shows the results of the depth-discontinuous straight edge extraction algorithm, DEdgeNet, and calibration network.

A. Dataset

This KITTI odometry dataset consists of 21 sequences from different scenarios. In each sequence, we only use data from the left color camera and LiDAR, with extrinsic parameters provided by [40].

a) DEdgeNet: DEdgeNet is a self-supervised network for extracting depth-discontinuous edge, Alg. 1 provides depth-discontinuous straight edges as the pseudo ground truth. The point cloud is projected to the depth map via the ground truth extrinsic matrix.

b) Calibration network: The calibration network predicts the rotation and translation errors between the camera and LiDAR. We add a random deviation ΔT within the set range.

B. Extraction of Depth-discontinuous Edges

The results of extracted depth-discontinuous edges are visualized in Fig. 5. The first row is the original RGB image. The second and the third row are computed by the Alg. 1 and DEdgeNet, respectively. The images in the second row consist of discrete straight edges and the joints are not smooth. The edges predicted by DEdgeNet are continuous and smooth, which contain more complete and richer features. Since DEdgeNet learns features from extracted straight edges, and processing starts from the original image that are not restricted by straight edges. In Alg. 1, we try to draw lines of diffffernt widths on the heatmap train DEdgeNet. During the experiments, we found that different line widths have a great impact on the edge extraction of the network. When we set the line width to 1, DEdgeNet cannot extract clear edges. Instead, the edges are thin and virtual. If the line width is set to 3, there are many white noise points on the heatmap. When the line width is set to 2, clear and accurate results can be obtained.

C. Network for Calibration

Our calibration network is trained in five error ranges $\{(1.5m,20^\circ),\ (1.0m,10^\circ),\ (0.5m,5^\circ),\ (0.2m,2^\circ),\ (0.1\text{m},\ 1^\circ)\}$. And we randomly initial extrinsic matrix in the maximum error range (translation error is $\pm1.5m$ in each axis, angle error is $\pm20^\circ$ in each direction). Results for multi-ranges are shown in Table I and comparisons with other learning-based calibration algorithms are shown in Table II. Our method achieves a mean translation error of 0.247cm (x, y, z: 0.258cm, 0.330cm, 0.153cm) and a mean angle error of 0.028°(roll, pitch, yaw: 0.052°,0.014°,0.018°).

Compared to LCCNet[12], our method trains a new feature extraction network DEdgeNet and keeps the features of RGB image untouched. LCCNet repeatedly encodes RGB images wastes computing power, and our method has obtained similar results with the fixed RGB image features.

Fig. 5. The first row is the original RGB images. The second row is the extracted depth-discontinuous straight edge heatmaps. The third row is the heatmap of depth-discontinuous edge predicted by DEdgeNet.

TABLE I

THE RESULTS OF THE MULTI-RANGE NETWORK ITERATION

Results of multi-range		Translation(cm)				Rotation(°)			
		Et	X	Y	Z	ER	Roll	Pitch	Yaw
After $20°/1.5m$ network	Mean	18.253	12.473	8.974	7.640	0.771	0.098	0.255	0.241
	Median	16.613	13.012	10.296	7.361	0.580	0.121	0.280	0.223
	Std.	8.491	8.313	3.818	5.924	0.970	0.070	0.087	0.100
After $10°/1.0m$ network	Mean	11.880	4.658	4.423	3.589	0.585	0.150	0.110	0.175
	Median	10.807	3.271	3.490	3.817	0.439	0.099	0.148	0.133
	Std.	5.137	2.735	1.993	0.874	0.745	0.137	0.095	0.145
After $5°/0.5m$ network	Mean	6.006	0.882	0.755	1.308	0.400	0.104	0.092	0.064
	Median	5.538	0.387	0.714	1.412	0.299	0.082	0.082	0.031
	Std.	2.865	1.000	0.483	0.785	0.649	**0.046**	0.052	0.062
After $2°/0.2m$ network	Mean	1.889	0.945	0.397	2.074	0.284	0.111	0.049	0.026
	Median	1.500	0.760	0.373	2.103	0.213	0.075	0.040	0.033
	Std.	1.624	0.929	0.301	1.074	0.632	0.094	0.053	0.021
After $1°/0.1m$ network	Mean	**1.109**	**0.247**	**0.330**	**0.153**	**0.159**	**0.052**	**0.014**	**0.018**
	Median	**0.802**	**0.218**	**0.307**	**0.137**	**0.109**	**0.070**	**0.010**	**0.013**
	Std.	**1.137**	**0.242**	**0.147**	**0.125**	**0.624**	0.086	**0.013**	**0.008**

The translation error and angle error are randomly initialized in $(-1.5m, 1.5m)$ on each axis and $(-20°, 20°)$ in each direction, respectively. This table shows that our error becomes smaller after each network and it achieves a mean translation error of 0.247cm and a mean angle error of 0.028°.

TABLE II

THE COMPARISON RESULTS WITH OTHER LEARNING-BASED CALIBRATION ALGORITHMS

Method	Mis-calibrated range	Translation(cm)				Rotation(°)			
		mean	X	Y	Z	mean	Roll	Pitch	Yaw
Regnet[14]	$[-1.5m, 1.5m]/[-20°, 20°]$	6	7	7	4	0.28	0.24	0.25	0.36
Calibnet[13]	$[-0.2m, 0.2m]/[-10°, 10°]$	4.34	4.2	1.6	7.22	0.41	0.18	0.9	0.15
Lccnet[12]	$[-1.5m, 1.5m]/[-20°, 20°]$	0.297	0.262	**0.271**	0.357	**0.017**	**0.020**	**0.012**	0.019
Ours	$[-1.5m, 1.5m]/[-20°, 20°]$	**0.247**	**0.258**	0.330	**0.153**	0.028	0.052	0.014	**0.018**

V. CONCLUSIONS

This paper presents a two-stage approach to solve the calibration problem between the camera and LiDAR. We consider the correspondence between 2D and 3D, using a traditional edge feature extraction algorithm combined with neural networks, which only encodes the reprojected depth map and uses the feature matrix of RGB image from DEdgeNet. Our method simplifies the calibration process but still requires iterative optimization. When projected by a random extrinsic matrix, the initial depth map has fewer projected points and result depends on the environment images are captured. The experimental results show the superior performance of our method.

REFERENCES

[1] C. Abah, A. L. Orekhov, G. L. H. Johnston, and N. Simaan, "A multi-modal sensor array for human–robot interaction and confined spaces exploration using continuum robots," *IEEE Sensors Journal*, vol. 22, no. 4, pp. 3585–3594, 2022.

[2] Z. Huang, C. Lv, Y. Xing, and J. Wu, "Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11 781–11 790, 2021.

[3] F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "A multi-modal mapping unit for autonomous exploration and mapping of underground tunnels," in *2018 IEEE Aerospace Conference*, 2018, pp. 1–7.

[4] C. Park, S. Kim, P. Moghadam, J. Guo, S. Sridharan, and C. Fookes, "Robust photogeometric localization over time for map-centric loop closure," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1768–1775, 2019.

[5] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," *Carnegie Mellon University*, 2005.

[6] M. A. Munoznbanon, F. A. Candelas, and F. Torres, "Targetless camera-lidar calibration in unstructured environments," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2020.

[7] J. Jeong, Y. Cho, and A. Kim, "The road is enough! extrinsic calibration of non-overlapping stereo camera and lidar using road information."

[8] Z. Pusztai and L. Hajder, "Accurate calibration of lidar-camera systems using ordinary boxes," in *IEEE International Conference on Computer Vision Workshop*, 2018.

[9] J. Kummerle and T. Kuhner, "Unified intrinsic and extrinsic camera and lidar calibration under uncertainties," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[10] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments," 2021.

[11] P. Moghadam, M. Bosse, and R. Zlot, "Line-based extrinsic calibration of range and image sensors," in *IEEE*, 2013.

[12] X. Lv, B. Wang, Z. Dou, D. Ye, and S. Wang, "Lccnet: Lidar and camera self-calibration using cost volume network," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021.

[13] G. Iyer, R. K. Ram., J. K. Murthy, and K. M. Krishna, "Calibnet: Geometrically supervised extrinsic calibration using 3d spatial transformer networks," 2018.

[14] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "Regnet: Multi-modal sensor registration using deep neural networks," *arXiv e-prints*, 2017.

[15] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 2004.

[16] G. Pandey, J. Mcbride, S. Savarese, and R. Eustice, "Extrinsic calibration of a 3d laser scanner and an omnidirectional camera," *IFAC Proceedings Volumes*, vol. 43, no. 16, pp. 336–341, 2010.

[17] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna, "Lidar-camera calibration using 3d-3d point correspondences," 2017.

[18] W. Wang, S. Ken, and K. Nobuo, "Reflectance intensity assisted automatic and accurate extrinsic calibration of 3d lidar and panoramic camera using a printed chessboard," *Remote Sensing*, vol. 9, no. 8, p. 851, 2017.

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

[20] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[21] J. K. Huang and J. W. Grizzle, "Improvements to target-based 3d lidar to camera calibration," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2020.

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

[23] J. Beltrán, C. Guindel, and F. García, "Automatic extrinsic calibration method for lidar and camera sensor setups," 2021.

[24] 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," 2021.

[25] G. Pandey, J. R. Mcbride, S. Savarese, and R. M. Eustice, "Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information," *AAAI Press*, 2012.

[26] Z. Taylor and J. Nieto, "Automatic calibration of lidar with camera images using normalized mutual information."

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

[28] T. Ma, Z. Liu, G. Yan, and Y. Li, "Crlf: Automatic calibration and refinement based on line feature for lidar and camera in road scenes," 2021.

[29] X. Liu, C. Yuan, and F. Zhang, "Fast and accurate extrinsic calibration for multiple lidars and cameras," 2021.

[30] R. Ishikawa, T. Oishi, and K. Ikeuchi, "Lidar and camera calibration using motions estimated by sensor fusion odometry," *IEEE*, 2019.

[31] C. Park, P. Moghadam, S. Kim, S. Sridharan, and C. Fookes, "Spatiotemporal camera-lidar calibration: A targetless and structureless approach," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1556–1563, 2020.

[32] Q. Ma, Z. Goh, S. Ruan, and G. S. Chirikjian, "Probabilistic approaches to the (axb = ycz) calibration problem in multi-robot systems," *AUTONOMOUS ROBOTS*, no. 7, 2018.

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

[34] R. G. V. Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, "Lsd: A line segment detector," *Image Processing On Line*, vol. 2, no. 4, pp. 35–55, 2012.

[35] K. Huang, Y. Wang, Z. Zhou, T. Ding, and S. Gao, "Learning to parse wireframes in images of man-made environments," *IEEE*, 2018.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," p. 770–778, 2016.

[38] D. Sun, X. Yang, M. Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," 2017.

[39] A. Kendall, M. Grimes, and R. Cipolla, "Convolutional networks for real-time 6-dof camera relocalization." 2015.

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