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

Zero-training LiDAR-Camera Extrinsic Calibration Method Using Segment Anything Model

Zhaotong Luo[1,2], Guohang Yan[1,†], Xinyu Cai[1] and Botian Shi[1]

***Abstract*— Extrinsic calibration for LiDAR and camera is an essential prerequisite for sensor fusion. Recently, automatic and target-less extrinsic calibration has become the mainstream of academic research. However, geometric feature-based methods still have requirements on the scene. Deep learning methods, while achieving high accuracy and good adaptability, rely on large annotated dataset and need additional training. We propose a novel LiDAR-camera calibration method by using the Segment Anything Model(SAM) without additional training. With the automatically generated masks, we optimize the extrinsic parameters by maximizing the consistency score of the point attributes that fall on each mask. The point cloud attributes include intensity, normal vector and segmentation class. Experiments on different real-world dataset demonstrate the accuracy and robustness of our proposed method. The code is available at https://github.com/OpenCalib/CalibAnything.**

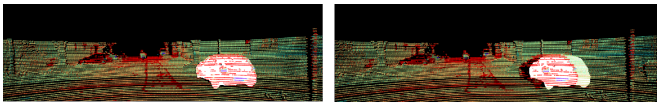
I. INTRODUCTION

Camera and LiDAR are the two main types of sensors used in self-driving vehicles. The complementary nature of the two sensors makes them a favored combination in autonomous driving tasks such as depth completion [1], object detection [2] and object tracking [3]. In order to fuse data from these two sensors, calibration is needed for time synchronization and spatial alignment. In this paper, we focus on the extrinsic calibration, which is to obtain the rigid transformation between the camera coordinate system and LiDAR coordinate system. The accuracy of extrinsic parameters fundamentally limits the performance of subsequent tasks. Thus, much effort has been made to handle this problem from different perspectives.

Early methods used artificial targets with special patterns that can be easily detected [4]–[8]. They can achieve high precision, but require manual work. Since the extrinsic parameters will change during daily use, a target-less and automatic method is needed for re-calibration. For this purpose, some methods exploit the geometric features in natural scenes such as lines [9]–[11] and vanishing points(VPs) [12], [13], which often exist in structured scenarios. In order to further reduce the requirements for the scene, learning-based approaches take the stage with the assistance of large-scale datasets. It can adapt to general scenes and achieve high accuracy. However, simple convolutional neural network [14] has weak generalization ability and poor interpretability. Although geometric constraints are added in some ways [15]–[17], they still need to be trained on a large well-labeled dataset and faces accuracy drop under dataset variations.

These problems can be alleviated with the advent of foundation model. The Segment Anything Model(SAM) [18] is a foundation model for image segmentation, demonstrating impressive zero-shot capabilities. Considering the usage of segmentation in calibration [19], [20], we propose a novel LiDAR-camera calibration method based on SAM. It doesn't require additional training and is more general for different scenarios.

We first use SAM to perform semantic segmentation on the entire image and get a set of masks automatically. Instead of finding definite correspondence between the point cloud and the image pixel, we use the consistency of the point cloud on each mask to measure multi-modal registration. The point cloud attributes include intensity, normal vector and segmentation class. As shown in Fig.1, with the correct extrinsic, the intensity of points projected on the car mask have higher consistency. For normal vector, the point cloud projected on the plane mask(such as the ground mask) has similar normal direction. In addition, we get the segmentation class of the points by plane fitting and euclidean clustering. The points of a separate object such as a vehicle will be grouped together, so the cluster ID also have consistency inside a mask. We design an objective function by these three properties, then we use it to optimize the extrinsic by search method.



(a) (b)

Fig. 1 The point cloud projected on a car mask with the correct extrinsic (a) and wrong extrinsic (b). The color represents the intensity value of the points.

In comparison to geometric feature-based methods [21], our method has better adaptability for different scenarios. Compared to learning-based method [17], we don't require extra training on a well-labeled dataset. Compared with other segmentation-based methods [19], [22], we avoid finding determinate correspondences between the image pixels and the point cloud, which only can be captured in limited types of object. Experiments on different dataset have demonstrated the generality and accuracy of our method.

The contributions of this work is listed as follows:

† Corresponding author.
[1] Zhaotong Luo, Guohang Yan, Xinyu Cai and Botian Shi are with Autonomous Driving Group, Shanghai Artificial Intelligence Laboratory, Shanghai, China. {luozhaotong, yanguohang, caixinyu, shibotian}@pjlab.org.cn
[2] Zhaotong Luo is with Tsinghua Shenzhen International Graduate School, Tsinghua University, Beijing, China, and also an intern in Autonomous Driving Group, Shanghai Artificial Intelligence Laboratory, Shanghai, China. luozt21@tsinghua.org.cn

1) We propose an automatic target-less LiDAR-camera extrinsic calibration method by using SAM. It is more general to scenarios and does not require training.
2) The objective function is established by the consistency of the three attributes of the points that fall on one mask, making the methods more robust and accurate.
3) The method was verified on real-world datasets and the code is released to benefit the community.

II. RELATED WORK

In general, the calibration methods can be divided into target-based and target-less types. Conventional target-less methods use geometric features or maximizing mutual information. By taking advantage of large-scale datasets, learning-based methods are developed to provide a rather accurate estimation with less scene requirements, roughly divided to regression and segmentation types.

A. Target-Based Methods

This type of method requires artificial targets. The targets are specially-designed in color, shape and reflectivity, make it easily detectable. The mainstream of targets are rectangular boards with specific pattern on it, such as checkerboard [4], circular grid [23] and Apriltag [6]. Because the horizontal edge of rectangle may not intersect with LiDAR scans, objects of other shape such as sphere [24], polygon board [25] are also proposed. By building strong correspondences between points-points or points-plane, this type of method generally achieves high precision. However, it needs human intervention at a low or high level.

B. Target-free Methods

Instead of custom-made target, some methods seek for geometric features in natural scenes. The most used features are lines or edges. There are generally two steps. Firstly the lines in the image are detected by an edge detector [10] or segmentation network [21]. The lines in the point cloud are mainly obtained by range discontinuity [10], [11] and intensity difference [21]. Then the many-to-many correspondences between lines are aligned according to its location [10], intensity and influential range [9]. Besides direct line features, [12], [13] use vanishing points to estimate the rotation matrix. It requires at least two VPs in the scene.

To reduce the restrictions on the scene, some methods utilize mutual information to measure the multi-modal registration, including gradient [26], intensity of the point cloud and the gray value of image [27], [28]. While above methods requires zones of mutual visibility, motion-based approach estimates the ego-motion of each sensor separately and solve the extrinsic by hand-eye model [29] or minimizing the projection error [30]. The accuracy of calibration is limited by the result of odometry.

C. Learning-based methods

The simple paradigm of learning-based methods is using an end-to-end network to estimate the extrinsic parameter with the input of RGB image and depth image. RegNet [14] first introduces the Convolutional Neural Networks(CNNs) to regress the 6 DoF parameter. In order to improve the transfer ability of the model, geometry constraints are added in the loss function. CalibNet [15] trains its network by maximizing the geometric and photometric consistency of the images and point clouds. RGGNet [16] considers the Riemannian geometry and utilizes a deep generative model. LCCNet [17] exploits the cost volume layer for feature matching and predicts the decalibrated deviation from initial calibration to the ground truth.

Despite of end-to-end networks, learning-based segmentation is used as part of the pipeline. [22] performs semantic segmentation respectively on pictures and point clouds, and then matches the centroids of a class of objects in 2D and 3D points. Due to the sparsity of the point cloud, [19] combined multiple frames of LiDAR data together, requiring for a high-precision positioning device. Because of the ambiguity of point cloud segmentation, some methods only perform segmentation on the image. [31] calibrates the extrinsic by maximizing the number of point cloud fell on the segmented foreground area in image. [20] uses instance segmentation to obtain object edges and define the loss function by the depth discontinuity. These methods can only predict objects of specific classes and establish limited correspondences. Besides, the segmentation network has transferbility problem over different dataset. Our method uses SAM to segment various types of objects in the image. For point cloud, we exploit the reflectivity and normal vectors in addition to segmentation.

III. METHODOLOGY

A. The Overview of The Method

The whole process can be divided into three parts. For an input image, we use SAM to generate the masks automatically. For point cloud, we implement normal estimation, clustering and intensity normalization to generate corresponding attribute of each point. Then the optimization target is to make the points that fall on one mask has close attribute value. We design an objective function to evaluate the consistency, which is used to optimize the extrinsic by search method. Fig.2 shows the pipeline of our proposed method.

B. Pre-processing

1) Image Segmentation: SAM is first applied on the entire image to get a number of masks of different instances. Since we use the consistency of the point cloud, we want the segmentation to be more fine-grained. So the parameters for model inference is adjusted to obtain more masks with less overlapping areas. The masks are annotated as $\mathbb{M} = \{M_i \mid i = 1, ..., m\}$. Each mask is a binary matrix of the same size as the image. $M_i(u, v) = \{0, 1\}$ denotes whether the pixel (u, v) belongs to instance i.

2) Point cloud Pre-processing: We first remove the points that are definitely invisible from the viewpoint of the camera, based on the initial extrinsic. For a dense point cloud, we downsample it by a voxelized grid approach to reduce

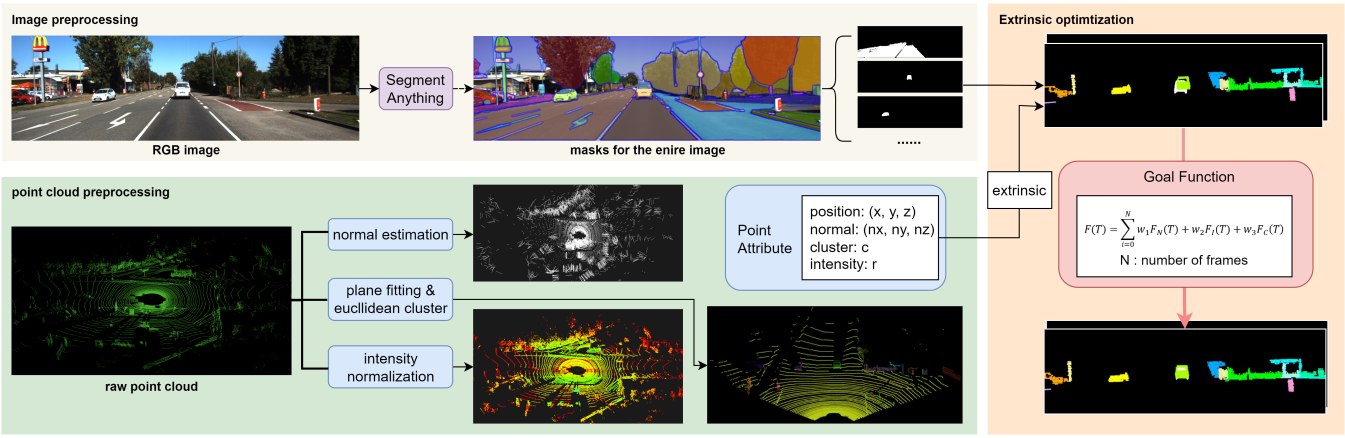


Fig. 2 Approach Overview. For image preprocessing, Segment Anything Model is used to generate masks of the entire image. For point cloud, we implement normal estimation, clustering and intensity normalization to generate corresponding attribute of each point. In the optimization stage, we design an objective function which measure the consistency of the three attributes.

subsequent calculations. Then we did normal estimation, intensity normalization and clustering.

There are a number methods [32], [33] that can be directly used for normal estimation. The normal direction of a point on the surface is approximated as the normal of a plane tangent to the surface. It is calculated by the eigenvectors and eigenvalues of a covariance matrix created from a number of nearest neighbors of the query point. The intensity of the point cloud is normalized to [0,1].

Besides, the segmentation class is obtained by plane fitting and clustering. We first apply plane fitting by RANSAC algorithm to extract large planes in the scene, such as the ground and walls. Then we implement euclidean clustering [32] to the remaining point cloud and get clusters of separate objects like vehicles and trees. Only the clusters that have enough points are remained. We assign a number c to the point, indicating which cluster it belongs to.

The final attributes of the point cloud include the position, normal vector, intensity and segmentation class:

$$P = \{x, y, z, \vec{n} = (n_x, n_y, n_z), r, c\} \tag{1}$$

C. Extrinsic Optimization

1) Objective function: We assume the intrinsic parameters of camera are already known and stay constant. By the extrinsic T, the point $P(x, y, z)$ in the LiDAR frame can be projected on the image as $p(u, v)$:

$$\lambda \, [u \ v \ 1]^T = KT \, [x \ y \ z \ 1]^T = KTP \tag{2}$$

For each mask M, we get a set of points falls on it, denoted as S:

$$S = \{P \mid M(p) = 1, p = KTP\} \tag{3}$$

We denote $N = \|S\|$ as the number of points in S. To measure the consistency of points in S, three functions f^N, f^I, f^C are designed respectively for normal vector, intensity and segmentation.

For normal vector, the function f^N is the average value of the pairwise dot product of all vectors in S. We first construct the matrix A (3xn) and B (nxn):

$$A = [\vec{n}_1, \vec{n}_2, ...\vec{n}_N], \ B = A^T A \tag{4}$$

$\vec{n}_i$ is the normal vector of points in S. f^N is defined as:

$$f^N = \frac{1}{N^2} \sum_{i=0}^{N} \sum_{j=0}^{N} |B_{i,j}| \tag{5}$$

The intensity function f^I is calculated by the variance of all intensity values in S:

$$f^I = 1 - \frac{1}{N} \sum_{i=0}^{N} (r_i - \frac{1}{N} \sum_{i=0}^{N} r_i)^2, \ r_i \leftarrow P_i \in S \tag{6}$$

For segmentation class, we first count the points of each class in the set S as $[n'_0, n'_1, ..., n'_C]$, where n'_i is the number of points in cluster i. C is the number of clusters. Then this array is sorted from the largest to the smallest as $[n_0, n_1, ..., n_C]$. The consistency score is calculated as:

$$f^C = \frac{1}{N^C} \sum_i k^i n_i, (i = 0, 1, ..., C) \tag{7}$$

$$N^C = \sum_i n_i, (i = 0, 1, ..., C) \tag{8}$$

If the category is more concentrated, the consistency score will be higher. k is set to 0.5 by experience.

The above function f^N, f^I, f^C measures the consistency score of one mask. The overall score of a image is calculated by the weighted score of all masks. The functions of all masks are denoted as F^N, F^I, F^C:

$$F^X = \sum_{i=1}^{m} w_i^X f_i^X f^A(N_i) \tag{9}$$

Here $X \in \{N, I, C\}$ for a more concise representation. $N_i = \|S_i\|$. The weight w_i^X of each mask is the number of points fall on it.

$$w_i^X = \frac{N_i}{\sum_{i=1}^{m} N_i} \tag{10}$$

$f^A(\cdot)$ is a compensate function for the sparsity of the point cloud. Far away areas or occluded areas are projected to few point clouds. Because fewer points tend to have higher consistency even under the wrong extrinsic, it is less reliable. So we punish the mask with little point cloud. The function is a monotone increasing function of the number of points:

$$f^A(N_i) = 1 - k_1 * N_i^{-k_2} \tag{11}$$

The final objective function is the linear combination of the three consistency function:

$$F(T) = w_1 F^N(T) + w_2 F^I(T) + (1 - w_1 - w_2) F^C(T) \tag{12}$$

Basically, it can be regarded as a function of the extrinsic parameters. The hyper-parameter k_1, k_2, w_1, w_2 are decided in the experiment.

2) Optimization: As above, the objective function can evaluate the alignment between the image and the point cloud with the variable T. Since we don't use definite correspondences, the function does not necessarily reach its maximum value at the correct extrinsic. In order to avoid mismatching, multiple frames can be used to build stronger constraints:

$$T = \arg\min_{T} \sum_{i}^{num} -F_i(T) \tag{13}$$

Because the function is computed by the points that fall on the mask, the function is discontinuous. Besides, we can find in the experiment that the objective function has many local extrema. It is not suitable for traditional gradient based optimization algorithm. Here we use the Nelder-Mead algorithm [34], which is widely used for derivative-free optimization. It defines the initial search space as a simplex, and then shifts and shrinks towards the function minimum. We assume a rough extrinsic is known and the search space is defined by a certain area around it. Then we uniformly sample some points in this area as the starting points of the search algorithm. After getting a number of optimized extrinsic parameters, the extrinsic with the lowest function values are regarded as the final estimate.

IV. EXPERIMENTS

A. Implementation

We conduct experiment on the two large-scale open source datasets: KITTI odometry benchmark [35] and Nuscenes [36]. For Kitti odometry dataset, we use the point clouds from a Velodyne HDL-64 LiDAR and images from the rectified left RGB camera in 5 selected sequences (00, 04, 05, 07, 13). For Nuscenes dataset, we use the data from the Velodyne HDL32E top LiDAR and front camera in 10 selected sequences (0001-0010). The calibration files provided by the dataset are regarded as the ground truth. The image segmentation process runs on a NVIDIA GeForce GTX 1660 Ti GPU and other processes are performed on an Intel Core i7-11700 CPU.

B. Preliminary Experiments

1) The effect of each function: We first verify the independent role of normal vector, intensity and clustering in the objective function. The function $F^N(T), F^I(T), F^C(T)$ in Eq. 12. are plotted by only varying one degree in a certain range and keep the other degrees as the ground truth. In the experiment, we let $\Delta roll$ vary in $[-5°, 5°]$ and let Δx vary in $[-0.5m, 0.5m]$. To verify the effect of multiple frames, 1 frame, 5 frames, and 10 frames in Kitti Sequence 00 are considered in the functions.

As shown in Fig. 3, the functions get the maximum value when the extrinsic is close to the ground-truth ($\Delta T \to 0$). In comparison, the intensity function has larger fluctuations and deviations to the ground-truth value, so its effect of measuring the degree of matching is worse. This may be because the intensity is affected by the distance, incident angle and noise. However, by using more frames, the curves of intensity function also become smooth and have higher convexity. It's reasonable to inference that using the three functions together and using more frames can make the method more robust. In the following experiment, 5 frames and 10 frames are used to test the final accuracy.

2) Ablation study for hyper-parameters: We did ablation study to determine the compensate parameter k_1, k_2 in Eq. 11 and the weight w_1, w_2 in Eq. 12. For other hyper-parameters, they are not the key to the method and are only set as empirical values.

The three function F^N, F^I, F^C share the same compensate parameters. Since the rotation has a larger impact on the projection, we use the rotational error as a scalar value to compare the effects of different k_1, k_2. For a pair of k_1, k_2, the angular error is defined as:

$$\frac{1}{9} \sum_{\alpha}^{\phi,\theta,\psi} \sum_{X}^{N,I,C} \| \arg\max_{\alpha} F^X(T_{gt} \Delta T(\alpha)) \| \tag{14}$$

Here ϕ, θ, ψ represent the roll, pitch and yaw angle. 20 frames in different sequences of Kitti are used to calculate this error when k_1, k_2 varies at a reasonable range. As shown in 4, the error achieves the smallest when $k_1 = 2, k_2 = 0.3$.

For each w_1, w_2 combination, the objective function F is determined. The average angular error is calculated as:

$$\frac{1}{9} \sum_{\alpha}^{\phi,\theta,\psi} \| \arg\max_{\alpha} F(T_{gt} \Delta T(\alpha)) \| \tag{15}$$

We also test 20 frames. The mean error under different w_1, w_2 settings is shown in Fig. 4. According to it, we set $w_1 = 0.35, w_2 = 0.2$. The settings are used in the following experiment for the Kitti and Nuscenes dataset.

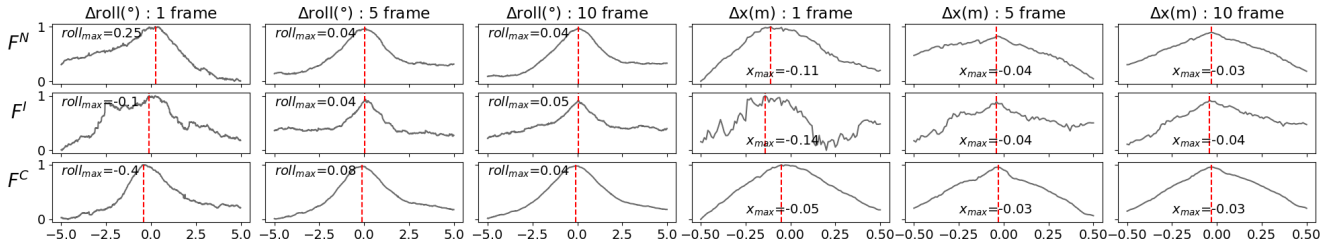


Fig. 3 The functions $F^{N,I,C}(T_{gt}\Delta T)$ are plotted by changing one degree of freedom in ΔT and keeping other dimensions zero. We let roll angle vary in $[-5°, 5°]$ (left 3 columns) and x vary in $[-0.5m, 0.5m]$ (right 3 columns). The sampling interval is 0.01° and 0.01m. The red line annotates the value of the variable when the function achieves the max value.

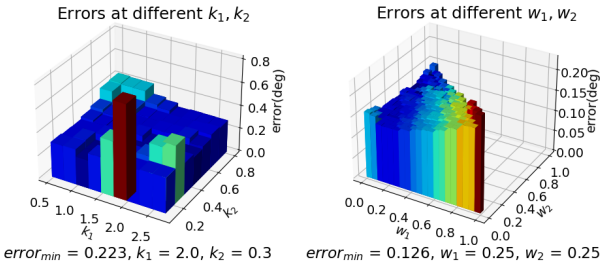


Fig. 4 The average angle error when using different hyper-parameters. k_1 is sampled in [0.5, 3] at the interval of 0.25. k_2 is sampled in [0.1, 0.8] at the interval of 0.1. w_1, w_2 are sampled in [0, 1] at the interval of 0.05.

3) The properties of the final objective function: The properties of the objective function is essential for the choice of optimization method. To better illustrate it, we change two degrees in the extrinsic parameters and plot the objective function. As shown in Fig. 5, it can be seen that the function has many local extrema. Therefore, when optimizing F, the start search point should be sampled in different locations in the search area.

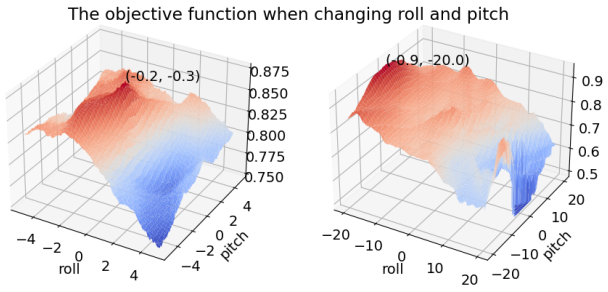


Fig. 5 The objective function $F(T * \Delta T)$ was plotted when roll and pitch vary in [-5, 5]° (left) and in [-20, 20]° (right). The maximum point is marked.

When the angles changes at $[-5, 5]°$, the maximum value of the function appears near the ground-truth. However, when the range expands to $[-20, 20]°$, the maximum value of the function appears far from the real value. This is because our method does not use exact matching relationship. When the deviation of the extrinsic parameters is too large, only a small number of points can be projected onto the image, resulting in unreliable consistency scores.

Therefore, in order to determine the feasible search range, we tested the average angular errors in Eq. 15 at different search scope, as shown in Fig. 6. For rotation, the error becomes larger when the search range is close to 15 degrees. For translation, the error has a step-like rise near 0.7m. In the following experiment, we test the accuracy when the initial error ranges within 5 degrees and 0.5m. The results show that the method is effective in this search space. We argue that an initial guess with such accuracy can be easily obtained from the CAD design or the motion-based hand-eye calibration.

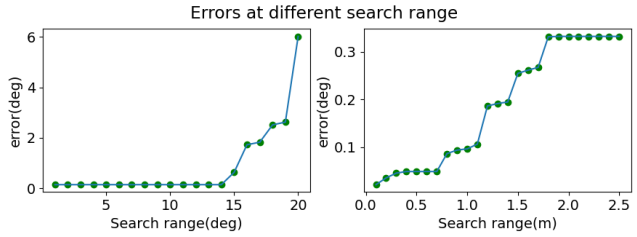


Fig. 6 The mean angular errors at different search range. It can be seen that the error has a step-wise increase when the search range is larger than a certain value.

C. Performance Comparison

We compare our method with a mutual-information(MI) based method [37] and a semantic-based method [19]. Additionally, we implement a motion-based method by pose estimation [38], [39] and hand eye calibration [40]. We also tried to compare with [41], but it requires depth-continuous edges, which rarely appear in outdoor natural scenes. We found it not applicable in our data. The MI-based method provides an automatic and a manual way for coarse calibration. We first use the automatic way. If it fails, we turn to the manual way. The semantic-based method and our method require an initial guess. Here we give the same range of initial error. The rotation error is randomly sampled in $[-5°, 5°]$ and the translation is sampled in $[-0.5m, 0.5m]$.

The MI-based method and semantic-based method require a dense point cloud for better registration. In order to meet this requirement, we concatenate a number of frames before and after the current frame into one point cloud by the LiDAR pose. The number of frames used in aggregation is represented by N_f. A aggregated point cloud and an image create a matching pair. The number of pairs used

Table. I. The MAE of different methods on Kitti and Nuscenes

	Methods	Np x Nf	Type	Rotation(deg)			Translation(m)		
				Roll	Pitch	Yaw	X	Y	Z
KITTI	Motion	50x1	Mean	1.304	1.232	1.384	0.321	0.278	0.356
			Std	0.292	0.347	0.376	0.149	0.102	0.160
	[37]	4x50	Mean	0.359	0.225	0.353	0.069	0.122	0.157
			Std	0.142	0.122	0.215	0.056	0.149	0.097
	[19]	1x50	Mean	0.393	0.341	0.370	0.202	0.203	0.144
			Std	0.212	0.170	0.169	0.076	0.085	0.094
	Ours	5x1	Mean	0.332	0.253	0.419	0.121	0.097	0.103
			Std	0.135	0.174	0.166	0.096	0.093	0.080
		10x1	Mean	**0.192**	**0.173**	**0.156**	**0.056**	**0.050**	**0.063**
			Std	0.158	0.093	0.134	0.040	0.044	0.062
NUSCENES	Motion	50x1	Mean	1.335	1.590	1.380	0.322	0.253	0.365
			Std	0.362	0.293	0.292	0.183	0.179	0.185
	[37]	4x100	Mean	0.456	0.405	0.419	0.165	0.156	0.143
			Std	0.243	0.295	0.185	0.121	0.138	0.109
	[19]	1x100	Mean	0.537	0.490	0.438	0.256	0.220	0.277
			Std	0.293	0.293	0.242	0.177	0.187	0.195
	Ours	5x5	Mean	0.389	0.424	0.444	0.120	0.151	0.143
			Std	0.307	0.390	0.210	0.125	0.102	0.119
		10x5	Mean	**0.202**	**0.340**	**0.198**	**0.109**	**0.099**	**0.130**
			Std	0.128	0.245	0.115	0.074	0.055	0.122

Np is the number of pairs used in one time calibration. Nf is the number of frames accumulated to one point cloud.

for calibration is denoted as N_p. N_p pairs make a group. Although different number of LiDAR frames are used in each method according to its requirement, the total number used in our method is lesser than others. For each method, 15 groups of data of different scenarios in Kitti and 15 groups of data in Nuscenes are tested. For the methods that need initial guess, each group was tested 3 times with random initial error. The results are analyzed according to the Mean Absolute Error(Mean) and standard deviation(Std) of each degree of freedom in the extrinsic parameters. The results are shown in Tab.I.

It can be seen that our proposed method achieves a better accuracy and stability than other method. The MI-based method generates LiDAR intensity images and the semantic-based method render the segmented point cloud into a image. Even if multiple frames of point cloud is accumulated, there still exist sparse area, which will introduce errors. Instead of computing pixel-level losses, our proposed method compute the consistency of a set of point cloud, making it more adaptable to point cloud sparsity. Besides, compared with the semantic based method, our method can use more types of object. By using 10 pairs of image and point cloud, the MAE of the proposed method is around $0.1°$ and $0.05m$ for Kitti and $0.2°$ and $0.1m$ for Nuscenes. All methods perform worse on Nuscenes dataset than the Kitti dataset, which is caused by the lesser channels of LiDAR lasers. The motion-based method has a rather large error. However, the test proves that it is enough for providing an initial guess for our method.

As shown in Fig. 7, the clustered point cloud is projected onto the segmented image with the extrinsic before and after calibration. We can see that the plants, billboards, walls and other independent objects can all establish correspondences. We also plot the calibration error distribution of our method in Fig. 8. It can be seen that most of the rotation errors are within $0.5°$ and most of the translation errors are less than 0.2 m. It's obvious that increasing the number of pairs reduces calibration errors.

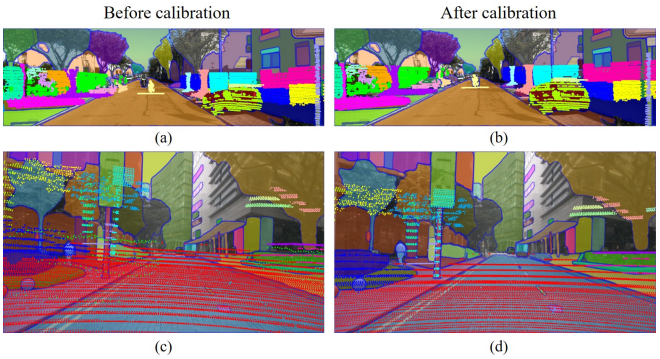


Fig. 7 The point cloud is projected to the segmented image by the extrinsic before calibration and after calibration in Kitti dataset(above) and Nuscenes dataset(below).

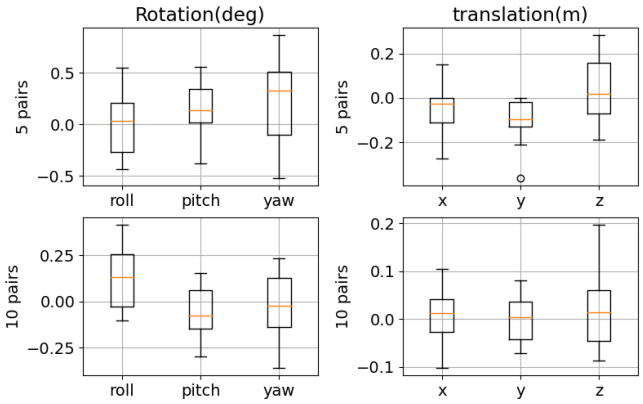


Fig. 8 The error distribution of our method in Kitti dataset by using 5 pairs and using 10 pairs.

V. CONCLUSIONS

In conclusion, we propose a novel LiDAR-camera calibration method using Segment Anything and point cloud consistency. Our approach can adapt to infrastructure scenarios without requiring additional training on a well-labeled dataset. Although an initial guess is needed, we can use the CAD design and continuously optimize it for online calibration. Experiments on real-world dataset demonstrate our method is accurate and robust to be applied to the online calibration during the operation of autonomous vehicles.

VI. ACKNOWLEDGEMENT

The research was supported by Shanghai Artificial Intelligence Laboratory, the National Key R&D Program of China (Grant No. 2022ZD0160104) and the Science and Technology Commission of Shanghai Municipality (Grant No. 22DZ1100102).

REFERENCES

[1] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3288–3295.

[2] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, Y. Lu, D. Zhou, Q. V. Le, *et al.*, "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.

[3] A. Asvadi, P. Girao, P. Peixoto, and U. Nunes, "3d object tracking using rgb and lidar data," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 1255–1260.

[4] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2301–2306.

[5] S. Verma, J. S. Berrio, S. Worrall, and E. Nebot, "Automatic extrinsic calibration between a camera and a 3d lidar using 3d point and plane correspondences," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3906–3912.

[6] Y. Xie, R. Shao, P. Guli, B. Li, and L. Wang, "Infrastructure based calibration of a multi-camera and multi-lidar system using apriltags," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 605–610.

[7] L. Grammatikopoulos, A. Papanagnou, A. Venianakis, I. Kalisperakis, and C. Stentoumis, "An effective camera-to-lidar spatiotemporal calibration based on a simple calibration target," *Sensors*, vol. 22, no. 15, p. 5576, 2022.

[8] 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, 2022.

[9] J. Kang and N. L. Doh, "Automatic targetless camera–lidar calibration by aligning edge with gaussian mixture model," *Journal of Field Robotics*, vol. 37, no. 1, pp. 158–179, 2020.

[10] P. Moghadam, M. Bosse, and R. Zlot, "Line-based extrinsic calibration of range and image sensors," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3685–3691.

[11] Z. Chai, Y. Sun, and Z. Xiong, "A novel method for lidar camera calibration by plane fitting," in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2018, pp. 286–291.

[12] I. Stamos, L. Liu, C. Chen, G. Wolberg, G. Yu, and S. Zokai, "Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes," *International Journal of Computer Vision*, vol. 78, no. 2-3, p. 237, 2008.

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

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

[15] 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)*, 2018, pp. 1110–1117.

[16] K. Yuan, Z. Guo, and Z. J. Wang, "Rggnet: Tolerance aware lidar-camera online calibration with geometric deep learning and generative model," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6956–6963, 2020.

[17] 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*, 2021, pp. 2894–2901.

[18] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[19] A. Tsaregorodtsev, J. Muller, J. Strohbeck, M. Herrmann, M. Buchholz, and V. Belagiannis, "Extrinsic camera calibration with semantic segmentation," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 3781–3787.

[20] P. Rotter, M. Klemiato, and P. Skruch, "Automatic calibration of a lidar–camera system based on instance segmentation," *Remote Sensing*, vol. 14, no. 11, p. 2531, 2022.

[21] 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.

[22] 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.

[23] J. Domhof, J. F. Kooij, and D. M. Gavrila, "An extrinsic calibration tool for radar, camera and lidar," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8107–8113.

[24] T. Tóth, Z. Pusztai, and L. Hajder, "Automatic lidar-camera calibration of extrinsic parameters using a spherical target," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8580–8586.

[25] 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.

[26] Z. Taylor, J. Nieto, and D. Johnson, "Multi-modal sensor calibration using a gradient orientation measure," *Journal of Field Robotics*, vol. 32, no. 5, pp. 675–695, 2015.

[27] 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." in *AAAI*, 2012.

[28] ——, "Automatic extrinsic calibration of vision and lidar by maximizing mutual information," *Journal of Field Robotics*, vol. 32, no. 5, pp. 696–722, 2015.

[29] 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)*, 2018, pp. 7342–7349.

[30] 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.

[31] Y. Zhu, C. Li, and Y. Zhang, "Online camera-lidar calibration with sensor semantic information," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4970–4976.

[32] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz*, vol. 24, pp. 345–348, 2010.

[33] A. Boulch and R. Marlet, "Fast and robust normal estimation for point clouds with sharp features," in *Computer graphics forum*, vol. 31, no. 5. Wiley Online Library, 2012, pp. 1765–1774.

[34] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.

[35] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[36] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[37] K. Koide, S. Oishi, M. Yokozuka, and A. Banno, "General, single-shot, target-less, and automatic lidar-camera extrinsic calibration toolbox," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 301–11 307.

[38] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[39] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[40] F. Furrer, M. Fehr, T. Novkovic, H. Sommer, I. Gilitschenski, and R. Siegwart, "Evaluation of combined time-offset estimation and hand-eye calibration on robotic datasets," in *Field and Service Robotics: Results of the 11th International Conference*. Springer, 2018, pp. 145–159.

[41] 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.