Iterative Closest Point of Object-based Point Clouds for Multi-View Extension

Junhyeong Ryu[1], Christina Suyong Shin[2], Jeongyeup Paek[1]

Department of Computer Science & Engineering, Chung-Ang University, Seoul, Republic of Korea[1]
Department of Computer Science, University of Southern California, Los Angeles, United States[2]
{rjh6883, jpaek}@cau.ac.kr[1], cshin956@usc.edu[2]

***Abstract*—Importance of LiDAR-based vehicle detection techniques for ensuring safe driving is increasing. However, LiDAR data from a single vehicle may have occlusions caused by obstacles, presenting a challenge. Eliminating these occluded areas through point cloud registration obtained from two vehicles can enhance autonomous driving systems. However, matching every point for point cloud registration requires a significant computational process, making real-time processing challenging. This paper proposes a novel technique that tackles this issue by introducing the *iterative closest point* (ICP) of *object-based point clouds*, enabling fast registration of point clouds scanned by two vehicles. The proposed approach demonstrates the ability to reduce the number of points significantly, resulting in a faster registration process compared to the conventional ICP approach.**

***Index Terms*—Iterative Closest Point (ICP), Point Cloud Registration, Multi-View Extension, Cooperative Perception**

I. Introduction

Autonomous vehicle driving relies on various sensors, including *light detection and ranging* (LiDAR), radio detection and ranging (RADAR), stereo cameras, and ultrasonic sensors, to perceive the surrounding environment. Among these sensors, LiDAR is studied extensively due to its higher accuracy compared to others. However, when recognizing the environment through LiDAR scanned from a single vehicle, occluded areas inevitably occur due to obstacles like nearby vehicles. Minimizing these occluded areas is crucial for reducing potential risks in autonomous driving.

To eliminate occluded areas, it is possible to acquire and register the point clouds of another vehicle that may have the view of the occluded area. In recent studies, endeavors are being made to achieve point cloud registration using various approaches involving artificial intelligence and the *iterative closest point* (ICP) [1] algorithm. However, these approaches are computationally intensive, leading to slow registration that are far from real-time. In addition, previous studies on point cloud expansion using ICP mainly focuses on recognizing the shape or characteristics of the surrounding area and expanding the point cloud based on that information [2], [3]. These methods have limitations, particularly on straight roads with little variation in the surroundings or areas with many similar objects.

To overcome these limitations, we propose *object-based iterative closest point* (O-ICP), a method of rapidly registering two point clouds by detecting dynamic objects and using selected point clouds around the object. With O-ICP, it is possible to pinpoint points based on the positions of dynamic objects and leverage the dynamic objects and surrounding features to extend point clouds around vehicles, offering a faster registration process compared to the conventional ICP method by significantly reducing the number of points that have the greatest impact on the speed of the ICP process.

II. Related Work

This section discusses the existing approaches on multi-view registration and object recognition in the point cloud.

A. Point Cloud Registration

ICP [1] is an iterative algorithm that registers two point clouds by minimizing the distances between points in the point clouds. However, it faces challenges when the initial poses of two point clouds do not match, resulting in synthesis with different local minima. Moreover, use of all points in repetitive operations leads to time-consuming computations.

To address these issues, several approaches have been proposed. Fast Point Feature Histogram [4] tackles initial pose problems by employing feature comparison with multi-dimensional vectors. TEASER [5] addresses the initial pose problem and speeds up the process by utilizing *Truncated Least Squares* costs that are insensitive to initial estimates. Additionally, some methods utilize learning-based models. PointNet [6] registers using point-wise features, while Indoor Multi-Robot Cooperative Mapping [3] achieves registration by learning point-wise geometric features and mapping local maps to global maps. FastReg [2] tackles the problem by learning correspondence using point-wise feature encoders and employing graph-based attention networks. Despite these efforts, the current state-of-the-art remains insufficient for real-time multi-view registration.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A4A5034130 & No. 2021R1A2C1008840), and also by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-RS-2022-00156353) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation)

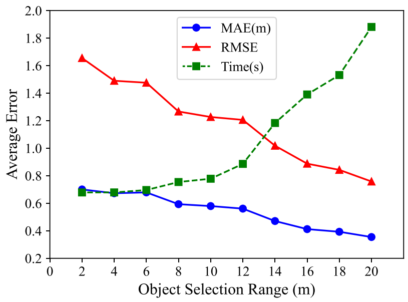
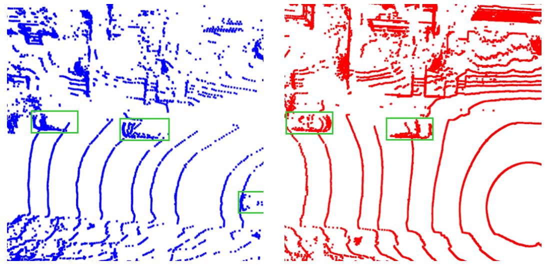


Fig. 1: Error rate by object selection range



(a) Left vehicle **(b)** Right vehicle

Fig. 2: Point cloud with the optimal selection distance for each vehicle

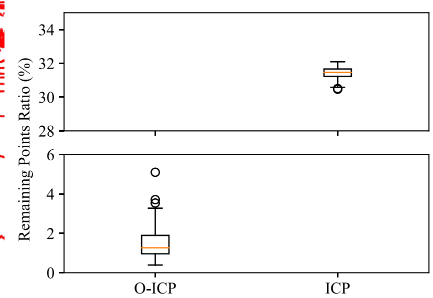


Fig. 3: Reduction in the number of point clouds by O-ICP

B. Point Cloud Object Detection

Recently, various techniques have been studied for object recognition in point clouds. VoxelNet [7] rapidly identifies objects by grouping point clouds into voxel units and incorporating location information with 3D convolution. However, due to the use of 3D information, it suffers from high computational complexity resulting in slow processing speed. Complex-YOLO [8] improves speed by converting the 3D point cloud into bird's eye view (BEV) and estimating object poses using a 2D operation with the Euler Region Proposal Network. While this approach enhances speed, accuracy still remains a concern. On the other hand, PointPillars [9] proposes a novel vertical encoder that leverages the vertical columns (pillars) of 3D point clouds, utilizing this encoder for feature encoding and enabling fast and accurate object detection through the use of 2D convolution layers. Therefore, our proposed O-ICP utilizes PointPillars for object detection.

III. Object-Based ICP (O-ICP)

In this section, we propose the O-ICP method for improved speed in point cloud registration.

The O-ICP procedure involves a sequence of steps: initial pose alignment, object detection, selection of objects around point clouds, ICP using the selected points, and subsequent transformation of remaining points via singular value decomposition (SVD). Initial pose alignment is determined based on the distances and angles between each vehicle's objects. By identifying the object with optimal distance and angle, a transformation matrix is applied to roughly align the positions to that of a chosen vehicle. During this process, due to the potential misidentification of objects and the absence of rotational matrices, precise alignment may not be achievable. However, subsequent ICP naturally resolves these issues.

To achieve object recognition, various methods can be employed, and in our approach, we utilize PointPillars which supports real-time object recognition. The recognized object information includes the center coordinates of vehicles and their extents. Through this, we can select crucial areas within the point cloud that encompass vehicles and utilize these areas for registration.

Next, we perform ICP on the point clouds of each vehicle's recognized object. By using only the points within the object's extent, the number of points is significantly reduced, resulting in faster registration. But, a challenge arises in distinguishing between the two vehicles since both point clouds were scanned from their respective perspectives, leading to an initial alignment problem. To tackle this, we designate a broader region than the object's extent to select point clouds of the object and its surrounding points. Fig. 1 plots the experiment result for finding the optimal range around the object. We determined that the optimal selection distance is 8 meters for achieving the lowest values of mean absolute error (MAE), root mean square error (RMSE), and registration time. We conducted our study based on this chosen distance.

Fig. 2 presents the process of selecting the optimal distance-based point cloud. Fig. 2a and Fig. 2b depict the selected point clouds by two vehicles recognizing the same object. Since the optimal distance from the vehicle center has been determined to be 8 meters, the point cloud around the vehicle is selected within a 16m x 16m square area centered around the vehicle's center. Through this selection, significant vehicle point cloud data and surrounding information are preserved, reducing the number of points in the overall point cloud while retaining distinctive features.

Fig. 3 represents the proportion of selected point clouds to the total point clouds for O-ICP, which were selected based on the optimal object-based distance, and ICP which only excluded points of the ground. O-ICP, which leverages object recognition to select points within the object's extent, retains an average of about 1.3% of the original point cloud. On the other hand, ICP, excluding ground points, retains approximately 31.4% of the entire point cloud. By selectively choosing points belonging to dynamic objects such as vehicles, O-ICP effectively reduces the number of points involved in the computation, leading to improved registration speed. Subsequently, after completing the ICP process, we utilize the obtained rotation and translation matrices to register the remaining points in the point cloud using a transformation based on SVD. By conducting ICP with a minimal set of points, the computational burden is significantly reduced.

To further enhance accuracy, we introduced an additional refinement step in the registration process. When the center points of two point cloud objects were within 1 meter of each other and there are three or more such pairs of points,

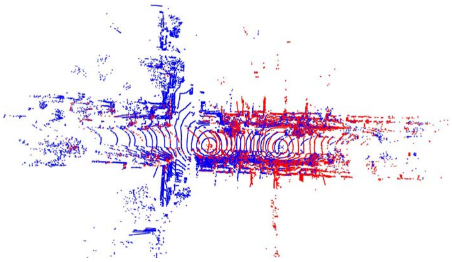


Fig. 4: Registration result of the point clouds from two vehicles.

Item \ Method	O-ICP (Ours)	ICP
Avg # of Points before Removal	51921.06	
Avg # of Reduced Points	693.19	16315.72
MAE(m)	0.59	3.01
RMSE	1.26	6.35
Time(ms)	754.22	5093.37

TABLE I: Comparison between O-ICP and ICP

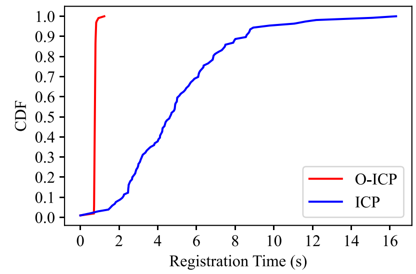


Fig. 5: Comparison of registration time between O-ICP and ICP

we perform additional iterations of the ICP using these point pairs as the reference. During this process, we only use points around the object's center, as the initial ICP has already accomplished rough alignment and registration. The reason for conducting an additional iteration only when there are three or more points is because, when there are two or fewer points, it can lead to a problem where one point becomes excessively close, causing instability in the registration.

IV. Evaluation

For the evaluation, the V2V4Real [10] dataset is used, which contains point cloud data simultaneously scanned by two vehicles on real roads. This dataset includes noise and variations that can occur in real-world scenarios, making it suitable for obtaining more accurate results compared to point clouds generated using simulators such as CARLA [11].

Fig. 4 presents the results of O-ICP using the V2V4Real dataset. It can be observed that the point clouds captured from each vehicle are well registered, and the extension is successful. To assess whether the O-ICP was performed correctly, error was measured using the distance between objects. Then, we use MAE and RMSE to compare O-ICP with the conventional ICP. The results can be found in Table I. For O-ICP, the average distance error between simple objects was around 0.59m, indicating a relatively accurate ICP with an RMSE of 1.26. On the other hand, the conventional ICP yielded an MAE of 3.01m, which is not significantly different, but the RMSE was 6.35, indicating a failure in proper registration. This is because, in the case of O-ICP, the object and its surrounding points are combined, leading to successful registration even with improper initial alignment. However for the conventional ICP, registration goal is to minimize the distances between all points, resulting in inadequate registration.

Finally, Fig. 5 plots the CDF of the time taken for point cloud registration. It can be observed that O-ICP exhibits significantly faster registration compared to ICP. Thus, the proposed method significantly improves the speed of point cloud registration while ensuring accurate alignment and enabling the detection of previously unseen objects through point cloud extension.

V. Conclusion

We proposed an object-based ICP (O-ICP) method to achieve fast registration of point clouds scanned from two vehicles. Our approach combines recognition of objects in point clouds with conventional ICP to reduce the number of points and improve the registration speed. Use of dynamic objects such as vehicles and the point cloud around the dynamic objects allows registration to be effective even in featureless straight roads where only surrounding vehicles are present. This method contributes to enabling real-time multi-view registration, which is essential for minimizing occlude areas during autonomous driving.

In future work, we plan to incorporate road shape recognition to achieve more accurate and faster registration of the point clouds. In addition, we plan to extend the field of view in shadow areas in real-time by leveraging cellular vehicle-to-everything (C-V2X) technology.

References

[1] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Transactions on pattern analysis and machine intelligence*, no. 5, pp. 698–700, 1987.

[2] E. Arnold, S. Mozaffari, and M. Dianati, "Fast and Robust Registration of Partially Overlapping Point Clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1502–1509, 2021.

[3] Z. Zhu, W. Jiang, L. Yang, and Z. Luo, "Indoor Multi-Robot Cooperative Mapping Based on Geometric Features," *IEEE Access*, vol. 9, pp. 74 574–74 588, 2021.

[4] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," in *IEEE international conference on robotics and automation*, 2009, pp. 3212–3217.

[5] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and Certifiable Point Cloud Registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2020.

[6] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[7] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499.

[8] M. Simon, S. Milz, K. Amende, and H.-M. Gross, "Complex-YOLO: Real-time 3D Object Detection on Point Clouds," *arXiv preprint arXiv:1803.06199*, 2018.

[9] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast Encoders for Object Detection from Point Clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[10] R. Xu, X. Xia, J. Li, H. Li, S. Zhang, Z. Tu, Z. Meng, H. Xiang, X. Dong, R. Song *et al.*, "V2V4Real: A Real-world Large-scale Dataset for Vehicle-to-Vehicle Cooperative Perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 712–13 722.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.