Improvement of Sequential SLAM by Considering Semi-dynamic Objects in Indoor Dynamic Environments

Yuhei Oshikubo
Cource of Precision Engineering
Chuo University
Tokyo, Japan
oshikubo@sensor.mech.chuo-u.ac.jp

Keigo Akiba
Cource of Precision Engineering
Chuo University
Tokyo, Japan
akiba@sensor.mech.chuo-u.ac.jp

Sarthak Pathak
Dept. of Precision Mechanics
Chuo University
Tokyo, Japan
pathak@mech.chuo-u.ac.jp

Yonghoon Ji
Graduate School for Advanced Science and Technology
Japan Advanced Institute of Science and Technology
Ishikawa, Japan
ji-y@jaist.ac.jp

Kazunori Umeda
Dept. of Precision Mechanics
Chuo University
Tokyo, Japan
umeda@mech.chuo-u.ac.jp

***Abstract*—In this paper, we propose a sequential SLAM method considering semi-dynamic objects that is robust in dynamic indoor environments. Generally, accuracy of SLAM decreases in dynamic environments and environments with few geometric features. Therefore, the proposed method utilizes semantic information in addition to geometric information via object detection, and extracts point clouds that are effective for localization by considering the attributes and relationships of the objects. Moreover, the method can extract more features even in environments with few static objects and clearly identify the use of each object point cloud for SLAM by adopting the concept of semi-dynamic objects. The effectiveness of the proposed method is verified by experiments.**

***Index Terms*—SLAM, object detection, scan matching, mobile robot, dynamic environments**

I. Introduction

In recent years, as the working population declines, robots are being used to replace human tasks. Many autonomous robots have already been deployed in specific spaces such as airports, restaurants and factories, where they are used for a variety of purposes, including transportation, security, and cleaning. Autonomous robots in these environments often use pre-built, high-precision maps. In other words, the map must be rebuilt each time the environment changes such as a construction site.

Simultaneous Localization and Mapping (SLAM) is often used to construct maps. When SLAM is classified by sensor, there are typically LiDAR SLAM [1], [2] and Visual SLAM [3]–[5]. Besides these, there are SLAM using WiFi [6], sonar [7] and so on. In some cases, sensor fusion is used to overcome situations that would be difficult with a single sensor.

However a static environment is often a prerequisite in the SLAM described above.This is due to two major problems with general SLAM in a dynamic environment. First, the frequent mismatch of correspondences between frames can degrade accuracy for localization. Second, a phenomenon called "flying ghosts" [8] occurs. This phenomenon is that unnecessary point clouds of dynamic objects such as people remain in the constructed point cloud map, which affects mapping. Moreover, in environments with poor geometry features such as straight corridors, localization may fail due to degeneracy, which occurs when the self-location is not uniquely determined because the geometric features are similar. In [9], dynamic and static objects were predefined, and a system was constructed to cope with dynamic environments by removing dynamic point clouds and using static objects as landmarks through deep learning detection. However, some of the defined static objects, such as PCs and chairs, had the potential to move and could not be guaranteed to be static. Furthermore, since there are generally few static objects that do not completely change position over time, approaches [10] that removes all potentially moving objects could break localization in environments with poor geometry features.

Thus, we introduce the concept of semi-dynamic objects [11] and propose a robust and accurate SLAM method for indoor dynamic environments. The definition of semi-dynamic objects is described in detail in section II-B. This method can be used to easily build maps of frequently changing environments, and the resulting environmental maps are expected to be used as preliminary information for self localization estimation by autonomous mobile robots.

II. Proposed Method

A. Outline

Fig. 1 shows flow of the proposed method. First, RGB images and 3D point clouds are acquired in multiple frames using an RGB-D camera. Wheel odometry obtained from

979-8-3503-2099-2/23/$31.00

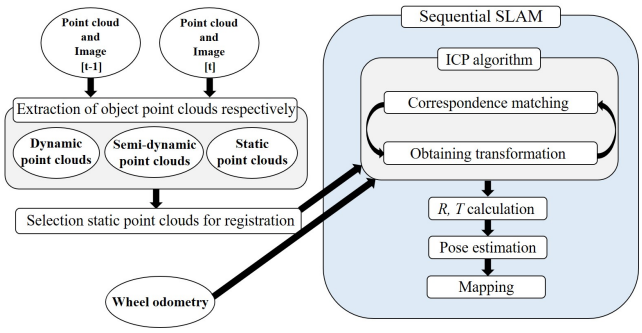


Fig. 1. Flow of the proposed method

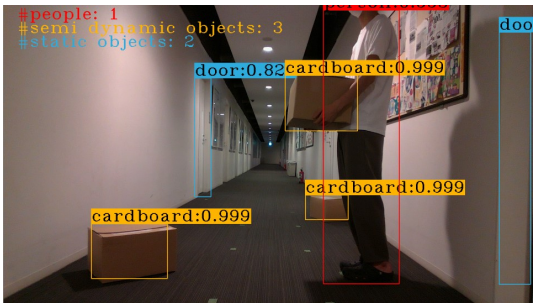


Fig. 2. Detection result by YOLO trained custom dataset

different frames is also acquired to use the initial position for the scan matching. Next, object point clouds are extracted from the acquired point clouds using image-based object detection. This object detection method is described in section II-C. Dynamic point clouds are then removed, leaving only static point clouds that are effective for positioning. This method is explained in section II-E. The static point cloud contains not only 3D coordinate information, but also information on the object and its color. The static point cloud is used for point cloud registration by ICP (Iterative Closest Point) [12] , and the rotation matrix R and the translation matrix T are calculated. Finally, R and T are used for localization and mapping.

B. Definition

We classify objects into three categories based on their frequency of movement: static objects, semi-dynamic objects, and dynamic objects. Dynamic objects are defined as those that move frequently such as humans, semi-dynamic objects are those that move occasionally such as chairs [11], and static objects are those whose positions are fundamentally unchanged such as walls. In our experiments, we consider humans as dynamic objects, cardboard boxes as semi-dynamic objects, and doors as static objects. As we assume an indoor environment in our study, we disregard external factors such as wind, and assume that semi-dynamic objects move only due to the influence of dynamic objects. In actual construction sites, cardboard boxes and carts do not move on their own, and it can be said that they move only when acted upon by dynamic objects such as humans or robots.

C. Object Detection

In this method, object detection was performed on RGB images acquired by an RGB-D camera using YOLOv4 [13]. The objects which we want to detect in this study had not been trained previously, so we were newly trained. The human, cardboard, and door were trained using 4200 images, which were expanded from the 175 images taken by the author by adding left-right inversion, sesame noise, and brightness changes to the images. Fig. 2 shows an example of the output results, indicating that a person, a cardboard box, and a door were detected, respectively. The information obtained from YOLOv4 results is used to extract object point clouds as described in section II-D.

D. Point Cloud of Each Object

The range image can be converted into a 3D point cloud. Therefore it is possible to extract a point cloud corresponding to the object region including the rectangular region containing the background in the image by matching the RGB image and the range image obtained in section II-C. However, since the point cloud includes the background behind it as well, clustering is performed after downsampling. The point cloud obtained from the RGB-D camera is dense, and if processed as is, the calculation time for clustering and scan matching becomes long. Therefore, downsampling is performed to reduce the computation cost. In this method, voxel grid downsampling is used to equalize the density of the point cloud. The clustering method adopts Euclidean clustering using the Euclidean distance between points, which enables exclusion of the point cloud of walls to some extent.

Point clouds are also converted for the areas where objects are not detected in the image, and after downsampling, they are stored in memory. These point clouds are not used for localization, but are left as point clouds used for map building.

E. Selection of Point Clouds for Registration Considering Semi-dynamic Objects

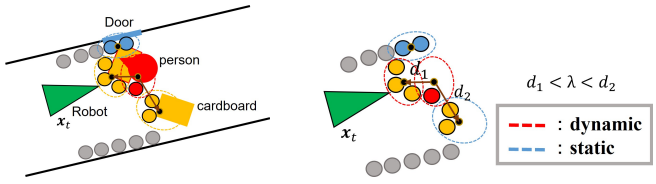
As mentioned in section II-B, whether a semi-dynamic object is moving or not is considered to be caused by a dynamic object in the indoor case. We focuse on the distance between the semi-dynamic object and the dynamic object. Therefore, if there is a dynamic object near a semi-dynamic object, the semi object is determined to be moving.

Fig. 3 shows a top view of the 3D space. The assumption is that we know what object each point corresponds to via object detection. The points represent the point cloud acquired from the sensor. The blue points correspond to doors, the orange points to cardboard boxes, and the red points to people. Also, the green triangle represents the robot.

First, the center of gravity of each object is calculated by averaging the 3D coordinates of the object points. The distance from the center of gravity of the dynamic object point cloud to the center of gravity of the semi-dynamic object is then calculated. If the Euclidean distance between the center of

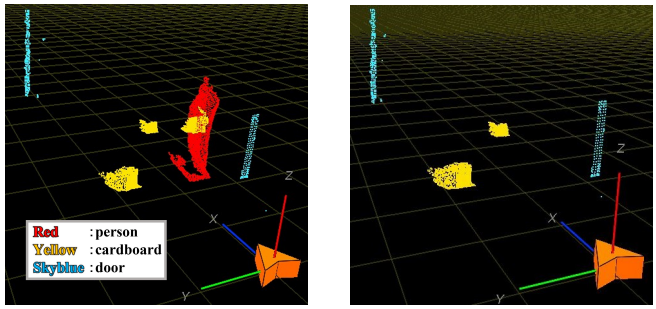
gravity of the dynamic object point cloud and the center of gravity of the semi-dynamic object point cloud is greater than a certain threshold value λ, the semi-dynamic object point cloud is used as the static point cloud for the alignment calculation. Conversely, if the distance is less than the threshold, the semi-dynamic object point cloud is considered to be a dynamic point cloud and is removed, and is not used for either positioning or map construction. The threshold value of λ is empirically set to 1.0 m. Fig. 3(b) shows that a semi-dynamic object is considered dynamic or static by the distance between the centers of gravity. In the figure, the points surrounded by the blue dashed line indicate that the group of points is considered static, while the points surrounded by the red dashed line indicate that the group of points is considered dynamic. When the distance d_1 is less than λ, it can be dynamic. On the other hand, when the distance d_2 is more far than λ, it can be static. Here, the point clouds judged to be static are used for positioning, i.e., localization. The same process is performed for each frame in sequence, with the static point clouds obtained in adjacent frames used as input for registration by ICP, and map construction based on that localization. Wheel odometry obtained from adjacent frames is used for initial positioning before ICP.

Fig. 4(a) and 4(b) show the actual data before and after point cloud selection for positioning, respectively. Fig. 4(a) is the result of the process described in section II-D based on the information as shown in Fig. 2 and the range image. Fig. 4(b) shows that the point cloud of a person and a cardboard box held by the person, which existed in Fig. 4(a), have been removed by the processing described in this section. This removes the negative impact of the dynamic point cloud on SLAM.



(a) Environments (b) Process

Fig. 3. Explanation of selection of static point clouds for registration



(a) Before processing (b) After processing

Fig. 4. Example of selection of static point clouds for registration

III. Experiment

A. Experimental Conditions

We conducted experiments to verify whether the accuracy has improved compared to conventional methods. Also, this experiment was conducted in a scenario where a person carries a cardboard box in a linear corridor as shown in Fig. 5(a), which is assumed to be dynamic and low geometric feature environments. There are three cardboard boxes, pre-defined as semi-dynamic objects, two are placed on the ground and the other was carried by a person along the way. Next, we explain Fig. 5(c). The numbers represent the positions of each object in each frame. The red, yellow, blue, and green symbols represent a person, cardboard, door, and robot, respectively. The robot was moved forward 1.0 m at a time by five times, using marks previously placed on the floor, and six measurements were taken. If the correction amount by localization using ICP is extremely large, the point cloud registration is judged as a failure, and only wheel odometry is used for localization.

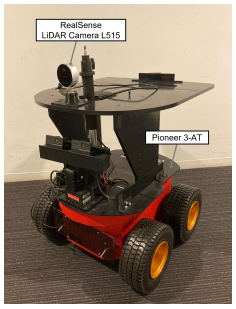
The evaluation metrics were the robot pose error (RSE) and the map construction result. The RSE was quantitatively evaluated by the Euclidean distance between the true position and the estimated position. Comparisons were made for cases where human removal processing was not performed, cases where human removal processing was performed [6], and cases where the proposed method was used. In all cases, the ICP implemented in the Point Cloud Library (PCL) was used for close localization. The map construction results were compared between the case without the human removal process and the case with the proposed method to qualitatively evaluate the presence or absence of unnecessary point clouds. The difference between the map construction results of the proposed method and those of the human removal SLAM [14] was not shown in the experimental results because it depends on the definition of dynamic and semi-dynamic objects. For semi-dynamic objects classified as dynamic, there is no problem with not using them in both localization and mapping. However, for semi-dynamic objects classified as static, there is a possibility that they may move with temporal changes even if they were static and present at any position during SLAM. If localization is performed using a pre-build map that includes semi-dynamic objects and those objects have moved to a different location than they were during SLAM, there is a risk that the accuracy of localization will decrease. For these reasons, the proposed method did not use point clouds of semi-dynamic objects for mapping, regardless of whether they were static or dynamic during SLAM.

In this study, a Pioneer 3-AT by Adept MobileRobots was used as the mobile robot, and RealSense LiDAR Camera L515 by Intel as an RGB-D camera was fixed it as shown in Fig. 5(b). The mobile robot was controlled by the author using a laptop PC.

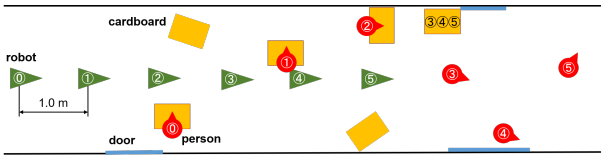
B. Experimental Results

Fig. 6 and Table I show the results of the robot pose error whose units is meters. From these results, it can be said that

(a) Environments



(b) Equipments



(c) Scenario

Fig. 5. Experimental conditions

the proposed method is more accurate localization than the compared methods in this experimental scenario. This may be thanks to the fact that the proposed method was able to utilize a larger number of static point clouds for localization than the other methods. However, all methods failed to localize with ICP in the fifth frame. For the other two methods, the accumulated errors became larger, and they fell into the local minimum, resulting in failure of correct positioning. In the case of the proposed method, the number of objects that can be used for positioning is reduced because a cardboard box cannot be observed in the fifth frame due to the viewing angle of the sensor.

Next, we compare the results of the constructed map. Fig. 7 shows the results of map construction without human removal and with the proposed method. Fig. 7(a) shows that the constructed point cloud includes people and cardboard boxes, while Fig. 7(b) shows that the point clouds of people and cardboard boxes were almost completely removed. This shows that the proposed method is robust to dynamic environments.

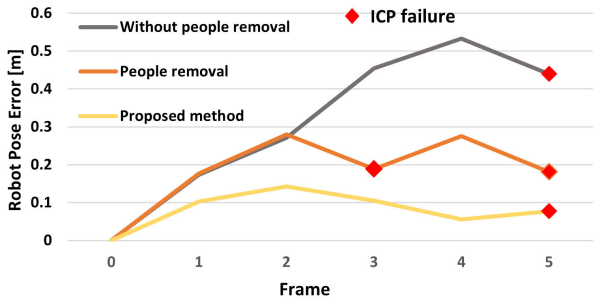
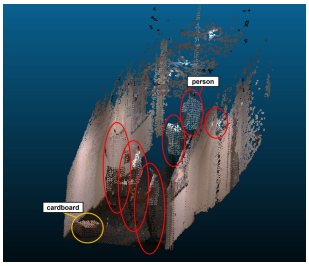


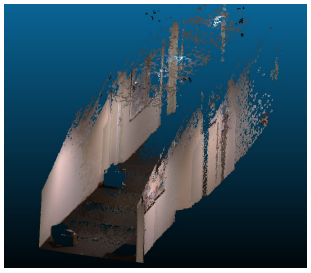
Fig. 6. Robot pose error

TABLE I
ROBOT POSE ERROR FOR EACH METHOD

Method	0	1	2	3	4	5
Only ICP	0.000	0.173	0.271	0.455	0.533	0.440
Akiba's [14]	0.000	0.177	0.280	0.189	0.276	0.181
Ours	0.000	0.103	0.143	0.106	0.056	0.077



(a) Without people removal



(b) Proposed method

Fig. 7. Constructed maps by each method

IV. CONCLUSION

In this paper, we proposed an sequential SLAM method that extracts effective static point clouds for localization based on the relationship between the attributes of objects detected using YOLO and the distances between objects in 3D space in dynamic indoor environments, and demonstrated its usefulness for SLAM. Experimental results show that incorporating and using the concept of semi-dynamic objects contributes to the robustness of SLAM in indoor dynamic environments.

However, the fact that errors accumulate over long distances when used sequentially is an issue for this method. It is necessary to optimize not only sequential processing, but also the entire processing comprehensively, including loop closure, taking into consideration the handling of semi-dynamic objects. In the future, we will conduct experiments under various scenarios to verify the effectiveness of this method.

REFERENCES

[1] Tixiao Shan, Brendan Englot,"LeGO-LOAM: lightweight and ground-optimized Lidar odometry and mapping on variable terrain," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4758-4765, 2018.

[2] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguère, Jens Behley, Cyrill Stachnis,"SuMa++: efficient LiDAR-based semantic SLAM," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4530-4537, 2019.

[3] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton and Olivier Stasse, "MonoSLAM: real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 6, pp. 1052-1067, 2007.

[4] Raúl Mur-Artal, Juan D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, Vol.33, No.5, pp.1255-1262, 2017.

[5] Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, Wolfram Burgard, "3-D mapping with an RGB-D camera," *IEEE Transactions on Robotics*, Vol. 30, No. 1, pp. 177-187, 2014.

[6] Brian Ferris, Dieter Fox, Neil Lawrence, "WiFi-SLAM using Gaussian process latent variable models,", *Proceedings of the 20th international joint conference on Artifical intelligence*, Vol. 7, pp. 2480–2485, 2007.

[7] Jinwoo Choi, Sunghwan Ahn, Wan Kyun Chung, "Robust sonar feature detection for the SLAM of mobile robot,"*IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3415-3420, 2005.

[8] M. Arora, L. Wiesmann, X. Chen and C. Stachniss, “Mapping the static parts of dynamic scenes from 3D LiDAR point clouds exploiting ground segmentation,” *European Conference on Mobile Robots (ECMR)*, pp. 1-6, 2021.

[9] Fanguwei Zhong, Sheng Wang, Ziqi Zhang, China Chen, Yizhou Wang,“Detect-SLAM: Making object detection and SLAM mutually beneficial,” *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[10] Wenbo Liu, Wei Sun, Yi Lu,“DLOAM: real-time and robust LiDAR SLAM system based on CNN in dynamic urban environments,”*IEEE Open Journal of Intelligent Transportation Systems*, 2021.

[11] Hongjun Zhou, Shigeyuki Sakane,“Localizing objects during robot SLAM in semi-dynamic environments,” *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2008.

[12] Paul J. Besl, Neil D. McKay, “A method for registration of 3-D shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp.239-256, 1992.

[13] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, “YOLOv4: Optimal speed and accuracy of object detection,” arXiv:2004.1093, 2020.

[14] Keigo Akiba, Ryuki Suzuki, Yonghoon Ji, Sarthak Pathak, Kazunori Umeda, “Performance improvement of ICP-SLAM by human removal process using YOLO,” Applied human informatics, Vol.5, No.1, pp.1-13, 2023.