Applied human informatics, vol.5 No.1

Performance Improvement of ICP-SLAM by Human Removal Process Using YOLO

Keigo AKIBA[1] Ryuki SUZUKI[1] Yonghoon JI[2] Sarthak PATHAK[3] Kazunori UMEDA[3]

[1] Course of Precision Engineering, School of Science and Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan
{akiba,r.suzuki}@sensor.mech.chuo-u.ac.jp
[2] Course of Advanced Science and Technology, School of Materials Science / Intelligent Robotics Area, Japan Advanced Institute of Science and Technology (JAIST), Asahidai 1-1, Nomi, Ishikawa, 923-1211, Japan
ji-y@jaist.ac.jp
[3] Department of Precision Engineering, Faculty of Science and Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan
pathak@mech.chuo-u.ac.jp, umeda@sensor.mech.chuo-u.ac.jp

Abstract. In this paper, we propose a novel iterative closest point (ICP)-based simultaneous localization and mapping (SLAM) approach that can build robust map information even in indoor environments where humans coexist. Several SLAM methods that have been studied so far assume a stationary environment. But there are challenges in operating in a dynamic environment with moving objects such as humans. Specifically, when a mobile robot constructs a map in an environment where humans coexist nearby, humans cause false matching in alignment sensor data. Furthermore, human occlusion also makes it difficult to construct a map with high accuracy. Therefore, we propose a human removal process that utilizes You Look Only Once (YOLO) to detect humans in image data. In this paper, by using this process with ICP-SLAM, we aim to improve the accuracy of map construction in an environment where humans coexist nearby. In our experiments, we verified the accuracy of map construction in comparison with conventional methods. This experiment is conducted in an indoor corridor where humans coexist nearby. Although we used ICP-SLAM for verification this time, the human removal process can be adapted to other SLAMS.

Keywords. SLAM, ICP, mobile robot, human removal process, YOLO, map construction

Received: Jul. 20 2022; Revised: Dec 11 2022; Accepted: Feb. 8 2023

1 Introduction

The use of autonomous mobile robots to replace human workers is currently attracting attention. Specifically, these robots are being introduced into indoor environments where humans coexist nearby. This purpose is for transportation and guidance in factories and airports. In recent years, a remarkable shortage of manpower has been caused by the decrease in the working population. Therefore, the importance of these robots is expected to continue to increase and the automation by robots is remarkable. However,

autonomous movement with high efficiency is essential because these robots are required to work at the same level or higher than humans in the real environment. For this reason, it is necessary to construct a map in advance and operate using it. Specifically, maps are used as advanced information for navigation and to improve the accuracy of map construction by autonomous robots [1,2]. Therefore, prior map building by SLAM is an essential process for the operation of these robots.

A lot of studies for SLAM. Among them, ICP-based SLAM is often used. ICP is one of the methods for aligning two different 3D measurement data and calculates rigid body transformation parameters for aligning them. ICP-based SLAM approach uses only environmental shape information obtained from sensors [3,4,5,6]. In addition, many visual SLAM approaches that extract and utilize features from images acquired by cameras also have been proposed [7][8][9].

However, all of these methods assume a stationary environment and have limitations in applying to real environments. Therefore, in dynamic environments in which humans exist, false alignment and matching can occur in sensor data. In addition, since sensor data is defective due to human occlusion, the removal of human sensor data is necessary for highly accurate map construction. In conclusion, robust pre-mapping in response to human occlusion is an important issue for the introduction of autonomous mobile robots in real environments.

Therefore, to deal with humans, dynamic features are detected by using difference processing between adjacent frames, and dynamic objects including humans are dealt with by not using these features for matching.[10,11,12,13] Another method uses RGBD images to discriminate static and dynamic regions in the image between adjacent frames, and addresses dynamic objects by leaving only static regions.[14,15] However, these methods cannot detect humans when they are temporarily stationary. In that case, all the point clouds of the humans remain and are recognized as static features, resulting in false matching.

Yu et al. use semantic segmentation [16] for multiple object detection and dense map construction with human removal [17]. However, semantic segmentation for multiple object detection requires a large amount of training data and time.

Joan et al. and Berta et al. used YOLO[18] and Mask R-CNN[19] to extract dynamic features of detected regions and remove humans from images[20,21]. However, these methods require multiple frames to remove a human for a narrow processing area. Here, a narrow processing area refers to the size of the area where sensor data can be obtained per frame. In addition, the narrow search range of a single frame results in a large proportion of occlusions in the image when removing human occlusions. On the other hand, a preliminary map for an autonomous mobile robot requires a rough map of the entire environment rather than a detailed map, so a method to search a wide area with fewer frames is suitable. Where rough and detailed refer to the density of the point cloud.

In this paper, we propose a robust map construction system that can search a wide area with fewer frames in an indoor environment where humans coexist, by applying the human removal process based on object detection using YOLO to SLAM. The novelty of our method is that the results of object detection by YOLO are reflected in the point cloud acquired by LiDAR using point cloud correspondence, which enables human identification. We explain the method of reflecting the detection results in Chapter 3. In addition, by using an RGBD camera in this method, the results of object detection in the image can be easily reflected and handled with high accuracy. This allows only the portion of the point cloud acquired by LiDAR that requires human exclusion in the vicinity of the robot to be handled with the minimum amount of point cloud removal

processing required. If we use only LiDAR, a sensor data is only a point cloud, making human detection difficult. The measurement range of RGB-D camera is shorter than LiDAR. So the distance at which a human can be detected is limited. But this is not a problem when the human is located far from the robot, because the effect on map construction is little.

Furthermore, several studies of dynamic SLAM cannot detect humans when they are temporarily stationary. In that case, all the point clouds of the humans remain and are recognized as static features, resulting in false matching.

On the other hand, our method is robust because it can detect humans regardless of their movement status.

The rest of this paper is structured as follows: Section 2 discusses the outline of ICP-SLAM by human removal process by using YOLO, Section 3 gives the details of the proposed human removal process proposal. Section 4 details the ICP algorithm as a method to align two different 3D point clouds, Section 5 details the true map, the evaluation method, and the experimental results. Section 6 presents the conclusions and lines for future work.

2 Outline of the method

An overview of the proposed method is shown in Fig. 1. In this method, RGB-D images and 3D point cloud information acquired by the range image sensor and LiDAR on the robot are used for map construction. The RGB-D camera can acquire images and point clouds in close areas from the robot, while LiDAR can acquire point clouds in a wide area around the robot. In this paper, point clouds acquired by the RGB-D camera are denoted as point cloud $\boldsymbol{P_r}$, and point clouds acquired by LiDAR are denoted as point cloud $\boldsymbol{P_l}$. First, the position of the robot is updated by odometry.

Odometry is calculated by the angle of rotation of the wheels obtained from the robot's internal sensors. Next, the human removal process is performed on the point cloud $\boldsymbol{P_l}$. The process identifies the human by using images and removing the human point cloud. For human identification, we use YOLO, a fast object detection algorithm based on deep learning, to detect humans in images. Then, by mapping the image to the point cloud $\boldsymbol{P_r}$, the results of the human detection are also reflected in the point cloud $\boldsymbol{P_l}$, furthermore, the point cloud of the human is detected and removed. After that, we use the ICP (iterative closest point) algorithm to align the points and construct a map of the environment without point cloud data of humans.

In this method, the point cloud is downsampled using a voxel grid, which reduces the density of the point cloud and speeds up the point cloud processing.

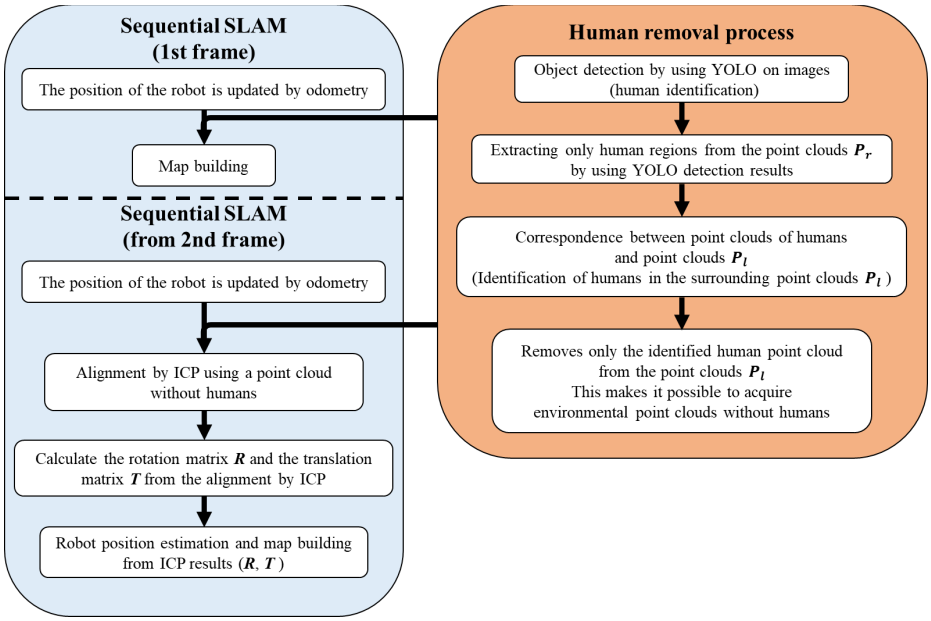


Fig. 1 Outline of the proposed method.

3 Human removal process

In this section, we discuss how to deal with human occlusion, which is essential for robust map construction in environments with humans. First, we perform human detection using YOLO to find the human area in the image, as shown in Fig. 2. The human region is a rectangular bounding box including the background. Next, from the range image acquired by the range image sensor, we extract only the point cloud $\boldsymbol{P_r}$ that belongs to the human area detected by YOLO. (Fig.3. (a)) Here, this human area includes the background. Thus, we use clustering and identify the background by the number of point clouds. Then the background is removed from the original human area as shown in Fig. 3. The point cloud data of the human is extracted. After that, this point cloud $\boldsymbol{P_l}$ of the human area obtained and the point cloud $\boldsymbol{P_l}$ obtained by LiDAR are matched for correspondence using the nearest neighbor search [22]. This correspondence allows the results of human identification in the image to be reflected in the point cloud obtained $\boldsymbol{P_l}$, enabling the detection of humans in the point cloud $\boldsymbol{P_l}$ [23].

Then, as shown in Fig. 4, the point cloud $\boldsymbol{P_l}$ of the surrounding environment without humans is obtained by removing the corresponding point cloud of humans from the point cloud $\boldsymbol{P_l}$.

In this method, the downsampling process is used as a preprocessing step for the point cloud. To map the point cloud $\boldsymbol{P_l}$ to the point cloud of a person, the threshold of the downsampling process is set individually in order to map more point clouds. Specifically, the density of the point cloud $\boldsymbol{P_r}$ camera, which is the search source, is increased and the density of the point cloud $\boldsymbol{P_l}$ is decreased. This allows for downsampling while allowing for many correspondences.

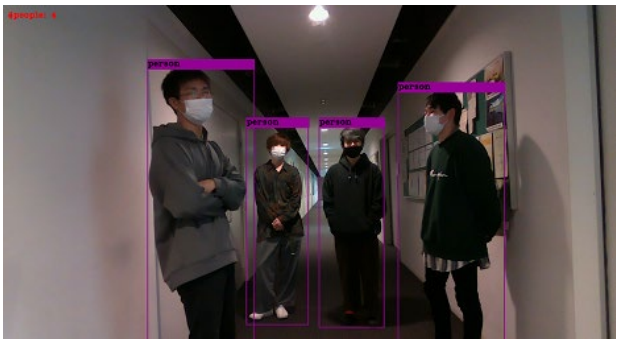
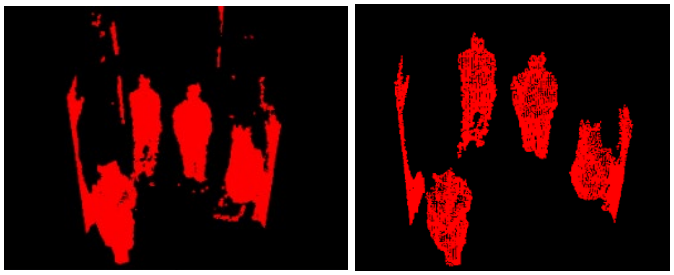
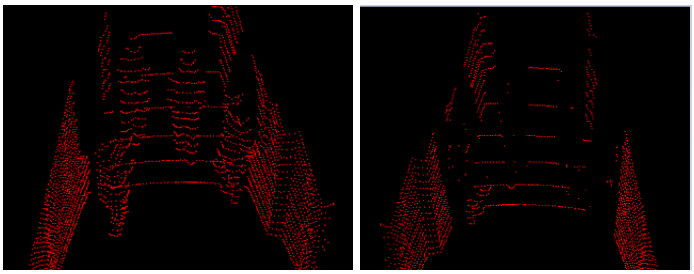


Fig.2 The result of human detection by YOLO.



(a) before clustering (b) after clustering

Fig.3 The result of background removal.



(a) before removal (b) after removal

Fig.4 Human removal process.

4 Alignment by ICP

In this paper, we use the ICP algorithm as a method to align two different 3D point clouds. In general, in SLAM for autonomous mobile robots, the robot position obtained by odometry is used as the initial position. However, errors accumulate in odometry, and as the distance traveled increases, the map becomes distorted. In this paper, we propose a method to reduce the error by using ICP.

The alignment of the point cloud is evaluated by the average of the distances between the points of each point and is attributed to the minimization problem. In this case, the point cloud measured in the current frame after human removal (source point cloud) is aligned with the point cloud measured in the previous frame (target point cloud). the evaluation formula of ICP algorithm is as follows.

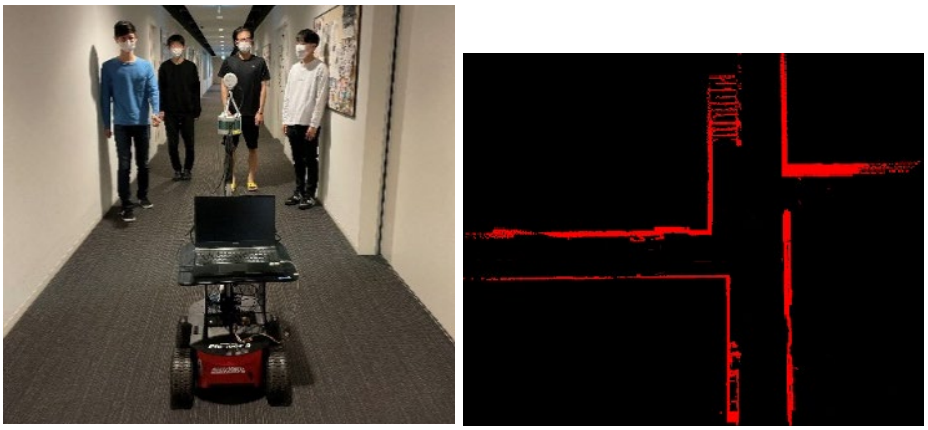
$$E = min \sum_{i=1}^{N} \left| \boldsymbol{p}_{k_i} - (\boldsymbol{q}_i \boldsymbol{R} + \boldsymbol{T}) \right|^2 \quad (1)$$

- E: sum of the squared distance (i.e., evaluation value)
- $\boldsymbol{p}$: a point in the source point cloud
- $\boldsymbol{q}$: a point in the target point cloud
- N: the number of points in the source point cloud (i.e., number of iterations)
- k_i: the reference scan data point corresponding to the point i in the source point cloud
- $\boldsymbol{R}$: the rotation matrix
- $\boldsymbol{T}$: the translation vector

Using the evaluation formula shown in Equation (1), the positioning accuracy is evaluated by the sum of squares $\boldsymbol{E}$ of the distance between points, moreover, the rigid body transformation parameter when e is minimum is obtained [3]. In this paper, the point cloud of the frame after human removal (source point cloud) and the point cloud of one frame before (target point cloud) are aligned by ICP. Then, the rotation matrix R and the translation vector T calculated from the ICP results are used to estimate the robot's self-position and construct a map at the same time.

5 Experiments

To validate the proposed method, we compared and evaluated the accuracy results of maps by odometry and ICP-SLAM without human removal, and ICP-SLAM with human removal. The accuracy of the map was evaluated by calculating the distance between each point on the map by each method from each point on the true map and taking the average of the sum of the distances as the error. In this experiment, as shown in Fig. 5 (a), we used a mobile robot (Adept Mobile Robots Pioneer-3DX), a range image sensor (Intel RealSense LiDAR Camera L515), and a LiDAR (Velodyne LiDAR's VLP-16). We fixed the range image sensor and LiDAR on the mobile robot. The range image sensor and LiDAR were fixed on the mobile robot for the measurement. The overhead view of the environment is shown in Fig. 5 (b) and the detail of the experimental environments are shown in Fig. 6 and Fig. 7 for experiment I and II respectively. The human movement paths for each experiment are shown in Fig. 8. The trajectory of the mobile robot in this experiment is shown by the yellow line in Fig. 9, Fig. 10, Fig. 11, and Fig. 12.

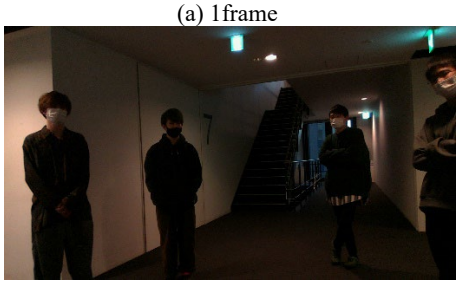


(a) State of experiment (b) Overhead view of the environment

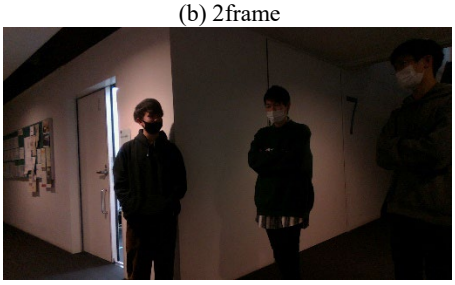
Fig.5 The experimental environment.

(a) 1frame

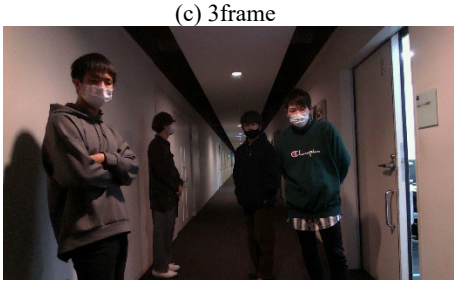
(b) 2frame



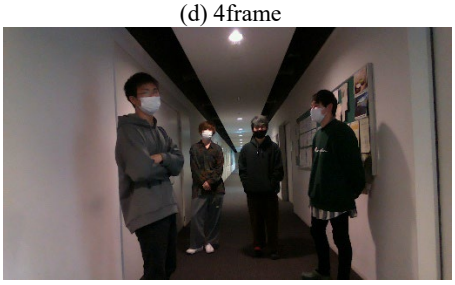
(c) 3frame



(d) 4frame



(e) 5frame

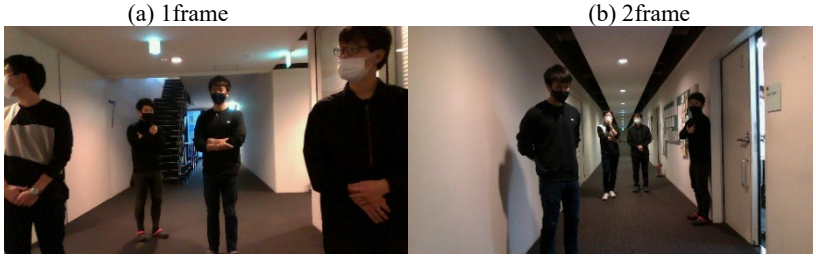


(f) 6frame

(g) 7frame

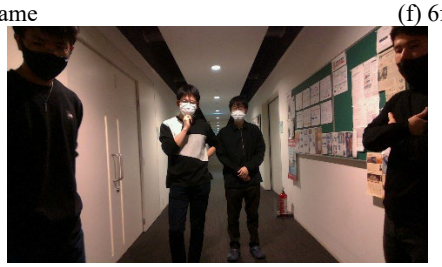
Fig.6 The experimental I environment

(a) 1frame (b) 2frame



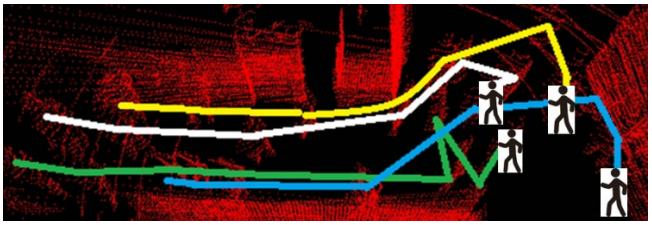
(c) 3frame (d) 4frame

(e) 5frame (f) 6frame

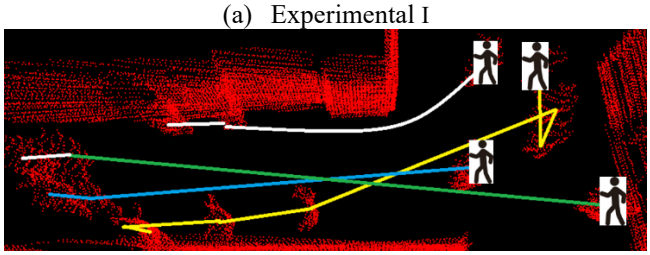


(g) 7frame

Fig.7 The experimental II environment.



(a) Experimental I



(b) Experimental II

Fig.8 Human movement paths.

5.1 Situation

As shown in Fig. 6 and Fig. 7, this experiment was conducted in the hallway on the seventh floor of Building No. 2 at Chuo University's Korakuen Campus for a total of seven frames in two ways. In the experiments, humans are always in the camera's field of view. In experiment I, as shown in Fig. 6, four humans were stationary in all frames. In Experiment II, as shown in Fig. 7, one of the four humans walked and the other three remained stationary in all frames. In both cases, the positions of the humans were changed in each frame to reproduce a dynamic environment that is closer to the real environment. In the experiments, the humans are always in the field of view of the RGB-D sensor. However, in a real situation, they may move in and out. To counter this, the use of multiple RGB-D sensors can help. This will be considered future work. Human movement paths are shown in Fig. 8. In the measurement experiment, the robot moves for approximately 10 seconds between each frame, and after moving, it stands still for approximately 5 seconds to take pictures. In this experiment, this action was repeated for a total of seven frames.

5.2 True map

The true maps used in the evaluation were created by manual positioning of the sensor data acquired in this experiment. Since there are two types of maps for each method, one with humans removed and one without, two types of true maps were prepared. So, error calculations were performed between the same types. Therefore, the accuracy of the environmental maps can be evaluated without the influence of human point clouds. Since both true maps are aligned only on the non-human environmental map portion, there is no human influence on creating the true map.

5.3 Evaluation method

Each point in the point cloud has 3D coordinates and indexes as information and is stored as sensor data. Therefore, the error of the map for each method is calculated by the distance between points with the same index in the true map.

The average of the sum of the distances between the points is calculated as the average of the errors, furthermore, the accuracy of the map construction is evaluated according to the errors.

In addition, there is a difference in the number of points due to the presence or absence of humans in each method. However, since two types of true maps are used, there is no effect of the difference in point cloud size during the evaluation.

5.4 Results

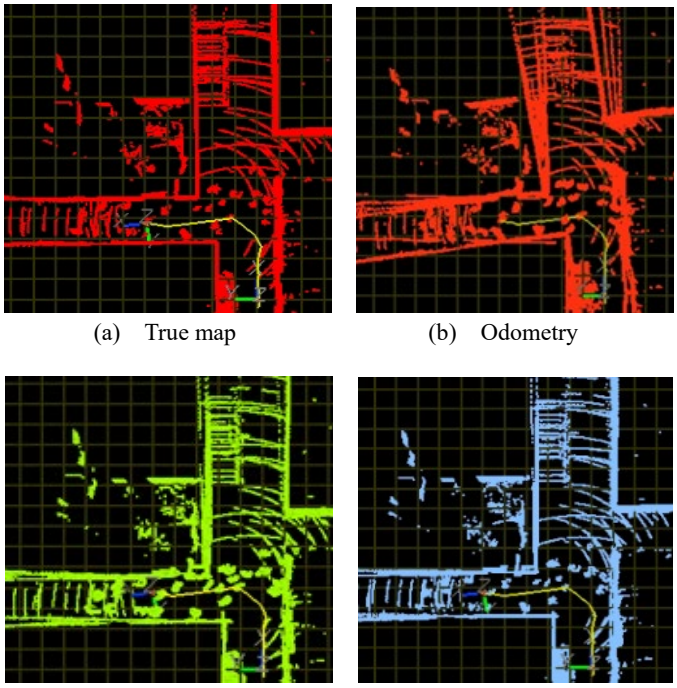
From the results shown in Tables 1 and 2, the proposed method with the human removal process has the smallest error and enables robust map construction. Map construction is performed separately based on this measurement data. The processing time for this process is approximately 10 to 20 seconds per frame. However, the processing time may vary slightly depending on the measurement data.

In Experiment II, the proposed method is highly robust to humans because not only stationary humans but also walking humans are present. Furthermore, Fig. 11 (a) and Fig. 12 (a) show that although many human point clouds are accumulated in the sensor data, the proposed method has considerably fewer human point clouds than the conventional method (Fig. 11(b) and Fig. 12 (b)). In other words, the proposed method is

suitable for operation in real environments where humans move frequently. The proposed method can be easily used as a preliminary information map by the human removal process.

On the other hand, in Fig. 9, the map by odometry appears to have the largest error. But as shown in Table 1, the error of odometry is smaller than that of the conventional ICP-SLAM. Because the robot's rotational movements are small and the percentage of points with large errors is quite small in Experiment I, this is probably the result of the overall smaller error.

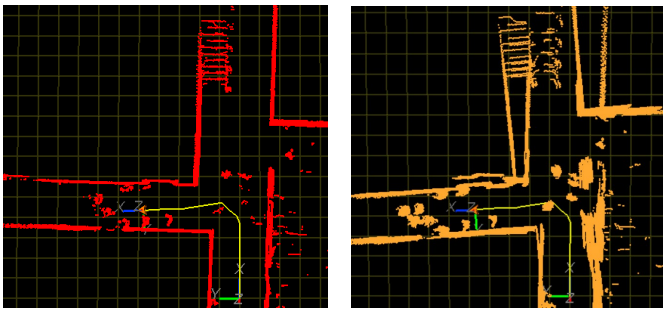
In the enlarged image of the proposed method shown in Fig. 11 and Fig. 12, a part of the human point clouds in the back of the image remains. Because a part of the human point clouds could not be removed due to the measurement range of the range image sensor. Concerning a walking human, there may be some misalignment between the point cloud $\boldsymbol{P_r}$ and point cloud $\boldsymbol{P_l}$. In such cases, some of the point clouds of humans may not be able to be matched and removed.



(a) True map (b) Odometry

(c) ICP-SLAM w/o human removal (d) ICP-SLAM w/ human removal

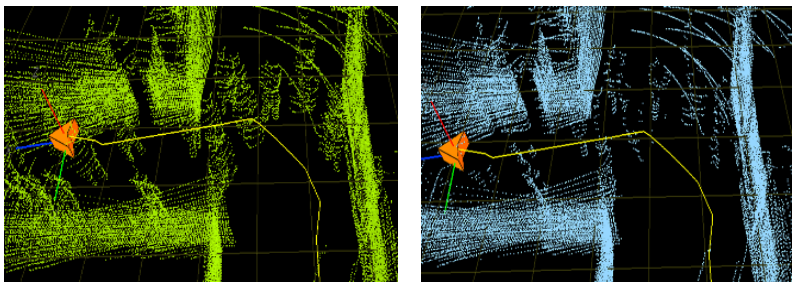
Fig.9 Map construction results by each method in Experiment I.



(a) True map (b) Odometry

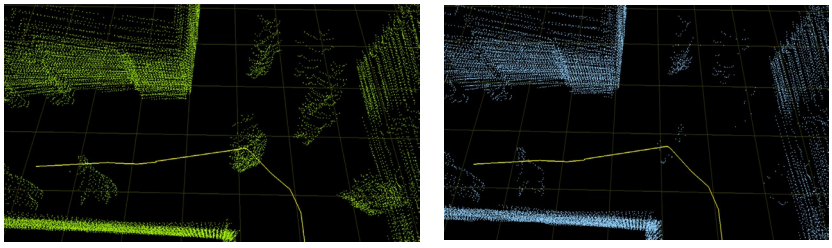
(c) ICP-SLAM w/o human removal (d) ICP-SLAM w/ human removal

Fig.10 Map construction results by each method in Experiment II.



(a) ICP-SLAM w/o human removal (b) ICP-SLAM w/ human removal

Fig.11 Enlarged map construction results in Experiment I.



(a) ICP-SLAM w/o human removal (b) ICP-SLAM w/ human removal

Fig.12 Enlarged map construction results in Experiment II.

Table. 1 Error of map information in Experiment I [m].

Odometry	0.462
ICP-SLAM w/o human removal	0.491
ICP-SLAM w/ human removal	0.269

Table. 2 Error of map information in Experiment II [m].

Odometry	0.478
ICP-SLAM w/o human removal	0.131
ICP-SLAM w/ human removal	0.097

6 Conclusions

In this paper, we proposed a method of applying the human removal process using object detection by YOLO to ICP-SLAM. By using this method to remove human data in indoor environments where humans are present, a highly accurate map construction was achieved.

For the verification of this method, a single RGB-D camera is used in this paper. But if we use four cameras, human detection can be performed in 360°.

In experiments, we verified the effectiveness of the proposed method for map construction and as preliminary information in a dynamic environment with many humans, which is quite close to the real environment. The results showed that the proposed method significantly outperforms conventional methods in terms of accuracy and robustness to humans in dynamic environments.

In addition, the effectiveness of the human removal process was verified using ICP-SLAM, but the proposed method can be used with other point cloud-based SLAMs as well.

On the other hand, the currently proposed method can only be used to identify humans among the YOLO identification objects. Therefore, in the future, we would like to expand the YOLO identification objects that can be used with this method to identify obstacles and stationary objects other than humans. This will enable the use of YOLO object detections as landmarks and the removal of obstacles to further improve the accuracy of map construction.

In addition, this method does not fully deal with the loss of sensor data caused by human occlusion. Therefore, we are currently working to extend this method to an active map construction system by path planning that interpolates missing areas.

References

1. M. Kimba, N. Machinaka, and Y. Kuroda, “Edge-Node Map-Based Localization without External Sensor Data,” In Proc. of the 1999 IEEE International Conference on Robotics and Automation (ICRA1999), pp. 1322-1328,1999.
2. R. Suzuki, Y. Ji, and K. Umeda, “Indoor SLAM based online observation probability using a hand-drawn map,” In Proc. of the 2022 IEEE/SICE International Symposium on System Integration (SII2022), pp. 695-698,2022.

3. P. J. Besl and N. D. McKay, “A Method for registration of 3-D shapes.” In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 239–256, 1992.
4. D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, “The Trimmed Iterative Closest Point algorithm,” In Proc. of IEEE International Conference on Pattern Recognition, pp.545–548, 2002.
5. G. Sébastien, and P. Xavier, “Multi-scale EM-ICP: A Fast and Robust Approach for Surface Registration,” In Proc. Of the European Conference on Computer Vision, pp.418–432, 2002.
6. S. Kaneko, T. Kondo, and A. Miyamoto, “Robust matching of 3D contours using iterative closest point algorithm improved by M-estimation,” Pattern Recognition, Vol.36, No.9, pp.2041–2047, 2003.
7. J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-Scale Direct Monocular SLAM,” In Proc. of European Conference on Computer Vision, pp.834– 849, 2014.
8. J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 3, pp. 611–625, 2017.
9. R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, “ORB-SLAM: A Versatile and Accurate Monocular SLAM System,” IEEE Transactions on Robotics, Vol.31, No.5, pp.1147–1163, 2015.
10. X. Chen et al., “SuMa++: Efficient LiDAR-based Semantic SLAM, ” In IROS, 2019.
11. W. Liu et al., “DLOAM: Real-time and Robust LiDAR SLAM System Based on CNN in Dynamic Urban Environments, ” IEEE Open Journal of Intelligent Transportation Systems, 2021.
12. Q. Lie et al., “LO-Net: Deep Real-time Lidar Odometry, ” In CVPR, 2019.
13. P. Pfreundschuh et al., “Dynamic Object Aware LiDAR SLAM based on Automatic Generation of Training Data, ” In ICRA, 2021.
14. E. Palazzolo et al., “ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals, ” In IROS, 2019.
15. R. Scona et al., “StaticFusion: Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments, ” In ICRA, 2018.
16. V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” In Proc. of IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No.12, pp. 2481-2495, 2017.
17. C. Yu, et al., “DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments,” In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1168-1174, 2018.
18. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition , pp. 779-788, 2016.
19. K. He, G. Gkioxari, P. Doll, and R. Girshick, “Mask R-CNN,” in Proc. of the IEEE Conf. International Conference on Computer Vision (ICCV), pp. 2980-2988, 2017.
20. J. C. V. Soares, M. Gattass, and M. A. Meggiolaro, “Visual SLAM in Human Populated Environments: Exploring the Trade-off between Accuracy and Speed of YOLO and Mask R-CNN,” International Conference on Advanced Robotics (ICAR), pp. 135-140, 2019.
21. B. Bescos, J. M. Fácil, J. Civera, and J. Neira, “DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes, ” In Proc. of IEEE Robotics and Automation Letters, Vol. 3, No. 4, pp. 4076-4083, 2018.
22. J. L. Bentley, “K-d Trees for Semidynamic Point Sets,” In Proc. of the 6th annual Symposium on Computational Geometry (SCG), pp. 187-197, 1990.
23. A. Dhall, K. Chelani, V. Radhakrishnan and K. M. Krishna, “LiDAR-camera calibration using 3D-3D point correspondences,” In arXiv preprint arXiv:1705.09785, 2017.