

Paper:

Development of Autonomous Navigation System Using 3D Map with Geometric and Semantic Information

Yoshihiro Aotani, Takashi Ienaga, Noriaki Machinaka, Yudai Sadakuni,
Ryota Yamazaki, Yuki Hosoda, Ryota Sawahashi, and Yoji Kuroda

Meiji University

1-1-1 Higashimita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan

E-mail: {ce62001, ykuroda}@meiji.ac.jp

[Received March 1, 2017; accepted June 22, 2017]

This paper presents an autonomous navigation system. Our system is based on an accurate 3D map, which includes “geometric information” (e.g., curb, wall, street tree) and “semantic information” (e.g., sidewalk, roadway, crosswalk) extracted by environmental recognition. By using the semantic map, we can obtain the suitable area to keep away from undesired places. Furthermore, by comparing the map with real-time 3D geometric information from LIDAR, we obtain the robot position. To show the effectiveness of our system, we conduct a 3D semantic map construction experiment and driving test. The experiment results show that the proposed system enables accurate and highly reproducible localization and stable autonomous mobility.

Keywords: autonomous navigation robot, 3D semantic map, localization

1. Introduction

In recent years, as robots have attracted focus, the development of robots that can substitute for human work, such as delivery and security, has been active. Situations such as crossing and running along narrow roads are expected in the city environments where these robots will work. For a robot to safely run in such an environment, an autonomous navigation system that integrates elements such as environmental recognition, localization, and path plan, is essential.

In general, conventional navigation systems create a geometric map with obstacle information, such as buildings and trees, within the driving environment from the geometric information obtained by LIDAR to generate a stable localization. With such a navigation system, Burgard et al. succeeded in long-distance autonomous travel of 3.2 km, within an urban environment where people live [1]. However, that is not sufficient for coexistence with humans. A navigation system that follows the rules of human society using visual information such as cross walks and signs is desired.

With the recent rise of deep learning, there has been

remarkable development in recognition technology based on visual information; it is now possible to know not only what object is in an image [2], but also what kind of environment it is in [3].

We suggest an autonomous navigation system that uses a 3D map, which is created in advance. The map is created by “geometric information” from the point cloud, which is acquired from LIDAR, and “semantic information” from deep learning based on visual information.

This system implements a travel ability suitable for an environment by localization, comparing the geometric information obtained from real-time data and from the preliminary map. At the same time, it generates a path plan from the runnable field at its point.

2. System Architecture

2.1. Autonomous Mobile Robot INFANT

In this section, we will describe the structure of the Integrated Foundations for Advanced Navigation Technology (**Fig. 1**) (INFANT), which we are developing. INFANT has differential two-wheel drive and a locker bogie structure, with a size of 0.60 m (W) × 0.85 m (D) × 1.45 m (H). We use USB to connect the PC to control the motors.

The sensors equipped are LIDAR (HDL-32e), a stereo camera (ZED), an AMU (Silicon Sensing Systems AMU-1802BR), and a wheel encoder. We use two laptop computers (Intel Core i7-3630 QM 2.40 GHz, 15.6 GB of RAM) to process the sensors and calculate the autonomous navigation.

2.2. Autonomous Navigation System

The schematic diagram of the autonomous navigation system is shown in **Fig. 2**. In this system, the 3D map with geometric and semantic information is constructed from sensor information acquired before hand. With this, the robot not only avoids obstacles but also follows the human rules such as crossing the road. The “geometric information” here shows obstacles that are physically unable to travel, and “semantic information” shows the information needed to follow the human rules. By com-



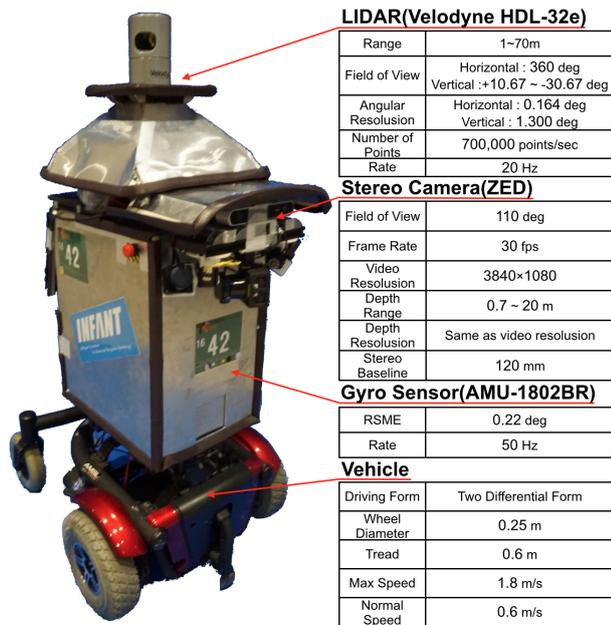


Fig. 1. INFANT.

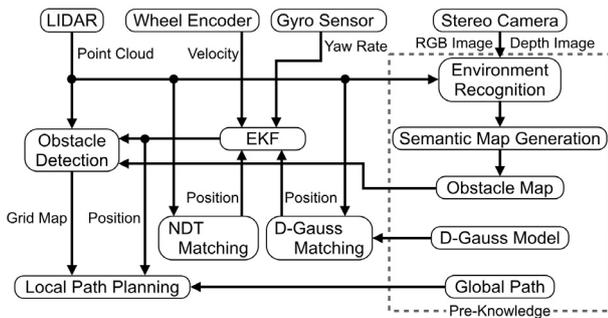


Fig. 2. System architecture.

binning a 3D map that includes such information and the information acquired by the sensors in real-time, it estimates its position and calculates its path plans to make a stable travel that fits the environment.

3. 3 Dimensional Map Construction

In this section, we will describe the construction of a 3D map, which includes the geometric and semantic information. For the construction of the geometric map, we used our own 3D geometric map construction method [4]. In this method, we construct a map based on graph theory. For the construction of the graph, we used Iterative Closest Point (ICP) [5], and for the optimization of this graph we used GraphSLAM [6, 7]. In addition, we constructed a highly precise 3D geometric map using pedestrian detection [8], which used the LIDAR point cloud; we removed any point clouds that were detected as a pedestrian.

In our system, by attaching semantic information based on environment recognition, as described later, a 3D map

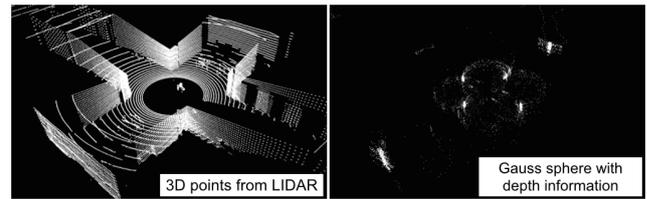


Fig. 3. LIDAR points and gauss sphere with depth information.

that includes geometric and semantic information is constructed.

4. Localization

This section describes the localization of this system. Generally, error accumulates in localization with gyro sensors and odometry via the slips of a gap and the wheel of the azimuth. Therefore, in this system, we estimated the azimuth and pose in the map coordinated by the Depth Gauss Sphere (D-Gauss Sphere) matching and Normal Distribution Transform (NDT) matching to minimize accumulation error. We generated stable localization by integrating the estimated results using the Extended Kalman Filter (EKF) [9].

4.1. Depth-Gauss Sphere Matching

The D-Gauss Sphere, which is provided by the Gaussian representation on the point cloud with normal information, can treat one face (e.g., wall surface) as a set of one point. Therefore, we could reduce the number of point clouds being processed and generate an alignment of a point cloud without impairing real-time performance. In addition, comparing the D-Gauss Sphere generated from real-time LIDAR data and the model of the D-Gauss Sphere placed on the 3D map, which is mentioned above, will create an azimuth and pose estimate in the map coordinates. Fig. 3 shows the 3D point cloud from the LIDAR and the D-Gauss Sphere. In the following sections, we will describe the principal component extraction [10], azimuth estimation, and pose estimation with the D-Gauss Sphere.

4.1.1. Extraction of Principal Component

Shimizu et al. [10] carried out clustering in the D-Gauss Sphere and determined it as principal components. However, this method has problems that are susceptible to the number of point clouds: for the plane that is near the sensor, it is easy to take a principal component, and for the far-off plane, it is difficult to take a principal component. It is desirable to constitute principal components by a far-off plane because the plane that is near the sensor easily detects noises by dynamic objects such as pedestrians and easily influences the result of the localization. Therefore, in this paper, we will calculate the *cost* based on Eq. (1)

for each point cloud constituting a D-Gauss Sphere.

$$cost = \lfloor d \rfloor = \max\{n \in \mathbb{Z} | n \leq d\} \quad \dots \quad (1)$$

where d expresses the distance to each point. Z and n are natural namers. In Eq. (1), the higher the $cost$, the farther apart the plane and sensor is. Afterward, we reprinted each point to obtain the value of $cost$ and performed a clustering based on Euclidian distance. When the number of the point cloud constituting each cluster exceeds an established value, we assume it as one principal component.

4.1.2. Azimuth Estimation

In this section, we will describe the azimuth estimation. First, we apply the extraction process described in the previous section to the D-Gauss Sphere of a model located on a 3D map and a query that we generated in real-time.

Next, we will compare the principal components of the model and the query and then calculate the relative azimuths for them one by one. Fig. 4 is a conceptual diagram. The resemblance degree calculation uses the following Eq. (2).

$$s_{ij} = \frac{\vec{m}_i \cdot \vec{q}_j}{|\vec{m}_i| |\vec{q}_j|} \quad \dots \quad (2)$$

when \vec{m}_i, \vec{q}_j express the vectors from the sensor origin of the model and the query to the principal components, respectively, and s_{ij} expresses the degree of similarity. We calculate the degree of similarity for the principal component vector in the model and the query according to this expression and perform a correspondence charge account for the pair with the highest similarity. After all of the associations, we calculate the angle between the two vectors, the model and the query. We then obtain the relative angle for the model by taking their average. In addition, we do not associate them when the similarity is less than the established value.

4.1.3. Position Estimation

The position estimation is calculated by the principal components for the D-Gauss Sphere of the model and a query similar to that described in the previous section. We obtain the robot's position with the principal components. In other words, in the relative position calculation of the model and the query, we can only use certain principal components in positioning to the normal direction of the principal components. Therefore, the position estimation of the D-Gauss Sphere is executed to the axis of the normal direction for the principal components from the sensor origin. We used the ICP algorithm to align the principal components of the model and the query on the axis and calculate each axis' position movement vector of the model and the query $\vec{\epsilon}_x, \vec{\epsilon}_y$. When the number of principal components on the axis is less than three points, we will calculate the position movement vector using Nearest Neighbor Search (NNS). Fig. 4 shows the conception diagram of the position estimation. The position movement

vector when using NNS is calculated by Eq. (3).

$$\vec{\epsilon}_x = \frac{\sum \vec{\epsilon}_{xi}}{N_x}, \vec{\epsilon}_y = \frac{\sum \vec{\epsilon}_{yj}}{N_y} \quad \dots \quad (3)$$

where, N_x, N_y is the number of principal components in each axis. $\vec{\epsilon}_{xi}$ and $\vec{\epsilon}_{yj}$ represent the displacement vector associated using NNS in each axis. Finally, we can obtain the relative movement vector $\vec{\epsilon} (\vec{\epsilon}_x, \vec{\epsilon}_y)$ for the placement position of the model by adding each vector.

4.2. Normal Distributions Transform Matching

The D-Gauss Sphere matching estimates the robot's pose by the normal vector of the location (e.g., wall surface). Thus, it does not work efficiently in an environment with no planes. Therefore, we used NDT [11] matching for localization using the prior 3D geometric map and real-time data. As shown in Fig. 5, NDT is a kind of scan matching method that uses voxel grids. The voxel grid describes the point cloud as normal distributions and associates two different point clouds using normal distributions.

To lower the computational cost when driving, we restricted the points for matching to fences, walls, curbs, etc., and limited the scanning range to 20 m². At this time, with the real-time data, we cut out the point cloud to a range of approximately 20 m² from a 3D map in compliance with the position and orientation estimated by the EKF.

4.3. Extended Kalman Filter

EKF [9] is used to integrate the localization results and the gyro odometry. In EKF, the state s is represented by the probability distribution, and the estimation result is obtained by repeating the two-step process of prediction and observation update. This system predicts the position and the orientation.

4.3.1. Prediction Step

In the prediction step, as following Eq. (4), the state \bar{s}_t is predicted based on the nonlinear function $g(u_t, s_{t-1})$.

$$\bar{s}_t = g(u_t, s_{t-1}) + \epsilon_t \quad \dots \quad (4)$$

where, u_t is the motion model representing the linear speed by the wheel encoder and the swing speed by the AMU at time $(t-1, t]$, s_{t-1} is the state at time $t-1$, and ϵ_t is the observation noise of the sensor.

4.3.2. Measurement Updating Step

In the observation updating step, the estimated value using the D-Gauss Sphere or the NDT is compared with the estimated value in the prediction step, and state s_t is updated based on Eq. (5).

$$s_t = \bar{s}_t + K_t(z_t - \bar{s}_t) \quad \dots \quad (5)$$

where \bar{s}_t is the predicted state, K_t is the Kalman gain, and z_t is the observation model estimated by the D-Gauss Sphere and NDT.

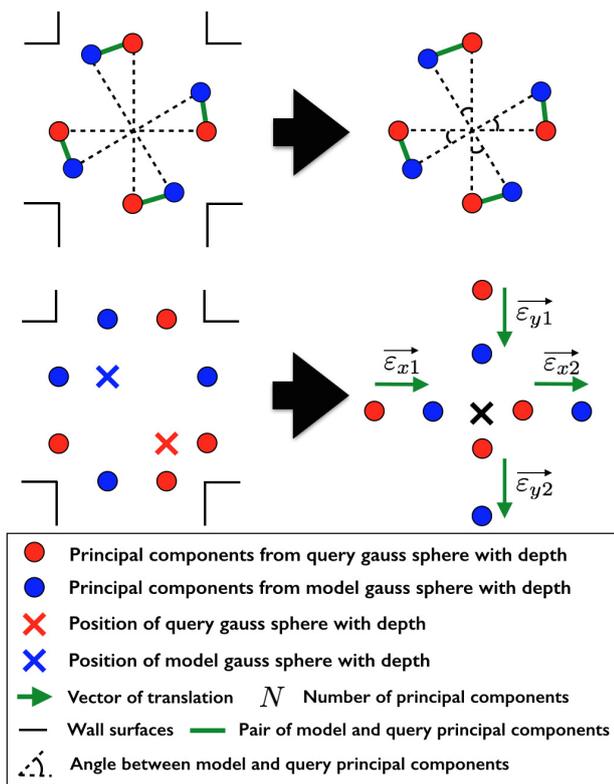


Fig. 4. Estimating relative position by matching principal components.

5. Environment Recognition

This section describes the environmental recognition method. For a robot to safely travel autonomously, obstacles that cannot be passed through (e.g., walls, curbs) must be avoided. However, that is not sufficient in human living environments. A robot must be able to act according to human rules such as lawns and pedestrian crosswalks. Therefore, the former is recognized with LIDAR, which has high ranging accuracy, and the latter is recognized from the images acquired by the stereo camera. By performing a 3D restoration from these sources, the position of the object is recognizable. In addition, to delete unnecessary dynamic objects (e.g., pedestrians, animals) while constructing 3D maps, recognition is performed using geometric information acquired from LIDAR.

5.1. Obstacle Recognition

For obstacles with a large height change (e.g., walls, fences), we use the conventional method of obstacle detection named Min-Max [12]. However, in outdoor conditions, it is difficult to recognize obstacles with a small height change, such as curbs, owing to noises generated by the vehicle’s body vibration from the waves of the road surface. We then estimate the curvature σ based on the variance-covariance matrix obtained by applying principal component analysis [13] to a neighborhood point cloud around a certain point, thereby detects a small height change.

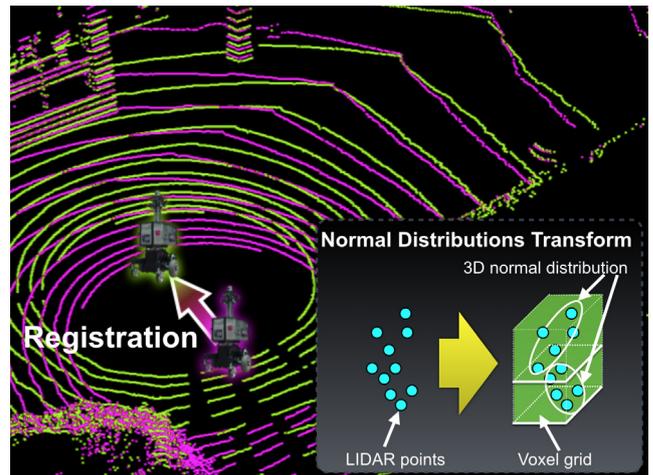


Fig. 5. Example of normal distribution transform.

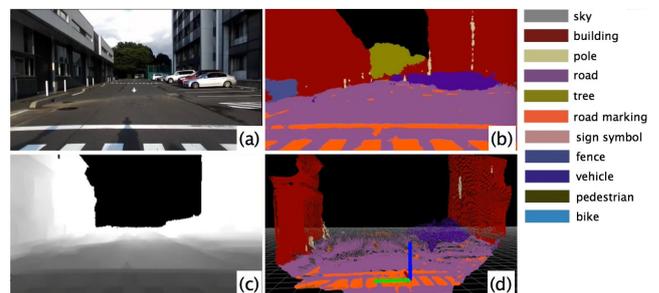


Fig. 6. Environment recognition. (a) RGB image, (b) classification result, (c) depth image, (d) point cloud with label.

5.2. Environmental Recognition

When using only information obtained from LIDAR, it is difficult to divide it into a travelable area and an impassible area by areas with different material such as lawns and pedestrian crosswalks. Therefore, we recognized a driving environment by the obtained visual information from the stereo camera to place semantic information in the point cloud. We applied the method proposed by Badrinarayanan et al. [14] for image classifications. This method is an improved version of the Convolutional Neural Network (CNN), which can perform multi-class classification on a pixel basis. To classify the input images by the softmax function, we execute up-samplings to the feature map obtained by performing convolution and pooling on the input images until the size becomes the same as the input image. For our system, as shown in Fig. 6, we classified 11 classes for example, “Road,” “Road Marking,” and “Tree” related to the outdoor driving environment.

For the learning of CNN, the CamVid dataset [15] provided by Cambridge University was used.

Furthermore, to grasp the position of each pixel in 3D space, an image segmented based on the depth image was three-dimensionally restored. The conversion of points (u, v) on the image to points (X, Y, Z) on the space is per-

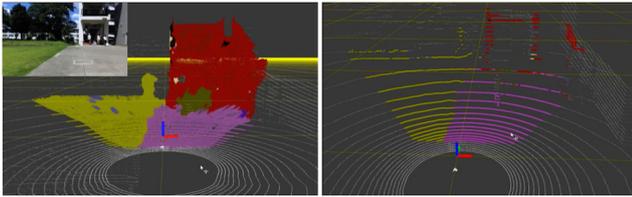


Fig. 7. Associating semantic information with 3D point cloud from LIDAR (left: raw data right: 3D point cloud with semantic information).

formed based on Eq. (6).

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6)$$

where f is the focal length, c_x and c_y are the optical centers, R_{ij} is the rotation matrix, and T_i is the pose vector. An application example of this method is shown in **Fig. 6**.

However, it is difficult for the stereo camera to remain focused on grasping the position with high stability. Therefore, we will associate the point cloud obtained by the above method with the point cloud acquired by LIDAR.

First, in the point cloud of the stereo camera, a point corresponding to an arbitrary point of LIDAR is searched by the k -Nearest Neighbor Method. At that time, as Eq. (7), we calculate the *score* sum of each color by weight, according to the Euclidean distance from the point of LIDAR. We then give the highest *score* color to each point of LIDAR.

$$score = \frac{1}{\sqrt{(S_x - L_x)^2 + (S_y - L_y)^2 + (S_z - L_z)^2}} \quad (7)$$

where S represents the 3D point of the stereo camera, and L represents the 3D point of LIDAR. **Fig. 7** shows the result of giving semantic information of the stereo camera to the point cloud of LIDAR at 1 scan.

5.3. Local Obstacle Map Construction

We used a two-dimensional grid map as an obstacle map that showed the travelable area to be used for path plans. Integrating the 3D map and real-time sensor information identifies the travelable area and compresses the space into two dimensions. Regarding the real-time data, the travelable area is estimated by the obstacle detection mentioned above. On the other hand, for the 3D map, we set the travelable area based on obstacle detections and the “road” according to semantic information. From this, the map supplements the dead angle of the sensor. In addition, the robot detects small steps and curbstones, which are difficult to detect as obstacles if we use only the real-time sensor data. Furthermore, by using the semantic information contained in the 3D map, it is also possible to act according to human rules such as crossing at a crosswalk like a human being.

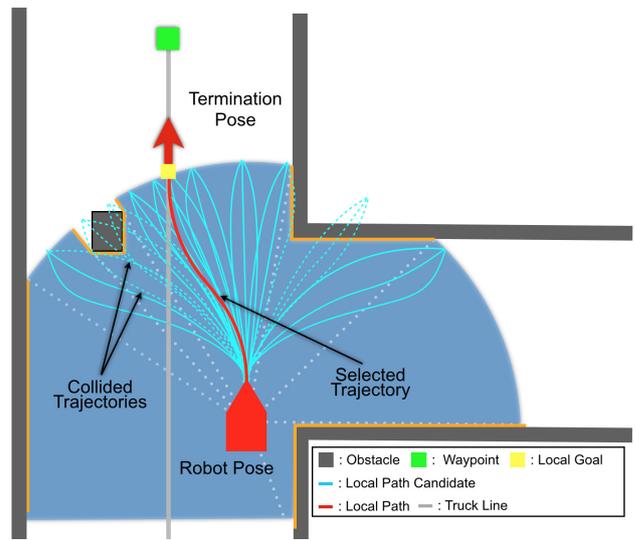


Fig. 8. Local path planning.

6. Path Planning

In this section, we will describe global path planning and local path planning, which are the guidelines by which the robot will travel. A waypoint is arranged as a global path in advance along the travel route on the 3D map. By generating a local path according to the route based on the obstacle map mentioned above, efficient and stable operation is enabled.

In the local path planning, the trajectory generation is performed using the methods from [16], [17], and [18], proposed by Feruguson, Haward, et al., which consider the motion model of the robot. At each point on the orbit, route planning is performed such that the robot follows the state shown in Eq. (8). The motion model parameters considered in the process of calculating the trajectory are the maximum acceleration, maximum curvature of the trackable trajectory, and maximum curvature change.

$$\mathbf{x}_t = [x_t \ y_t \ \theta_t \ \kappa_t \ v_t] \dots \dots \dots (8)$$

where \mathbf{x}_t represents the state of the robot at the time, and x_t , y_t , and θ_t are the position and posture of the robot, respectively. κ_t is the orbital curvature, and v_t is the forward speed.

By minimizing the costs calculated in Eqs. (9), (10), and (11), the optimal trajectory is selected from the trajectory candidates, which are generated considering the motion model.

$$P = \sqrt{\delta x^2 + \delta y^2} \dots \dots \dots (9)$$

$$H = |\Delta\theta| \dots \dots \dots (10)$$

$$Cost = aP + bH \dots \dots \dots (11)$$

where P and H are the variation between the current pose and the terminal attitude. a and b are arbitrary constants. The outline is shown in **Fig. 8**.



Fig. 9. Driving test routes.

7. Experimental Result

To show the usefulness of our system, we conducted experiments at Meiji University's Ikuta Campus, including 3D map construction, localization, environment recognition, and path planning. In addition, the autonomous travel experiment, as shown in Fig. 9, which was performed on a course 1282 m long, including both indoor and outdoor environments. The outdoor operating speed was set to 0.7 m/s; indoors, as a consideration to pedestrians, we set the speed to 0.4 m/s.

7.1. 3D Semantic Map Construction

Figure 10 shows the result of the 3D map construction for the entire Ikuta Campus from the proposed method. We can see that this is constructed from the semantic information, such as roads (purple), buildings (red), road markers (orange), and lawns (green), which are imparted for the point cloud in Fig. 6. In addition, Fig. 10 shows a diagram comparing the proposed method (A) and the actual environment (B). From this, we can see that the semantic information is correctly assigned.

7.2. Localization

In this section, we evaluated the difference in the localization with pose and position. The comparison results are the results from both the NDT and D-Gauss Sphere matching.

The experiment environment is shown in Fig. 11; an indoor environment (A) with walls in various directions, an outdoor environment (B) surrounded by trees and flowers, and another outdoor environment (C) made of parallel buildings. In terms of azimuth estimation, we evaluated the difference in direction estimation when turning 360° at a speed of 0.2 rad/s. In terms of position estimation, we evaluated the difference between the traveled distance and the actual distance when traveling 10 m forward at a speed of 0.5 m/s. In each environment, we performed 5 trials and calculated each mean error and standard deviation. The results of the azimuth estimation and position estimation experiment are shown in Tables 1 and 2. From these results, we know that in all environments (A), (B), and (C), by using the proposed method, the azimuth estimation had approximately 1° of error, and the position estimation had approximately 0.2 m of error.

This is because in each environment, the weight of the odometry in EKF, D-Gauss, and NDT is appropriately calculated. This occurs when a low trustable estimation was made because a valid feature could not be obtained.

7.3. Environmental Recognition and Path Planning

Figure 12 shows an example of obstacle recognition. The red points mark the obstacles detected by the Min-Max method, which uses the objects' height information, and the blue points mark the obstacles detected by curvature methods, which uses the results from the principal component analysis to a neighborhood point cloud around

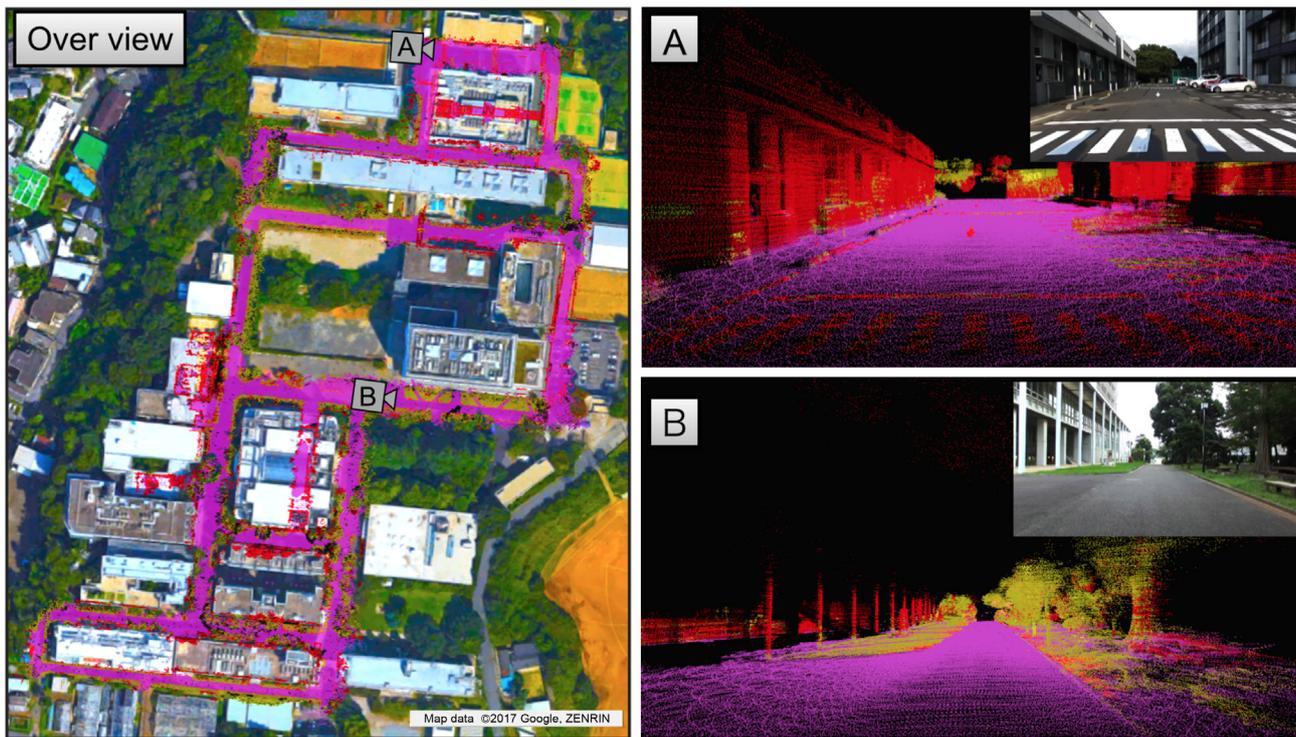


Fig. 10. Details of 3D semantic map.



Fig. 11. Experiment environment of localization.

Table 1. Azimuth estimate experimental result.

Unit : [deg]

Experiment Environment	Proposed Method	D-Gauss	NDT
A	0.559 (SD=0.622)	1.083 (SD=5.423)	1.812 (SD=7.816)
B	1.369 (SD=0.036)	2.238 (SD=0.754)	3.018 (SD=1.265)
C	0.487 (SD=0.346)	0.499 (SD=0.388)	11.31 (SD=12.24)

Table 2. Position estimate experimental result.

Unit : [deg]

Experiment Environment	Proposed Method	D-Gauss	NDT
A	0.230 (SD=0.087)	0.292 (SD=0.091)	0.171 (SD=0.062)
B	0.197 (SD=0.062)	0.427 (SD=0.219)	0.151 (SD=0.128)
C	0.156 (SD=0.115)	0.337 (SD=0.048)	0.153 (SD=0.067)

a certain point. The green points mark the detected obstacles with prior semantic information. Fig. 13 shows an example of the path plan avoiding obstacles. We found that an optimal path was generated even if there were obstacles on a route that was set in advance. This path follows the prior path but generates another path around the obstacle so that the robot will avoid it.

7.4. Driving Test

In this experiment, we ran the course shown in Fig. 9 five times.

Figure 14 shows the robot’s motion path for five laps. In addition, Table 3 shows the distance of the start and end points in all five running experiments, and Fig. 15 shows the robot’s state at its end point. From Fig. 14, we could verify that the robot could run stably on the same course on every lap. As shown in Table 3, even if the robot traveled autonomously over 1 km, with our system,

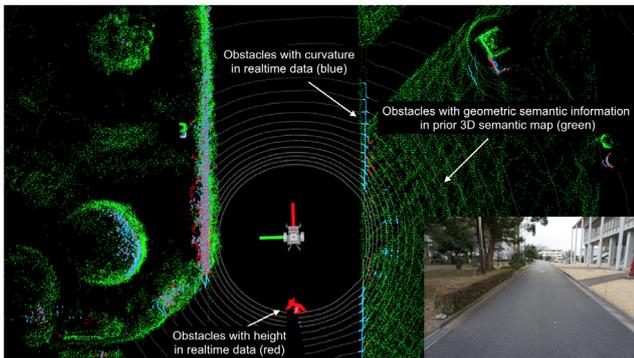


Fig. 12. Example of the obstacle recognition.

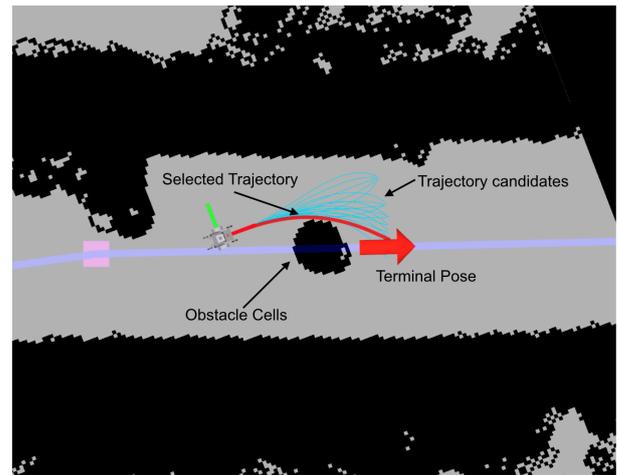


Fig. 13. An example of path planning to avoid obstacles.

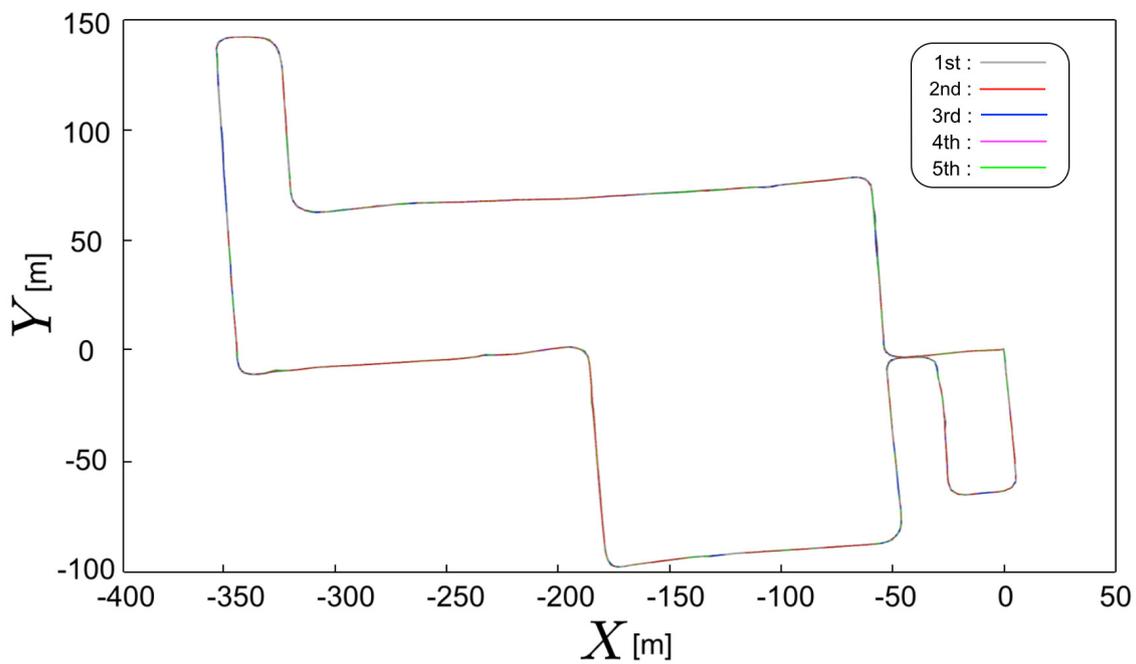


Fig. 14. Trajectory by our navigation system.

Table 3. The distance between the start and the goal.
Unit : [deg]

1st	2nd	3rd	4th	5th	Average
0.422	0.336	0.355	0.358	0.325	0.359

the robot could return to the starting point within an average of 0.359 m.

8. Conclusion and Future Work

In this paper, we proposed a navigation system using a 3D map, which includes geometric information and se-

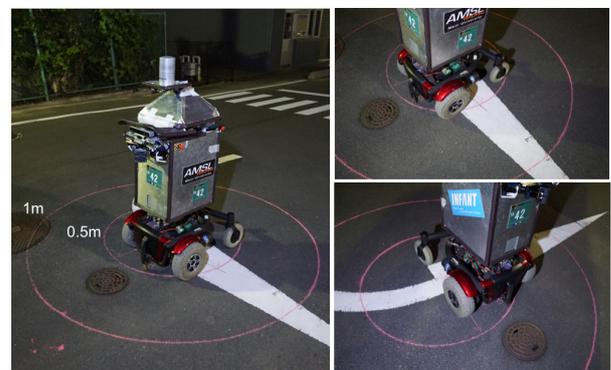


Fig. 15. An example of the robot goal position.

semantic information, to enable stable autonomous driving for a robot under urban environments. The experimental results show that our system is capable of repeatable localization and autonomous navigation by utilizing information such as lawns and crosswalks, which are difficult to grasp with geometric information.

However, because our system relies on a 3D map, which must be created beforehand, it is difficult to autonomously travel in places where the robot first visits. Therefore, localization and the environment recognition based on real-time sensor data will be a future problem.

References:

- [1] R. Kümmerle et al., "Autonomous robot navigation in highly populated pedestrian zones," *J. of Field Robotics*, Vol.32, No.4, pp. 565-589, 2015.
- [2] A. Krizhevsky, S. Ilya, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012.
- [3] B. Zhou et al., "Learning deep features for scene recognition using places database," *Advances in neural information processing systems*, 2014.
- [4] T. Hagiwara, D. Katakura et al., "Development of a stable navigation system in the known environment by using autonomous 3D map construction," *The 16th SICE System Integration Division Annual Conf.*, pp. 811-814, 2015 (in Japanese).
- [5] S. Rusiniewicz and M. Levoy, "Efficient variants of the ICP algorithm," *Proc. of the Int. Conf. on 3-D Digital Imaging and Modeling (3DIM)*, pp. 145-152, 2001.
- [6] G. Grisetti, R. Kümmerle, H. Strasdat, and K. Konolige, "g2o: A general Framework for (Hyper) Graph Optimization," *Technical report*, 2011.
- [7] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based SLAM," *IEEE Intelligent Transportation Systems Magazine*, Vol.2, pp. 31-43, 2010.
- [8] D. Katakura and Y. Kuroda, "3D LIDAR-based Detection and Tracking of Multiple human," *The 21st Robotics Symposia, Nagasaki*, 2016 (in Japanese).
- [9] S. Thrun et al., "Probabilistic Robotics," *MIT Press*, Vol.1, pp. 201-215, 2005.
- [10] S. Shimizu and Y. Kuroda, "High-Speed Registration of Point Clouds," *The 19th Robotics Symposia, Hyogo*, 2014 (in Japanese).
- [11] P. Biber and W. Straber, "The normal distributions transform: A new approach to laser scan matching," *Proc. 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2743-2748, 2003.
- [12] S. Thrun et al., "Stanley: The robot that won the DARPA Grand Challenge," *Proc. of the JFR*, Vol.23, No.9, pp. 661-692, 2006.
- [13] F. Neuhaus et al., "Terrain drivability analysis in 3D laser range data for autonomous robot navigation in unstructured environments," *Emerging Technologies and Factory Automation (ETFA)*, pp. 1-4, 2009.
- [14] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *arXiv preprint arXiv:1511.00561*, 2015.
- [15] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, Vol.30, No.2, pp. 88-97, 2009.
- [16] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments," *Proc. of the JFR*, Vol.25, pp. 939-960, 2008.
- [17] T. M. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *Proc. of the IJRR*, Vol.26, pp. 141-166, 2007.
- [18] T. M. Howard and C. J. Green, "State space sampling of feasible motions for high-performance mobile robot navigation in complex environments," *Proc. of the JFR*, Vol.25, pp. 325-345, 2008.



Name:
Yoshihiro Aotani

Affiliation:
Master Student, Department of Mechanical Engineering, School of Science and Technology, Meiji University

Address:
1-1-1 Higashimita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan

Brief Biographical History:
2016 Received the Bachelor degree in Mechanical Engineering from Meiji University



Name:
Takashi Ienaga

Affiliation:
Master Student, Department of Mechanical Engineering, School of Science and Technology, Meiji University

Address:
1-1-1 Higashimita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan

Brief Biographical History:
2016 Received the Bachelor degree in Mechanical Engineering from Meiji University



Name:
Noriaki Machinaka

Affiliation:
Bachelor Student, Department of Mechanical Engineering, School of Science and Technology, Meiji University

Address:
1-1-1 Higashimita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan



Name:
Yudai Sadakuni

Affiliation:
Bachelor Student, Department of Mechanical Engineering, School of Science and Technology, Meiji University

Address:
1-1-1 Higashimita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan



Name:
Ryota Yamazaki

Affiliation:
Bachelor Student, Department of Mechanical Engineering, School of Science and Technology, Meiji University

Address:
1-1-1 Higashimita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan



Name:
Yuki Hosoda

Affiliation:
Bachelor Student, Department of Mechanical Engineering, School of Science and Technology, Meiji University

Address:
1-1-1 Higashimita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan



Name:
Ryota Sawahashi

Affiliation:
Bachelor Student, Department of Mechanical Engineering, School of Science and Technology, Meiji University

Address:
1-1-1 Higashimita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan



Name:
Yoji Kuroda

Affiliation:
Professor, Department of Mechanical Engineering, School of Science and Technology, Meiji University

Address:
1-1-1 Higashimita, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan

Brief Biographical History:

1994 Received Dr. of Engineering from The University of Tokyo
1995- Assistant Professor, Meiji University
2000- Associate Professor, Department of Mechanical Engineering, Meiji University
2004-2005 Visiting Professor, Department of Mechanical Engineering, Massachusetts Institute of Technology
2013- Professor, Department of Mechanical Engineering, Meiji University

Main Works:

• "6DOF Localization Using Unscented Kalman Filter for Planetary Rovers," *Advanced Robotics*, Vol.24, No.8-9, pp. 1199-1218, 2010.

Membership in Academic Societies:

- The Japan Society of Mechanical Engineers (JSME)
 - The Robotics Society of Japan (RSJ)
 - The Japan Society of Naval Architects and Ocean Engineers (JSNAOE)
 - The Institute of Electrical and Electronics Engineers (IEEE)
-