https://doi.org/10.20965/jrm.2015.p0401

Paper:

Integrated Autonomous Navigation System and Automatic Large Scale Three Dimensional Map Construction

Yusuke Fujino, Kentaro Kiuchi, Shogo Shimizu, Takayuki Yokota, and Yoji Kuroda

Department of Mechanical Engineering, Meiji University
1-1-1 Higashimita, Tama-ku, Kawasaki 214-8571, Japan
E-mail: amsl-racing@meiji.ac.jp
[Received March 16, 2015; accepted June 6, 2015]

The method we propose for constructing a large three-dimensional (3D) map uses an autonomous mobile robot whose navigation system enables the map to be constructed. Maps are vital to autonomous navigation, but constructing and updating them while ensuring that they are accurate is challenging because the navigation system usually requires accurate maps. We propose a navigation system that explores areas not explored before. The proposed system mainly uses LIDARs for determining its own position – a process known as localization – or the environment around the robot – a process known as environment recognition – for creating local maps and for avoiding mobile objects – a process known as motion planning. We constructed a detailed 3D map automatically using autonomous driving data to improve navigation accuracy without increasing the operator's workload, confirming the feasibility of the proposed method through experiments.

Keywords: autonomous navigation robot, human recognition, automatic three dimensional map construction

1. Introduction

Research and development on mobile robots is proceeding worldwide, focusing on robots that move autonomously in urban environments. The objective of such research is to have robots replace human beings in work such as delivery, security, and driving in urban environments. Many difficulties arise for mobile robots navigating autonomously, such as static obstacles – buildings and roadsides – and dynamic obstacles – pedestrians and cars –, and high-rise buildings is the reason why finding out its own position by using global positioning systems (GPS) is very difficult. These problems are why detailed maps are often made by hand a priori using light detection and ranging (LIDAR), cameras, or GPS [1–3]. Navigation systems based on detailed a priori map information could increase autonomous mobile robot efficiency and safety because these let the system know where robots are and what they see. Putting such navigation systems to practical use requires time and labor by operators and robots in advance to determine locations. Changes in building or road locations, for example, require even more work to update map information [4]. Using autonomous mobile robots in urban environments requires that we overcome difficulties such as the above. To do this, we set a goal of making an autonomous navigation system that guesses and judges things using general concepts in the same way as human beings do [5]. Specifically, maps including the place information are helpful to understanding the environment. We hold that autonomous navigation systems without precise prior information, using map construction and autonomous renewal by robots alone, use constructed maps. In such cases, map information should include static obstacles and considering that most moving obstacles in urban environments are human beings then human beings should be recognized as such. Because of this, autonomous mobile robots move more efficiently, safely, and smoothly in areas which robots have already been – without repeating or increasing operators' work. We propose a multiple flexible autonomous navigation system that does not require prior detailed information or automatic large three-dimensional (3D) map construction or renewal using sensor data and robot's pose. Results of our experiments demonstrate the validity of our proposed method.

2. System Architecture

2.1. Autonomous Mobile Robot INFANT

The configuration of our mobile robot, known as Integrated Foundations for Advanced Navigation Technology (INFANT), is detailed in [6]. The HDL-32e LIDAR is used for localization and environment recognition. We also used the Silicon Sensing Systems AMU-1802BR to determine the pose and a wheel encoder to obtain travel speed. The robot has a differential GPS (MicroStrain) to determine position but only for the operator to determine the robot's position. Sensor processing and autonomous mobile calculation are done by a laptop computer on the robot using an Intel® Core i7-c3630 QM 2.40 GHz and RAM 3.8 GB. PCs, the motor controller and sensors communicate over the Ethernet using TCP/IP.

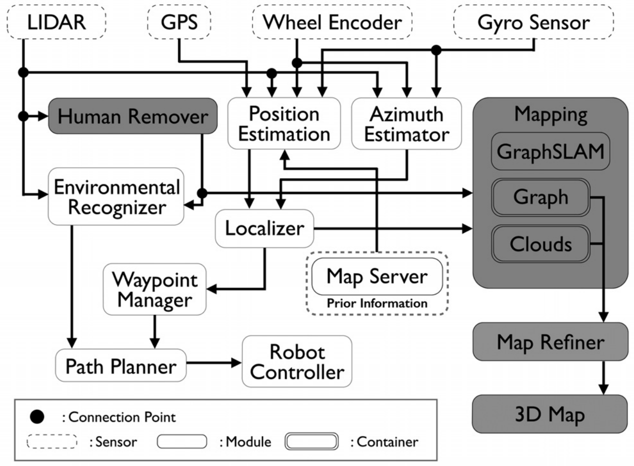

Fig. 1. System architecture of our autonomous system running on INFANT. The background of the new module is shaded.

2.2. Autonomous Navigation System

Figure 1 shows the autonomous navigation system architecture that runs in unknown environments where prior information is gotten easily from the Internet and real-time data from the robot is used without the need for previous detailed information. This detailed information is a specific environment map made by the operator on the spot using the LIDAR. In unknown environments, the plotted point (waypoint) indicating the global position is given as prior information so that the robot determines the rough direction to the target (global path planning) and follows road surfaces automatically by creating local maps (local path planning). Local planning consists mainly of environment recognition, localization and path planning. In environment recognition, it searches for the static obstacles (road edges and walls) and dynamic obstacles (human). Localization estimates the robot's position using a digital map, LIDAR, AMU and initial position. Additionally, to know the global position helps to figure out the target area and exploring area. To reduce accumulated error such as slippage, the global position is determined by the matching of the point cloud from LIDAR and electronic map. In path planning using the results of environment recognition and localization, it creates a rough route to the waypoint and selects the smooth path from several local path candidates for avoiding obstacle with best final pose for effectiveness and safety.

2.3. Autonomous Large Scale 3D Map Construction

A 3D map tracing the robot's is constructed by inputting robot movement and environment shape information. Shape information is used as underlying information. For this, this information should consist of static information alone by having information on dynamic obstacles removed. In this system, the living environment is the main work area for the autonomous mobile robot and pedestrian are the large majority of mobile obstacles. A human remover is established between environmental sensors and map construction and from this, we construct a 3D map using only static obstacles. A constructed partial 3D maps are handed over to the map construction module and integrated by common territory among each partial maps. The large 3D map is then constructed.

3. Localization

One of our navigation system's main functions involves describing localization in map coordinates. Navigation system accuracy based on localization may be strongly controlled by results of localization. For these reasons, high multiplicity and flexibility are important factors in localization. Adapting to this requirement, we propose integrating the following two methods: (i) local localization state estimation (dead reckoning) using wheel odometry, (ii) global localization using a digital map that contains latitude and longitude. Additionally, localization in map coordinates consists of azimuth estimation and translational position estimation in map coordinates. Digital maps from the Internet are given to both processes as prior information. Specifically, by comparing shape information of buildings from the digital map and the real world around the robot, the position and azimuth of robot's are estimated in map coordinates. Thanks to map-provision services, e.g., Google maps, easily acquired digital maps of living environments ensure a multiplicity of localizations. We detail flexibility of each estimation and modules. Note that the digital map we use is from Google maps [a] and that the map resolution of the map is 0.238 m per pixel.

3.1. Azimuth Estimation Using a Gaussian Map

Most surface of buildings in the map in our living environment are flat, so we assume that the real digital map shape and model match. This suggests that azimuth of the robot be estimated from the distribution of surrounding surface directions even where robot cannot measure own position from artificial satellite. We extract predominant surface data from shape data obtained from the LIDAR, and compare this surface with digital map to estimate the azimuth. The azimuth estimation flow is normal estimation from point cloud, Gaussian map generation with Gaussian representation, and predominant normal extraction. A final comparison is made by using the work of Shimizu [7]. In this process, classifying a Gaussian map based on normals of point cloud, the median and variation is derived from each cluster. The processing flow for extracting the dominant normal on Gaussian sphere is described in [6]. Note that after clustering, the dominant rate is given to each superior normal using the number of elements. The dominant rate is the rate of point cloud regarded as the same group in acquired point cloud. If this ratio for all dominant normals is not enough or the variation of each cluster is high, it is necessary to reduce the probability of the estimated normal value in the next matching process. In real urban environments, there are some places that the LIDAR can find any of the buildings inside the sensor area. In such areas, it is possible that estimating the azimuth using the predominant normal may not be activated correctly, but we solve this problem by integrating with a gyro sensor.

3.2. Translational Position Estimation by Map Matching

Azimuth error accumulated from odometry is corrected by azimuth estimation using the Gaussian map mentioned above. In the sections that follow, we detail how to correct accumulated translational error. Images used in map matching are digital with depicted building shapes (model image) and point clouds acquired from the LIDAR for each 3 m. The distance between points and the robot are 30 m or less (query image) and the amount of position calibration is calculated by searching the spot where the model and query images correspond. General-purpose template matching using intensity is not enough, however, for finding the correct relationship between two images, so we calibrated the position by using formula evaluation (Eq. (1)) to calculate the similarity of the model and query images.

$$Score = A\frac{Q_{fit}^{u,v}}{Q_{plot}} + B\left(1 - \left(\frac{d^{u,v}}{w}\right)^2\right) \quad . \quad . \quad . \quad . \quad (1)$$

(u,v) is a coordinate of pixels, Q_{plot} is the total number of wall pixels detected as a wall of a building in a query image, $Q_{fit}^{u,v}$ is the number of pixels which the shape of building in model image and the wall surface in query data are matched in pixel coordinate (u,v). $d^{u,v}$ is the distance between (u,v) and the estimated position of the robot converted in the model image. w is a region to be searched in model data and this is also denoted as the distance from the robot not yet estimated in the model image.

4. Environment Recognition and Path Planning

The most important task for the autonomous robot in an urban environment is avoiding obstacles. Obstacles are divided into two groups – static and dynamic. Dynamic obstacles especially interrupt robot safety, so we must know the movement and instance of dynamic obstacles for planning movement to avoid them. When the robot recognizes the environment or constructs the environment map while running, only static information are expected as sensor data. In fact, using dynamic information may decrease the accuracy of environment recognition and of the map. Recognizing human beings in the environment is critical to recognizing the environment and constructing the environment map. For these reasons, we use human recognition from the point cloud and use shape features in motion planning. Point cloud of pedestrians which have a large majority of obstacles in the urban environment removed are used in environment recognition, environment map construction and 3D map construction to improve accuracy, as mentioned below.

Table 1. Extracting features.

No.	Dimension	Description of the feature
f_1	1	Distance to the centroid of the cluster
f_2	1	Number of points contained in the cluster
f_3	3	Size of the cluster
f_4	6	Three dimensional covariance of the cluster
f_5	6	The normalized moment of inertia tensor
f_6	9	Two dimensional covariance of upper and bottom body
f_7	30	Width, length and curvature in each block
f_8	143	Histogram of the point number in the grid
f_9	143	Histogram of the normal gradient in the grid

4.1. Human Recognition

This section focuses on human recognition using the 3D point cloud from the LIDAR and shape feature. We only require a solid object to detect human beings, so the point cloud of the ground surface is excluded by using Min-Max. Projecting acquired points on two dimensional grid maps, individual cells compare the highest and lowest values of points in each cells. If the difference is below the threshold, points in the voxel are discarded; in other words, only 3D object points are extracted. Remaining points are then clustered by Euclidean distance. Generated clusters contain objects of different sizes in the environment, e.g., human beings, trees, or walls. These objects are regarded as in the same cluster if a human being is close to these objects. This means that people walking in the group or standing by the objects gathered as the same cluster because of short distance between clusters. Dividing clusters is thus an important technique for recognizing human beings correctly. Principal component analysis (PCA) and a bounding box are used for this process. These clusters are then used to extract the nine shape features shown in **Table 1**. These features are high dimension, so it is undesirable to determine by the threshold alone. We use a support vector machine (SVM) [8] trained by feature vector calculated from shape information of object and construct the classifier. Note that training data is created from the 3D point cloud from the LIDAR. Positive data are a human being who is standing or walking. Negative data are the other 3D objects (**Fig. 2**). **Figs. 3** and **4** show the ROC and precision-recall curves used to evaluate the classifier trained by the extracted features. Test data consist of positive human and negative other object data and differs from the training data. With the observation value alone, however, false detection may occur due to the sensor rate and environment noise. To avoid this problem, we predicted and tracked target movement, using the motion model to estimate the cluster movement and track cluster to weigh based on the previous recognition rate.

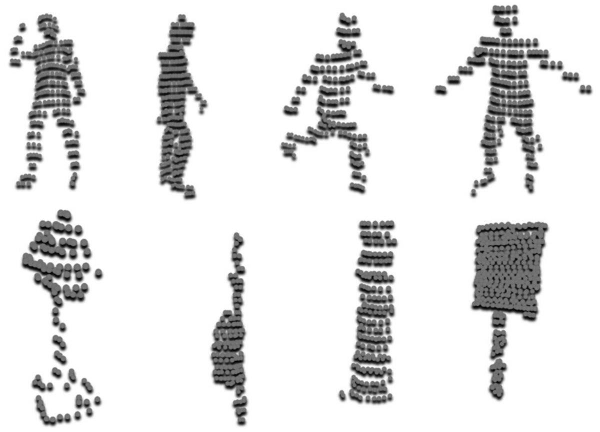

Fig. 2. The figure above shows positive data and that below shows negative data.

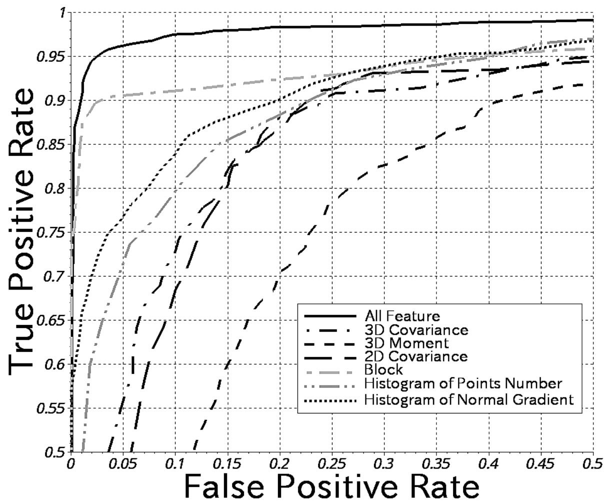

Fig. 3. ROC curves.

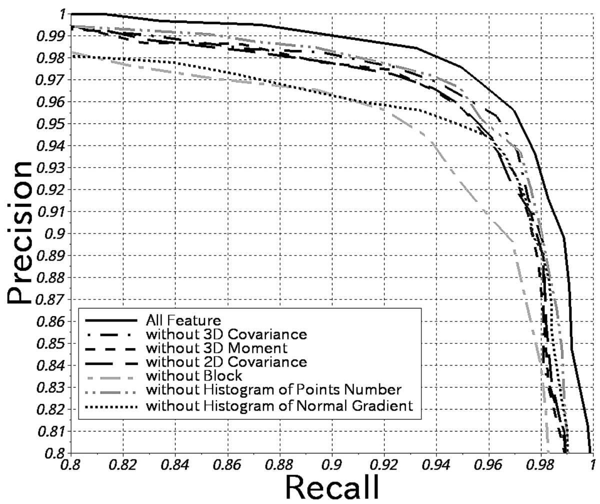

Fig. 4. Precision/recall curves.

4.2. Environment Recognition

We recognize the environment by using a LIDAR. Variety exists in real-world regions that robots cannot enter. Similar to other processes, environmental recognition requires flexibility to adapt to these situations, which change over time. In our local map construction, regions that robots cannot enter are divided into three parts – obstacles with large height change, obstacles with little height change and different material load surfaces. Using the optimum technique in individual regions may improve flexibility in recognizing where robots can enter and integrating each technique at the suited situation to be more multiple system. Note that as the path planning below is executed in 2-dimensional grid cells, environment recognition represents obstacles by 2-dimensional grid cell.

Static obstacles are assumed to exist in our living environment as walls and road edges. Min-Max [9] used in previous obstacle recognition is selected for use with obstacles with large height change, but its use with obstacles with little height change is difficult to recognize because of noise from robot movement and ground roughness. To solve this problem, we use PCA [10] with points near around the interest point and estimate curvatures calculated from variance-covariance matrix. By changing the likelihood of obstacles found by using both techniques dynamically, regions that robots cannot enter could be estimated.

4.3. Path Planning and Motion Selection

With path planning and motion selection, a local environment map is constructed by using the environment recognition above. A global path toward a waypoint is generated by using a D*Lite algorithm [11]. After a global path is generated, local path candidates are generated and a path is selected with smoothness, no fear of interference by obstacles, and considering the robot's last pose to ensure efficient safe movement.

To follow the global path generated by the D*Lite algorithm, we set a target trajectory point on the global path and select a point on the global path and in distance d as target position coordinate (x, y). The trajectory generated toward the target point coordinate without considering robot motion model may deviate from the trajectory, so in this approach we use our proposed trajectory generation [12]. This enables us to plan 5 degrees of freedom in Eq. (2) at every point in the trajectory. Motion parameters considered while calculating the trajectory are max acceleration, the max curvature to trace and the max difference of curvature.

$$\mathbf{x}_t = \begin{bmatrix} x_t & y_t & \theta_t & \kappa_t & v_t \end{bmatrix}^T \quad \ldots\ldots\ldots (2)$$

$\mathbf{x}_t$ is the robot state in time t, x_t, y_t, θ_t are the position and pose of the robot, κ_t is trajectory curvature, and v_t is forward velocity. The processing flow for generating trajectories while following a waypoint is described in [6]. Marked in the grid are obstacles judged as impossible to run by local map construction. Trajectories without interfering obstacles are saved as candidates and that with high adaptability with the global path is selected.

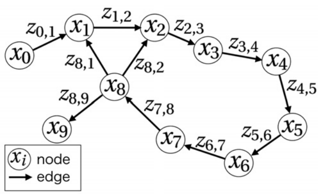

Fig. 5. Graph.

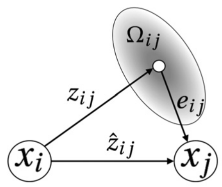

Fig. 6. Detailed graph.

5. Automatic 3D Map Construction

In this section, we describe constructing a large scale map. Optimization and generation of the graph is essential for map construction. Our system created a highly accurate map using the point cloud which removed human [13].

5.1. Graph-Based SLAM

Here we illustrate graph-based SLAM [14, 15]. This is a localization method containing odometry and sensor data proposed by Giorgio Grisetti. The estimated position and sensor data are kept by graph structure. As **Figs. 5** and **6** show, a graph consists of nodes and edges. Node $\mathbf{x}$ is the state of the robot in the corresponding frame. The relative position between nodes is expressed in edges and has relative position data z_{ij} and matrix Ω_{ij}, which has the weight of each component in a diagonal element. This is used while calculating the information matrix of each edge while estimating the maximum position likelihood. In the optimization process, maximum likelihood position $\mathbf{x}^*$ makes objective function $F(\mathbf{x})$ in Eq. (3) the minimum.

$$F(\mathbf{x}) = \sum_{i,j \in C} \mathbf{e}_{ij}^T \Omega_{ij} \mathbf{e}_{ij} \quad \text{. . . . .} \quad (3)$$

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} F(\mathbf{x}) \quad \text{. . . . .} \quad (4)$$

$$\mathbf{e}_{ij} = \hat{z}_{ij}(\mathbf{x}_i, \mathbf{x}_j) - z_{ij} \quad \text{. . . . .} \quad (5)$$

$\hat{z}_{ij}$ is the relative state between temporary estimated position $\mathbf{x}_j$ and $\mathbf{x}_i$ and z_{ij} is an observed value of state $\mathbf{x}_j$, $\mathbf{x}_i$. Next is the minimization of objective functions in Eq. (4). To optimize objective function $F(\mathbf{x})$, which is nonlinear to estimated position $\mathbf{x}$, it is solved by regarding the objective function as a topically linear function and using repetitive calculations around temporary estimated position $\mathbf{x}$. If appropriate estimated position $\breve{\mathbf{x}}$ is established, the error function around the position is expressed as shown below.

$$\mathbf{e}_{ij}(\breve{\mathbf{x}} + \Delta\mathbf{x}) \simeq \mathbf{e}_{ij}(\mathbf{x}) + J_{ij}\Delta\mathbf{x} \quad \text{. . . . . . .} \quad (6)$$

Taylor expansion is done until the first member to regard error function $\mathbf{e}_{ij}$ around estimated position $\breve{\mathbf{x}}$ as linear. J_{ij} is a Jacobian matrix around estimated route $\breve{\mathbf{x}}$ of error function $\mathbf{e}_{ij}$. With this in mind, the component of objective function $F_{ij}(\breve{\mathbf{x}} + \Delta\mathbf{x})$ around initial estimated route $\breve{\mathbf{x}}$ is expressed as follows:

$$\begin{aligned} & F_{ij}(\breve{\mathbf{x}} + \Delta\mathbf{x}) \\ & = \mathbf{e}_{ij}(\breve{\mathbf{x}} + \Delta\mathbf{x})^T \Omega_{ij} \mathbf{e}_{ij}(\breve{\mathbf{x}} + \Delta\mathbf{x}) \\ & = (\mathbf{e}_{ij}(\breve{\mathbf{x}}) + J_{ij}\Delta\mathbf{x})^T \Omega_{ij} (\mathbf{e}_{ij}(\breve{\mathbf{x}}) + J_{ij}\Delta\mathbf{x}) \\ & = c_{ij} + 2b_{ij}\Delta\mathbf{x} + \Delta\mathbf{x}^T H_{ij} \Delta\mathbf{x}. \quad \text{. . . . . .} \quad (7) \end{aligned}$$

$$\left(\begin{array}{ccc} c_{ij} & = & \mathbf{e}_{ij}^T \Omega \mathbf{e}_{ij} \\ b_{ij} & = & \mathbf{e}_{ij}^T \Omega J_{ij} \\ H_{ij} & = & J_{ij}^T \Omega J_{ij} \end{array} \right)$$

Objective function $F(\breve{\mathbf{x}} + \Delta\mathbf{x})$ is a summation of component $F_{ij}(\breve{\mathbf{x}} + \Delta\mathbf{x})$, and is given as follows:

$$\begin{aligned} & F(\breve{\mathbf{x}} + \Delta\mathbf{x}) \\ & = \sum_{i,j \in C} \left(c_{ij} + 2b_{ij}\Delta\mathbf{x} + \Delta\mathbf{x}^T H_{ij} \Delta\mathbf{x} \right) \\ & = c + 2b\Delta\mathbf{x} + \Delta\mathbf{x}^T H \Delta\mathbf{x}. \quad \text{. . . . .} \quad (8) \end{aligned}$$

By replacing $\Delta\mathbf{x}$ with $\Delta\mathbf{x}^*$ to make the first derivative $\Delta\mathbf{x}$ of objective function $F(\breve{\mathbf{x}} + \Delta\mathbf{x})$ as 0, $\Delta\mathbf{x}^*$ is expressed as follows:

$$H\Delta\mathbf{x}^* = -b. \quad \text{. . . . .} \quad (9)$$

In Eq. (9), we must select an efficient computational method such as LU decomposition to calculate the inverse matrix to be implemented because H is a nondense matrix most of the time. From Eq. (9), calculated $\Delta\mathbf{x}^*$ updates the estimated route $\breve{\mathbf{x}}$ as mentioned below. This calculation only ends if the desired accuracy is satisfied.

$$\breve{\mathbf{x}} = \breve{\mathbf{x}} + \Delta\mathbf{x}^* \quad \text{. . . . .} \quad (10)$$

5.2. ICP

The iterative closest point (ICP) [16] is used to correct the position of two sets of point clouds and to calculate edges containing relative position data among two frames in our system. The ICP estimates highly accurate relative positions by using repetitive calculations when two sets of point clouds are in a suitable initial relative position. If the initial relative position is far from the final estimated relative position, however, registration of our method fails because of the relationship between the points of the coordinates that compose each set of point clouds. By searching for the congruent point of interest in the ICP, points of coordinates that are a short distance away are selected. Not giving a suitable initial position may cause mistaken matching and collapse registration. In other words, stabilized registration requires a suitable initial position. With our method, odometry becomes a useful initial relative position for the initial position.

5.3. Loop Closure

Using the ICP enables us to estimate a high accurate edge from adjoined nodes, but as a running course become longer, estimation error accumulates. To solve this problem, the ICP is executed with two nodes that reach the same region to generate another edge. After this, conducting Graph-based SLAM may correct accumulated error.

6. Experimental Result

We use autonomous mobile experiments to prove the feasibility of our system in the real environment include both indoor and outdoor types.

6.1. Autonomous Navigation

6.1.1. Environmental Recognition and Path Planning

Figure 7 shows generated path results based on the environmental recognition in **Fig. 8**. In this environmental recognition, our system detected obstacles with large height change, little height change and a different material load surface. Path planning also followed the preset route at first and in the step for generating a trajectory, thus creating a path toward the middle of the traversable area based on the local map. These results show that our environmental recognition and path planning are effective.

6.1.2. Localization

Figure 8 shows localization results. As shown in **Fig. 8**, our localization performs in various environments without fail.

6.1.3. Human Recognition

Figure 9 shows human recognition in experiments. At top in **Fig. 9**, a cluster estimated to be a human being is surrounded by a bounding box. It is color-coded for each estimated cluster. The point cloud from which the cluster of the human being has been removed is shown at bottom. These results show that recognizing and removing a human from the point cloud work successfully.

6.2. 3D Map Construction

To demonstrate the usefulness of our 3D map construction, we selected 3 routes from experiments to create a partial map and combined them into a large map. Just as in the autonomous navigation experiments, these routes include both inside and outside. The estimated trajectory and map construction process are shown in **Fig. 10**, and the detailed large 3D map constructed is shown in **Fig. 11**. As shown in **Fig. 10**, the whole map has less distortion than an aerial photograph thanks to the position collection process. **Fig. 11** shows that the detailed 3D map has high accuracy in the limited place.

7. Conclusions and Future Work

We have proposed autonomous navigation without prior detailed map information and automatic large 3D map construction. Autonomous navigation consists mainly of localization, environmental recognition, and path planning. Localization using azimuth estimation and translational position estimation showed the robustness by using LIDAR and digital map. In environmental recognition, a robot could detect obstacles by integrating height information with curvature information while the robot avoiding obstacles such as moving like pedestrian, trees and road edges. In human recognition, the robot calculated high dimensional parameters from shape information provided by the LIDAR and judged whether a shape was human by using an SVM. In path planning, our system moved safely and smoothly by using global and local path planning. Global path planning shows the approximate direction the robot should go by waypoints and local path planning follows the global path and avoids surrounding obstacles based on the robot's kinematic model. In automatic large 3D map construction, our system created a highly accurate map using sensor information from which human beings had been removed. Based on the relative distance between two points calculated by odometry, our system estimated both the relative position and relative angle by using the ICP, enabling us to automatically construct the map so that it revised cumulative error by optimization using graph-based SLAM. When the robot revisited, our system cloud updated and integrated the map highly accurately by reprocessing the above process. In experiments in various environments, we demonstrated the usefulness of our proposed autonomous navigation system. It became possible to automatically construct a large highly accurate 3D map. In the future, we will develop a construction method for semantic maps.

References:

[1] N. Akai, K. Inoue, and K. Ozaki, "Autonomous navigation based on magnetic and geometric landmarks on environmental structure in real world," J. of Robotics and Mechatronics, Vol.26, No.2, pp. 158-165, 2014.

[2] J. Eguchi and K. Ozaki, "Development of the Autonomous Mobile Robot for Target-Searching in Urban Areas in the Tsukuba Challenge 2013," J. of Robotics and Mechatronics, Vol.26, No.2, pp. 166-176, 2014.

[3] T. Shioya, K. Kogure, and N. Ohta, "Minimal Autonomous Mover: MG-11 for Tsukuba Challenge," J. of Robotics and Mechatronics, Vol.26, No.2, pp. 225-235, 2014.

[4] L. Gomes, "Hidden Obstacles for Google's Self-Driving Cars," MIT Technology Review, August 28, 2014.

[5] Q. V. Le, "Building high-level features using large scale unsupervised learning," Acoustics, 2013.

[6] T. Yokota et al., "Development of the Versatile Autonomous Navigation System," The 15th SICE System Integration, Tokyo, 2014 (in Japanese).

[7] S. Shimizu and Y. Kuroda, "High-Speed Registration of Point Clouds by using Dominant Planes," The 19th Robotics Symposia, pp. 453-458, 2014 (in Japanese).

[8] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," ACM Trans. on Intelligent Systems and Technology, 2011.

[9] 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.

[10] 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.

[11] S. Koenig and M. Likhachev, "D* Lite," Proc. of the AAAI, pp. 476-483, 2003.

[12] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments," Proc. of the JFR, Vol.25, pp. 939-960, 2008.

[13] T. Yokota and Y.Kuroda, "LIDAR-based Human Recognition using Shape Features," The Robotics and Mechatronics Conf. 2014, 3P1-K04, 2014 (in Japanese).

[14] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g^2o: A general framework for graph optimization," pp. 3607-3613, 2011.

[15] G. Grisetti, R. Kummerle, and C. Stachniss, "A tutorial on graph-based SLAM," IEEE Intelligent Transportation Systems Magazine, Vol.2, pp. 31-43, 2010.

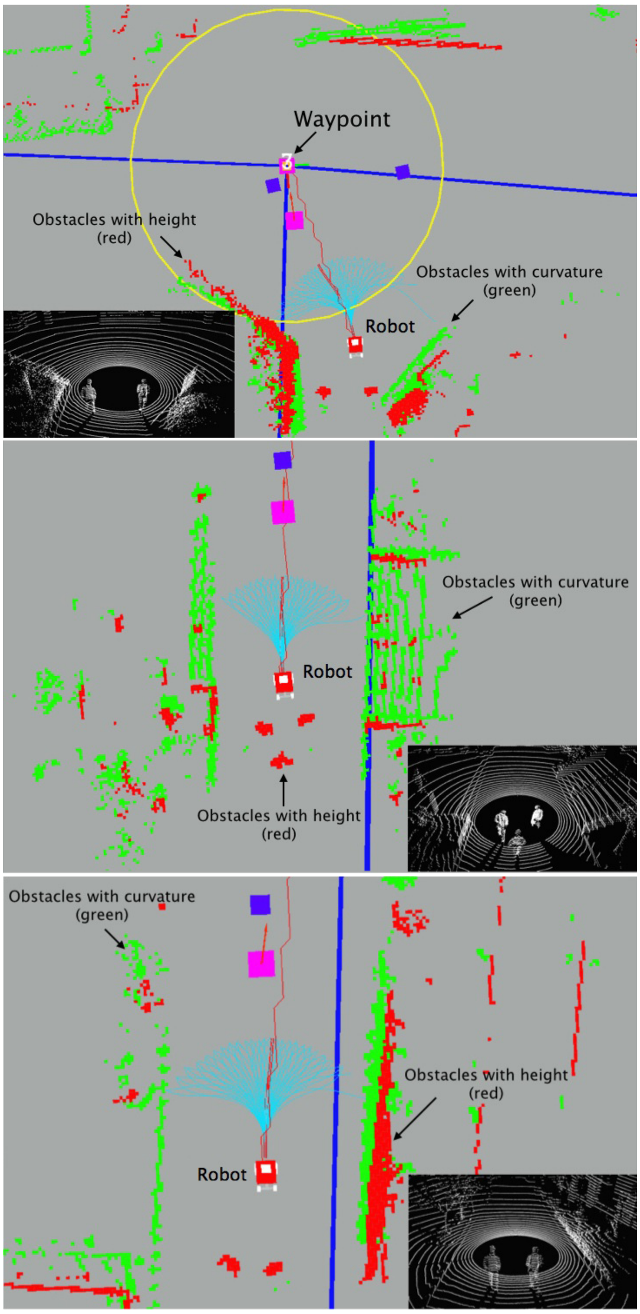

Fig. 7. Environment recognition and path planning results.

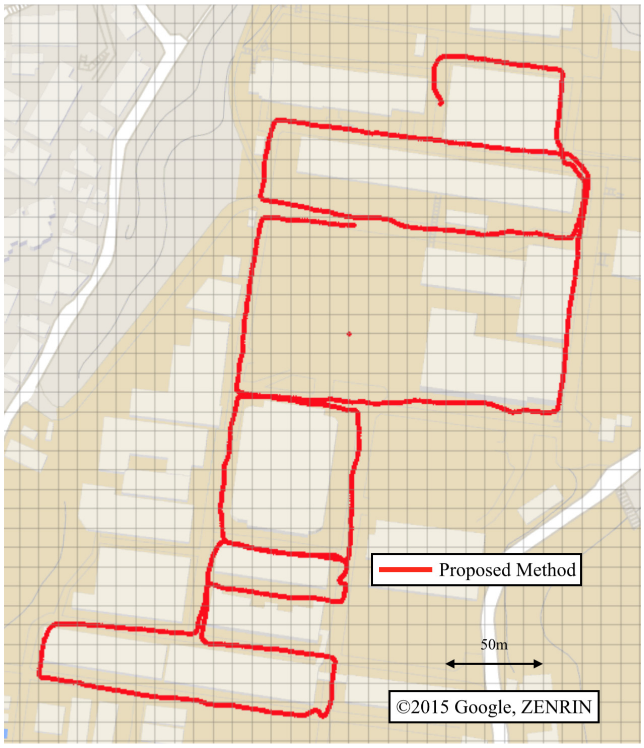

Fig. 8. Results of localization.

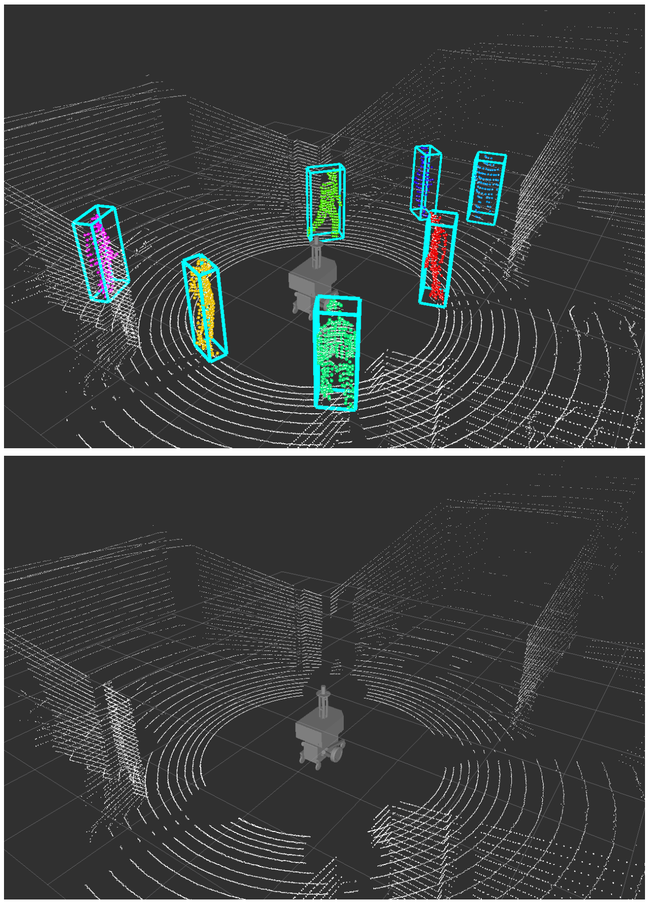

Fig. 9. Human-being remover.

[16] S. Rusinkiewicz 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.

Supporting Online Materials:

[a] https://www.google.co.jp/maps/ [Accessed February 25, 2015]

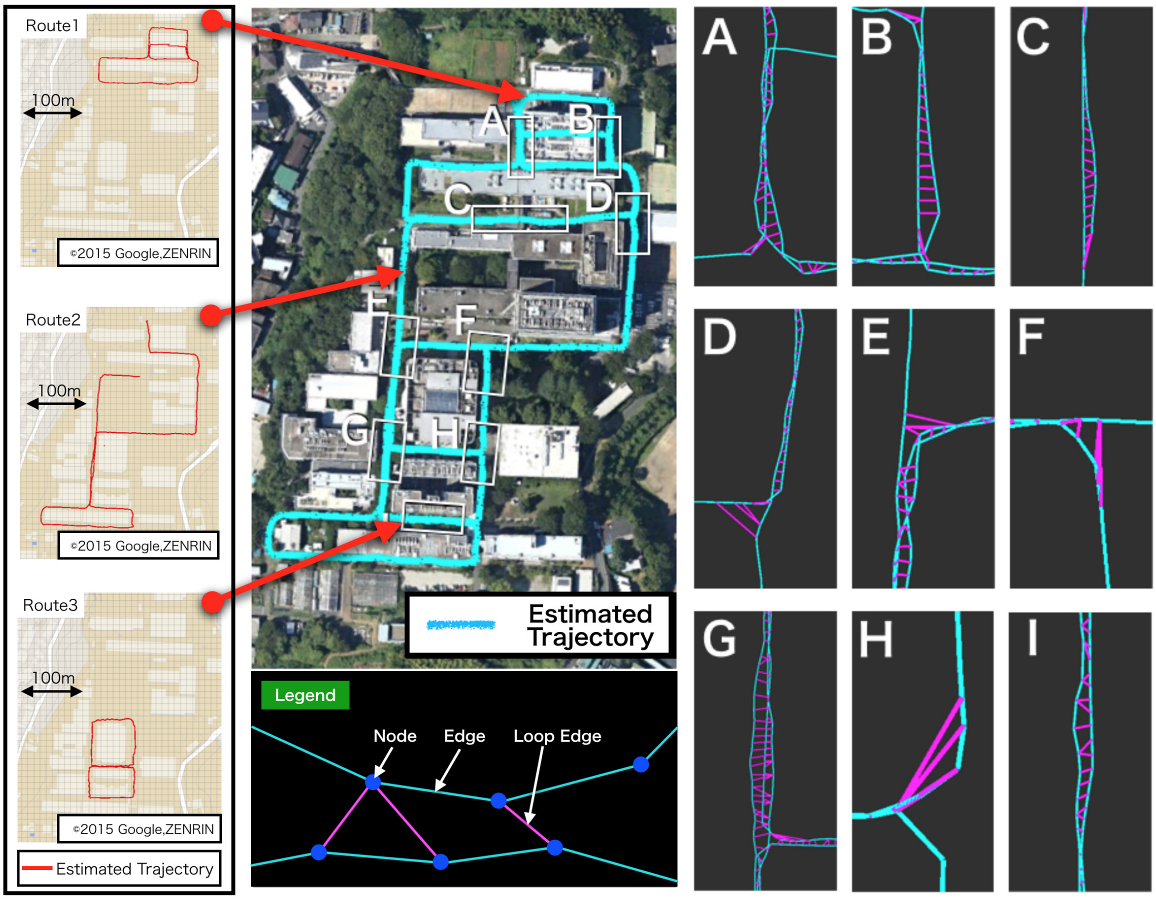

Fig. 10. Large-scale 3D map.

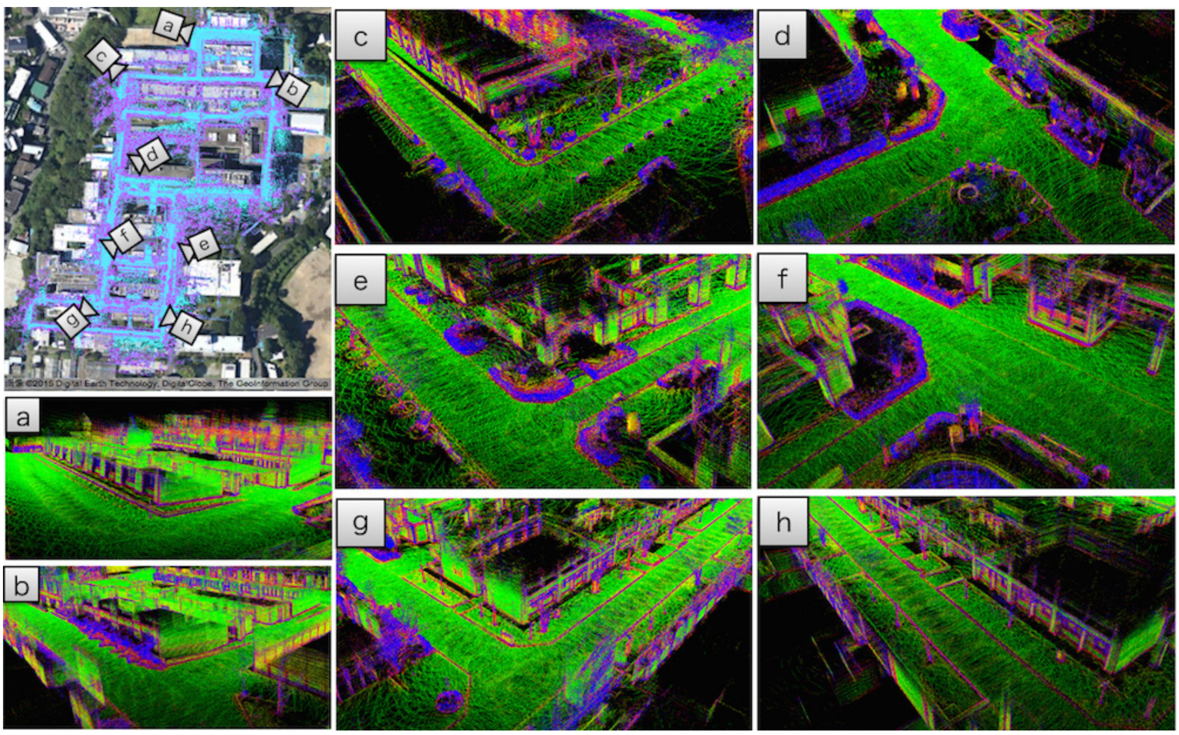

Fig. 11. Detailed constructed map.

Name:
Yusuke Fujino

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:
2014 Received the Bachelor degree in Mechanical Engineering from Meiji University

Membership in Academic Societies:
- The Robotics Society of Japan (RSJ)

Name:
Kentaro Kiuchi

Affiliation:
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:
2013 Received the Bachelor degree in Mechanical Engineering from Meiji University
2015 Received the Master degree in Mechanical Engineering from Meiji University

Membership in Academic Societies:
- The Japan Society of Mechanical Engineers (JSME)

Name:
Shogo Shimizu

Affiliation:
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:
2013 Received the Bachelor degree in Mechanical Engineering from Meiji University
2015 Received the Master degree in Mechanical Engineering from Meiji University

Membership in Academic Societies:
- The Japan Society of Mechanical Engineers (JSME)

Name:
Takayuki Yokota

Affiliation:
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:
2013 Received the Bachelor degree in Mechanical Engineering from Meiji University
2015 Received the Master degree in Mechanical Engineering from Meiji University

Membership in Academic Societies:
- The Japan Society of Mechanical Engineers (JSME)

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)