3-D MAP GENERATION BY A MOBILE ROBOT EQUIPPED WITH A LASER RANGE FINDER

Takumi Nakamoto, Atsushi Yamashita, and Toru Kaneko

Department of Mechanical Engineering, Shizuoka University
3-5-1 Johoku, Hamamatsu-shi, Shizuoka, 432-8561, Japan
Phone: +81-53-478-1604, Fax: +81-53-478-1604
E-mail: {f0530055, tayamas, tmtkane}@ipc.shizuoka.ac.jp

ABSTRACT

When an autonomous mobile robot acts in constructed environments, a map that has the information about obstacles is necessary for the robot. If a map is given, however, the mobile robot cannot start its action without the information of its present location on the map. Therefore, techniques of map generation and self-localization are important for mobile robots. In this paper, we propose map generation and self-localization methods by a mobile robot equipped with a laser range finder. The robot measures the distance between obstacles around it with a laser range finder at multiple positions, and generates a map by integrating these range information. The integration is realized by using a dead-reckoning (odometry) method and ICP (Iterative Closest Point) algorithm together. We also propose a method for estimating the present location of the robot on the generated map. Experimental results have shown the validity of the proposed method.

1. INTRODUCTION

Introduction of autonomous mobile robots such as guard robots and nursing robots is expected with development of robot industry in recent years. Generally, autonomous mobile robots need the information of obstacle positions when they work in a field. One of the methods for showing the information of obstacle positions is a two-dimensional (2-D) plane map of surrounding environment. A robot can search out the safe path if it knows the information of obstacle positions from a 2-D plane map of surrounding environment. However, a 2-D plane map does not necessarily exist. Therefore, when a mobile robot has no map, it needs to generate a 2-D plane map by itself [1]. Additionally, a mobile robot cannot avoid obstacles that are not on the same plane with a 2-D plane map. In this case, three-dimensional (3-D) map generation of surrounding environment becomes important for a robot [2-3]. In this paper, we propose a method for 3-D map generation of environment.

Another problem is estimation of the present location of a mobile robot on a map. Mobile robots cannot start their action if they do not know the present location on a map. In this case, a robot has to estimate a present location only from the information acquired with its sensors [4-10].

Therefore, we also propose a method for estimating a present location of a mobile robot on a generated map.

2. SENSING METHOD

The mobile robot is equipped with a laser range finder (LRF) (Fig.1). A LRF can acquire the information of distance to surrounding obstacles. The robot is equipped with two LRFs in order to acquire the 3-D information of obstacle positions (Fig.2). The first LRF (LRF1) is installed in parallel with a floor. The robot measures the distance to the surrounding environment that has the same height with LRF1 by using LRF1. As a result, the information of distance to obstacles can be acquired in the horizontal cross section in the same height of laser light (Fig.3). The second LRF (LRF2) is leaned to a floor. The robot acquires the information about obstacle positions in various heights by using LRF2. The robot can acquire 3-D information about surrounding environment while it moves. We assume that the mobile robot moves only on plane floors. Moving objects do not exist in environments at the time of measurement. In our method, 2-D plane maps are generated by using the sensing result of LRF1. 3-D maps are constructed by using the 2-D information in environment acquired with the LRF1 and the 3-D information acquired with the LRF2.

Fig.1 Laser range finder

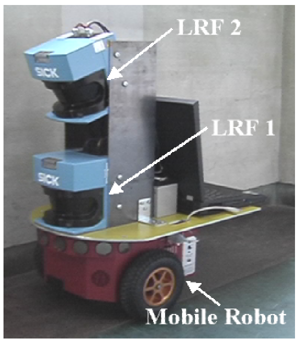
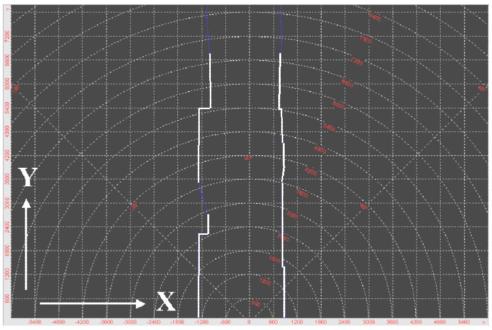


Fig.2 Mobile robot

(a) Sensing range



(b) Sensing result

Fig.3 Sensing with a LRF

3. MAP GENERATION

3.1 Integration of Measurement Data

Map generation is performed using the information acquired by two LRFs on the mobile robot. However, it is difficult to acquire the distances between the robot and all obstacles in the environment from one position, because there are limitations in a measurable range of the LRF and problems of occlusions. Therefore, the robot measures environment from multiple positions while it moves. And the map is generated by integrating each measurement data acquired by the LRFs. It is necessary to estimate the positional and orientational relationship among all measurement positions in order to generate whole map of the environment by integrating all measurement data.

A dead-reckoning (odometry) is one of self-localization methods of robots by using an inner sensor of a wheeled robot. However, a slipping of each wheel causes positional and orientational errors of the robot position and orientation, because this method estimates its traveling distance only from the information about rotational angle of each wheel. Furthermore, a dead-reckoning cannot correct an accumulated error, and positional and orientational errors accumulate according to the increase of the traveling distance. Therefore, the error inevitably arises in the positional and orientational relationship of each measurement data acquired in multiple positions, and the distorted map will be generated if map generation is performed only using a dead-reckoning. In this case, positional and orientational errors caused by a dead-reckoning must be corrected by using external sensors.

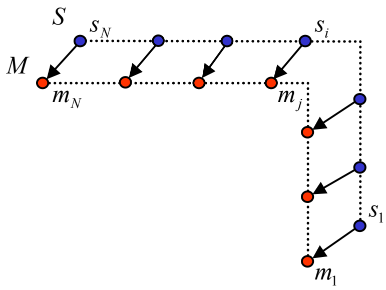


Fig.4 ICP algorithm

In this paper, we use a LRF that can measure direct distance to obstacles as an external sensor. The error correction of self-localization is executed by aligning each measurement data. Alignment of each measurement data is performed by using the measurement results whose parts overlap with each other.

As to the alignment problem of sensing results, there are a lot of methods. We use the method based on ICP (Iterative Closest Point) algorithm [11], because IPC algorithm is one of the most efficient and powerful method that can carry out direct alignment of the measurement data.

3.2 ICP Algorithm

ICP algorithm is an alignment method of multiple point groups. It minimizes the error function by iterative computation using the overlapping area measured by the LRF at multiple positions. Here, let M and S be point groups, respectively. Point group M is measurement data acquired after the robot moves, while point group S is data before it moves. We define the point $\boldsymbol{m}_j$ $(1 \le j \le N)$ that is the nearest point to a point $\boldsymbol{s}_i$ $(1 \le i \le N)$ in point group M is corresponding points about a point $\boldsymbol{s}_i$ in point group S (Fig.4). Parameters $(\mathbf{R}, \mathbf{t})$ indicate the orientational and positional relationship between point group M and S. In other words, these parameters mean the movement of the robot. $\mathbf{R}$ is a rotation matrix of 2×2 (or 3×3), and $\mathbf{t}$ is a translational movement vector. The parameters $(\mathbf{R}, \mathbf{t})$ can be gained by minimizing E_1 that is the square sum of the distance between each corresponding point (Equation (1)).

$$E_1(\boldsymbol{R}, \boldsymbol{t}) = \sum \left\| \boldsymbol{m}_j - (\boldsymbol{R}\boldsymbol{s}_i + \boldsymbol{t}) \right\|^2 \tag{1}$$

This algorithm uses corresponding point groups within a constant distance between each corresponding point (Fig.5).

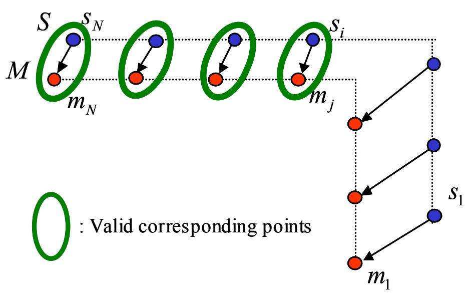


Fig.5 Corresponding point groups

The alignment of two groups is performed by this algorithm even if the numbers of points in point group M and S differ from each other. N_1 and N_2 are the numbers of points in point group S and M, respectively. If the numbers of point $\boldsymbol{m}_j$ is larger than that of point $\boldsymbol{s}_i$ ($N_1 < N_2$), the alignment is performed under the condition that there is the point $\boldsymbol{m}_j$ that is not selected as the corresponding point in point group S (Fig.6).

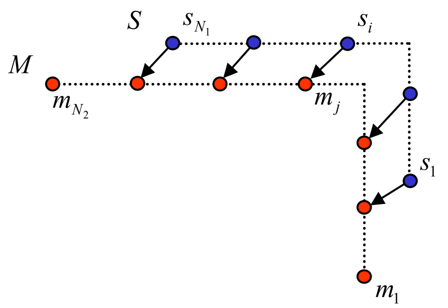


Fig.6 The number of points $\boldsymbol{m}_j$ is larger

This algorithm can treat the situation where the numbers of point $\boldsymbol{s}_i$ is larger than that of point $\boldsymbol{m}_j$ ($N_1 > N_2$). Let the point $\boldsymbol{s}_i$ that is the nearest point to point $\boldsymbol{m}_j$ be the corresponding point about point $\boldsymbol{m}_j$. In this case, there is point $\boldsymbol{s}_j$ that is not selected as the corresponding point in point group M (Fig.7).

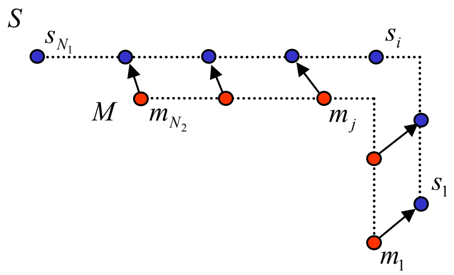


Fig.7 The numbers of points $\boldsymbol{s}_i$ is larger

3.3 Modified ICP Algorithm

The original ICP algorithm is not able to detect correct corresponding points if the distance between each measurement data is large (Fig.8). In this case, the alignment cannot be performed correctly.

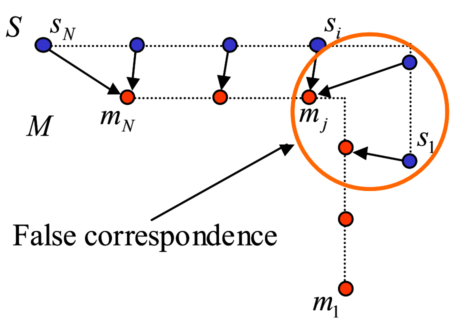


Fig.8 False correspondence

Therefore, we use modified ICP algorithm that adopt point-to-line correspondence instead of point-to-point correspondence (Fig.9).

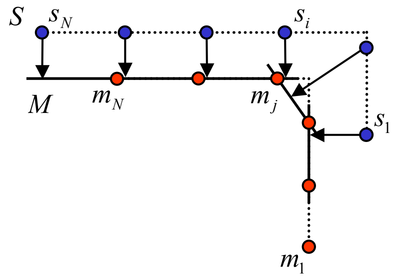


Fig.9 Modified ICP algorithm

Specifically, let two points $\boldsymbol{m}_j$, $\boldsymbol{m}_k$ ($1 \leq j \leq N$, $1 \leq k \leq N$) that is the nearest point to point $\boldsymbol{s}_i$ ($1 \leq i \leq N$) in point group M be corresponding points about point $\boldsymbol{s}_i$ in point group S. The straight line that consists of the two points $\boldsymbol{m}_j$, $\boldsymbol{m}_k$ is used instead of corresponding points. This algorithm calculates the parameters (**R**, **t**) to minimize E_2 that means the square sum of the distance between each point and straight line (Equation (2)).

$$E_2(\boldsymbol{R},\boldsymbol{t}) = \sum \frac{\left|(\boldsymbol{R}\boldsymbol{s}_i + \boldsymbol{t})(\boldsymbol{m}_k - \boldsymbol{m}_j)_\perp + \boldsymbol{m}_k \boldsymbol{m}_{j\perp}\right|^2}{\left\|\boldsymbol{m}_k - \boldsymbol{m}_j\right\|^2} \qquad (2)$$

where $\boldsymbol{m}_\perp$ means the vector which is perpendicular to $\boldsymbol{m}$, and its size is $|\boldsymbol{m}|$.

In our method, the initial values of parameters (**R**, **t**) are decided with information acquired by the dead-reckoning method. The iterative computation is executed before E_2 becomes smaller value than a threshold value.

4. SELF-LOCALIZATION

Mobile robots cannot start their action without the information of its present location on the map. Therefore, it is necessary for the robots to estimate the present location on the map without giving a robot's initial location.

In this paper, self-localization is performed by searching for the location where the measurement result of the surrounding environment is similar to the shape of the environment in the known map. If only one location is searched, this location on the map is the present location. If multiple locations are searched, these locations on the map are regarded as the present location candidates of the robot.

We define that point group S is the measurement data at that location and point group M means the points on the known map. Let the point $\boldsymbol{m}_l$ ($1 \leq l \leq N$) that is the nearest point to point $\boldsymbol{s}_d$ ($1 \leq d \leq N$) in point group M be the corresponding point about point $\boldsymbol{s}_d$ in point group S. The robot counts the number of points whose distances between each corresponding point ($|\boldsymbol{s}_d - \boldsymbol{m}_l|$) is less than the threshold value (Fig.10). If the number of these points is larger than the threshold value at that location, the robot regards that location as the present location candidates on the known map. When there are multiple present location candidates, the robot moves from the present location to another location and measures surrounding environment there. The present location candidates after movement are calculated by performing the same processing using the measurement data. The correct location of the robot has consistent relationship between the first and the second location candidates. If there are multiple consistent location, and above process is repeated until only one consistent present location exists.

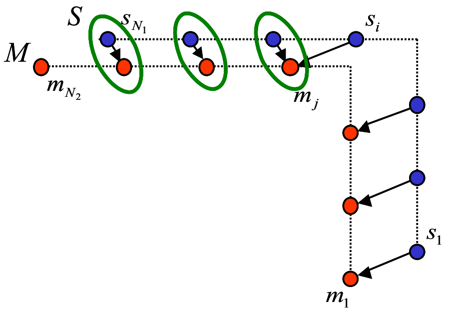
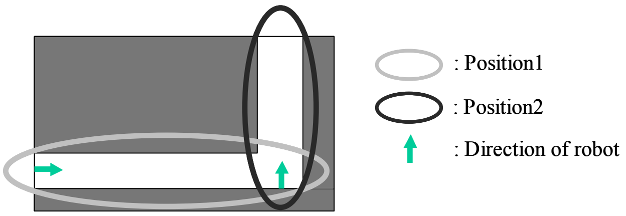


Fig.10 Count of corresponding points

5. EXPERIMENT

5.1 Map Generation

Our mobile robot used two LRF (SICK, LMS 200-30106) that can measure distance by the propagation time of the pulse of laser light, and can measure direct distance between obstacles around the robot and the robot itself. The maximum sensing ranges of distance and angle are 180deg and 30m, respectively. The maximum error of the measurement is 4cm. The resolution of sensing angle is 0.5deg. The robot acquired 27 measurement data by traveling the passage of L-shape of the indoor environment in total (Fig.11).



(a) Whole passage

(b) View from position 1 **(c) View from position 2**

Fig.11 L-shape corridor

Figure 12 shows the alignment result of two measurement data only with the dead-reckoning method. Figure 13 shows the result with our method.

These results show that the correct alignment cannot be performed only with a dead-reckoning. On the other hand, the correct alignment is performed with our method.

Figure 14 shows the map that includes all the measurement data with the dead-reckoning method, while Fig. 15 shows the map that is generated with our method. Table 1 shows the quantitative result about L (passage length) and θ (passage angle).

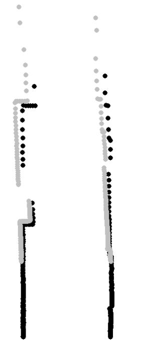


Fig.12 Dead-reckoning

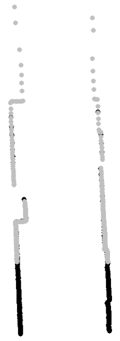


Fig.13 Our method

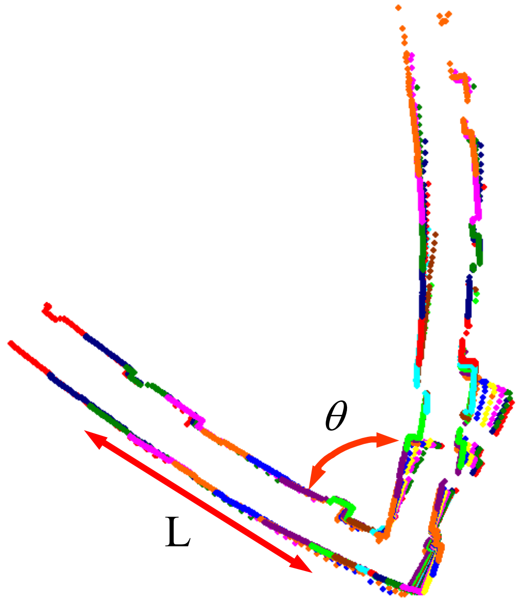


Fig.14 Dead-reckoning

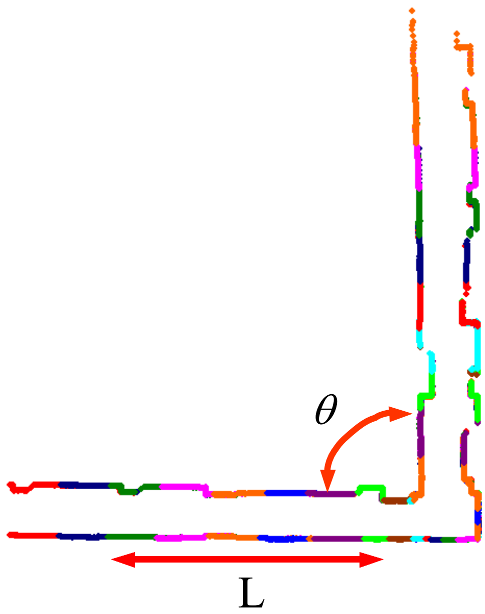


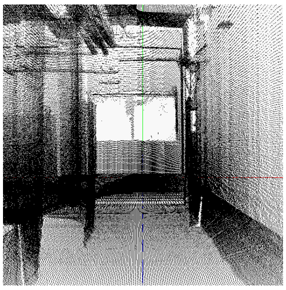
Fig.15 Our method

Table 1 Measurement accuracy

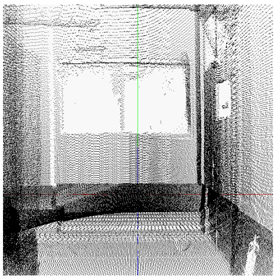
	L[mm]	θ[deg]
True value	9600	90
Dead-reckoning	10385	70
Our method	9830	89

From these results, it is verified that the accuracy of the map that is generated with our method is very high, although the map that is generated only with the dead-reckoning method is distorted.

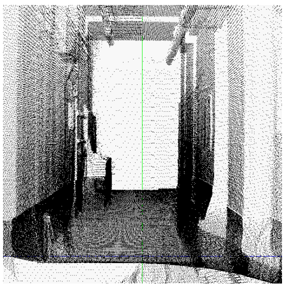
Our method can also generate 3-D map by using two LRFs (Fig.16).


(a) 3-D Map 1


(b) Position 1


(c) 3-D Map 2


(d) Position 2


(e) 3-D Map 3

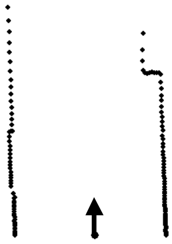

(f) Position 3

Fig.16 Generated 3-D map

5.2 Self-Localization

We also verified the validity of self-localization method by experiment. We displaced the robot after it generated the map of the environment.

Figure 17 (a) shows the measurement data at the location after the displacement, and (b) shows the result of self-localization. The direction of the arrow expresses the estimated direction of the robot.


(a) Sensing result

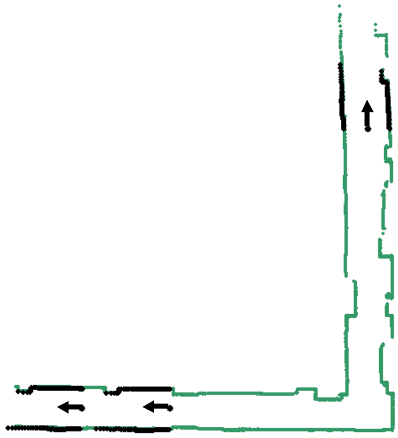
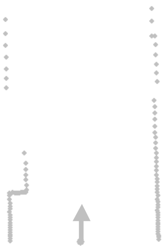

(b) Self-localization

Fig.17 Result of self-localization 1

This result shows that there are multiple present location candidates and the robot cannot estimate its position and orientation uniquely. Therefore, the robot decided to move to realize self-localization. In this case, it rotated 180 degrees from the present location, measured the distance to the surrounding environment, and performed self-localization again (Fig.18).


(a) Sensing result

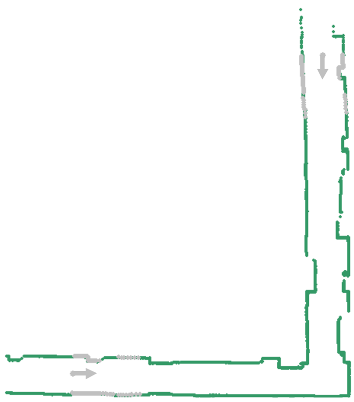

(b) Self-localization

Fig.18 Result of self-localization 2

There are three location candidates in the first sensing and two location candidates in the second sensing. The consistent present location can be determined by integrating these two results (Fig.19).

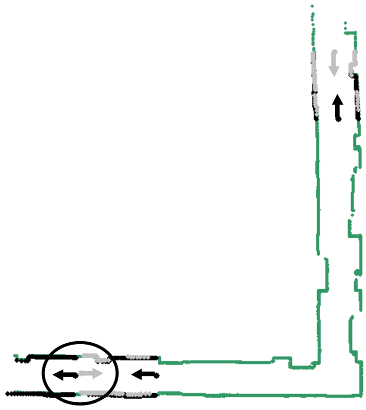


Fig.19 Results of self-localization 1 and 2

From this result, it is verified that our method can work well even if there are multiple similar obstacles around the robot.

6. CONCLUSION

In this paper, we propose the method of generating a 3-D map by using the mobile robot equipped with two LRFs. We also propose self-localization method of the robot. The robot generates the environmental map and estimates its position and orientation by utilizing the method based on ICP algorithm.

From the experimental results, the validity of our map generation method and self-localization method is verified.

As a future works, the computation time of the map generation must be improved. We have to construct the map generation method in dynamic environment, too.

7. ACKNOWLEDGMENT

This research was partly supported by CASIO Science Promotion Foundation, Japan.

8. REFERENCES

[1] Sebastian Thrun: "Robotic Mapping: A Survey," *CMU-CS-02-111*, 2002.

[2] Sebastian Thrun, Wolfram Burgard and Dieter Fox: "A Real-Time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3D Mapping," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pp.321-328, 2000.

[3] Sebastian Thrun, Dirk Hahnel, David Ferguson, Michael Montemerlo, Rudolph Triebel, Wolfram Burgard, Christopher Baker, Zachary Omohundro, Scott Thayer and William Whittaker: "A System for Volumetric Robotic Mapping of Abandoned Mines," *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pp.4270-4275, 2003.

[4] John J. Leonard and Hugh F. Durrant-Whyte: "Simultaneous Map Building and Localization for an Autonomous Mobile Robot," *Proceedings of the 1991 IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pp.1442-1447, 1991.

[5] Michael Montemerlo, Sebastian Thrun, Daphne Koller and Ben Wegbreit: "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.

[6] Nicola Tomatis, Illah Nourbakhsh and Roland Siegwart: "Hybrid Simultaneous Localization and Map Building: A Natural Integration of Topological and Metric," *Robotics and Autonomous Systems*, Vol.44, pp.3-14, 2003.

[7] Masahiro Tomono: "A Scan Matching Method using Euclidean Invariant Signature for Global Localization and Map Building," *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pp.866-871, 2004.

[8] Masahiro Tomono: "Robust Robot Localization and Map Building using a Global Scan Matching Method," *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.1059-1064, 2004.

[9] Masahiro Tomono: "3-D Localization and Mapping Using a Single Camera Based on Structure-from-Motion with Automatic Baseline Selection," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp.3353-3358, 2005.

[10] Masahiro Tomono: "Efficient Global Scan Matching Using Saliency-based Scan Point Resampling," *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.1433-1438, 2005.

[11] Paul J. Besl and Neil D. McKay: "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.14, No.2, pp.239-256, 1992.