

Indoor SLAM based on line observation probability using a hand-drawn map

Akiba Keigo (✉ akiba@sensor.mech.chuo-u.ac.jp)

Chuo University - Korakuen Campus: Chuo Daigaku - Korakuen Campus <https://orcid.org/0000-0001-7046-8272>

Ryuki Suzuki

Chuo University - Korakuen Campus: Chuo Daigaku - Korakuen Campus

Yonghoon JI

JAIST: Hokuriku Sentan Kagaku Gijutsu Daigakuin Daigaku

Sarthak Pathak

Chuo University Faculty of Science and Engineering Graduate School of Science and Engineering: Chuo Daigaku Rikogakubu Daigakuin Rikogaku Kenkyuka <https://orcid.org/0000-0002-5271-1782>

Kazunori Umeda

Chuo University Faculty of Science and Engineering Graduate School of Science and Engineering: Chuo Daigaku Rikogakubu Daigakuin Rikogaku Kenkyuka <https://orcid.org/0000-0002-4458-4648>

Research Article

Keywords: SLAM, Mobile robot, Hand-drawn map, Line observation

Posted Date: January 3rd, 2023

DOI: <https://doi.org/10.21203/rs.3.rs-2336414/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

RESEARCH

Indoor SLAM based on line observation probability using a hand-drawn map

Keigo Akiba^{1*}, Ryuki Suzuki¹, Yonghoon Ji², Sarthak Pathak³ and Kazunori Umeda³

Abstract

In this paper, we propose a novel method to build indoor map information by utilizing a hand-drawn map as prior information is given. So far, previous studies using the hand-drawn map have been limited to robot pose estimation and navigation. In our approach, finding the correspondence between the shape of the real environment and the shape of the hand-drawn map is performed to build accurate map information. In addition, even if the estimation of the robot pose on the hand-drawn map fails, our method can continue to build the map by re-estimating the robot's pose on the hand-drawn map based on the previous corresponding information. In the simulations, we verified the accuracy of the built map using three hand-drawn maps.

Keywords: SLAM; Mobile robot; Hand-drawn map; Line observation

1 Introduction

The use of autonomous mobile robots to replace human workers is currently attracting attention. In recent years, a remarkable shortage of manpower has been caused by a decrease in the working population. Therefore, the importance of these robots is expected to continue to increase and automation by robots. In our immediate surroundings, autonomous mobile robots have been introduced in many indoor facilities such as factories, shopping malls, and airports for movement and guidance. Simultaneous localization and mapping (SLAM) technology is indispensable for the operation of autonomous mobile robots. SLAM is a technology that generates a map of the environment for robot operation by simultaneously constructing a map and estimating the robot pose. In order to improve the accuracy of map building, several methods that utilize prior information have been proposed. However, it is difficult to obtain detailed maps or prior information with accurate dimensions in large-scale facilities such as airports. In general, floor maps posted in facilities show the location of rooms and other facilities in an easy-to-understand layout; however, the scale, shape, and aspect ratio are often incorrect, making it difficult to use them as prior information. Therefore, novel methods using prior information that is available to everyone and highly versatile are required. Machinaka *et al.* estimated the robot pose based on

Google Maps as the prior information [1]. However, this method requires time to prepare the prior information and adjust the parameters which are tedious tasks.

On the other hand, it is relatively easy to prepare a hand-drawn map based on human prior knowledge. Bahram *et al.* suggested Monte Carlo localization (MCL) [2] that is possible on the hand-drawn map in order to estimate not only the robot pose but also the scale of the map at the same time [3]. Additionally, many methods for operating autonomous mobile robots using the hand-drawn map have been proposed [4, 5, 6]. However, these methods are limited to the navigation and localization of mobile robots on the hand-drawn map, and no mapping has been implemented.

In this respect, we propose a map building framework for indoor use that calculates the probability of line observation on the hand-drawn map and using it for SLAM. Specifically, we estimate the robot pose using the error of straight lines in which lines in the real environment correspond with lines on the hand-drawn map, as shown in Fig. 1. In addition, we use the probability of line observation to re-estimate the robot pose. As described above, the use of hand-drawn maps for map building not only enables highly accurate map building but also allows the environment to be understood in advance. This has the advantage of automating all route planning and robot control performed by humans during map construction.

The remainder of this paper is organized as follows. Section II discusses the overall structure of the pro-

*Correspondence: akiba@sensor.mech.chuo-u.ac.jp

¹Graduate of Science and Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan

Full list of author information is available at the end of the article

posed framework for mapping with the hand-drawn map, and calculations of the probability of line observation is explained in Section III. Section IV presents the re-estimation of the robot pose on the hand-drawn map with probability of line observation. Section V describes the method for the correcting the robot pose in SLAM. Our proposed method and the simulation results are discussed in Section VI. Finally, Section VII presents the conclusions of this article and future work.

2 Overview

The proposed method assumes a wheeled mobile robot equipped with a laser range finder (LRF). The overall process of the proposed framework is shown in Fig. 2. The proposed method can be divided into two main parts: the process on a hand-drawn map and the process in the real environment.

The process on the hand-drawn map is as follows. First, the straight lines are extracted from the hand-drawn map by using the probabilistic Hough transform. Next, we use the method by Bahram *et al.* [3] to estimate the robot pose on the hand-drawn map. It uses particles including map scale information as state variables to estimate the robot pose and the scale of the hand-drawn map at the same time. Then, we apply ray tracing to acquire the pseudo-range data. At this time, correspondences between straight lines on the hand-drawn map and the range data are simultaneously calculated. This correspondence is calculated by nearest neighbor search by Euclidean distance. Meanwhile, if the MCL on the hand-drawn map [3] is not performed correctly, the recovery process to re-estimate the robot pose on the hand-drawn map is executed. When the estimated robot position reaches a wall or other non-running area on the hand-drawn map, this MCL cannot be performed correctly. The detailed process is presented in Section IV.

The process in the real environment is as follows. First, we extract straight lines using random sample consensus (RANSAC) from the actual range data observed by the LRF mounted on the robot. Then, we calculate the correspondence between the straight lines on the hand-drawn map and the straight lines observed by the robot. After that, we calculate the observation probabilities for each straight line by finding the correspondence between the pseudo-range data and the actual range data. A detailed description of the line observation probabilities is given in Section IV. Next, the actual robot orientation and position are respectively estimated based on straight lines and a scan matching process. The detailed estimation methods are described in Section IV. Finally, map building is performed based on the estimated actual robot position and orientation.

3 Hand-drawn map processing

This section presents MCL-based pose estimation process of the robot on a hand-drawn map and the re-estimation process when the estimated pose is corrupted.

3.1 Monte Carlo localization in hand-drawn maps

MCL in hand-drawn maps is able to output the robot pose x_t and the scale s_t of the hand-drawn map at the same time. The process can be divided into prediction, updating, resampling, and estimation. There are some differences in the likelihood calculations in the prediction and updating processes, compared to conventional MCL, as follows.

3.1.1 Prediction process

The prediction process shifts the distribution of particles based on the control input u_t and the motion model to obtain the state x_t at the current time. In addition, the scale s_t of the hand-drawn map is predicted. In this process, an uncertainty is applied to each component of x_t^i that follows a specific distribution. Each prediction step is expressed as follows.

$$x_t^i = F_k x_{t-1}^i + s_t^{i-1} (u_t + G_t w_t^i) \quad (1)$$

where x_t^i is the state of the i -th particle. F_k is Matrix of linear models for state transitions of the system, u_t is control input, and G_t is Matrix of a noise model for time transitions. w_t^i is its noise following a multivariate normal distribution.

$$s_t^i = s_{t-1}^i + e^1 \quad (2)$$

where e^i is noise of the i -th particle relative to the scale of the hand-drawn map.

3.1.2 Update process

The update process takes into account the scale of the hand-drawn map in the likelihood calculation. The measurement model $p(\frac{z_t}{s_t^i} | x_t)$ and likelihood calculation model are shown in Eq. (3). Unlike the general likelihood calculation model, the scale s_t of the hand-drawn map is taken into account for the measured values z_t and the likelihood function $g(p, \mu, s)$. Specifically, the likelihood is calculated by using each distance value from ray tracing and observations measurement to obtain a measurement model. Here, the distance values from ray tracing consider the scale s_t . In addition, in the linear observation probability explained in Chapter 4, the observation probability of

a straight line is calculated by using the distance between each point of the pseudo-observation data by ray tracing and the observation data in the real environment for the likelihood function in Eq. (3).

$$p\left(\frac{z_t}{s_t^i} \mid \mathbf{x}_t^i\right) = p\left(\frac{z_{t-1}}{s_{t-1}^i} \mid \mathbf{x}_{t-1}^i\right) g(l, \mu, s)$$

$$g(p, \mu, s) = \frac{1}{\sqrt{2\pi\sigma_g^2}} \exp\left\{-\frac{\left(\frac{l_t^i(j)}{s_{t-1}^i} - \mu_t(j)\right)^2}{2\sigma_g^2}\right\} \quad (3)$$

where $\mu_t(j)$ is distance (observations) and $l_t^i(j)$ is distance (ray-tracing). σ_g^2 is distance dispersion. $g(p, \mu, s)$ is likelihood function at a distance of one ray.

3.2 Recovery from failure on hand-drawn map

When the MCL on the hand-drawn map [3] is not performed well, it is possible to recover the robot pose based on the straight lines considered to have been observed in the previous frame (i.e., probability $f(r)$ in Eq. (4) exceeds the threshold). When the estimated robot position reaches a wall or other non-running area on the hand-drawn map, this MCL is not performed correctly. The recovery process is divided into two parts: the preprocessing of the hand-drawn map, and the determination of the observation points to redistribute particles for MCL [3] based on observation points.

In the pre-processing, candidate observation points R_i are spread out at random. Then, we register the indices of the lines on the hand-drawn map that can be observed at each candidate observation point as shown in Fig. 3. Here, black and gray areas represent the walls (i.e., occupied areas) and unoccupied areas, respectively. Yellow circles represent candidate observation points, and red arrows represent pseudo-range data.

Next, in the process of determining observation points, we calculate which line on the hand-drawn map is being observed at each frame. This means that even if the MCL [3] is not successful, the robot pose is likely to be in the vicinity of the point where it can observe the line on the hand-drawn map that is observed just before. Therefore, we can recover the robot pose by finding multiple observation points that can observe the straight line on the hand-drawn map observed by the robot in the previous frame, spreading particles around these points, and executing MCL [3] again.

4 Line observation probability

In this method, the following equation for calculating the probability $f(r)$ of the existence of each line is used to determine whether each line on the hand-drawn map has been observed.

$$f(r) = \frac{1}{\sqrt{2\pi\sigma_g^2}} \exp\left\{-\frac{(r^i - sl_i)^2}{2\sigma^2}\right\} \quad (4)$$

where r_i and l_i are the i -th points of the actual range data and the pseudo-range data from the robot pose on the hand-drawn map, respectively. Index i denotes the order of the corresponding range data. σ is the uncertainty of the LRF. s is the scale of the hand-drawn map estimated in each frame. The correspondence between the actual range data and the pseudo-range data is obtained as the index of the range data. When the observation probability in Eq. (4) exceeds the threshold, the straight line is considered to have been observed.

5 Sequential SLAM by pose correction

With the observed lines on the hand-drawn map and the lines observed in the real environment, the following equation is used to correct the robot orientation.

$$\theta = \frac{1}{N} \sum_{i=1}^M (\theta_{k_i} - \theta_i) \quad (5)$$

where θ is the amount of correction for the robot in the rotation, and k_i is the index of the line on the hand-drawn map corresponding to the i -th observation line. M is the number of lines for which correspondence has been calculated. Next, the robot position is also corrected by the iterative closest point (ICP) [7] which is a well-known scan matching process. The amount of movement in the translational is calculated as the sum of the squares of the distances between the range data E , as follows.

$$E = \min \sum_{i=1}^N \|\mathbf{n}^T (\mathbf{y}_{u_i} - (\mathbf{x}_i + \mathbf{T}))\|^2 \quad (6)$$

where x and y are the range data measured in the previous frame (i.e., the target range data) and the current frame (i.e., the source range data) respectively. n is the normal vector. N and u_i are the number of points in the source range data and the index of the source range data corresponding to the i -th point in the target range data, respectively. T is the vector of translational components of the transformation matrix.

6 Experimental Evaluation

6.1 Experiment Conditions

To verify our method, we conducted simulations in the environment shown in Fig.4 (a) and Fig.5 (a). Hand-drawn maps were easily created using the Windows 10 built-in software “Paint.” We defined one pixel in created hand-drawn maps as 10 cm. The maximum range distance of the virtual LRF was set to 50 m, and the error of the virtual encoder was defined by using a normally distributed random number.

Fig.4 (b) shows the hand-drawn maps used in this simulation (Fig. 4(a)). In hand-drawn map A, curves are conspicuous because of the messy drawing. On the other hand, the hand-drawn map B has a discrepancy in the horizontal ratio of the shape on the left as compared to the simulation environment. Hand-drawn map C also has discrepancies in the horizontal ratio of the left shape and the vertical ratio of the path leading to the upper region.

The hand-drawn maps used in the experiment with the environment in Fig. 5(a) are shown in Figs. 5(b). Compared to the environment in Fig. 4(a), a more complex, vast, and the real indoor environment was assumed. The hand-drawn map shown in Fig. 5 (b) has deviations in the horizontal proportions of the upper left road and the vertical proportions of the road leading from near the center to the lower side.

6.2 Simulation results

In the simulations, we evaluated the accuracy of our sequential SLAM process in comparison with SLAM by odometry and conventional ICP. Figures 6 7 show the constructed maps. Tables 1 2 show the average deviation of the constructed maps from the true values. In simulations 1 and 2, the errors exceeded 1 m when the hand-drawn maps were not used. On the other hand, when hand-drawn maps were used, the error was around 0.3 m in both cases. These results show that the use of hand-drawn maps can greatly reduce the error.

First, we describe the experiment in the environment shown in Fig. 4 (a). In the case of using hand-drawn map A and hand-drawn map B, maps were built without the failure of MCL [3]. However, in the case of using the hand-drawn map C, the MCL [3] broke down in the middle region because the aspect ratio of the corridor on the hand-drawn map differs greatly; thus, the recovery process described in Section IV was performed as shown in Fig. 8. Here, the green line represents the line exceeding the threshold of the observation probability. Red points are particles for the MCL process.

As a result, a reliable map was built by our proposed framework using the hand-drawn map. Note that a

previous study [4] mentioned that robot pose estimation cannot be done correctly when the aspect ratio of the real environment and the hand-drawn map is significantly different. On the other hand, when hand-drawn map B was used, the MCL [3] did not break down, although there was a large discrepancy in the aspect ratio as shown in Fig. 6(c). This is because there is no significant difference in the aspect ratio of the map, although the size of the entire map is different.

Next, we describe the experiment in the environment shown in Fig. 5 (a). As shown in Fig.7 (b), (c), and (d), highly accurate map building was achieved for both hand-drawn maps shown in Fig.5 (b). Although the size and aspect ratio of each hand-drawn map differed in some areas, there were no significant differences.

On the other hand, near the upper left corner in Fig. 5 (a), the robot pose on the hand-drawn map sometimes advanced into the inaccessible area, as shown in Fig. 9. In such a narrow area, even a slight deviation in the estimation of the robot pose on the hand-drawn map would cause the robot to move into inaccessible areas. In addition, if the pathway is wide and there are only a few observation lines on one side of the pathway, as shown in Fig. 5 (a), the re-estimation will not be accurate, and the method will fail. Therefore, if the area for robot operation in the real environment is narrow, we should draw a larger area on the hand-drawn map in that location. That way, there are fewer cases where the robot moves into the inaccessible area. Actually, the accessible area near the upper left turn is depicted widely in Fig. 5 (b), and (d). There were no cases where the robot moved into an inaccessible area when changing direction. From the above discussion, if the hand-drawn map is intentionally drawn with a wider running area, it was found that the system is less likely to fail.

From the above results, our proposed method is able to effectively make use of information contained in hand-drawn maps to achieve accurate SLAM. In particular, it achieved map construction with high accuracy in both the simple environment of simulation1 and the complex environment of simulation2. In addition, when the self-position estimation in the hand-written map was not successful, the re-estimation process corrected the error, demonstrating the high robustness of the system. In summary, we were able to demonstrate the high practicality of hand-drawn maps.

7 Conclusion

In this study, we proposed a novel SLAM scheme for indoor use based on the hand-drawn map by matching line information. In the simulations, we were able to

build maps with higher accuracy by using hand-drawn maps.

By using hand-drawn maps, which can be created by anyone and easily used as prior information, a highly versatile map construction method was realized. In particular, the practicality of hand-drawn maps was demonstrated by achieving highly accurate map construction in an environment similar to an actual, complex indoor environment.

In the future, we will start to develop a system that automatically builds a map of the real environment by obtaining the trajectory from the hand-drawn map using our proposed approach.

Declarations

Availability of data and materials

Not applicable

Competing interests

The authors declare that they have no competing interests.

Funding

Not applicable

Authors' contributions

KA developed the overall system, conducted the experiments, analyzed the data, and wrote the manuscript. RS devised the basic concept of the overall system and conducted the experiments. JY, SP and KU supervised the entire study and the writing of the manuscript. All authors read and approved the final manuscript.

Author details

¹Graduate of Science and Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan. ²Graduate School for Advanced Science and Technology, Japan Advanced Institute of Science and Technology (JAIST), 1-1 Asahida, Nomi, Ishikawa, 923-1211, Japan. ³Faculty of Science and Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan.

References

- Kimba, M., Machinaka, M., Kuroda, Y.: Edge-node map based localization without external sensor data. In: IFAC PapersOnLine, vol. 51, pp. 203–208 (2018)
- Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte carlo localization for mobile robots. In: In Proc. of IEEE International Conference on Robotics and Automation (ICRA), pp. 1322–1328 (1999)
- Behzadian, B., Agarwal, P., Burgard, W., Tipaldi, G.D.: Monte carlo localization in hand-drawn maps. In: In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4291–4296 (2015)
- Boniardi, F., Valada, A., Burgard, W., Thrun, S.: Autonomous indoor robot navigation using a sketch interface for drawing maps and routes. In: In Proc. of IEEE International Conference on Robotics and Automation (ICRA), pp. 2896–2901 (2016)
- Boniardi, F., Behzadian, B., Burgard, W., Thrun, S.: Robot navigation in hand-drawn sketched maps. In: In Proc. of IEEE European Conference on Mobile Robotics (ECMR), pp. 1–6 (2015)
- Chen, K., Vázquez, M., Savarese, S., Thrun, S.: Localizing against drawn maps via spline-based registration. In: In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 8521–8526 (2020)
- Low, K.L.: Linear least-squares optimization for point-to-plane icp surface registration. In: Technical Report TR04-004. University of North Carolina, Chapel Hill (2004)

Figure 1 Our proposed method.

Figure 2 Outline of the method.

Figure 3 Determination of observation points. (a) Indices of lines; (b) Observation points.

Figure 4 Simulation environment and hand-drawn maps. (a) Simulation environment1; (b) Hand-drawn maps for simulation.

Figure 5 Simulation environment and hand-drawn maps. (a) Simulation environment2; (b) Hand-drawn maps for simulation.

Figure 6 The determination of observation points. (a) True map; (b) Result of Hand-drawn map A; (c) Result of Hand-drawn map B; (d) Result of Hand-drawn map C; (e) Result of ICP-SLAM; (f) Result of odometry.

Figure 7 The determination of observation points. (a) True map; (b) Result of Hand-drawn map D; (c) Result of Hand-drawn map E; (d) Result of Hand-drawn map F; (e) Result of ICP-SLAM; (f) Result of odometry.

Figure 8 Recovery from failure. (a) Determination of observed straight line; (b) Particle redistribution; (c) Recovery by MCL [3].

Figure 9 Moving into a non-running area. (a) Hand-drawn map; (b) Actual environment.

Table 1 Recognition rate for each position of simulation1.

| Methods | Measurement error [m] |
|------------------|-----------------------|
| Odometry | 6.157 |
| ICP-SLAM | 2.720 |
| Hand-drawn map A | 0.270 |
| Hand-drawn map B | 0.297 |
| Hand-drawn map C | 0.300 |

Table 2 Recognition rate for each position of simulation2.

| Methods | Measurement error [m] |
|------------------|-----------------------|
| Odometry | 1.102 |
| ICP-SLAM | 2.165 |
| Hand-drawn map D | 0.334 |
| Hand-drawn map E | 0.331 |
| Hand-drawn map F | 0.341 |

Figures

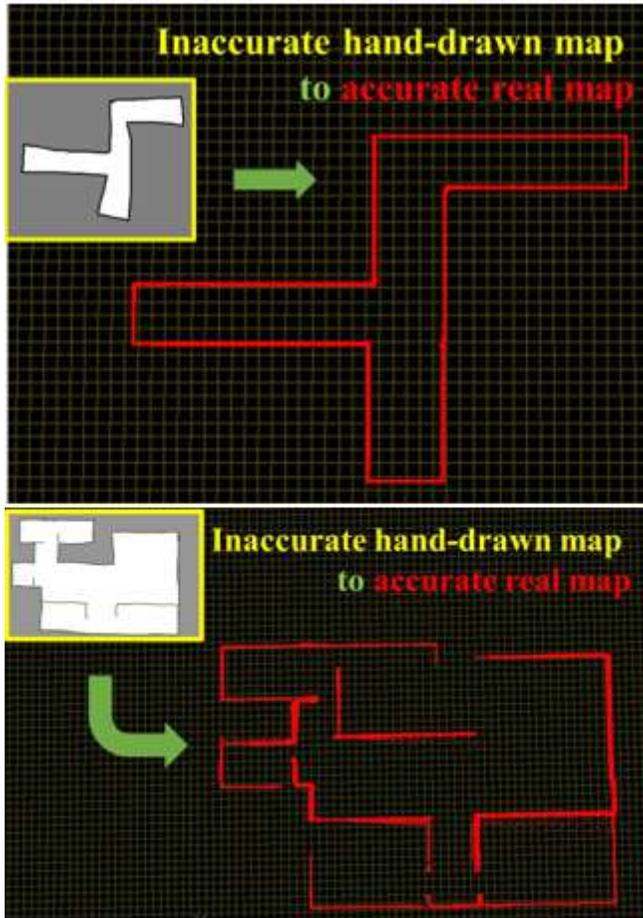


Figure 1

Our proposed method.

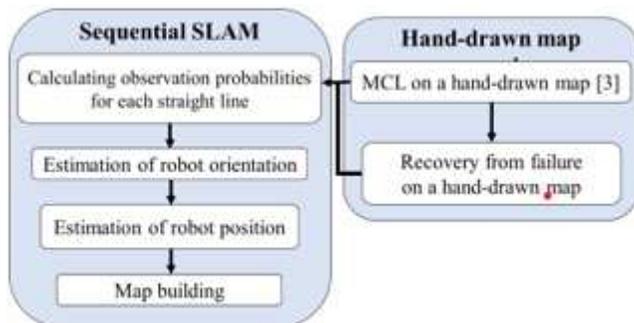


Figure 2

Outline of the method.

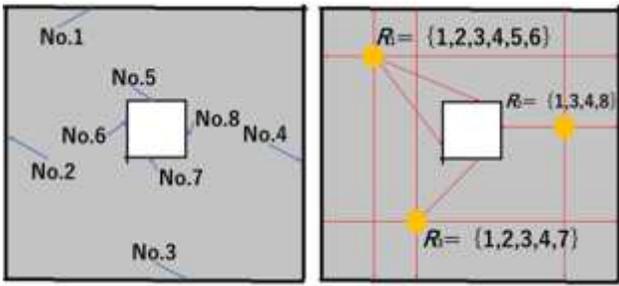


Figure 3

Determination of observation points. (a) Indices of lines; (b) Observation points.

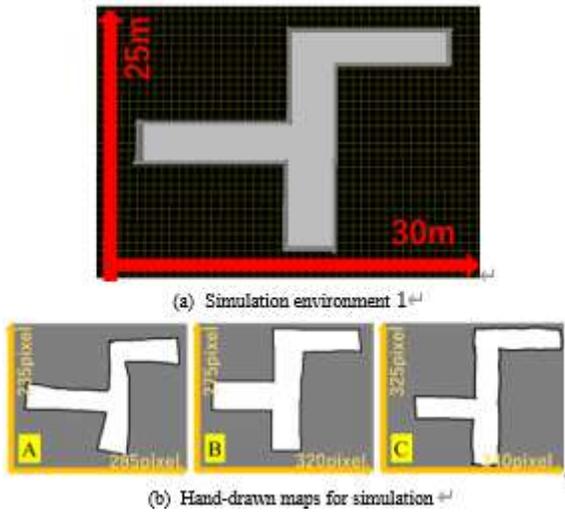
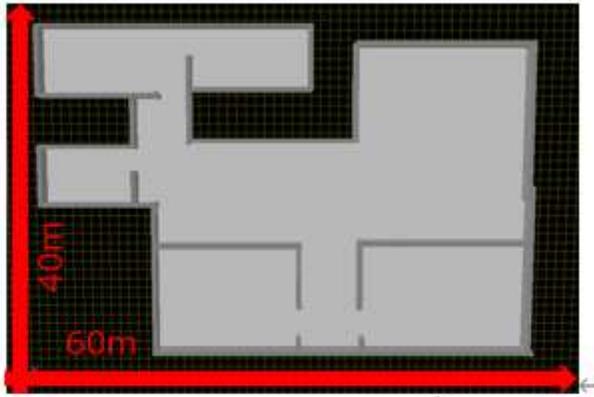
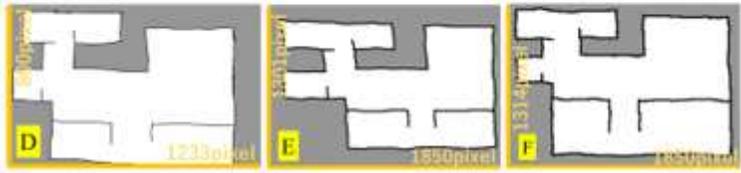


Figure 4

Simulation environment and hand-drawn maps. (a) Simulation environment1; (b) Hand-drawn maps for simulation.



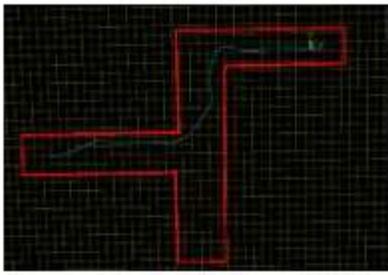
(a) Simulation environment 2 ←



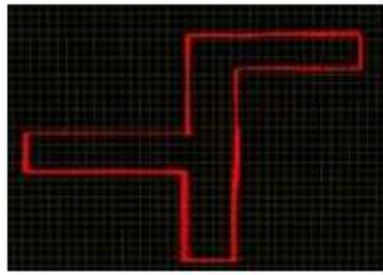
(b) Hand-drawn maps for simulation ←

Figure 5

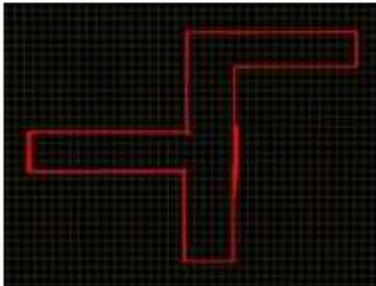
Simulation environment and hand-drawn maps. (a) Simulation environment2; (b) Hand-drawn maps for simulation.



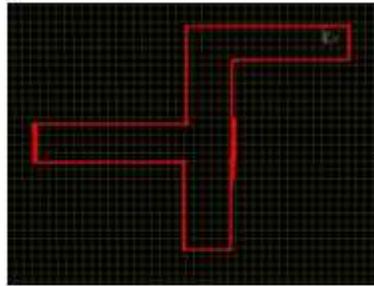
(a) True map



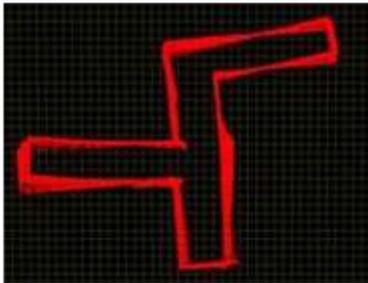
(b) Result of hand-drawn map A



(c) Result of hand-drawn map B



(d) Result of hand-drawn map C



(e) Result of ICP-SLAM



(f) Result of odometry

Figure 6. The determination of observation points.

Figure 6

The determination of observation points. (a) True map; (b) Result of Hand-drawn map A; (c) Result of Hand-drawn map B; (d) Result of Hand-drawn map C; (e) Result of ICP-SLAM; (f) Result of odometry.

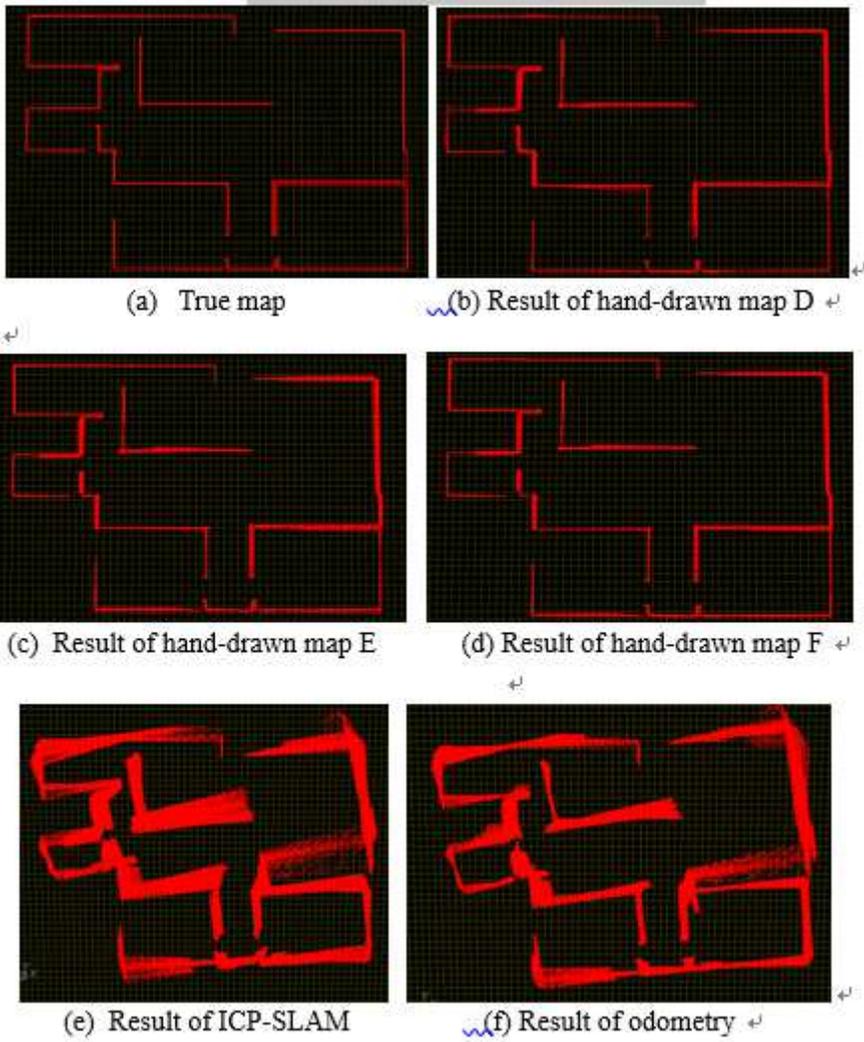


Figure 7

The determination of observation points. (a) True map; (b) Result of Hand-drawn map D; (c) Result of Hand-drawn map E; (d) Result of Hand-drawn map F; (e) Result of ICP-SLAM; (f) Result of odometry.

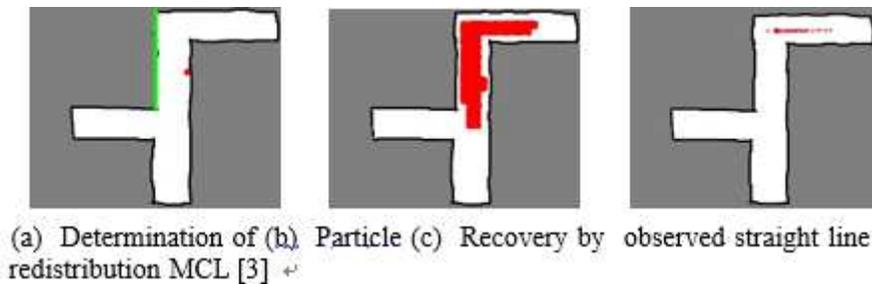


Figure 8

Recovery from failure. (a) Determination of observed straight line; (b) Particle redistribution; (c) Recovery by MCL [3].

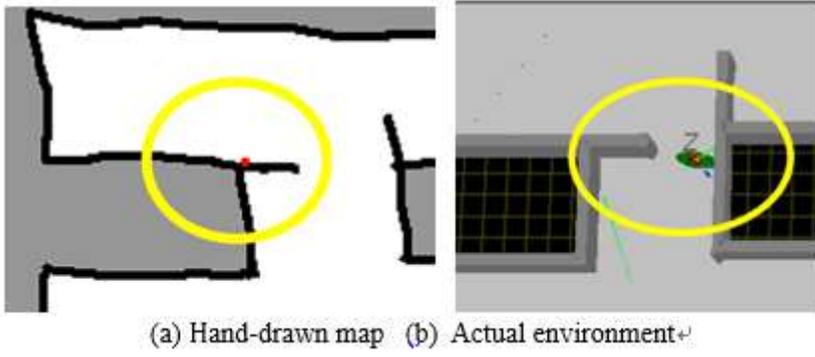


Figure 9

Moving into a non-running area. (a) Hand-drawn map; (b) Actual environment.