2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Outdoor Visual Localization with a Hand-Drawn Line Drawing Map using FastSLAM with PSO-based Mapping

Keisuke Matsuo and Jun Miura
Department of Computer Science and Engineering
Toyohashi University of Technology

***Abstract*— This paper deals with a navigation of a mobile robot in a campus environment using a hand-drawn line drawing building map. Hand-drawn maps often include various types of uncertainty such as incorrect size/position and missing objects, thereby making it difficult to establish correspondence between objects in the map and sensory data. We solve this problem using a SLAM approach with an input hand-drawn map being an initial estimate. The proposed method combines a FastSLAM with a particle swarm optimization for map refinement. The method has been successfully applied to a stereo-based localization in a real scene.**

***Index Terms*— Outdoor navigation, Hand-drawn map, Line drawing map, Particle filter, Particle swarm optimization.**

I. Introduction

Localization is one of the fundamental functions of mobile robots. Vision has increasingly been used as a sensor for outdoor localization and navigation [3]. Previous works on visual localization in outdoor environments can roughly be divided into two categories: landmark-based ones [18], [7], [5], [8] and view-based ones [2], [14]. These works are basically learning-based, that is, the localization is performed using a learned map. This means that the data of the environment must be collected *before* localization.

People often use a line drawing map and can navigate themselves where they have never been. Although only a part of objects such as buildings are usually written in the map, people can determine their position with respect to such objects by matching their knowledge of the map with observations of the environment. Since mobile robots in the future are expected to communicate with people as in the way people do, localization using a line drawing map will be an important capability.

Yun and Miura [22] proposed a localization method using a line drawing building map with uncertainty. They used line segments in the image and vertical planes in stereo data as features for localization. To cope with uncertain correspondence, they adopt a multi-hypothesis approach, which is relatively complicated and costly. Leung, Clark, and Huissoon [11] proposed a localization method based on the matching between detected wall features and the line features extracted from aerial images in conjunction with particle filter. Senlet and Elgammal [19] used aerial images as a prior knowledge of sidewalk positions. Kümmerle et al. [9] developed a graph SLAM method based on a similar idea and an accurate 3D range sensor. Parsley and Julier [17] proposed a general framework of exploiting the use of various prior information with uncertainty in SLAM. These works use relatively accurate maps for localization or SLAM.

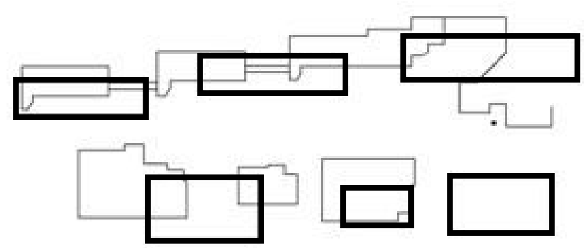


Fig. 1. An example of hand drawn map (bold) superimposed on an accurate map (thin). Building sizes and positions are largely deviated from the true values, but their qualitative placements are correct.

Line drawing maps we usually use have various types of uncertainties. Fig. 1 shows an example of hand-drawn map of our campus. One student of our laboratory drew this map with a GUI-based tool and under the direction to draw only main buildings, given a rough scale of the environment, but without seeing any maps or aerial images (that is, only with his memory)[1]. The bold lines in the map indicate outlines of buildings, superimposed on the true outlines drawn with thin lines. We can guess this map can provide enough information for a human visitor to the university to reach the destination (i.e., one of the buildings) as long as the almost correct starting point is given. It is, however, very challenging for a robot to localize and navigate itself using such a map due to various uncertainties included in the map.

Some previous works deal with navigation using hand-drawn maps. Chronis and Skubic [1] proposed to use sketched maps for directing robot navigation. A route is extracted from a sketched path in the form of landmark states with corresponding actions (e.g., "turn the next corner"). This work is adequate for a simple environment, but does not deal with real complex environments. Yun and Miura [23] analyzed hand-drawn maps and extracted four key uncertainties in them, that is, those in *dimension*, *position*, *shape*, and *existence*, all of which are included in the above map shown in Fig. 1.

We divide the above uncertainties into two categories: existence and geometric. The existence uncertainty means some of objects are not drawn in the map; the above map

[1]Note that whether a map is drawn by freehand is not an issue here. The important point is that this map is drawn without any references.

includes only main buildings but not other objects such as small structures and trees. The geometric uncertainty is the one regarding geometric properties of buildings.

We have developed a stereo-based Monte Carlo localization method [13] that can cope with the existence uncertainty. The method uses a map which describes ***geometrically-correct*** outlines of buildings, and addresses the issue of making correspondence between the lines and stereo data. We proposed to use a view-based object recognition for extracting only stereo data from buildings for better correspondence. The method is extended in this paper so that it can cope also with the geometric uncertainty, especially ones in dimension and position.

The proposed method takes a SLAM-based approach. A hand-drawn line drawing map is used as an initial estimate for building position and dimension and is gradually refined through a stereo-based SLAM process in accordance with the estimation of robot trajectory. We implement the method using a FastSLAM combined with a Particle Swarm optimization (PSO) for map refinement. This novel algorithm called FastSLAM/PSOM (FastSLAM with PSO-based Mapping) has been successfully applied to outdoor localization using fairly rough line drawing maps.

The rest of the paper is organized as follows. Sec. II briefly explains our previous localization method. Sec. III explains a PSO-based mapping and the FastSLAM/PSOM algorithm. Sec. IV shows experimental results. Sec. V describes conclusions and discusses future work.

II. STEREO-BASED MONTE CARLO LOCALIZATION USING A LINE-DRAWING MAP

This section briefly explains our stereo-based Monte Carlo localization method [13]. The issue here is how to make correspondence between visual data and a line drawing map.

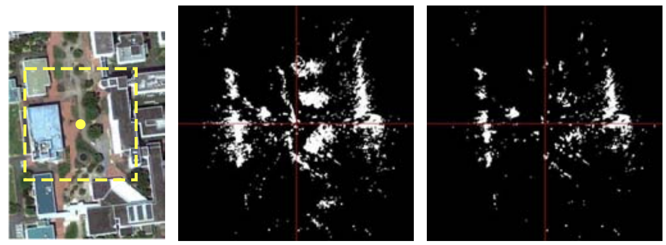
A. Stereo data acquisition and building extraction

Lines in the map represent building wall positions. It is therefore necessary to reliably detect walls for map matching. In indoor corridor scenes, it is relatively easy to extract wall positions using conventional 2D laser range finders. In usual outdoor scenes, however, buildings are sometimes obstructed by various objects such as trees and bushes, and may be difficult to extract reliably due to a fixed-height scanning plane. High-definition 3D laser range finders can be used instead, but they are still expensive.

We therefore use a stereo camera (Point Grey Research, Bumblebee XB3, $66\,[deg.]$ horizontal field of view) for range measurements. The camera is mounted on a pan-tilt head (FLIR Motion Control Systems PTU-D46) and stereo data are taken at six panning positions while the robot is stopping. This gives the robot an omnidirectional view for reliable localization.

To extract stereo data only from buildings, we use an SVM-based classifier proposed by Miura and Yamamoto [14], which has been shown to be effective in view-based localization under various weather and seasons. We divide an input image into 16×16 windows and classify each

Fig. 2. Building extraction results. Windows with purple "X" marks are judged as building.



(a) Scene and observation position.

(b) Local map without building extraction.

(c) Local map with building extraction.

Fig. 3. Effect of building extraction on the local map.

window using several image features; for building, we use the normalized color (r, g, b), an edge density, the peak value of the voting in Hough transform, and the variance of edge directions. Fig. 2 shows some building extraction results. Although some regions other than buildings are also extracted, the overall result is acceptable because a complete classification is not a necessary condition for Monte Carlo localization.

B. Local map generation

The obtained stereo data are converted into a 2D local grid map for matching with a line drawing map. The local map is robot-centered and the size of each cell is $0.1\,[m] \times 0.1\,[m]$. Each of stereo data points, which is originally represented in the camera coordinates, is transformed into the robot local coordinates and then voted on the grid map. Each cell accumulates the votes. Finally, a Gaussian smoothing is applied to the local map to consider the discretization of the grid map. Fig. 3 shows the local map with and without building extraction, generated from the image set shown in Fig. 2; many of the data from large trees are deleted correctly in Fig. 3(c).

C. Monte Carlo localization

The state vector to be estimated is represented by a 2D robot pose in the world coordinates. We use a standard Monte Carlo localization algorithm [20]. Likelihood of each particle is given by comparing the local grid map and the input line drawing map. Using the position of a particle, lines visible from that point are calculated and mapped onto the local map using the pose of the particle. The values in the local map are first summed up where the mapped lines exist. This summed value is discounted by the effective ratio of lines, which is the ratio of pixels on the lines with non-zero values to those with zero values, to be used as a likelihood.

III. FastSLAM with Particle Swarm Optimization-based Mapping

Localization using a hand-drawn building map requires correspondence between buildings in the map and those in the scene. Since the map includes a large amount of uncertainty, however, this correspondence is not easy to establish without correction of the map. We thus apply a SLAM (simultaneous localization and mapping) approach to solve this problem, with the input hand-drawn map being an initial estimate.

A. FastSLAM

The full SLAM problem is to estimate the following posterior [20]:

$$p(\boldsymbol{x}_{1:t}, \boldsymbol{m} \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}), \tag{1}$$

where $\boldsymbol{x}_{1:t}$ is a sequence of robot poses, $\boldsymbol{m}$ is a map, $\boldsymbol{z}_{1:t}$ is a sequence of observations, and $\boldsymbol{u}_{1:t}$ is that of control commands. We then factorize this expression as follows:

$$\begin{aligned} &p(\boldsymbol{x}_{1:t}, \boldsymbol{m} \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}) \\ &\quad = p(\boldsymbol{x}_{1:t} \mid \boldsymbol{z}_{1:t}, \boldsymbol{u}_{1:t}) \cdot p(\boldsymbol{m} \mid \boldsymbol{x}_{1:t}, \boldsymbol{z}_{1:t}). \end{aligned} \tag{2}$$

This factorization decomposes the full SLAM problem into two sequential estimation problems: (1) estimating the robot *path* using observation and control sequences and (2) making a map with *known* robot path and observations. This greatly reduces the computational cost of the SLAM problem. Particle filters based on this factorization are called ***Rao-Blackwellized particle filters***, and SLAM algorithms using this type of particle filters are collectively called FastSLAM [16], [4].

B. Particle Swarm Optimization

Bayes filters such as the Kalman filter are usually used for the map estimation part of FastSLAM. The initial distribution of the map parameters (e.g., landmark positions) is given from an initial guess or some background knowledge; this distribution should include the true parameter in it in the Bayesian formulation. In our hand-drawn map-based navigation, however, building positions and shapes in the map may sometimes be very largely deviated as shown in Fig. 1, and if we use a distribution large enough to cover possible deviation of building parameters, it may fail to converge due to a too large search space. We therefore propose to use Particle Swarm Optimization (PSO) [6], [21] as an approximation of map state estimate.

PSO is an optimization algorithm based on the evolutionary computation paradigm and can efficiently search a large space for optimal values. There are several attempts to apply PSO for SLAM or mapping problems. PSO is used for map estimation [12] or pose estimation [15] for an alternate estimation of pose and map. Lee et al. [10] used PSO to improve the proposal density in the motion estimation phase of FastSLAM. We use PSO for map refinement in the FastSLAM framework.

PSO controls a set of particles in a state space by adjusting the velocity vector of each particle. Each particle makes use of its history as well as the knowledge gained by the swarm as a whole. The outline of a basic PSO algorithm is as follows [21]:

1) Start with an initial set of particles, distributed according to the initial knowledge.
2) Calculate the velocity vector for each particle in the swarm.
3) Update the position of each particle using the velocity vector.
4) Go to step 2 and repeat until a termination condition is satisfied.

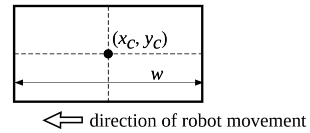


Fig. 4. Parameters representing a building.

The position update is carried out by:

$$\boldsymbol{x}^i_{k+1} = \boldsymbol{x}^i_k + \boldsymbol{v}^i_{k+1}, \tag{3}$$

where $\boldsymbol{x}^i_{k+1}$ represents the position of particle i at iteration $k+1$ and $\boldsymbol{v}^i_{k+1}$ represents the corresponding velocity vector.

A commonly used scheme for updating the velocity vector is:

$$\boldsymbol{v}^i_{k+1} = w\boldsymbol{v}^i_k + c_1 r_1 (\boldsymbol{p}^i - \boldsymbol{x}^i_k) + c_2 r_2 (\boldsymbol{p}^g_k - \boldsymbol{x}^i_k), \tag{4}$$

where r_1 and r_2 are random numbers between 0 and 1, $\boldsymbol{p}^i$ is the best position found by particle i so far, and $\boldsymbol{p}^g_k$ is the best position in the swarm at time k.

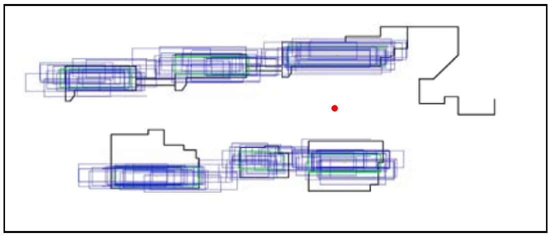
There are three parameters in eq. (4). The inertial parameter w determines PSO's global/local behavior. Two ***trust*** parameters, c_1 and c_2, determine how much confidence the current particle has in itself (c_1) and in the swarm (c_2).

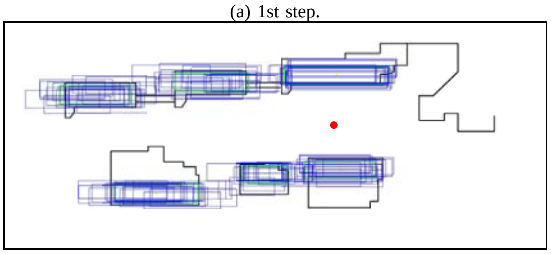
C. PSO for map estimation

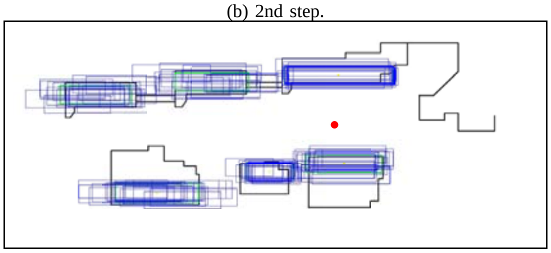
We here deal with a scene where buildings are placed in two rows, as shown in Fig. 1, and the robot moves between them. We then represent each building by its position (x_c, y_c) and width w as shown in Fig. 4. Each PSO particle moves in this three-dimensional search space. The initial swarm (i.e., a set of PSO particles) is determined by a Gaussian sampling using the values in the input hand-drawn map as means. The variances are determined by analyzing a set of map samples actually drawn by several people.

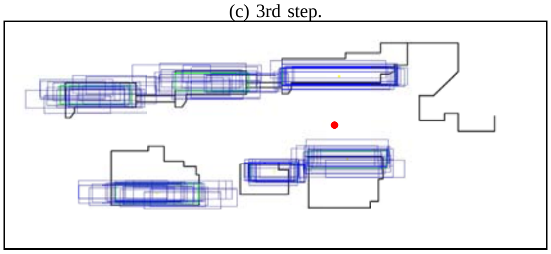
PSO is performed as one of the steps of the FastSLAM. We choose buildings which are considered visible from the current robot position, and perform PSO only for them; particle sets for the other buildings are kept unchanged. The visibility is judged based on the mean distance between the robot and a building and on the measurable range of stereo.

Since the PSO iteration is one of the steps in the outer SLAM iteration, it is not desirable to run PSO until a complete convergence; more information will possibly be obtained in subsequent observations. We therefore set a maximum number of PSO iterations in order to keep a certain level of diversity of PSO particles after each observation.


(a) 1st step.


(b) 2nd step.


(c) 3rd step.


(d) 4th step.

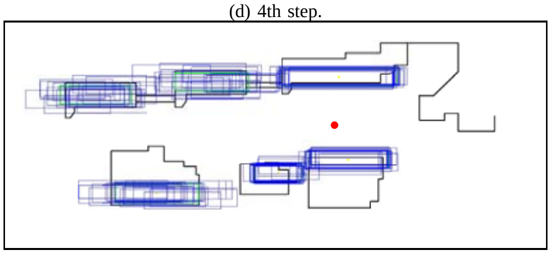

(e) 5th step.

Fig. 5. Building parameter estimation using PSO. The rightmost two buildings and the lower-middle one are considered visible and updated.

Fig. 5 shows an example of PSO process. The robot position is indicated by a red mark and only three buildings on the right are judged as visible and are refined. We used the following parameter values: $w = 0.5$, $c_1 = 0.8$, $c_2 = 1.0$. The number of PSO particles in one swarm was 10 and the maximum number of iteration was set to 10.

D. FastSLAM/PSOM algorithm

We call our new SLAM algorithm FastSLAM/PSOM (FastSLAM with PSO mapping). This algorithm manages two kinds of particles, SLAM particles and PSO particles. Each SLAM particle holds a robot pose and a map. A map is a set of swarms, each of which consists of PSO particles representing the corresponding building parameters.

Algorithm 1: FastSLAM/PSOM

1 Input a hand-drawn map;
2 **for** *each SLAM particle* **do**
3   Generate the initial set of swarms by a probabilistic sampling;
4 Get observation and odometry;
5 **for** *each SLAM particle* **do**
6   Sample the current pose based on the odometry reading;
7   Determine a set $B_{visible}$ of visible buildings;
8   **for** *each building* $b \in B_{visible}$ **do**
9   Run PSO;
10   Update the best latest map;
11   Calculate the likelihood based on the current pose and the latest map;
12 Resample the SLAM particles;
13 Go back to *line* 4;

TABLE I

PROCESSING TIME PER FRAME.

# of PSO particles	# of SLAM particles	processing time [sec.]
10	50	26
10	100	46
10	150	64
20	50	50
20	100	106
20	150	126

Alg. 1 shows the algorithm of FastSLAM/PSOM, which is similar to usual FastSLAM algorithms, but has an inner loop for PSO map estimation.

To calculate the likelihood (*line* 11), we use the *best* latest map for each SLAM particle. The best map is generated by extracting the PSO particle with the highest likelihood from each swarm and compiling them into one map.

In resampling step (*line* 12), we avoid to generate infeasible estimates by setting constraints on particle status, that is, we do not accept SLAM particles which cause the following two cases: (1) two buildings overlap with each other; (2) the robot intersects with one of the buildings.

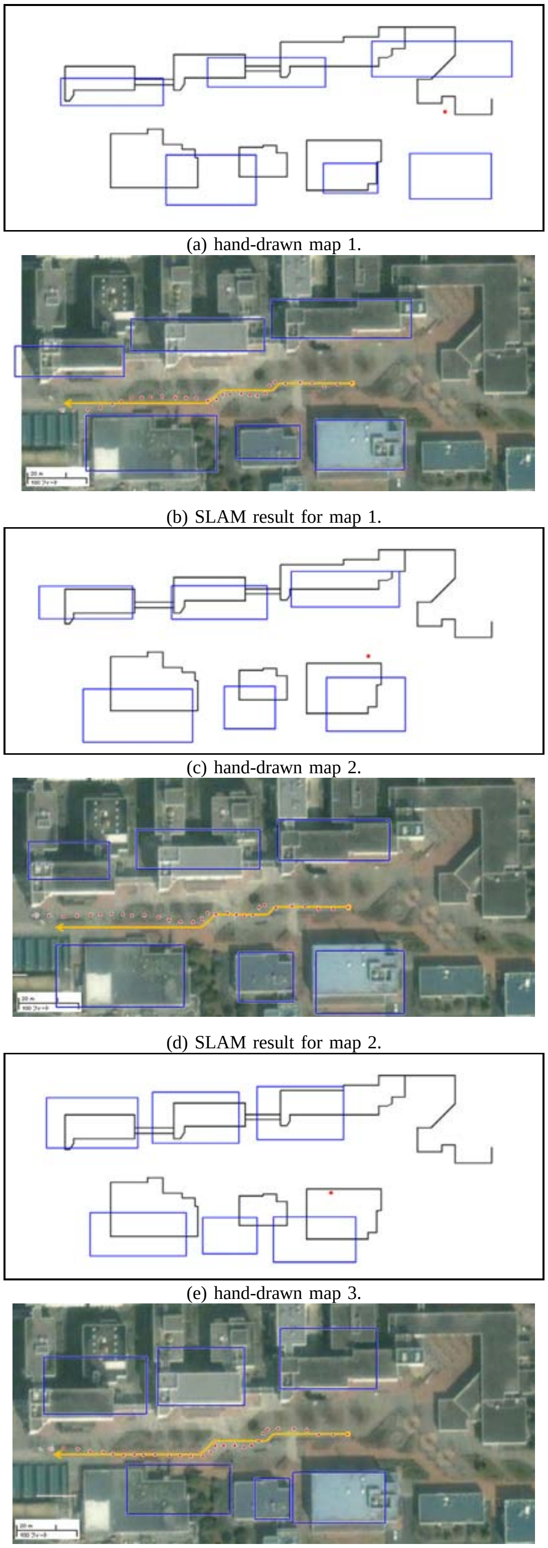
IV. EXPERIMENTAL RESULTS

We perform two experiments. One is off-line localization using the data acquired by moving an actual robot in our campus. The other is simulation-based analysis of the FastSLAM/PSOM algorithm.

Table I compiles the processing time per one frame with various combination of the numbers of SLAM and PSO particles. The processing time is almost proportional to the numbers because particles can basically be processed in parallel and the number of PSO iterations is fixed. In the experiments below, we use 100 and 20 particles for SLAM and PSO, respectively.

A. Localization results for real data

We asked a few students of our laboratory to draw a part of our campus using a GUI-based drawing tool. What they



(a) hand-drawn map 1.

(b) SLAM result for map 1.

(c) hand-drawn map 2.

(d) SLAM result for map 2.

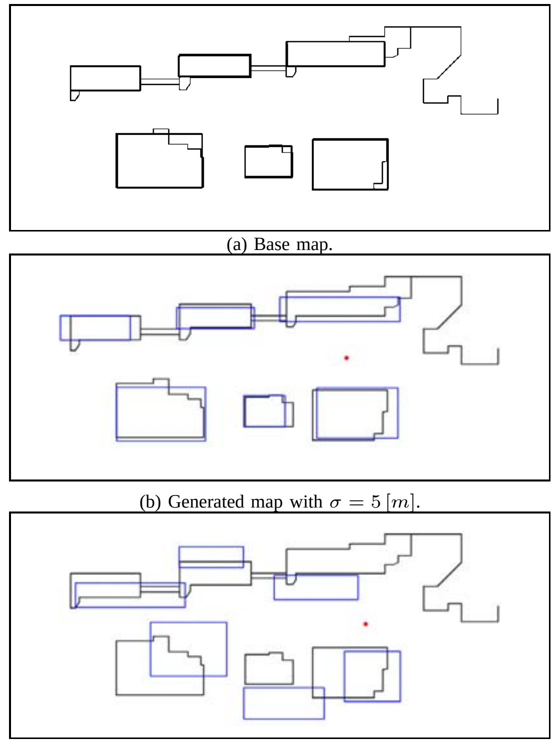
(e) hand-drawn map 3.

(f) SLAM result for map 3.

Fig. 6. SLAM result for three hand-drawn maps.

drew are six specified buildings and one landmark which is used as a starting position. They drew a map only with their memory (without seeing any maps or aerial images), but the scale of the map was given, indicated in the drawing tool.

Fig. 6 shows the SLAM results for three input hand-drawn maps. Each figure shows an input hand-drawn map superimposed on a true map or a SLAM result superimposed on an aerial image. In spite of large uncertainties in the input maps, the building sizes and positions are considered to be recovered reasonably well.



(a) Base map.

(b) Generated map with $\sigma = 5\,[m]$.

(c) Generated map with $\sigma = 12.5\,[m]$.

Fig. 7. Base map and generated maps with added uncertainty.

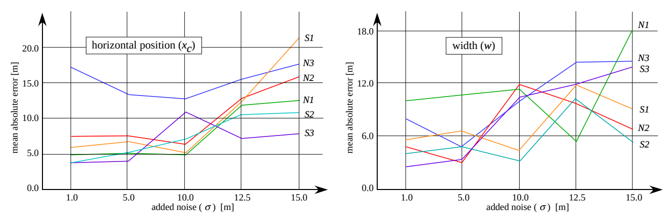


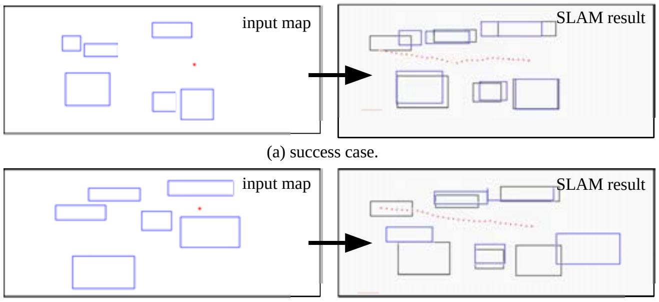
Fig. 8. Mean absolute errors of the horizontal position (x_c) and width (w) of the buildings for different added noises. Building labels, $N1$, $N2$, and $N3$, correspond to the upper three buildings from right to left in Fig. 7(a), and $S1$, $S2$, and $S3$ for the lower three ones.

B. Quantitative evaluation of FastSLAM/PSOM

We then try to quantitatively evaluate the performance of FastSLAM/PSOM by providing it maps with different magnitudes of uncertainty. We here use simulated observation and motion data for controlling the amount of uncertainty and for getting the true value.

We generated uncertain maps by adding perturbations to a base map, which is also approximated map with rectangular building shapes but is almost geometrically correct. Perturbations are added to position (x_c, y_c) and width w and controlled by a standard deviation. Fig. 7 shows the base map and some of generated maps with different uncertainties.

Fig. 8 shows the change of the averaged absolute error



(a) success case.

(b) failure case.

Fig. 9. A successful and a failed SLAM result for maps with $\sigma = 15.0\,[m]$.

TABLE II

SUCCESS RATIO WITH DIFFERENT MAP UNCERTAINTIES.

σ of added noise $[m]$	1.0	5.0	10.0	12.5	15.0
success/trials	5/5	5/5	4/5	1/5	1/5

of the horizontal position and the width of six buildings according to the increase of the added errors, up to about 25% of building widths, where we can see an increasing tendency of the errors.

We also examined the success ratio of the proposed method for different added errors. We here judge the SLAM successful when all estimated buildings overlap with the corresponding true ones. Fig. 9 shows a successful and a failed SLAM result. Table II shows the success ratio drops as the uncertainty increases. Using more particles certainly improves the success ratio, but requires more computation.

V. CONCLUSIONS AND DISCUSSION

We have developed a stereo-based outdoor localization method using a hand-drawn line drawing building map with large uncertainties. The method adopts FastSLAM with a particle swarm optimization for map refinement. This new method, FastSLAM/PSOM, has been shown to be effective for outdoor localization for an actual mobile robot.

Navigation of a mobile robot further requires path planning and obstacle avoidance. We suppose that a path to take is also described in a hand-drawn map. Since the path is usually specified with respect to buildings, and since the building position and size are refined through the SLAM process, the path should also be modified accordingly. It is also necessary to speed up the computation of the FastSLAM/PSOM for navigation by, for example, employing a rigorous parallel computation. Other future work include recognition of largely-deformed (i.e., not uniformly scaled) or scale-free maps, and use of landmarks in the map.

ACKNOWLEDGMENT

The authors would like to thank Dr. Junji Satake for his help in experiments. This work is supported in part by Grant-in-Aid for Scientific Research (No. 21300075) from JSPS.

REFERENCES

[1] G. Chronis and M. Skubic. Sketch-based Navigation for Mobile Robots. In *Proceedings of 2003 IEEE Int. Conf. on Fuzzy Systems*, pp. 284–289, 2003.

[2] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *Int. J. of Robotics Research*, Vol. 27, No. 6, pp. 647–665, 2008.

[3] G.N. DeSouza and A.C. Kak. Vision for Mobile Robot Navigation: A Survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 2, pp. 237–267, 2002.

[4] G. Grisetti, C. Stachniss, and W. Burgard. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. In *Proceedings of 2005 IEEE Int. Conf. on Robotics and Automation*, pp. 2432–2437, 2005.

[5] K. Irie, T. Yoshida, and M. Tomono. Mobile Robot Localization Using Stereo Vision in Outdoor Environments under Various Illumination Conditions. In *Proceedings of 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 5175–5181, 2010.

[6] J. Kennedy and R.C. Eberhart. Particle Swarm Optimization. In *Proceedings of 1995 IEEE Int. Conf. on Neural Networks*, pp. 1942–1948, 1995.

[7] J. Klippenstein and H. Zhang. Quantitative Evaluation of Feature Extractors for Visual SLAM. In *4th Canadian Conf. on Computer and Robot Vision (CRV '07)*, pp. 157–164, 2007.

[8] M. Kronfeld, C. Weiss, and A. Zell. Swarm-supported Outdoor Localization with Sparse Visual Data. *Robotics and Autonomous Systems*, Vol. 50, No. 2, pp. 166–173, 2010.

[9] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard. Large Scale Graph-Based SLAM using Aerial Images as Prior Information. *Autonomous Robots*, Vol. 30, No. 1, pp. 25–39, 2011.

[10] H.-C. Lee, S.-K. Park, J.-S. Choi, and B.-H. Lee. PSO-FastSLAM: An Improved FastSLAM Framework using Particle Swarm Optimization. In *Proceedings of 2009 IEEE Int. Conf. on Systems, Man, and Cybernetics*, pp. 2763–2768, 2009.

[11] K.Y.K. Leung, C.M. Clark, and J.P. Huissoon. Localization in Urban Environments by Matching Ground Level Video Images with an Aerial Image. In *Proceedings of 2008 IEEE Int. Conf. on Robotics and Automation*, pp. 551–556, 2008.

[12] W. Low, R. Nagarajan, and S. Yaacob. Visual based SLAM using Modified PSO. In *Proceedings of 6th Int. Coilloquium on Signal Processing and Its Applications*, pp. 1–5, 2010.

[13] K. Matsuo, J. Miura, and J. Satake. Stereo-Based Outdoor Localization using a Line Drawing Building Map. In *Proceedings of 2011 European Conf. on Mobile Robots*, pp. 309–314, 2011.

[14] J. Miura and K. Yamamoto. Robust View Matching-Based Markov Localization in Outdoor Environments. In *Proceedings of 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2970–2976, 2008.

[15] M.R. Mohammadi and S.S. Ghiday. Integrated PSO and Line Based Representation Approach for SLAM. In *Proceedings of 2011 ACM Symp. on Applied Computing*, pp. 1382–1388, 2011.

[16] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of AAAI National Conf. on Artificial Intelligence*, pp. 593–598, 2002.

[17] M.P. Parsley and S.J. Julier. Towards the Exploitation of Prior Information in SLAM. In *Proceedings of 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2991–2996, 2010.

[18] F.T. Ramos, J. Nieto, and H.F. Durrant-Whyte. Recognising and Modeling Landmarks to Close Loops in Outdoor SLAM. In *Proceedings of 2007 IEEE Int. Conf. on Robotics and Automation*, pp. 2036–2041, 2007.

[19] T. Senlet and A. Elgammal. Satellite Image Based Precise Robot Localization on Sidewalks. In *Proceedings of 2012 IEEE Int. Conf. on Robotics and Automation*, pp. 2647–2653, 2012.

[20] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

[21] G. Venter and J. Sobieszczanski-Sobieski. Particle Swarm Optimization. In *Proceedings of 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conf.*, pp. AIAA 2002–1235, 2002.

[22] J. Yun and J. Miura. Multi-Hypothesis Localization with a Rough Map using Multiple Visual Features for Outdoor Navigation. *Advanced Robotics*, Vol. 21, No. 11, pp. 1281–1304, 2007.

[23] J. Yun and J. Miura. A Quantitative Navigability Measure of Rough Maps. *J. of Robotics and Mechatronics*, Vol. 21, No. 1, pp. 95–103, 2009.