The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

A Hybrid Approach to RBPF Based SLAM with Grid Mapping Enhanced by Line Matching

Wei-Jen Kuo, Shih-Huan Tseng, Jia-Yuan Yu, and Li-Chen Fu

Abstract—**In this paper, we present a novel data structure representing the environment with occupancy grid cells while each grid map is associated with a set of line features extracted from laser scan points. Due to the fact that line segments are principal elements of artificial environments, they provide considerable geometric information about the environment which can be used for enhancing the accuracy of localization. Orthogonal characteristic of line features is the key issue to guarantee the consistency of the SLAM algorithm by allowing us to deal with lines that are parallel or perpendicular to each other. This behavior allows us to sample robot poses more correctly. As a result, the proposed algorithm can close bigger loops with the same number of particles. Experimental results are carried out using SICK LMS-100 laser scanner which has a maximum range of 20*m* and Pioneer 3DX mobile robot mapping an indoor environment with the size of 40*m* × 47*m*.**

I. Introduction

SIMULTANEOUS localization and mapping (SLAM) of mobile robots has been a fundamental requirement for robust robotic navigation. It is also considered to be a chicken and egg problem, that is, a robot needs a consistent map for localization, in order to acquire such a map, precise estimation of robots' location is required. This makes SLAM a hard problem which necessitates searching for solution in a high-dimensional space. A number of different solutions had been proposed, most of which aiming at the issue of how to represent, process, store, and retrieve the map information.

The main idea of this paper resides in exploiting the characteristics of a structured indoor environment via adding an additional line database that stores line segments extracted from the map associated with each individual particle of Rao-Blackwellized particle filter (RBPF). Therefore, each particle updates an individual set of line segments extracted from their map known as a line database. Direction of line segments are then used in the localization stage for calculating weight of each particle. Due to the structure of indoor environments, particles with more extracted lines parallel or perpendicular to the dominating direction of the lines in its' database has greater chance to express the correct map of the environment.

Manuscript received March 1, 2009. This work was supported by National Science Council, Taiwan, under Grant NSC 97-2218-E-002-015.

Wei-Jen Kuo is with the Department of Computer Science and Engineering, National Taiwan University (e-mail: r96922113@ ntu.edu.tw).

Shih-Huan Tseng is with the Department of Computer Science and Engineering, National Taiwan University (e-mail: d96922018@ntu.edu.tw).

Jia-Yuan Yu is with the Department of Computer Science and Engineering, National Taiwan University (e-mail: r97922097@ntu.edu.tw).

Li-Chen Fu is with Department. of Computer Science and Engineering and Department of Electrical Engineering, National Taiwan University (Phone +886-2-23622209, e-mail: lichen@ntu.e.du.tw).

A. Background and Related Work

In recent years, solutions to the SLAM problem have became more and more mature, and methods for solving the problem can be classified by the underlying techniques applied to estimate the robot pose as well as the map representation used to describe the environment. The most common approach to estimate the robot trajectory and landmark locations are extended Kalman filter (EKF) and particle filter (PF); where two state of the art mapping data structures are grid map and landmark map, respectively.

A dominating approach to the SLAM problem that made use of EKF was introduced by Smith *et al.* [1]. In that paper, EKF is applied to incrementally estimate the posterior distribution over robot pose along with landmark location. However, EKF SLAM had problems with data association and also required time complexity quadratic to the number of landmarks. An alternative approach, *FastSLAM*, was addressed by Montemerlo [2]. He extended the framework of RBPF introduced by Murphy [3] to factorize EKF SLAM problem into a problem with robot's localization problem and another with landmark's location estimation, which are respectively solved by particle filter and Kalman filter. This allows the complexity of *FastSLAM* to scale logarithmically with the number of landmarks.

Each particle of the RBPF represents a potential trajectory of the robot and a map of the environment. Eliazar and Parr [4] introduced a purely laser based algorithm, *DP-SLAM,* to apply the concept of RBPF on an approach based on grid map without any assumption on landmarks. *DP-SLAM* introduces a new map representation called distributed particle (DP) mapping, which is able to update and maintain hundreds of candidate maps and robot poses efficiently via a particle ancestry tree. Another similar work on grid based RBPF was addressed by Grisetti [5]. In that approach, a sensor aware proposal distribution and adaptive resampling are applied to increase the performance of RBPF with grid maps.

Besides grid and landmark map representation, a Closed Line Segment (CLS) map which only uses line segment as elements was introduced by Zhang [6] in year 2000. The line segment in fact provides considerable geometric information of the laser scan, that can be used for fast localization and mapping. Another approach for constructing the line-based map can be found in [7], where the author also provides closed form formulas for line fitting. Furthermore, Nguyen *et al.* (2006) also proposed a lightweight SLAM called Orthogonal SLAM (*OrthoSLAM*) [8], which reduces the complexity by mapping only lines that are parallel or perpendicular to each other, and such characteristic pertains to the main structure of most indoor environments. In

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

contrast to that, by associating a line database with each particle, we can reduce the number of particles needed, and moreover, increase the accuracy of mapping.

Although matching line segment is an effective means to enhance the efficiency of localization, it is rarely used collectively with particle filters to exert its influence. In this paper, we propose a RBPF based SLAM where each particle carries an individual grid map, associated with a set of line features collected during the mapping process. This allows us to utilize the rich line features of structured indoor environment, and furthermore being able to maintain and update the map efficiently with fewer particles.

B. System Overview

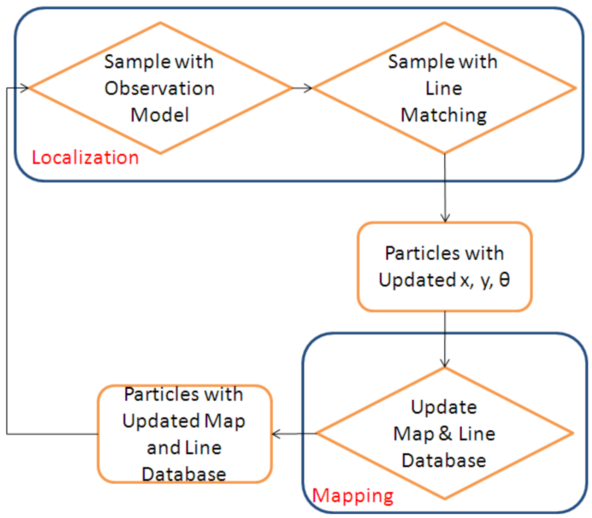

Fig. 1 Flowchart of our proposed algorithm

Figure 1 shows the flowchart of our proposed algorithm. The orange diamond blocks are the main functions of the SLAM process, and orange rectangle blocks represent outputs of main functions. The upper half of Fig. 1 consists of components which build up the localization part of our algorithm, and particles are weighted and resampled during this phase. Output particles are used in the mapping part, as shown in the lower half of the figure, to update maps and line databases associated with particles. Finally, a new set of resampled particles are created and the algorithm repeats itself.

C. Paper Structure

This paper is organized as follows. Section II briefly explains how RBPF can be used to solve the SLAM problem. We describe the implementation details of the hybrid approach in Section III. The experimental results are then shown in Section IV. Finally, Section V draws the conclusion and address future works.

II. Mapping With RBPFs

RBPF have been introduced as an effective means to solve the simultaneous localization and mapping (SLAM) problem. The beauty of RBPF lies in the individual map maintained by each particle, meaning multiple hypotheses of the robot trajectory are held as candidates which will eventually converge to a few trajectories similar to the real one. The main idea is that RBPF maintains a joint distribution $p(x_{1:t}, m|z_{1:t}, u_{1:t-1})$ over the environment maps m and the robot trajectory $u_{1:t-1}$ with a particle filter. This distribution can be estimated by observations $z_{1:t}$ and control measurements $u_{1:t-1}$ obtained by sensors and wheel encoders on the mobile robot. Since the map and line segment estimation is considered on the path estimation, each particle maintains its own map along with a set of line segments L extracted from range readings. Furthermore, we can make use of the factorization:

$$p(x_{1:t}, m|z_{1:t}, u_{1:t-1}) = p(x_{1:t}|z_{1:t}, u_{1:t-1})p(m, |x_{1:t}, z_{1:t}) \quad (1)$$

This Rao-Blackwellization approach using the factorization above allows us to estimate the robot trajectory and environment map separately. The trajectory from time step 1 to t can be calculated by estimating the posterior $p(x_{1:t}|z_{1:t}, u_{1:t-1})$ over the trajectory from time step 1 to t-1, which can be done by applying the particle filter.

III. Hybrid Approach

The hybrid approach we propose here is composed of mapping with two different map structures, namely, line segment map and grid map. Although two kinds of map representations are used, line segment map actually plays a role of correcting the grid map instead of serving as an independent map itself. Therefore, no extra processing is required for the extracted line segments to form a well knit map, and moreover storage of line segment map becomes rather simple, i.e., a database of line parameters is associated with each particle.

A. Feature Enhanced RBPF Mapping

Particle filter is a simulation-based method that tracks targets with partially observable state. It maintains a weighted set of sample states $S_t = \{s_1, s_2, \dots, s_M\}$ called particles. Each particle carries a weight value that represents the reliability of the state of the environment which it stands for. The process of particle filter can be summarized as the following four steps:

1) *Sampling*: In this step, a new generation of particles S_t is obtained by applying the motion model to the previous generation of particles S_{t-1}. The motion model we use can be found in [9] where an Expectation Maximization (EM) framework is used to learn parameters of the model.

2) *Weighting*: An importance weight $w_t^{(i)}$ is given to each ith particle of the current generation based on the measurement models $p(s_t^{[i]}|z_t, m^{[i]})$ and $p(x_t^{(i)}|L_t^{(i)}, L^{(i)})$. Here, $L^{(i)} = \{\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_N\}$ is the set of the line segments in the database and $L_t^{(i)} \in \{\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_k\}$ is the set of lines perceived at time t. Line segments in $L_t^{(i)}$ are compared to the line segments in $L^{(i)}$ according to their parameters in Hough space. The

details of weighting will be explained in the next few sections.

3) *Resampling*: A new set of particles of the current generation is randomly drawn with replacement from S_t with probability proportional to its weight. Since particles with mismatched line direction results in very low weight, it is not likely to be resampled.

4) *Map Estimation*: For each particle, an individual map and an updated database of line segments is computed with $p(m^{(i)}, L^{(i)} | x_{1:t}, z_{1:t}, L_t^{(i)})$ based on the trajectory $x_{1:t}^{(i)}$ estimated by that particle. Moreover, the line databases associated with each particle is also update according to $z_{1:t}$ and $L_t^{(i)}$.

In this paper, we propose an innovative weighting method which considers the alignment of dominating direction line segments with the extracted line segment of each particle during the weighting step. The benefit of such modification to the algorithm reduces the risk of orientation mismatch. This is implemented by modifying RBPF so that each map carried by individual particles is associated with a set of line segments, known as a "line database," extracted from the previously built map. The following sections show how lines are detected and weighted.

B. Line Detection

The indoor environment contains many artificial landmarks, such as walls and corners. These landmarks are easy to extract and matching can also be straight forward while using an appropriate line segment representation. A comparison of line extraction algorithms can be found in [10]. In our approach, a line segment **l** is defined by the following set of parameters: $\mathbf{l} = \{\rho, \theta, l, (x_{start}, y_{start}), (x_{end}, y_{end})\}$, where ρ and θ are parameters of Hough space; l, (x_{start}, y_{start}), and (x_{end}, y_{end}) are length and endpoints of a line segment respectively. The structure of particles is depicted in Table 1, where k_i is the number of extracted lines in particle *i*, which may differ between particles according to the estimated robot path. So far, we can find a number of works aiming to detect line segments and other features with the use of laser range finder [11][12]. Our feature detection procedure includes the following steps.

TABLE 1
Particle structure of the hybrid approach

	robot pose	**map**	**line 1**	**line 2**	**…**	**line k_i**
particle 1	$x\ y\ \theta$	m	$\rho_1\theta_1 l_1 \dots$	$\rho_2\theta_2 l_2 \dots$	…	$\rho_{k_1}\theta_{k_1} l_{k_1} \dots$
particle 2	$x\ y\ \theta$	m	$\rho_1\theta_1 l_1 \dots$	$\rho_2\theta_2 l_2 \dots$	…	$\rho_{k_2}\theta_{k_2} l_{k_2} \dots$
			…			
particle *N*	$x\ y\ \theta$	m	$\rho_1\theta_1 l_1 \dots$	$\rho_2\theta_2 l_2 \dots$	…	$\rho_{k_N}\theta_{k_N} l_{k_N} \dots$

1) *Range data segmentation*:

Preprocessing of line segments is done before line extraction, where consecutive scan points are clustered into the same segments based on the assumption that points belong to the same object as long as the distance between them is less than a given threshold. Since sensor LMS-100 has a maximum range of 20*m*, range readings farther than this are ignored, and therefore such reading also break segments.

2) *Least square line fitting*:

Each segment is then fitted into a line on the x-y plane by performing the least square line fitting algorithm. An estimation of line parameters in *ax*+*by*+*c*=0 is calculated during this step. Furthermore, line segments with line fitting error larger than $E(a,b,c)$ are eliminated. Line parameters can be calculated by applying the following equations:

$$a = \sum_{i=1}^{N_p} x_i \sum_{i=1}^{N_p} y_i^2 - \sum_{i=1}^{N_p} y_i \sum_{i=1}^{N_p} x_i\, y_i \tag{2}$$

$$b = \sum_{i=1}^{N_p} y_i \sum_{i=1}^{N_p} x_i^2 - \sum_{i=1}^{N_p} x_i \sum_{i=1}^{N_p} x_i\, y_i \tag{3}$$

$$c = (\sum_{i=1}^{N_p} x_i\, y_i)^2 - \sum_{i=1}^{N_p} x_i^2 \sum_{i=1}^{N_p} y_i^2 \tag{4}$$

Here, we fit one segment at a time, and N_p is the number of points in the segment we are currently dealing with. The more points there is in a segment, the more precise it will fit with the line *ax*+*by*+*c*=0. With these parameters, the line fitting error can be obtained by (4)

$$E(a,b,c) = \sum_{i=1}^{N} (-\frac{a}{b}x_i - \frac{c}{b} - y_i)^2 \tag{5}$$

3) *Hough parameter calculation*:

Since least square line fitting algorithm is heavily affected by outliers, we apply a Hough transform which has a voting space being restricted such that $|\rho - \rho_v|$ and $|\theta - \theta_v|$ are smaller than a threshold value. Here, ρ and θ are initial Hough parameters obtained from line parameters *a*, *b*, and *c*. The initial Hough parameters can be obtained from the line parameters as follows:

$$\rho = \frac{c}{\sqrt{a^2 + b^2}} \qquad \theta = tan^{-1}\left(-\frac{b}{a}\right) \tag{6}$$

C. Feature Association

Matching score on both grid map and feature location are considered when particles are being weighted. The weighting is done by checking which particle makes a more correct guess of feature association, and particles with the wrong feature association will eventually disappear in the resampling process.

Choi *et al.* proposed a line feature based SLAM [13] that associates line features through a weighted Euclidean distance measure in Hough space and the length of overlap between lines, and this same method is also applied in our implementation. Our line matching procedure starts with matching of the set of lines extracted from current sensor readings $\boldsymbol{L}_s$ with lines in the database $\boldsymbol{L}_d$. If the extracted line $\boldsymbol{L}_s^{(i)}$ matches with line $\boldsymbol{L}_d^{(j)}$ in the database, the line parameters $\rho, \theta, l, (x_{start}, y_{start})$, and (x_{end}, y_{end}) of $\boldsymbol{L}_d^{(j)}$

are updated in order to let the two line segments be merged. The line merging algorithm consists of the following steps:

1) *Search* :

For each line segment $\boldsymbol{L}_{\mathrm{s}}^{(\mathrm{i})}$ in $\boldsymbol{L}_{\mathrm{s}}$ a complete search through $\boldsymbol{L}_{\mathrm{d}}$ is performed, and once a closest match is found in the database and the difference between parameters ρ and θ of the closest matched lines are lower than a given threshold value, we regard the two line segments belonging to the same line, and therefore we merge the two lines. If line segment $\boldsymbol{L}_{\mathrm{s}}^{(\mathrm{i})}$ isn't found in the database, and its length exceeds 5*m*, $\boldsymbol{L}_{\mathrm{s}}^{(\mathrm{i})}$ is very likely to be a newly discovered line, and therefore we insert it into the database.

2) *Update* :

Line merging is the most important part of updating the database. The following equation shows the updated parameters of $\boldsymbol{L}_{\mathrm{d}}^{(\mathrm{j})}$:

$$\rho_{d,new}^{(j)} = \frac{count * \rho_{d,old}^{(j)} + \rho_s^{(i)}}{count + 1} \tag{7}$$

$$\theta_{d,new}^{(j)} = \frac{count * \theta_{d,old}^{(j)} + \theta_s^{(i)}}{count + 1} \tag{8}$$

A variable *count* is used to keep track of the number of times the line segment $\boldsymbol{L}_{\mathrm{d}}^{(\mathrm{j})}$ is updated for. Two endpoints are chosen after searching for the longest distance among $(x_{start}, y_{start})_d^{(j)}$, $(x_{end}, y_{end})_d^{(j)}$, $(x_{start}, y_{start})_s^{(i)}$, and $(x_{end}, y_{end})_s^{(i)}$. The two farthest points are then projected onto the line with Hough parameters $\rho_{d,new}^{(j)}$ and $\theta_{d,new}^{(j)}$, forming the new endpoints as shown in Fig. 2.

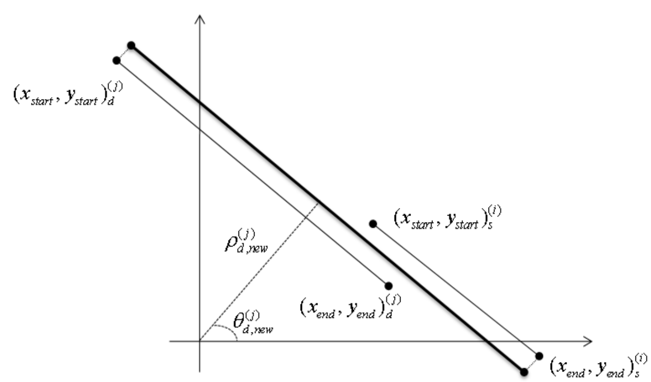

Fig. 2 Two line segments $\boldsymbol{L}_{\mathrm{d}}^{(\mathrm{j})}$ and $\boldsymbol{L}_{\mathrm{s}}^{(\mathrm{i})}$ are merged, forming a new line

3) *Pruning* :

Some line segments might be detected only once or twice due to sensor and localization errors. These undesirable data are pruned away during this step. If the *count* of a certain line segment is smaller than a predefined threshold, a cleanup action is performed every, say, 10 iterations. This pruning action is important, since each particle carries an individual set of lines, and we prefer to keep the number of line segments as low as possible.

D. Particle Weighting

The weight of each particle comes from two principal factors. One of them is similar to that concerning observation model proposed in [4], where each laser scan is given a weight value w_{grid} ranging from 0 to 1 after a laser penetration model is applied to the estimated robot pose on the grid map constructed so far. The other weighting factor w_{line} is obtained by matching line segments extracted from the current scan to those in the database with (9).

$$w_{line} = \frac{total\ length\ of\ matched\ lines\ \ at\ time\ t}{total\ length\ of\ detected\ lines\ at\ time\ t} \tag{9}$$

Since matching line direction has the most significant effect on weighting particles, line segments with $\left|\theta_s^{(i)} - \theta_d^{(j)}\right|$, $\left|\theta_s^{(i)} - \theta_d^{(j)} - \frac{\pi}{2}\right|$, or $\left|\theta_s^{(i)} - \theta_d^{(j)} + \frac{\pi}{2}\right|$ smaller than a given threshold value are considered as matched lines.

With these two weighting factors, the current pose is obtained by finding the *i*th particle which maximizes the value $w_{grid}^{(i)} \times w_{line}^{(i)}$. The grid map and line database are assumed to be conditionally independent, and this somehow works properly. As a result, integrating line direction analysis into the weight computation allows us to discard those particles whose extracted line direction doesn't match the dominating line directions observed, since w_{line} is nearly 0 for a line direction mismatched.

IV. Experimental Results

We tested our algorithm with sensor logs generated by P3-DX robot and SICK LMS-100 laser rangefinder on a standard PC, equipped with 1.6GHz CPU and 2G memory. The program runs at real time with grid size set to 5cm and 50 particles used for all experiments. Communication between application programs are established on a client-server framework that run at 5Hz. The first experiment is made under a scenario where traditional particle filter performs more unstably, such as when robot makes sudden turns or passes narrow doorways. The results are shown in Fig. 3. Light blue circle represents location of the robot, and red dots scattered around the robots' location depicts the particle distribution. It can be seen from the result of experiments that the distribution of particles are more centralized while using our hybrid approach, which accordingly leads to reduction of orientation mismatch, as shown in Figs. 4(a.2), (b.2). This is because particles sampled via the motion model are likely to include incorrect orientations, causing mismatch between extracted line directions and the principal line directions. As a result of line direction matching in the hybrid approach, these particles are given relatively low weights, and therefore aren't likely to be resampled.

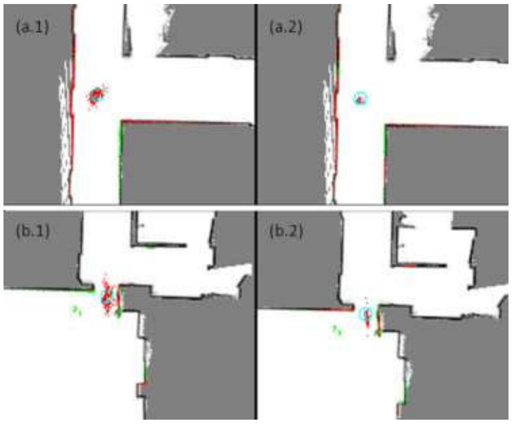

Fig. 3 The two figures on the left show the distribution of particles when using original RBPF; the other two on the right show distributions while taking line direction matching into account.

The improvement in handling loop closing is validated in the second experiment, which is carried out on sensor logs of the CSIE building in NTU. The size of the building is approximately $40m \times 47m$ and contains a perfect loop. Figure 4 shows the comparison of results with and without matching line directions. Red circle in the figure indicates the location where loop is closed. Since our hybrid approach keeps track of extracted line directions, and the direction of main corridors of the building are perpendicular to each other, the directional mismatch which causes loop closing failure hardly occurs.

Figure 5 depicts a comparison of the variance along x, y, and θ axes of robot poses while traveling along a 378-step path. When our approach is used (Fig. 5 right), the variance of y and θ seem to drop enormously, yet the variance on the major direction of the robot (in this case the x axis) doesn't improve, this is because less than two corner features were detected. The general SLAM used for this comparison is similar to the one in [4]. Note that for both methods, only 20 particles are used. Figure 6 verifies the accuracy of our hybrid approach by showing that the corridors map matches almost completely the two rectangles approximately representing the floor plan, that is, the ground truth.

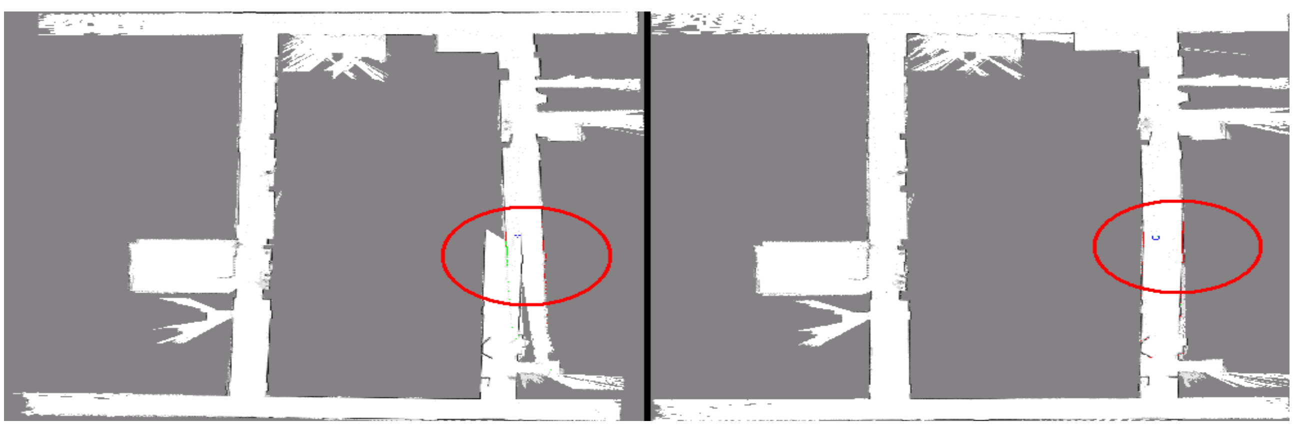

Fig. 4 SLAM in the CSIE building with general approach (left), and hybrid approach(right).

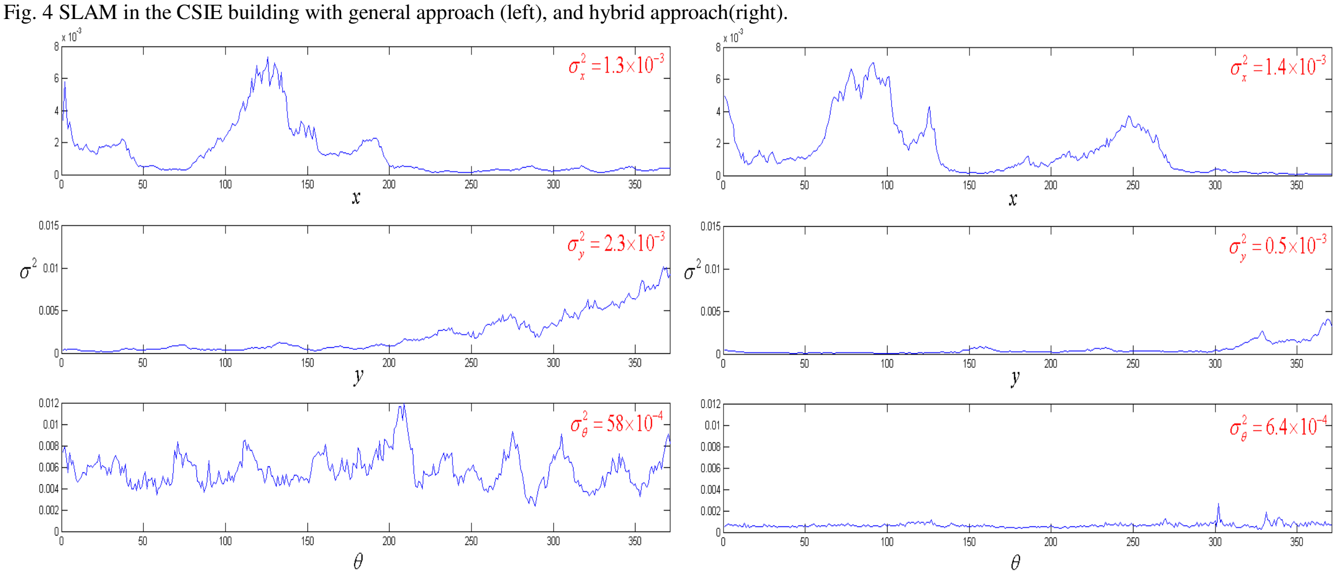

Fig. 5 (left)Portion of the grid map where data forming the curve is collected (right)Comparison of hybrid and general approach. Dash line shows the time interval where robot makes a rapid turn

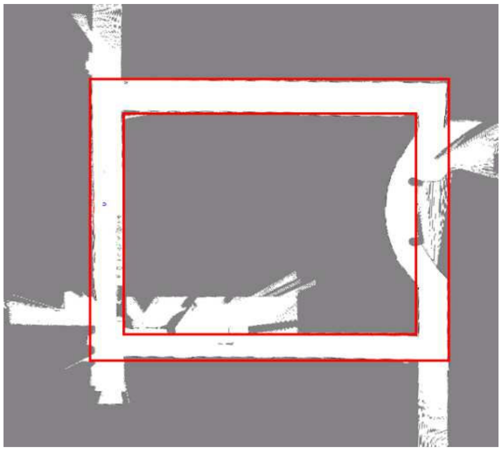

Fig. 6 Grid map compared with two rectangles representing the ground truth

V. Conclusion

In this paper, we have increased the accuracy and efficiency of Rao-Blackwellized particle filter SLAM algorithm by considering the direction of extracted lines when weighting particles. The main contribution of this paper verified by the experimental results via verifying with a real robot lies in that weighting according to our proposed method actually reduces the number of particles required to achieve precise localization and mapping. A consequence of this is of course more efficient and/or more precise SLAM algorithm can be expected.

The proposed line merging algorithm learns principal direction of line segments, and also stores representative line segments into the database as a basis of future matching.

Our future work includes integrating other features extracted from range data, such as corner, arc, and circle as a reference of weighting. Since these are strong information features, they are expected to make a considerable improvement to the accuracy of localizing the robot.

Acknowledgment

This work was supported by the National Science Council of the Republic of China under Grant NSC 97-2218-E-002-015.

References

[1] R. Smith, M. Self, and P. Cheeseman, “Estimating uncertain spatial relationships in robotics,” *Autonomous Robot Vehicles*, I.J. Cox and G.T. Wilfon, Ed. New York: Springer Verlag, pp. 167-193, 1990.

[2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” *Proc. AAAI Nat’l Conf. Artificial Intelligence*, 2002.

[3] K. P. Murphy. Bayesian map learning in dynamic environments. In NIPS-12, pages 1015– 1021, 2000.

[4] A. Eliazar and R. Parr, “DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks,” in *Proc. 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-03)*, Morgan Kaufmann 2003, pp. 1135–1142.

[5] G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with rao-blackwellized particle filters,” *IEEE Transactions on Robotics*, vol. 23, pp. 34-46, 2007.

[6] Zhang, L. and B. K. Ghosh. “Line segment based map building and localization using 2D laser rangefinder.” *Proceedings. ICRA 2000. IEEE International Conference on Robotics and Automation*, pp.2538-2543, 2000.

[7] S. Pfister, S. Roumeliotis, and J. Burdick, “Weighted line fitting algorithms for mobile robot map building and efficient data representation,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, pp. 1304–1311, Sep. 14-19 2003.

[8] Nguyen, V. Harati, A. Martinelli, A. Siegwart, and R. Tomatis, N., "Orthogonal SLAM: a Step toward Lightweight Indoor Autonomous Navigation," *2006 IEEE/RSJ*, *International Conference on Intelligent Robots and Systems,* pp.5007-5012, 9-15 Oct. 2006.

[9] A. I. Eliazar and R. Parr, “Learning probabilistic motion models for mobile robots” in *International Conference on Machine Learning*, p.32, 2004.

[10] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart, “A Comparison of Line Extraction Algorithms using 2D Laser Rangefinder for Indoor Mobile Robotics” *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005)*, pp. 1929-1934, 2-6 Aug, 2005.

[11] Joo Xavier, Marco Pacheco, Daniel Castro and Antnio Ruano “Fast line, arc/circle and leg detection from laser scan data in a Player driver” *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005.*pp. 3930-3935, 18-22 April 2005.

[12] A. Alempijevic, “High-speed feature extraction in sensor coordinates for laser rangefinders” in *Proceedings of the 2004 Australasian Conference on Robotics and Automation*, 2004.

[13] Young-Ho Choi, Tae-Kyeong Lee, and Se-Young Oh, “A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot,” *Autonomous Robots*, vol. 24, pp. 13-27, 2008.