Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

Simultaneous Localization and Mapping with Environmental Structure Prediction

H. Jacky Chang, C. S. George Lee*, Yung-Hsiang Lu and Y. Charlie Hu
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-2035
{chang26, csglee, yunglu, ychu}@purdue.edu
https://engineering.purdue.edu/ResearchGroups/DEAR

***Abstract*— Traditionally, the SLAM problem solves the localization and mapping problem in explored and sensed regions. This paper presents a prediction-based SLAM algorithm (called P-SLAM), which has an environmental structure predictor to predict the structure inside an unexplored region (i.e., look-ahead mapping). The prediction process is based on the observation of the surroundings of an unexplored region and comparing it with the built map of explored regions. If a similar structure is matched in the map of explored regions, a hypothesis is generated to indicate that a similar structure has been explored before. If the environment has repeated structures, the mobile robot can utilize the predicted structure as a virtual mapping, and decide whether or not to explore the unexplored region to save exploration time. If the mobile robot decides to explore the unexplored region, a correct prediction can be utilized to localize the robot and speed up the SLAM process. We also derive the Bayesian formulation of P-SLAM to show its compact recursive form for real-time operation. We have experimentally implemented the proposed P-SLAM in a Pioneer 3-DX mobile robot using a Rao-Blackwellized particle filter in real-time. Computer simulations and experimental results validated the performance of the proposed P-SLAM and its effectiveness in an indoor environment.**

I. Introduction

Simultaneous localization and mapping (SLAM) is a fundamental and complex problem in mobile robotics research. Traditionally, in a SLAM problem, a mobile robot explores and senses an unknown region, constructs a map and localizes itself. Researchers encountered many problems such as dead reckoning, noisy sensory measurements, failure data association and dynamic environment. These problems cause uncertainties and high-dimensional spaces in maps and robot poses, which require complex computations to solve them. Some researchers utilized various probabilistic techniques [1]–[8] to solve the SLAM problem and learn the probability distribution of the elements in the explored regions with little emphasis in the unexplored regions.

This work was supported in part by the National Science Foundation under Grant IIS-0329061. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

*This material is based upon work supported by (while serving at) the National Science Foundation.

This paper proposes a SLAM algorithm, called Prediction-based SLAM Algorithm (P-SLAM), which focuses on predicting the structure inside an unexplored region and utilizing the predicted information for SLAM. The P-SLAM has an environmental structure predictor to generate hypotheses of unexplored regions before the mobile robot actually explores them. Here the structure means objects or constructed walls that are in an environment. If we can predict the correctness of the structure in an unexplored region to a certain degree, then we can decide not to explore the unexplored region, and utilizes the predicted structure as a virtual mapping to save the exploration time. Furthermore, if we decide to explore the unexplored region, then a correct prediction will speed up the SLAM process in an environment with repeated structures. The effectiveness of P-SLAM depends on the characteristics of an environment such as repeated similar features, similar shapes or symmetric structures. We usually can find an indoor environment with these characteristics (e.g., straight walls, right-angle corners, similar rooms and symmetric layout in a building). With these characteristics, the P-SLAM utilizes its built map to predict unexplored regions.

A successful environmental structure prediction can be utilized in many mobile robotics research. In P-SLAM, we utilize it to save the exploration time and speed up the SLAM process. Our previous work has shown that coordinating an ant-like, multi-robot system by a simple environment estimation can explore an unknown area with energy-and-time efficiency [9]. Schroter *et al.* presented a detection-and-classification method of gateways in an office environment before a mobile robot fully observed these features [10]. This prediction scheme allows a mobile robot to obtain better localization and mapping results.

Most of the existing probabilistic approaches to the SLAM problem are all based on Bayesian filters [1], [11]. The Bayesian formulation results in a compact recursive form suitable for real-time computation. Since the proposed P-SLAM differs from the traditional SLAM algorithms in the utilization of predicted maps for localization and mapping, it will be of interest to derive the Bayesian formulation of P-SLAM and compare it with the Bayesian formulation of traditional SLAMs. Like the Bayesian formulation of traditional SLAM

0-7803-9505-0/06/$20.00 ©2006 IEEE

algorithms, the Bayesian formulation of P-SLAM also results in a compact recursive form, which incrementally solves the SLAM problem.

This paper is organized as follows. In Section II, we formulate the prediction problem as a search problem and a similarity measure problem. In Section III, we describe the environmental structure predictor. In Section IV, we present the Bayesian formulation of P-SLAM, and its implementation with a Rao-Blackwellized particle filter. In Section V, experimental results are presented. Conclusions are summarized in Section VI.

II. Problem Formulation

The central problem of the proposed P-SLAM is how to predict the structure of an unexplored region. The P-SLAM adopts the occupancy-grid map representation, and this allows us to easily distinguish both the explored and unexplored cells in an occupancy-grid map. The prediction process can be divided into four major steps: (i) Locate a target frontier cell to predict; (ii) Collect structure information nearby the target region; (iii) Search similar structures in the built map; (iv) Generate a hypothesis if a similarity match exists.

In the first step, a target cell f is selected for prediction. To prevent repeated or unnecessary predictions, we adopt the following two selection criteria: i) the target is a frontier cell (i.e., a cell between an explored region and an unexplored region), and ii) the cell is the next exploration goal. The first criterion guarantees that there is an unexplored region nearby, and the second criterion avoids unnecessary predictions because we want to utilize the predicted result immediately.

In the second step, the surrounding S_f of the target cell f is formed within a range d as shown in Fig. 1(a). The range d defines what is the surrounding and determines the prediction range. Next, the third step is formulated as a search problem: Given f and S_f, search a reference cell f' with the reference region $C_{f'}$ in the built map $\mathcal{M}$ such that

$$C_{f'} = \underset{C_{f'} \in \mathcal{M}, f' \neq f}{\arg\max} \{\Psi(C_{f'}, S_f)\} \quad (1)$$

where $\Psi(\cdot)$ is a similarity measure function. The corresponding f' and $C_{f'}$ are shown in Fig. 1(b). In the fourth step, if the similarity measure is larger than a pre-defined threshold, then we say that a match exits and a hypothesis is generated. As shown in Fig. 1(b), the generated hypothesis $H_{f'}$ corresponds to the unexplored region in Fig. 1(a). In Section III, we shall present the proposed environmental structure predictor, which includes the four steps described above.

After we obtain a hypothesis from the look-ahead mapping, the next question is how to utilize the hypothesis in SLAM. As shown in Fig. 2, the major difference between traditional SLAM algorithms and the P-SLAM is the look-ahead mapping. The hypothesis generated from the look-ahead mapping assists the localization process. The details on how to merge the look-ahead mapping into a traditional SLAM will be discussed in Section IV.

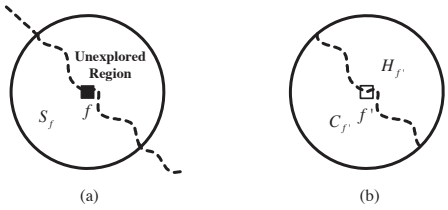


Fig. 1. The prediction of an unexplored region nearby a frontier cell. (a) The frontier cell, f, is marked as a bold square. The target region is inside the circle with a diameter d. The dash line is the boundary between the unexplored and explored areas. The upper right side of the boundary is indicated as the unexplored region. The surrounding region is S_f. (b) The reference cell in the built map is f' marked as a hollow square. The reference region, $C_{f'}$, is inside the circle with a diameter d. The dash line corresponds to the boundary of the target region in (a). The $H_{f'}$ is used to generate the hypothesis of the unexplored region in (a).

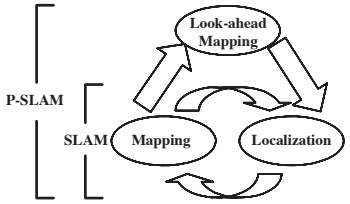


Fig. 2. Difference between the proposed P-SLAM and the traditional SLAM. The P-SLAM has a look-ahead mapping, which is not included in a traditional SLAM.

III. Environmental Structure Predictor

A. Search of a reference region

Since the possible reference region could exist in any place in the built map (i.e., position and rotation), the search space could be enormously large and could overwhelm our algorithm. To overcome this problem, we utilize image registration techniques to search a reference region. The image registration is a process of establishing point-by-point correspondence between two images of a scene [12]. The image registration consists of feature extraction, feature correspondence and calculating transformation matrices. We briefly describe the image registration method here.

1) Feature Extraction: Lines and corners are selected as our features because they are the elementary representation of walls, hallways or rooms. The line features are extracted from Hough transform, and the corner features are extracted from image gradients. Figure 3 shows an example of feature extraction from a partial mapping result.

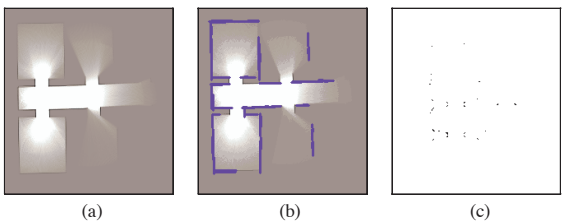
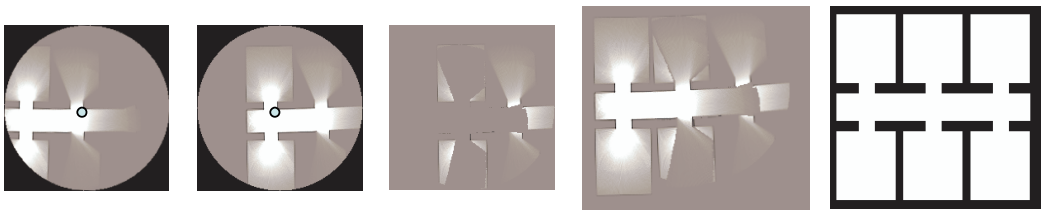


Fig. 3. Feature extraction. (a) The original occupancy-grid map. (b) The extracted line features. (c) The extracted corner features.



(a) (b) (c) (d) (e)

Fig. 4. (a) The selected target region. The small circle is the target control point. (b) The found reference region. The small circle is the reference control point. (c) The generated hypothesis. (d) The hypothesis is merged into the built map for comparing with Fig. 3(a). (e)The simulation environment ($23\text{m} \times 20\text{m}$).

2) Homogenous Transformation Matrix: We select two corner features as a pair of control points, where one point is in the target region and the other one is in the reference region. With the pair of control points, we establish a 3×3 homogenous-coordinate-transformation matrix and utilize it with the control points to obtain one possible aligned reference region, $\tilde{C}_{\tilde{f}'}$ where $\tilde{f}'$ is the reference cell.

B. Similarity measure

When we obtain one possible reference region, $\tilde{C}_{\tilde{f}'}$, and a target region, S_f, the next step is to calculate the similarity measure of these two regions. Consider the filters $I(\cdot)$ and $J(\cdot)$ for filtering out the occupied cells,

$$I(i,j) = \begin{cases} 1 & \text{if } S_f(i,j) \text{ and } \tilde{C}_{\tilde{f}'}(i,j) \text{ are occupied} \\ 0 & \text{else} \end{cases} \tag{2}$$

$$J(R,i,j) = \begin{cases} 1 & \text{if } R(i,j) \text{ is occupied} \\ 0 & \text{else} \end{cases} \tag{3}$$

where $R(i,j)$ is an input region and i, j are the position indexes of the cell. Using Eqs. (2) and (3), the similarity measure of S_f and $\tilde{C}_{\tilde{f}'}$ can be established as

$$\Psi(S_f, \tilde{C}_{\tilde{f}'}) = \frac{2 \times \sum_{i=1}^{d_s} \sum_{j=1}^{d_s} \mathrm{I}(i,j)}{\sum_{i=1}^{d_s} \sum_{j=1}^{d_s} J(S_f,i,j) + \sum_{i=1}^{d_s} \sum_{j=1}^{d_s} J(\tilde{C}_{\tilde{f}'},i,j)} \tag{4}$$

where d_s is the size of the input region obtained from $d_s = \lceil \frac{d}{\text{scale}} \rceil$, and the scale is the resolution of the map.

C. Hypothesis generation

With extracted features and homogenous transformation matrices, we search every possible reference region and calculate their similarity measures. We then obtain the reference region $C_{f'}$ with the highest similarity-measure value and utilize it to generate a hypothesis. An example of hypothesis generation is shown in Fig. 4. The original built map is shown in Fig. 3(a). The selected target region and the found reference region are shown in Fig. 4(a) and Fig. 4(b), respectively. The calculated similarity-measure value is 0.53, and the generated hypothesis is shown in Fig. 4(c). In Fig. 4(d), the prediction result is merged into the built map for comparison with Fig. 3(a). Comparing with the simulation environment Fig. 4(e), the prediction result shows that the environmental structure predictor correctly estimates the structures in the unexplored region. In addition, to prevent low-quality predictions, a threshold value of similarity measure is selected between 0.4 and 0.7. From extensive computer simulations, a similarity-measure value above 0.4 usually indicates the existence of a similar structure with a reasonable quality of matching.

IV. Prediction-based SLAM (P-SLAM)

In this section, we shall first review the Bayesian formulation of a traditional SLAM algorithm and then extend it to derive the Bayesian formulation of P-SLAM. Then we describe the implementation of P-SLAM with a Rao-Blackwellized particle filter.

A. Bayesian formulation of P-SLAM

Consider the set of robot locations, motion commands and observations, respectively, as follow:

$$X_k = \{x_0, x_1, \ldots, x_k\} = \{X_{k-1}, x_k\} \tag{5}$$

$$U_k = \{u_1, u_2, \ldots, u_k\} = \{U_{k-1}, u_k\} \tag{6}$$

$$Z_k = \{z_1, z_2, \ldots, z_k\} = \{Z_{k-1}, z_k\} \tag{7}$$

where k denotes the discrete-time index, x_k denotes the robot pose, u_k denotes the motion command, z_k denotes the observation, and x_0 is the initial location of the robot. Given the motion commands U_k and the observations Z_k, the SLAM problem is to calculate the distribution of $P(X_k, \mathcal{M}|Z_k, U_k)$. One possible way to estimate $P(X_k, \mathcal{M}|Z_k, U_k)$ is to consider all the possible maps and poses. However, the high-dimensional spaces in the maps and the poses make the SLAM problem intractable. To reduce the dimensionality of the search space, two assumptions are made – the motion model is Markov and the environment is stationary. Then $P(X_k, \mathcal{M}|Z_k, U_k)$ can be derived in a recursive form [13] expressed as

$$P(x_k, \mathcal{M}|Z_k, U_k) = \kappa \times P(z_k|x_k, \mathcal{M}) \times \int P(x_k|x_{k-1}, u_k) \times P(x_{k-1}, \mathcal{M}|Z_{k-1}, U_{k-1}) dx_{k-1} \tag{8}$$

where κ is a normalizing constant. Next, we consider the SLAM problem with hypotheses (i.e., predicted maps). The set of hypotheses is expressed as $H_k = \{h_1, h_2, \ldots, h_k\} = \{H_{k-1}, h_k\}$. The posterior probability of P-SLAM at time k is expressed as $P(x_k, H_k, \mathcal{M}|Z_k, U_k)$. We omit the derivation

due to the space limitation and obtain the recursive form of P-SLAM as

$$P(x_k,H_k,\mathcal{M}|Z_k,U_k) = \kappa \times P(z_k|\mathcal{M},x_k)P(h_k|\mathcal{M},x_k)\times \int P(x_k|x_{k-1},u_k)\times P(x_{k-1},H_{k-1},\mathcal{M}|Z_{k-1},U_{k-1})dx_{k-1}. \quad (9)$$

Comparing Eq. (9) with the traditional SLAM in Eq. (8), the difference is that the P-SLAM has an additional term $P(h_k|\mathcal{M},x_k)$, which represents the probability of observing h_k given $\mathcal{M}$ and x_k. Since h_k is a hypothesis and there is no direct way to calculate $P(h_k|\mathcal{M},x_k)$, we thus resort to utilizing the similarity measure as an approximation to $P(h_k|\mathcal{M},x_k)$. This is based on the property that a higher similarity-measure value will lead to a higher probability of observing h_k.

B. P-SLAM with a Rao-Blackwellized particle filter

We utilize a Rao-Blackwellized particle filter (RBPF) [14] and an occupancy-grid map as the backbone of P-SLAM. We also stabilize raw odometry readings by laser-scan matching before sending to the RBPF [15].

The procedures and notations in the RBPF are similar to [15], [16]. Each particle (n) is a tuple $\langle x_k^{(n)}, m_k^{(n)}, h_k^{(n)}, w_k^{(n)} \rangle$, where $x_k^{(n)}$ is the robot pose at time k, $m_k^{(n)}$ is the constructed map, $h_k^{(n)}$ is the hypothesis map, $w_k^{(n)}$ is the particle weight for re-sampling, and T_k is the exploration target at time k. The update of each particle at time k is as follow:

$$\begin{aligned} x_k^{(n)} &= A(u_{k-1}, x_{k-1}^{(n)}) \\ m_k^{(n)} &= M(z_k, x_k^{(n)}) + m_{k-1}^{(n)} \\ h_k^{(n)} &= V(m_k^{(n)}, h_{k-1}^{(n)}) + H(m_k^{(\text{best})}, x_k^{(\text{best})}, x_k^{(n)}, T_k) \\ w_k^{(n)} &= S(z_k, x_k^{(n)}, m_{k-1}^{(n)}, h_{k-1}^{(n)}) w_{k-1}^{(n)} \end{aligned} \quad (10)$$

where $A(\cdot)$, $M(\cdot)$, and $S(\cdot)$ are the action, map, and sensor models, respectively. More details about these models can be found in [16]. The hypothesis-validation model, $V(\cdot)$, validates the hypothesis by comparing $m_k^{(n)}$ and $h_{k-1}^{(n)}$, and removing explored parts from the hypothesis. For the hypothesis-generation model, $H(\cdot)$, it utilizes the built map from the current best particle, denoted by the superscript (best), and the next movement target T_k. Note that the hypothesis is only generated when a new target is selected and the similarity-measure value is higher than the threshold. In addition, the generated hypothesis for each particle is also adjusted to the relative pose of the best particle. This avoids extensive prediction in each particle since the map in each particle is similar to each other.

C. Worst-case scenarios

We identify two different cases of worst-case scenarios in P-SLAM. The first one is that an environment is totally irregular without any common structure. This results in no knowledge can be utilized from a built map, and no hypothesis can be generated. In this extreme case, the penalty is the computational cost of the look-ahead mapping of P-SLAM algorithm. The mapping result will be exactly the same as the embedded SLAM algorithm.

The second worst-case scenario occurs when the prediction is inaccurate or incorrect and the robot utilizes it in Eq. (10). According to Eq. (9), the updating strength of $h_{t-1}^{(n)}$ in $S(\cdot)$ is set to the similarity measure when it was generated. Hence, the influence of a hypothesis is determined by its similarity measure. If a hypothesis has a low similarity-measure value, the observation z_k overlapped with $m_{k-1}^{(n)}$ will dominate the particle weight update in Eq. (10). Therefore, the system will recover after several iterations of re-sampling process. However, if a hypothesis has a high similarity-measure value, or a robot suddenly moves into a new region with small observation overlapping with the built map, the hypothesis will dominate the particle weight update. Due to the bound of the action model, the mapping result will be locally correct. For global consistency, the mapping can be improved by post-processing with an algorithm proposed by Lu and Milios [17].

V. Experimental Results

A. Computer simulations

To validate the effectiveness and correctness of the proposed P-SLAM, we have performed computer simulations on Player/Stage [18], which is a popular mobile-robot simulator and control software. We simulated a P3-DX mobile robot with a SICK LMS-200 laser ranger, running at a top speed of 0.2m/sec. For the mapping parameters, the resolution of the occupancy-grid map was 0.1m, the surrounding range d for prediction was 10m, and the threshold of the similarity measure was 0.6. A symmetric square maze was selected as a testing environment to demonstrate the look-ahead mapping, and a normal indoor-office environment was selected to test the correctness of the prediction. Both of them have many common features/structures, which can be utilized in the P-SLAM.

1) Simulation in a symmetric square maze environment: In this simulation, we demonstrate the prediction capability of P-SLAM when the environment is highly regular. In addition, we show that a robot can save the exploration time by leaving out some predicted regions from exploration. As shown in Fig. 5(a), the selected square maze has an area of 625m^2 and has many repeated similar structures. The width of the square obstacles is 3.5m, and the corridor width is 2.2m. We performed a full coverage with a scan-line exploration strategy. It took 684 sec. to explore the environment, and the mapping result of the traditional SLAM is shown in Fig. 5(b). Instead of the full coverage, we navigated the robot movement to avoid exploring the predicted regions generated by P-SLAM. The exploration took 457 sec., and the mapping result is shown in Fig. 5(c). In Fig. 5(d), the hypotheses are "added" to the P-SLAM mapping result assuming that the environmental boundary is known. Figure 5(d) shows that the P-SLAM predicted the structures inside the unexplored regions reasonably well. Since we did not explore the predicted regions, the exploration time was reduced by 227 sec., which is a 33% reduction comparing to the full exploration. If the environmental boundary is unknown, then

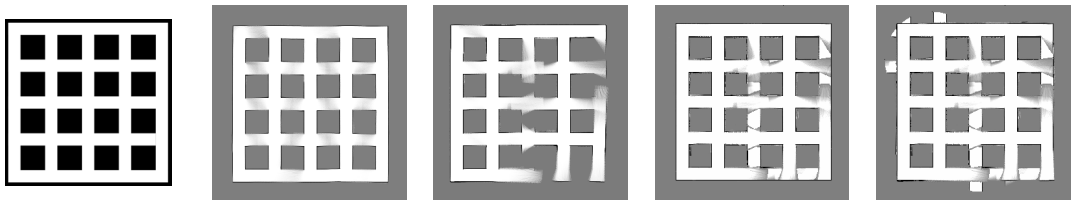


Fig. 5. (a) The square maze environment used in the computer simulation. (b) The mapping result of the full coverage with a scan-line exploration strategy. (c) The P-SLAM mapping result with designated robot movement. (d) The hypotheses are "added" to the P-SLAM mapping result with the environmental boundary known. (e) same as (d) when the hypotheses are "added" to the P-SLAM mapping result but without knowing the environmental boundary.

Fig. 5(d) will become Fig. 5(e), where some hypotheses are outside the square maze (e.g., upper-left corner). This is because the environmental structure predictor does not know the boundary and considers those regions as unexplored regions and generates hypotheses for them.

2) Simulation in an office environment: The environment was a blueprint of the SDR site B selected from the robotics data set repository [19]. During the exploration, 56 hypotheses were generated, and the covered area of the upper-half office environment has the size of about about 2000m^2. The total predicted size of the 56 hypotheses was 605.76m^2, which was 30% of the covered area.

To verify the correctness of the predicted hypotheses, we navigated the robot to explore the predicted regions. When we obtained the built map of the predicted area, we compared the built map and the hypotheses by utilizing the similarity measure described in Section III-B. The similarity-measure values of these hypotheses and the final built map varied from 0.10 to 0.86, and the distribution is shown in Fig. 6. In addition, there were 36 hypotheses with similarity-measure values higher than 0.4 and the average similarity measure was 0.51, which indicated the prediction was very effective in the office environment. Most of the successful predictions happened at the location of repeated structures such as the rooms or hallways. As shown in the final map in Fig. 7, some of the predicted regions cannot be reached by the robot, however, these regions can be removed by a post processing with a path-planning algorithm.

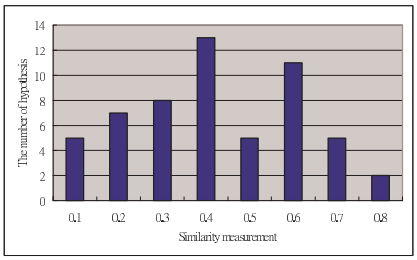


Fig. 6. The distribution of the number of hypothesis and the similarity measure with the built map.

B. Experiments on a Pioneer 3-DX robot

We implemented the proposed P-SLAM algorithm on a P3-DX mobile robot equipped with a SICK LMS-200 laser ranger. All the required computations were handled by a laptop with a 2.0GHz Pentium-M processor and 2GB memory. The number of particles in the RBPF was set to 200, which was tested for real-time execution in our system.

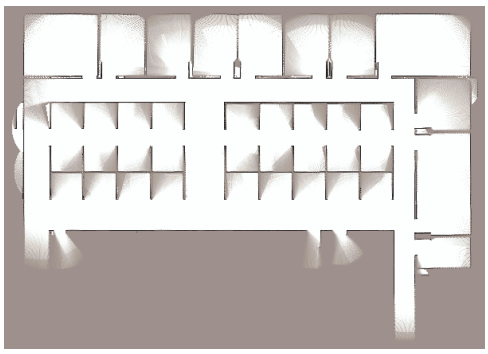


Fig. 7. The map of the office environment generated by the P-SLAM.

The testing environment was a corridor inside an office building as shown in Fig. 8. The corridor only has very few corner features and similar parallel lines. This resulted in no similar structures can be found due to the lack of corner features. Thus, we arbitrarily broke down the parallel lines into several segments by manually adding boxes in the corridor. These boxes provided additional corner features. With these boxes, the P-SLAM could detect the corridor with repeated similar structures.

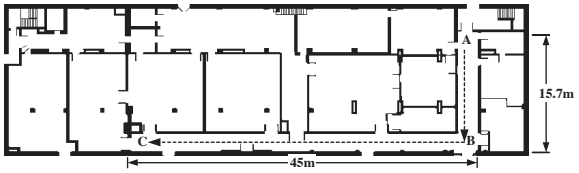


Fig. 8. The blueprint of the experimental environment. The mobile robot explored the 'L' shape corridor starting from location 'A' and ending at location 'C'.

Figures 9(a) and 9(b) show the final maps generated by the RBPF and by the P-SLAM, respectively. Comparing these two final maps, the P-SLAM provided a better result with less bending. The reason is that the map of corridor $\overline{AB}$ in Fig. 8 provides a good template for future predictions in corridor $\overline{BC}$. Hallway $\overline{AB}$ is a good template because $\overline{AB}$

is shorter, and the laser ranger can have additional laser readings at location 'B' for scan matching. Figure 10 shows how the generated hypothesis assists the traditional SLAM. The location marked by "I" indicates the built map without considering the hypothesis, and the location marked by "II" indicates the generated hypothesis. The hypothesis affected the update weight of the particle filter and resulted in a better map. We also examined the hallways by checking the tiles on the floor and found that the corridor is a structure of parallel lines. This result matches to our previous experience that a corridor is usually a pair of parallel lines instead of a pair of slightly bent lines. The result also reveals that a good built map is critical to P-SLAM.

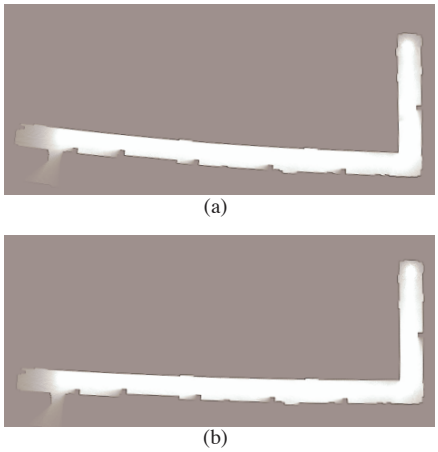


Fig. 9. (a) The final map generated by the RBPF. (b) The final map generated by the P-SLAM. The map of the corridor $\overline{BC}$ is less bending comparing to (a).

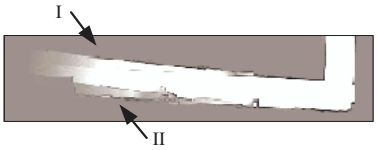


Fig. 10. The location marked by "I" indicates the built map without considering the hypothesis generated by the P-SLAM. The location marked by "II" indicates the generated hypothesis, which looks like the shadow of the built map at marked I.

VI. Conclusions

In this paper, we have presented the proposed P-SLAM algorithm. With the P-SLAM, a mobile robot can predict an unexplored region for look-ahead mapping. In addition, we have also derived the Bayesian formulation of P-SLAM, which merges the predicted maps into the traditional SLAM Bayesian formulation. The Bayesian formulation generalizes how to utilize the predicted information in the SLAM problem. Computer simulations have shown that the P-SLAM is very effective in an indoor environment. Finally, we implemented the P-SLAM on a Pioneer 3-DX mobile robot in real-time, and the experimental results showed that the P-SLAM improved the results of traditional SLAMs. It also reveals that a good built map or a good database is important for prediction.

References

[1] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.

[2] M. G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[3] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A. Ng., "Simultaneous mapping and localization with sparse extended information filters," in *Proc. of the Fifth Int. Workshop on Algorithmic Foundations of Robotics*, 2002.

[4] J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Proc. of IEEE Int. Workshop on Intelligent Robots and Systems*, 1991, pp. 1442–1447.

[5] J. Neira and J. Tardos, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. on Robotics and Automation*, vol. 17, pp. 890–897, Dec 2001.

[6] H. J. Chang, C. S. G. Lee, Y.-H. Lu, and Y. C. Hu, "A computational efficient slam algorithm based on logarithmic-map parititioning," in *Proc. of IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Sep. 2004, pp. 1041 – 1046.

[7] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. of the AAAI National Conf. on Artificial Intelligence*. Edmonton, Canada: AAAI, 2002.

[8] J. Meltzer, R. Gupta, M.-H. Yang, and S. Soatto, "Simultaneous localization and mapping using multiple view feature descriptors," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2004, pp. 1550–1555.

[9] H. J. Chang, C. S. G. Lee, Y.-H. Lu, and Y. C. Hu, "Energy-time-efficient adaptive dispatching algorithms for ant-like robot systems," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 4, April 2004, pp. 3294 – 3299.

[10] D. Schroter, T. Weber, M. Beetz, and B. Radig, "Detection and classification of gateways for the acquisition of structured robot maps," in *In Proc. of 26th Pattern Recognition Symposium*, Aug. 2004, pp. 553–561.

[11] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 1, Sept. 2003, pp. 842–849.

[12] A. A. Goshtasby, *2-D and 3-D Image Registration for Medical, Remote Sensing, and Industrial Applications*. Wiley Publishers, 2005.

[13] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Probabilistic algorithms and the interactive museum tour-guide robot minerva," *Int. Journal of Robotics Research*, vol. 19, no. 11, pp. 972–999, 2000.

[14] A. Doucet, J. de Freitas, K. Murphy, and S. Russel, "Rao-blackwellized particle filtering for dynamic bayesian networks," in *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2000, pp. 499–516.

[15] D. Hähnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003, pp. 27–31.

[16] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," in *Proc. of Robotics: Science and Systems*, 2005.

[17] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, pp. 333–349, 1997.

[18] R. T. Vaughan, B. P. Gerkey, and A. Howard, "On device abstractions for portable, reusable robot code," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2003, pp. 2121–2427.

[19] A. Howard and N. Roy, "The robotics data set repository (radish)," 2003. [Online]. Available: http://radish.sourceforge.net/