Sensor Planning for Mobile Robot Localization - A hierarchical approach using Bayesian network and particle filter -

Hongjun Zhou
Chuo University, Japan
zhou@indsys.chuo-u.ac.jp

Shigeyuki Sakane
Chuo University, Japan
sakane@indsys.chuo-u.ac.jp

***Abstract*— In this paper we propose a hierarchical approach to solving sensor planning for the global localization of a mobile robot. Our system consists of two subsystems: a lower and a higher layer. The lower layer uses a particle filter to evaluate the posterior probability of the localization. When the particles converge into clusters, the higher layer starts particle clustering and sensor planning to generate an optimal sensing action sequence for the localization. The higher layer uses a Bayesian network for the probabilistic inference. The sensor planning takes into account both localization belief and sensing cost. We conducted simulations and actual robot experiments to validate our proposed approach.**

***Index Terms*— Sensor planning, Localization, hierarchical approach, Particle filter, Bayesian Network**

I. Introduction

In recent years, global localization of mobile robots has been an active area of research and various methods have been developed. However, most of the studies have taken a passive approach [1] in the sense that the robot's actions are random and are not planned explicitly for efficient localization. Let us consider an example of mobile robot global localization in the environment as depicted in Fig. 1(a). The black circle on the right side shows the current pose of the robot. We assume that the robot has two candidate poses indicated by the white circle on the left and the black circle on the right, when the robot took some sensing actions. Then, if the robot takes a sequence of sensing actions along the path indicated by the dotted line, localization will make slow progress since the robot still finds difficulties in discriminating between the two possibilities with similar local structures in the paths. By contrast, if the robot takes a sequence of sensing actions downward along the other path (the solid line), localization could be achieved much faster by sensing the dissimilar structures. This example suggests that some inference of the environmental structure allows the robot to perform sensor planning for efficient localization. In this paper, we propose a hierarchical approach for performing sensor planning for global localization.

II. System Structure

As shown in Fig.2, the mobile robot estimates its pose by the hierarchical integration of the two layers. We initially decompose the environmental map into some large cells (see Fig.6(a)), and perform clustering of each cell's geometrical features to discriminate the classes of the cells. The geometrical features of the cells is extracted from range data that the robot gathered at the central point of each large cell. The lower layer uses a particle filter to evaluate the posterior probability of the localization. When the particles converge into clusters, the higher layer starts particle clustering and sensor planning to generate an optimal sensing action sequence for the lower layer's localization.

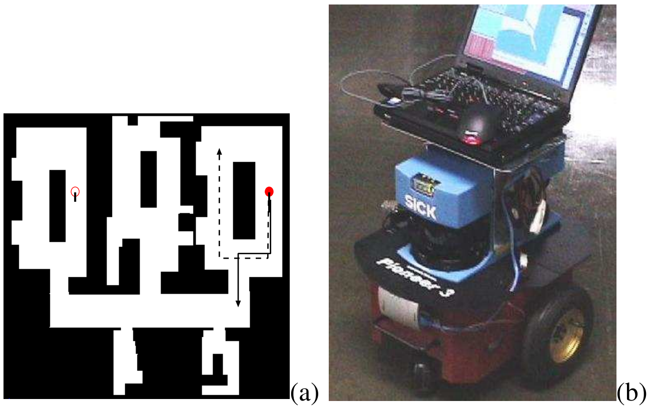


Fig. 1. (a) An example of sensor planning for mobile robot localization. (b) A Pioneer 3 mobile robot with two SICK laser range sensors.

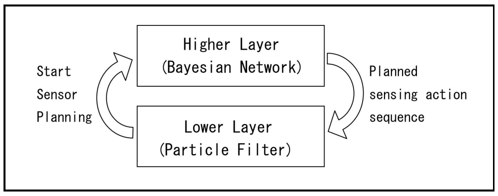


Fig. 2. A hierarchical architecture for mobile robot localization

The higher layer consists of a Bayesian network [10] which represents the contextual relationships between the geometrical features of the cells, the sensing actions, and the global localization beliefs. The Bayesian network allows us to predict the possible sensing actions and geometrical features of the next cell. Based on the predicted information, the system estimates gross localization, then calculates an integrated utility to evaluate each sensing action by taking into account the

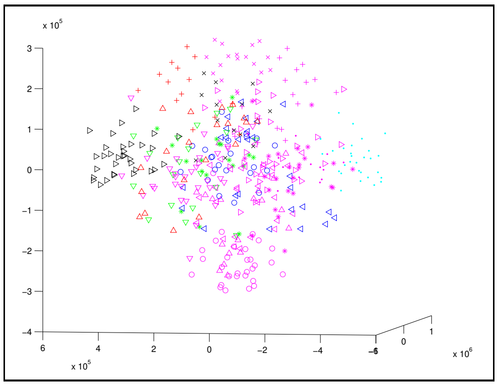


Fig. 3. Sensor data of each cell which was classified into 20 classes using K-means clustering. Classified data was projected in the first three eigenvectors. The graph shows represents different clusters with different symbols.

gross localization belief and the sensing cost. In this paper, the term *gross localization* means determining the large cell in which the robot is currently in, and the robot's orientation. The higher layer generates an optimal sensing action sequence and, in the planned sensing actions, the lower layer localizes the robot more accurately using a particle filter. Consequently, our hierarchical system features the benefits of both efficient and accurate localization.

III. Cell Decomposition of the Environment

The gross localization is based on large cell decomposition. To obtain the geometrical features of the local environment which a robot might come across in the navigation, the system decomposes the environment into cells. Figure 6(a) shows an example of the exact cell decomposition[11] which is performed twice, i.e., from left to right and from top to bottom.

In this paper, we decompose the environment using exact cell decomposition[11], we can also decompose the environment into large cells with same size if the robot environment is more complicated. The decomposition result does not effect the algorithms of geometrical features classification, sensor planning and global localization.

Our system uses a mobile robot (Pioneer 3) (Fig. 1(b)) with two laser range sensors (SICK, LMS200). The range data is generated covering a 360-degree view. Since the system employs a cell-based representation, the robot has to recognize the cell in which the robot is in. To solve this problem, the system performs clustering of the geometrical features of the cells. The robot gatherers range sensor data at the central point of every large cell with four compass directions, and classifies range data into classes using a K-means clustering algorithm [12]. Figure 3 shows an example of the results classified into 20 classes. Using these classified results, we can represent and recognize the geometrical features of the cells.

IV. Lower Layer for Accurate Localization

In the past decade, particle filter [13] has been applied to various areas such as visual tracking [14] and mobile robot localization [15]. It estimates the posterior probabilities of unobservable state variables from sensor measurements. In the application to mobile robot localization, a state typically represents a pose of the robot in the environment. Particle filter approximates the posterior using a group of samples (or particles) with weights that correspond to the likelihood of the states. In every time step, the particles are resampled based on the weight values. A particle filter for mobile robot localization is shown in Fig. 4. The pose of each particle simulates the possible pose of the mobile robot. To perform global localization, the initial particles have a uniform distribution in the free space of the environment (Fig. 4(a)). The initial sensing action of this example is random without planned action. After a sequence of sensing actions, the particles converge into clusters (Fig. 4(b), (c)). When the mobile robot turns left at an intersection, the particles conveniently converged into one cluster (Fig. 4(d)).

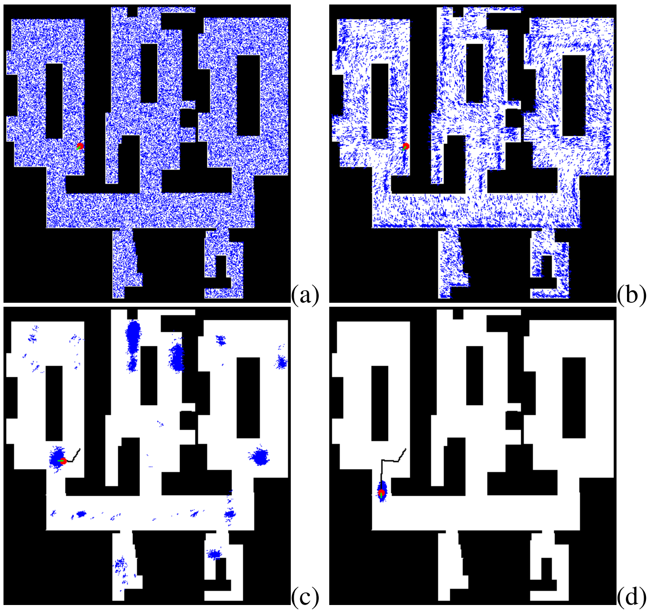


Fig. 4. A example of mobile robot localization using particle filter by random actions.

The role of the lower layer is twofold: to tell the higher layer when it should start sensor planning, and to compensate for the accuracy of the gross localization performed by the higher layer. The details are described in Section VI, which discusses our simulation experiments.

V. Higher Layer Control

A. Bayesian network design

The higher layer employs a Bayesian network for sensor planning. We use a Bayesian network as shown in Fig. 6. The nodes S_{t+1} and S_t denote the states at time $t+1$ and t, respectively. A state is a pose of the mobile robot at each large cell, and a state transition occurs when the robot moves to the next cell or changes its orientation. The higher layer performs gross localization.

We quantize the robot orientation in the cell into four directions (east, west, north and south), and the robot has four states in each cell. The node $Action$ denotes the robot motion while moving from one cell to the neighboring cell. The nodes O_{t+1} and O_t denote the observation (sensor measurement) obtained at states S_{t+1} and S_t, respectively. The observation is not raw sensor data at each cell but is the classified results of the sensor data (see Section III).

In the following experiments, the environment is decomposed into 103 cells and the robot has four possible orientations at each cell, so the robot has $103 * 4 = 412$ states. These states are represented by the same number of random variables and each variable has a probability, i.e., a real value from 0 to 1. The node S_t includes these 412 variables and the node S_{t+1} has 413 random variables which correspond to the 412 states plus an error state (exception or obstacle). The node $Action$ includes four random variables which correspond to four actions: go forward, go back, turn left, and turn right. The nodes O_{t+1} and O_t have 20 random variables as defined in Section III. The initial values of the node S_t are generated from the sum of the weights of particles in each particle cluster.

B. Conditional Probability Tables Learning of the Bayesian Network

The Bayesian network is represented in two factors [10], e.g. graph structure and CPTs. The graph structure of a Bayesian network represents the casual relations between the nodes. The graph structure of our Bayesian network were designed manually (see Fig. 6). CPTs are the conditional probability tables of the nodes can be learned from case data directly. Since our environment information data is complete, we use the $Maximum\ Likelihood\ Estimation$[16] method to learn the CPTs. For example, The conditional probability of node X's value x_i is $P(X = x_i|Z_1 = z_1, Z_2 = z_2, ..., Z_k = z_k)$, and $Z_1, Z_2, ..., Z_k$ is K parents nodes of node X. $z_1, z_2, ..., z_k$ are denoted by the values of nodes $Z_1, Z_2, ..., Z_k$. So the conditional probability $P(X = x_i|Z_1 = z_1, Z_2 = z_2, ..., Z_k = z_k)$ should be calculated as the following:

$$\begin{aligned} & P(X = x_i|Z_1 = z_1, Z_2 = z_2, ..., Z_k = z_k) \\ = & \frac{P(X = x_i, Z_1 = z_1, Z_2 = z_2, ..., Z_k = z_k)}{P(Z_1 = z_1, Z_2 = z_2, ..., Z_k)} \\ \simeq & \frac{N(X = x_i, Z_1 = z_1, Z_2 = z_2, ..., Z_k = z_k)}{N(Z_1 = z_1, Z_2 = z_2, ..., Z_k)} \end{aligned}$$

$N(X = x_i, Z_1 = z_1, Z_2 = z_2, ..., Z_k = z_k)$ denotes the number of training data cases which has $X = x_i, Z_1 = z_1, Z_2 = z_2, ..., Z_k = z_k$. $N(Z_1 = z_1, Z_2 = z_2, ..., Z_k = z_k)$ denotes number of the data case that has $Z_1 = z_1, Z_2 = z_2, ..., Z_k = z_k$ in the total environment data.

C. Soft evidence of Bayesian Network

In Section III, our system has classified the geometrical features of the cells using K-mean clustering. However,if the geometrical features of each cell are similar, the class boundaries may be ambiguous for recognizing the class. If we use the classification results directly as evidences to the Bayesian network, the inference could be too sensitive to estimate the robot pose. Since the Bayesian network allows us to use *soft evidence* [10] for inference, we use the nodes O_{t+1} and O_t as the soft evidences to cope with the uncertainties of the boundaries.

For example, we assume the node O_t has sensor data obtained by the robot in a cell i with an orientation m. We also assume the sensor measurement (denoted by P_{im}) is classified into a class n. The system calculates the distances between the P_{im} and a center point of each class, then sorts all of the distances. The system converts, by normalization, the distances to beliefs or probabilities of the classification to the cell. The probabilities are the soft evidences of the observation node O_t.

D. Inference for Sensor Planning

Our goal is to select an optimal sensing action sequence for localization. The evaluation of action sequence are performed based on the inference of the Bayesian network and a utility function (Eq. 10).

As shown in the following procedure, sensor planning for localization of the higher layer has four steps:

- Joint probability of nodes calculation
- Estimating Possible Actions
- Predicting Possible States
- Sensor Planning for Localization

1) Joint probability of Nodes Calculation: Initially, we calculate the joint probability of the Bayesian network's nodes (Eq. 1). The joint probability of all of nodes is calculated as the following formulations.

$$\begin{aligned} & P(A, S_t, S_{t+1}, O_t, O_{t+1}) && (1) \\ \overset{\text{CR}}{=} & P(A) * P(S_t|A) * P(S_{t+1}|S_t, A) * \\ & P(O_t|A, S_t, S_{t+1}) * P(O_{t+1}|A, S_t, S_{t+1}, O_t) \\ \overset{\text{CI}}{=} & P(A) * P(S_t|A) * P(S_{t+1}|S_t, A) * \\ & P(O_t|S_t, S_{t+1}) * P(O_{t+1}|S_{t+1}) && (2) \end{aligned}$$

Based on the chain rule (CR) of probability and the conditional independence (CI) relationships between the nodes, the joint probability Eq. 1 can rewrite in Eq. 2. In Eq. 2, the conditional probabilities are learned by the $Maximum\ Likelihood\ Estimation$ method which is described in section V-B.

2) Estimating Possible Actions: Using the joint probability we can estimate the conditional probability distribution of the node $action$ given the evidences of observation nodes O_t and current state node S_t as Eq. 4.

$$\begin{aligned} P(A = a|\hat{s}_t, \hat{o}_t) & \overset{\text{CI}}{=} P(A|S_t = \hat{s}_t) && (3) \\ & \overset{\text{Bayes}}{=} \frac{P(A = a, S_t = \hat{s}_t)}{P(S_t = \hat{s}_t)} && (4) \end{aligned}$$

where:

$$P(A=a, S_t=\hat{s_t}) = \sum_{(\hat{o_t}, \hat{o_{t+1}}, \hat{s_{t+1}})} P(S_{t+1}=\hat{s_{t+1}}, A=a, S_t=\hat{s_t}, O_t=\hat{o_t}, O_{t+1}=\hat{o_{t+1}})$$

and

$$P(S_t=\hat{s_t}) = \sum_{(a, \hat{o_t}, \hat{o_{t+1}}, \hat{s_{t+1}})} P(S_{t+1}=\hat{s_{t+1}}, A=a, S_t=\hat{s_t}, O_t=\hat{o_t}, O_{t+1}=\hat{o_{t+1}})$$

In Eq. 3, a, $\hat{s_t}, \hat{o_t}$ and $\hat{o_{t+1}}$ are the evidences of the nodes $Action$, S_t, O_t and O_{t-1}, respectively. We define the possible actions in $\bar{A} = \{a_1^\star, ..., a_N^\star\}$. Each possible action $a_n^\star \in \bar{A}$ must satisfy the following condition:

$$P(A=a_n^\star | S_t=\hat{s_t}, O_t=\hat{o_t}) > 0 \tag{5}$$

3) Predicting Possible States: The goal of this step is to predict the next states when the robot takes a possible action $a_n^\star \in \bar{A}$. Initially, we calculate the conditional probability distribution of node S_{t+1} given the evidences of the nodes S_t, O_t and one of possible action $a_n^\star \in \bar{A}$ using the following formulations.

$$P(S_{t+1}=\hat{s_{t+1}} | A=a_n^\star, \hat{s_t}, \hat{o_t}) \overset{\text{Bayes}}{=} \frac{P(S_{t+1}=\hat{s_{t+1}}, A=a_n^\star, \hat{s_t}, \hat{o_t})}{P(A=a_n^\star, \hat{s_t}, \hat{o_t})} \tag{6}$$

where:

$$P(A=a_n^\star, \hat{s_t}, \hat{o_t}) = \sum_{[\hat{s_{t+1}}, \hat{o_{t+1}}]} P(\hat{s_{t+1}}, A=a_n^\star, \hat{s_t}, \hat{o_t}, \hat{o_{t+1}})$$

and

$$P(S_{t+1}=\hat{s_{t+1}}, A=a_n^\star, \hat{s_t}, \hat{o_t}) = \sum_{[\hat{o_{t+1}}]} P(\hat{s_{t+1}}, A=a_n^\star, \hat{s_t}, \hat{o_t}, \hat{o_{t+1}})$$

Based on the inference of Eq. 6, we can obtain most-probable explanation $\hat{s_{t+1}^\star}$ that satisfies

$$\hat{s_{t+1}^\star} = argmax(P(\hat{s_{t+1}} | a_n^\star, \hat{s_t}, \hat{o_t}))$$

And the conditional probability of the most-probable explanation $\hat{s_{t+1}^\star}$ is calculated as the following formulation.

$$Prob(\hat{s_{t+1}^\star})_{a_n^\star} = P(\hat{s_{t+1}^\star} | a_n^\star, \hat{s_t}, \hat{o_t}) \tag{7}$$
$$= max(P(\hat{s_{t+1}} | a_n^\star, \hat{s_t}, \hat{o_t})) \tag{8}$$

We define the most-probable explanation $\hat{s_{t+1}^\star}$ is the predicted next state, and $Prob(\hat{s_{t+1}^\star})_{a_n^\star}$ is the probability of the predicted next state when the robot takes a possible action $a_n^\star$. Using the above formulations we can calculate all of $Prob(\hat{s_{t+1}^\star})_{a_n^\star}$ which correspond to each $a_n^\star \in \bar{A}$.

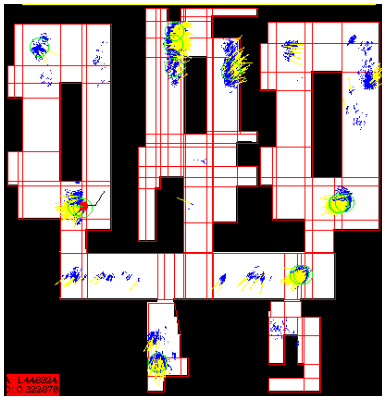


Fig. 5. Cell based particle clustering.

4) Sensor Planning for Localization: Based on the $Prob(\hat{s_{t+1}^\star})_{a^\star}$ and sensing cost, the mobile robot can find an optimal sensing action sequence for the global localization. We will describe it in detail in section VI-B.

VI. Simulation Experiments

A. Particle Clustering

Our system does not perform sensor planning at all times. The sensor planning starts when the particles converge into clusters. The system checks the averages and variances of the weights of the particles in all cells and at every resampling cycle. As shown in Fig. 5, when the average and the variance exceeds certain thresholds, the system starts to perform particle clustering and sensor planning.

To reduce the computational overhead, the system performs particle clustering based on cells. The system calculates the center of the particles in each cell and computes the distances between the center and the particles. Since variance of the distances indicates the density of the particles in each cell, we can locate clusters by thresholding the variances.

In Fig. 5, the circles show the locations of the particle clusters detected by the above procedures, and the bars on the particles show the orientations of the robot. The *higher layer* also starts sensor planning at this stage. Possible locations of the robot are focused on the cells with particle clusters. The probabilities of the possible locations are evaluated by the sum of the particle's weights in the cluster. The initial status of the *higher layer* is illustrated in Fig. 6(a). Candidate robot poses (white circle) are shown in the centers of the cells which have clusters, and the orientation of the robot is obtained from the average of the particles in the cluster. The probabilities of the clusters are calculated by the sum of the particles' weights. The gradation of the cell color in Fig. 6(a) shows the probability of the robot pose. That is, the darker (brighter) intensity corresponds to higher (lower) probability of the global localization.

B. Higher Layer Control for Sensor Planning

The particle filter performs filtering, predicting and smoothing the sensing data. However, it cannot control the actions of

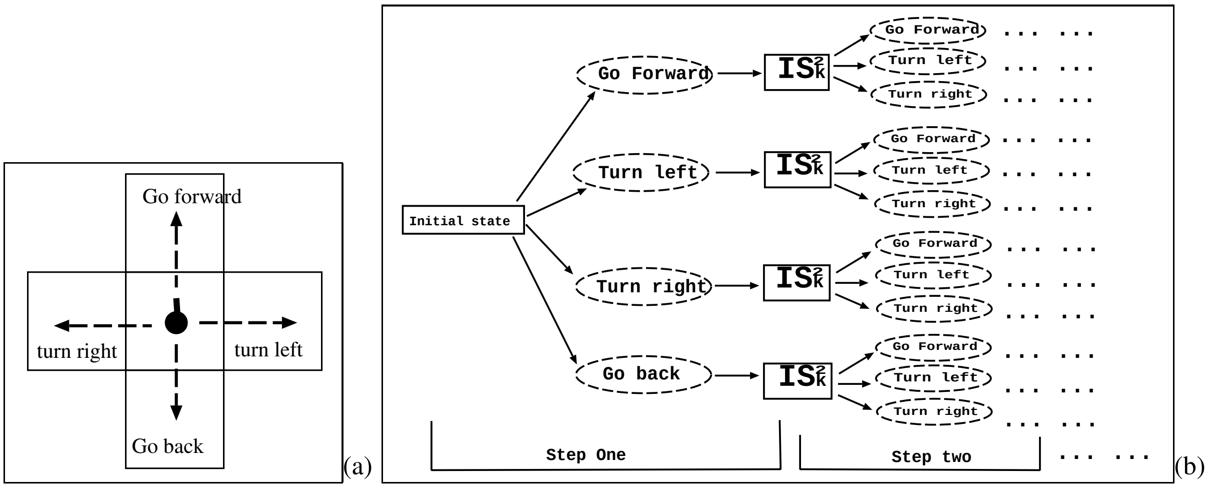


Fig. 7. (a) Four possible actions of the mobile robot in the sensor planning phase. (b) State transition during the sensor planning for mobile robot localization.

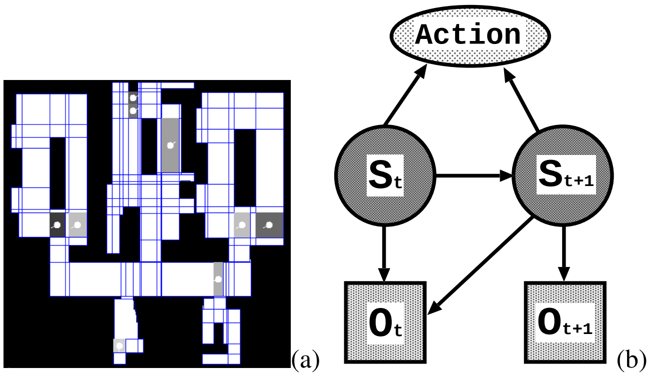


Fig. 6. (a) The initial status of the *higher layer control*. (b) The Bayesian network used in our system.

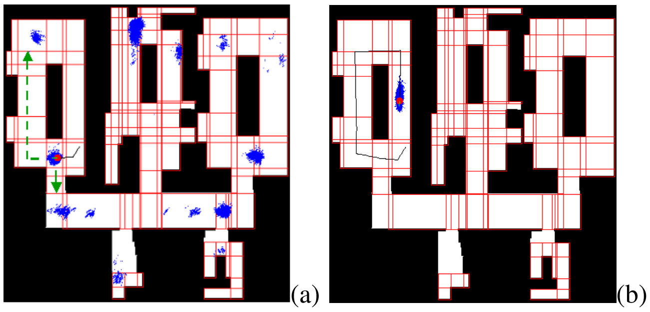


Fig. 8. (a) The mobile robot has two possible actions before the particles complete their converging. (b) If the mobile robot performs a forward action at the intersection, the particles need a long sequential action for complete convergence.

the mobile robot. If we only implement the particle filter for localization and do not consider the action control, random sensing actions may cause slow convergence of the particles. For example, in Fig. 8 (a), the robot faces an intersection before complete convergence. In Fig. 8 (b), if the robot takes a 'go forward' action at the intersection, the robot needs a long sequence of actions before convergence. In Fig. 4(d), the robot takes a 'turn left' action at the intersection, and the efficiency of localization is better than the sensing action of Fig. 8 (b). Therefore, to solve this problem, the *higher layer* is necessary to improve the efficiency of localization.

To reduce computational overhead of the planning, we assume there are four possible actions of the robot (e.g. *Go forward, Turn left, Turn right, and Go back*). (See Fig. 7(a)). After the particle clustering phase, the system locates some particle clusters. Cell-based sensor planning starts at this stage. The possible action sequences and state transition of each cluster during the sensor planning are shown in Fig. 7(b). In Fig. 7(b), $Initial\ state$ denotes one pose of the white circle (see Fig. 6(a)), which was found in the particle clustering phase. We use $IS_k \in \{IS_1, IS_2, ..., IS_K\}$ to denote one of $Initial\ state$. In the Fig. 6(a), there are nine $Initial\ states$, so the $K = 9$.

IS_k^2 is the pose of the robot, when the circle has finished a possible action, we call this step $Step\ one$. The white circles will move from $Initial\ state$ to the neighbor cells after $Step\ one$. When the circle takes a two-step action, the action sequences and state combinations are illustrated in $Step\ two$ of Fig. 7(b).

The system assumes the white circles perform all of the possible action sequences from the robot's initial pose, and estimates the gross global localization probabilities of the white circles using the Bayesian network, when the actions finished. We denote the average probability of all of the white circles to be '$AveBelief$', and '$Cost$' is defined by distance which the circle moved. For example, when the robot takes a possible action $a_n^\star$, the $AveBelief$ is calculated in the following equation.

$$AveBelief = \frac{\sum_{k \in K} Prob(s_{t+1}^{\hat{\star}})_{(a_n^\star, IS_k)}}{K} \tag{9}$$

We use $Prob(s_{t+1}^{\hat{\star}})_{(a_n^\star, IS_k)}$ to denote the predicted global localization belief of the robot, when $Initial\ state$ is IS_k,

Action	Average Probability (AveBelief)
Go forward	0.8010
Turn left	0.6754
Turn right	0.8630
Go back	0.8652

TABLE I

AVERAGE PROBABILITIES OF ALL OF THE PARTICLE CLUSTERS INFERRED BY THE BN WHEN THE SYSTEM PERFORMED ONE ACTION.

Action 1	Action 2	Average Probability (AveBelief)
Go forward	go forward	0.8862
Go forward	turn left	0.8558
Go forward	turn right	0.8010
Turn left	go forward	0.9460
Turn left	turn left	0.6754
Turn left	turn right	0.7623
Turn right	go forward	0.8803
Turn right	turn left	0.8761
Turn right	turn right	0.8735
Go back	go forward	0.7704
Go back	turn left	0.8091
Go back	turn right	0.7704

TABLE II

AVERAGE PROBABILITIES OF ALL OF PARTICLE CLUSTERS INFERRED BY THE BN WHEN THE SYSTEM PERFORMED TWO ACTIONS.

the Bayesian network is given soft evidences of the current state (S_t), observations (O_t, O_{t+1}), and robot takes a possible action $a_n^\star$, The calculation of $Prob(\hat{s_{t+1}^\star})_{(a_n^\star, IS_k)}$ is described in detail in section V-D.

$$\begin{aligned} Utility &= (1-\theta) * AveBelief + \frac{\theta}{Cost} \\ where: & \quad 0 < \theta < 1 \end{aligned} \tag{10}$$

Based on Eq. 10, by taking into account the balance between $AveBelief$ and $cost$, the system calculates the '$Integrated\ Utility$', and selects an optimal action sequence for the mobile robot localization based on the $Integrated\ Utility$.

For example, in Fig. 6(a), there are nine white circles which correspond to nine possible robot poses, and each possible pose corresponds to an $Initial\ state$ (see Fig. 7(b)). Initially, the system assumes that all circles perform '$Step\ one$'. As shown in Fig. 7, in this step, the robot has four possible actions. The average probabilities of nine circles (AveBelief) are shown in Table. I. We can see the $AveBelief$ is the largest, if the robot takes the action $go\ back$. If the largest $AveBelief$ does not exceed a certain threshold, the system has to perform the '$Step\ two$' or more, until it finds an $AveBelief$ which is larger than the threshold. In our experiments, we used 0.9 for the threshold. If the system finds $AveBelief$ which is larger than the threshold, an optimal action sequence will be selected by calculating the $Integrated\ Utility$ using the Eq. 10. The optimal action sequence must hold a largest $Integrated\ Utility$. Since, in the Table. I, the largest $AveBelief$ does not exceed the threshold (0.9), the system has to perform $Step\ two$. The system calculates the $AveBelief$s of nine circles, when they have finished "$Step\ one$ and $Step\ two$"(see Fig. 7). The calculated results are shown in Table. II. We see that the $AveBelief$ is the largest when the nine circles performed '$turn\ left$' and '$go\ forward$', and the largest $AveBelief$ (0.9460) exceeds the threshold (0.9). In Table II, we only find one $AveBelief$ which exceeds the threshold. So the action sequence, "$turn\ left$ then $go\ forward$", will be the optimal action sequence for localization. The simulation results in Fig. 8 and Fig. 4 show that the sensor planning is effective for the global localization.

VII. ACTUAL ROBOT EXPERIMENTS

As shown in Fig. 9, actual robot experiments of sensor planning for mobile robot localization were conducted. Initially, the particles were generated based on uniform distribution (see Fig.9(a)), then after some random sensing actions, the particles converged into clusters (Fig.9(b)). Based on the method described in Section VI-A, our system locates some particle clusters (see Fig. 9(c)). The sensor planner implemented in the higher layer starts at this stage. Through sensor planning by taking into account the trade-off between global localization belief and the sensing cost, the system obtains an optimal sensing action sequence for localization (see Fig. 9(d)). The particles completed their converging when the mobile robot finished the optimal sensing actions. The experimental results demonstrate the effectiveness of our proposed system.

VIII. PREVIOUS WORKS

Kristensen[2] proposed a sensor planning method based on a top-down decision tree algorithm. Miura and Shirai [9] defined an utility value based on sensing cost and sensing uncertainty to plan the sensor action for navigation. Jensfelt and Kristensen [3] used multiple hypothesis tracking, based on Kalman filtering, for active global localization. Basye et al.[6] built planning and control systems that integrated sensor fusion, prediction, and sequential decision-making using a temporal belief network (dynamic Bayesian network). Tomatis et al.[5] also proposed to integrate Kalman filter based metric map and POMDP based topological map. Fox et al.[4] proposed an *Active Markov Localization* to improve the efficiency of localization. The optimal sensing action sequence is selected based on an action selection function which taking into account the trade-off between increase of global localization probability entropy and the sensing cost. Zhou and Sakane [7], [8] proposed two systems to perform sensor planning for localization using Bayesian network.

IX. CONCLUSION

This paper presented a hierarchical approach for solving sensor planning for the global localization of a mobile robot. The environment is represented by the integration of a large cell based topological map and an occupancy grid map (grid size is 0.05 m). Our system consists of two subsystems:

a lower layer and a higher layer. The lower layer uses a particle filter to evaluate the posterior probability of localization based on the occupancy grid map. When the particles converge into clusters, the higher layer starts particle clustering and sensor planning to generate an optimal sensing action sequence for the localization based on the topological map. The higher layer uses a Bayesian network that represents contextual relationships between the geometrical features of the local environment, the sensing actions and the global localization beliefs. The sensor planning generates optimal action sequences by taking into account both the localization belief and the sensing cost. Simulations and actual robot experiments demonstrated the effectiveness of our proposed approach. Further improvements in actual environments are planned for future study.

REFERENCES

[1] S.Thrun, D.Fox, W.Burgard, and F.Dellaert, "Robust Monte Carlo Localization for Mobile Robots," *Artificial Intelligence*, 128(1-2), 2001.

[2] S. Kristensen, "Sensor Planning with Bayesian Decision Analysis," *PhD thesis*, Faculty of Technology and Science, Aalborg University, Aalborg, Denmark, 1996.

[3] P. Jensfelt and S. Kristensen, "Active Global Localization for a Mobile Robot Using Multiple Hypothesis Tracking," *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 5, pp.748-760, 2001.

[4] D. Fox, W. Burgard, and S. Thrun, "Active Markov Localization for Mobile Robots," *Robotics and Autonomous Systems*, Vol.25, pp.195-207, 1998.

[5] N. Tomatis, I. Nourbakhsh, K. Arras and R. Siegwart, "A Hybrid Approach for Robust and Precise Mobile Robot Navigation with Compact Environment Modeling," *Proc. of IEEE Int. Conf. on Robotics & Automation*, pp.1111-1116, 2001.

[6] K. Basye, T. Dean, J. Kirman, and M. Lejter, "A Decision-Theoretic Approach to Planning, Perception, and Control," *IEEE Expert*, Vol.7, No.4, pp.58-65, 1992.

[7] H. Zhou and S. Sakane, "Sensor Planning for Mobile Robot Localization using Bayesian Network Representation and Inference," *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.440-446, 2002.

[8] H. Zhou and S. Sakane, "Learning Bayesian Network Structure from Environment and Sensor Planning for Mobile Robot Localization," *Proc. 2003 IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent System (MFI2003), pp.76-81, 2003*.

[9] J. Miura and Y. Shirai, "Vision-Motion Planning for a Mobile Robot considering Vision Uncertainty and Planning Cost," *Proc. of 15th Int. Joint Conf. on Artificial Intelligence*, pp.1194-1200, 1997.

[10] F. V. Jensen, "Bayesian Networks and Decision Graphs," *Springer*, 2001.

[11] J.C. Latombe, "Robot Motion Planning," *Kluwer Academic Publishers*, 1996.

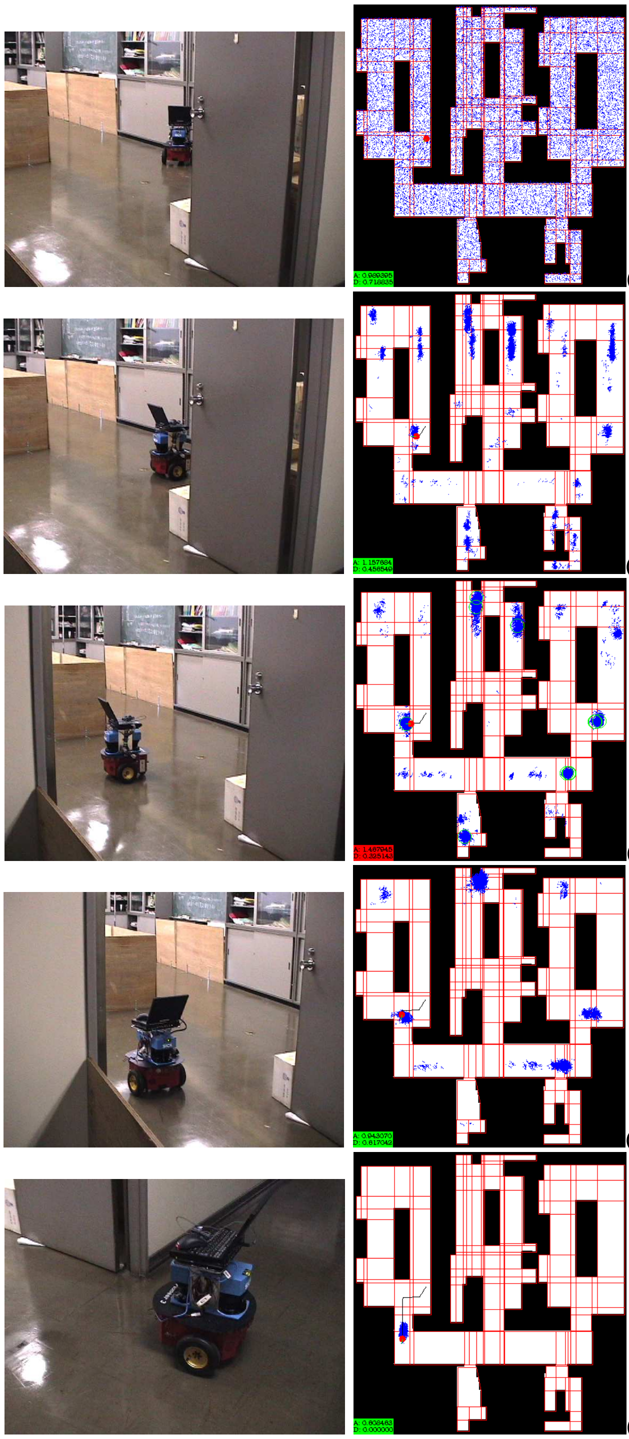
[12] Richard O. Duda, Peter E. Hart and David G. Stork, "Pattern Classification, 2/Edition," *Published by John Wiley & Sons Inc*, 2001.

[13] J. Liu and R. Chen. "Sequential monte carlo methods for dynamic systems. Journal of the American Statistical Association", 93:1032-1044, 1998.

[14] M. Isard and A. Blake. "Condensation: conditional density propagation for visual tracking. International Journal of Computer Vision", 29(1):5-28, 1998. In press.

[15] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. "Sequential Monte Carlo Methods In Practice". Springer Verlag, New York, 2001.

[16] D. Heckerman, "A Bayesian Approach to Learning Causal Networks," Technical Report MSR-TR-95-04, Microsoft Research, March, 1995.



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

Fig. 9. Sensor planning for mobile robot localization using actual robot (2).