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

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

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

Abstract—In this paper we propose a hierarchical approach to solve sensor planning for global localization of a mobile robot. The higher layer uses a Bayesian network which represents the contextual relation between the geometrical features of local environment, the robot sensing actions and the global localization beliefs. In the higher layer, the system allows sensor planning by taking into account the trade-off between global localization belief and the sensing cost to generate an optimal sensing action sequence. Through the optimal sequence of sensing action, the lower layer uses particle filter to efficiently and precisely localize the mobile robot. The simulation experiments show effectiveness of the proposed approach.

I. Introduction

The navigation and localization of a mobile robot is a traditional, fascinating research area. Many studies to date have focused on how to obtain an accurate map, and then how to match the sensing information of the robot to the map for localization. However, in robot navigation, the robot cannot always determine its unique pose only by local sensing information since the sensor is prone to errors and a slight change of the robot's pose deteriorates the sensing results. Therefore, many probabilistic approaches have been proposed to cope with uncertainties and to improve the robustness of the localization [1] [2].

The localization of a mobile robot involves estimating the pose of the robot in an environment. There are two classes of localization problem, position tracking and global localization. In the position tracking, a robot knows its initial position [3] and only needs to reduce uncertainty in the odometry reading. If the initial position is not known or the robot is kidnapped to somewhere, the problem is global localization, i.e., the mobile robot has to estimate its global position through a sequence of sensing actions[4]. In this paper, we deal with the global localization problem.

In resent years, the global localization is an active research area, however, the major localization approach is passive [4], the sensing actions for localization can not be controlled by taking into account the trade-off between global localization belief and the sensing cost. For example, as shown in Fig.1, mobile robot has two possible locations indicated by a white circle and a black circle. If the mobile robot takes the sensing action sequence shown as the solid line, the localization efficiency will be better than the other sensing action of the dotted line. To obtain the efficient sensing action for localization, the mobile robot have to perform sensor planning based on global environment information. However, less progress has been made in active sensing for the mobile robot localization, the passive localization approaches can not generate a sensing strategy for localization based on prediction of the further sensing information.

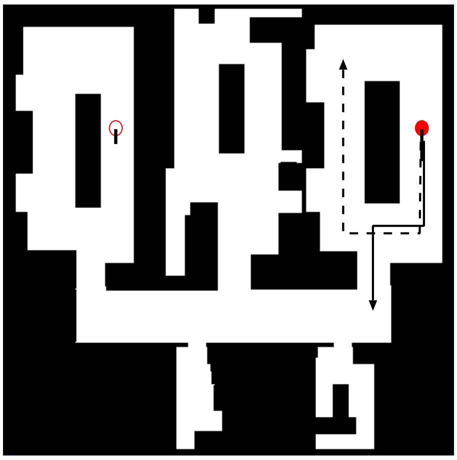

Fig. 1. An example of sensor planning for mobile robot localization

II. Previous works

A. Probabilistic Approaches for Global Localization

Some Bayesian approaches to mobile robot navigation and localization have been proposed. Thrun et al.[1][4] proposed localization of a mobile robot using a particle filter. The particle filter resamples and updates the belief of localization, and estimates the maximal posterior probability density for the localization. However, since the robot moves randomly without sensor planning, the system does not always assure efficient convergence of the particles if the environment has locally similar sensing patterns.

Some works used Bayesian networks for modeling the mobile robot navigation environment. Asoh et al.[6]

developed a system to combine local information for localization using a Bayesian network. They employed speech interaction to obtain evidences in the Bayesian network to decrease uncertainty of the localization. However, the system does not actively plan how the mobile robot should gather sensor information, and the Bayesian network structure is manually designed. Basye et al.[7] built planning and control systems that integrated sensor fusion, prediction, and sequential decision-making using a temporal belief network (dynamic Bayesian network). However, this POMDP (partially observable Markov decision process) approach cannot represent complex causal relations between the robot's actions and the environment's information. Tomatis et al.[14] also proposed an approach to integrate the Kalman filter based metric map and POMDP based topological map.

B. Sensor Planning and Active Localization of Mobile Robot

Sensor planning of mobile robots is also an active research area. Kristensen[10] proposed a sensor planning method based on a top-down decision tree algorithm. However, the utility-based Bayesian decision tree is too simple to evaluate the causal relations between local sensor information and the global localization. Miura et al.[11] defined an utility value based on sensing cost and sensing uncertainty to plan the sensor action for navigation, however, the system used a simple Bayesian rule.

A multiple hypothesis tracking approach has been used in active global localization[9]. However, the approach was based on Kalman filtering which assumes a model of linear dynamics with Gaussian noise.

Fox et al.[8] proposed an Active Markov Localization to improve the efficiency of localization. The optimal sensing action sequence was 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. However, in this system, the increase of global localization probability entropy was evaluated only by calculating the difference of the entropy between the current belief of the robot pose and the belief of all free positions in the environment. Therefor, the combined environment information which is sensed from the various place of the environment is not taking into account for the mobile robot localization. Zhou et al. [12], [13] proposed two approaches about sensor planning for localizaiton using Bayesian network. However, the proposed two systems can not solve complex environment. The system of this paper which generalizes the proposed sensor planning algorithms, and let it to adapt more general environment. In addition, the system is different from the proposed two systems, it can solve the localization not only efficiently but also accurately.

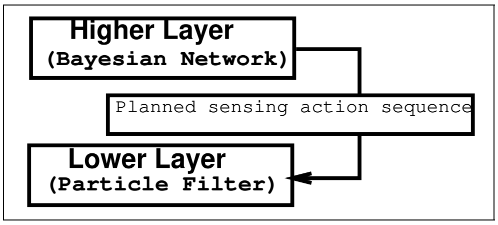

Fig. 2. A hierarchical architecture for the mobile robot localization

III. System Structure

As shown in Fig.2, the mobile robot estimates its pose by hierarchical integration of the two layer. Based on the geometrical features of the environment, the environment has been decomposed into some cells (see Fig.8), and clustering of the cell's geometrical features has been performed to allow to recognize the class of cells. In the hierarchical architecture, the lower layer uses particle filter to accurately localize the mobile robot. However, the random sensing action always affects the efficiency of the localization. To improve the efficiency of localization, the proposed integration system has a higher layer using Bayesian network[5] which represents the contextual relation between the geometrical features of the cells, the robot sensing actions, and the global localization beliefs. By taking into account the trade-off between global localization belief and the sensing cost, the higher layer allows to generate an optimal sensing action sequence for the particle filter localization.

IV. Cell Decomposition of Environment

The higher layer of the system has been used to represent the causal relation between the geometrical features of local environment, the robot sensing actions, and global localization belief. To obtain the geometrical features of local environment which a robot might come across in the navigation, the system decomposes the environment into cells. As shown in Fig. 8, exact cell decomposition[15] has been performed twice (from left to right and up to down).

The system assumes use of a mobile robot like Pioneer 3 (Fig. 3)with a laser. The laser range data is simulated, which covers 360 degrees view around each cell. Since the system is a cell based representation, the robot has to recognize which cell it in. To overcome this problem, the system performs clustering of the cell's geometrical features. The geometrical features are classified into 20 classes using K-means [19]. The classification result is shown in Fig. 4.

V. Lower Layer for Accurate Localization

In recent years, particle filter [17] has been widely applied in computer vision [18], mobile robot localization [16] applications. It estimates the posterior over unobservable state variable from sensor measurements. In mobile robot localization application, the state

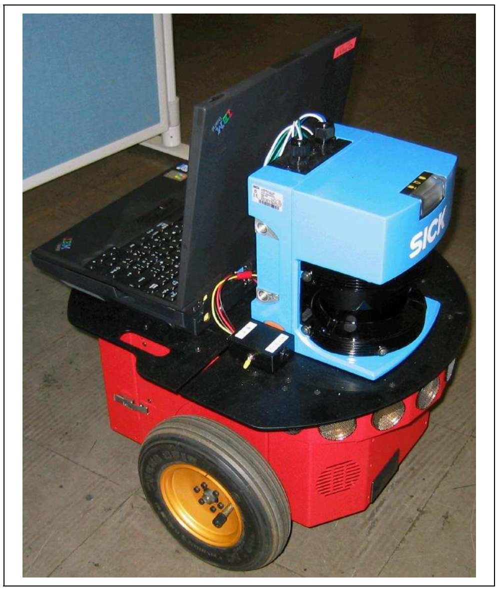

Fig. 3. Pioneer3 mobile robot of this system

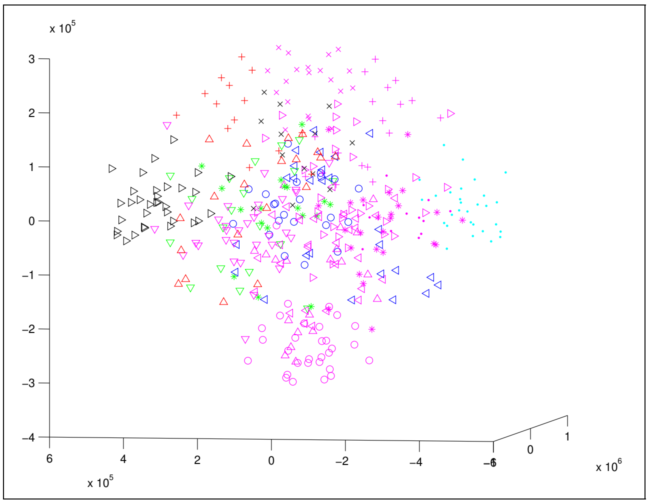

Fig. 4. Sensor data of every cell which was classified into 20 classes using K-means. (Classified data was projected in first eigenvectors and show different clusters with different symbols)

refers to the pose (location and orientation) of the mobile robot relative to its environment. The posterior probability of the mobile robot pose is simulated in particles distribution. A particle filter for mobile robot localization is shown as Fig. 5. The pose of every particle simulates the possible pose of the mobile robot. Since we want to perform a global localization, initially, the particles uniformly distributed in free space of environment (Fig. 5(a)). After some sequential sensing actions, the particles converge into some clusters (Fig. 5(b), (c)). When the mobile robot turns left at an intersection, the particles are converged into one cluster (Fig. 5(d)).

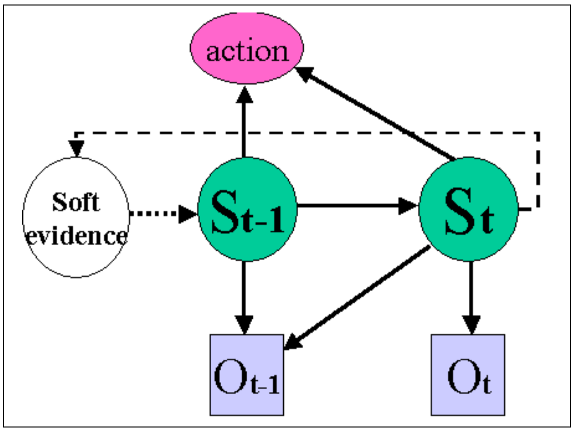

Fig. 6. The Bayesian network used in the system.

VI. Higher Layer Control

A. Bayesian network design

The higher layer employs a Bayesian network for the sensor planning, the Bayesian network has been designed as shown as Fig. 6. The node S_{t-1} and S_t denote the states at time $t-1$ and t, respectively. The state means the pose of the mobile robot at every cell. Since the higher layer is used to perform rough localization, we only define the robot has four directions (east, west, north and south), and the mobile robot only has four states in every cell. The node *Action* denotes the robot motion while moving from one cell to neighbor cell. The node O_{t-1} and O_t denote the observation (sensor measurement) which obtained at state S_{t-1} and S_t, respectively. The node *Soft evidence* denotes the probability which is sent from previous state (S_{t-2}).

The node S_{t-1} includes 412 probabilistic variables, which represent the robot's poses, while it is in every cell with four directions (up, down, left and right). The node S_t is same as the node S_{t-1} has 412 probabilistic variables, and with one error state (if it is a wall). The node *Action* includes 4 probabilistic variables, go forward, turn left, turn right, and go back. The node O_{t-1} and O_t have 20 probabilistic variables which defined at Section IV. The node *Soft evidence* is same as the node S_{t-1}, includes 412 probabilistic variables. However, since it is a soft evidence node, the initial value of this node allows to one set of probability value. In this system, we set the initial value of *Soft evidence* node by the weight of every particle clusters which classified in section IV.

B. Soft evidence of Bayesian Network

In the section IV, the system has classified the geometrical features of the cells using K-mean. However, since the geometrical features are similarity, the boundaries of the classes are too ambiguous to classifies the every class. If we use the results of the classification to be the observation of Bayesian network directly, the inference will be too sensitive to estimate the pose of

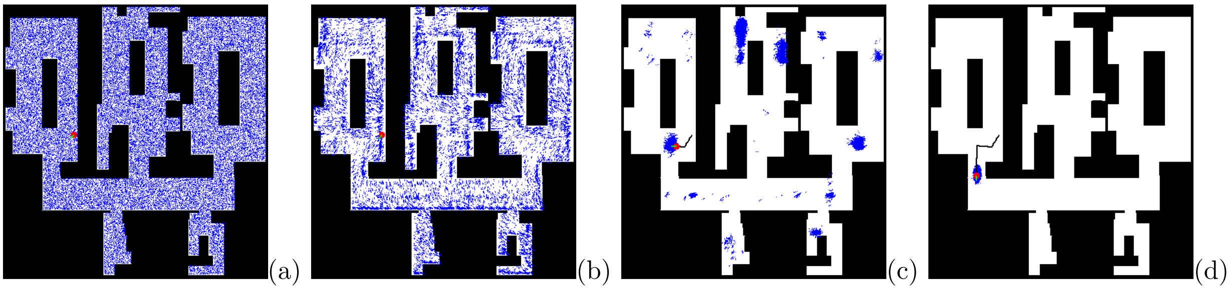

Fig. 5. A example of mobile robot localization using particle filter by random actions.

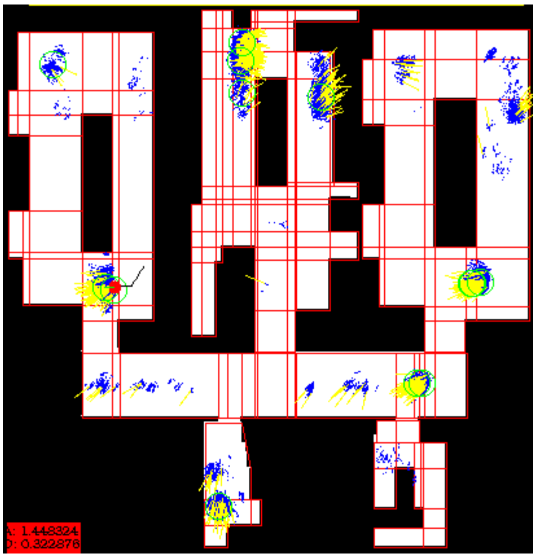

Fig. 7. Cell based particle clustering.

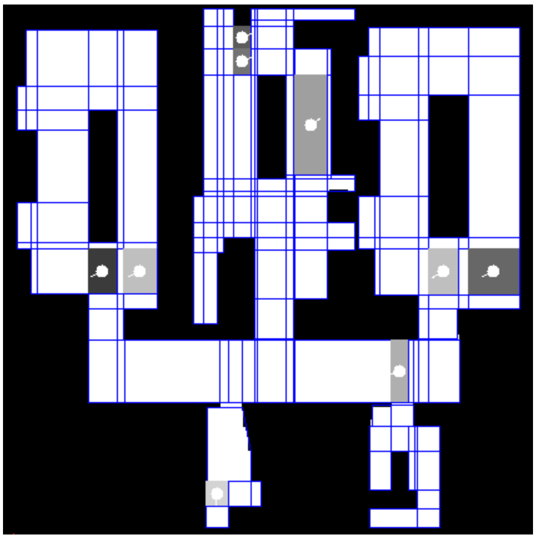

Fig. 8. The initial status of higher layer control.

the robot. Fortunately, the Bayesian network allows the soft evidence [5] to be used for inference. To overcome the uncertainty of the boundary, the node O_{t-1} and O_t have used the soft evidences. For example, assuming the node O_{t-1} has an observation which is the robot sensed at cell i with orientation m. The sensor measurement is named P_{im}, and is projected into class n. The system calculates the distances between the P_{im} and center point of every class, and sorts all of distances, then normalizes the distances to be probabilities. The probabilities are the soft evidence of the observation node O_{t-1}.

VII. Simulation Experiments

A. Particle Clustering

The higher layer control can be ran when the particles converge into some clusters. The system checks the averages and variances of the weight of particles in every resampling cycle, two thresholds of the average and the variance are use to determine timing to perform particle clustering and sensor planning.

As shown in the Fig. 7, the system performs particle clustering at this state, the average and variance of weight of particles are calculated and illustrated at down-left of Fig. 7. To reduce the computational cost, the system performs the particle clustering base on the cells. The system calculates the center of the particles in every cell, and computes the distances between the particles of each cell and the center. Then the system evaluates the distances variance of each cell. Based on a threshold of distance variance, we can find clusters of particles.

In Fig. 7, the circles are the location of the particle clusters which are found by the above rules, and bars on the particles are the orientations of the particles. The higher layer will be ran at this time. Since the possible locations of the robot will be focused on the cells which have particle clusters, and probabilities of every location are evaluated by the weight of particles, the initial status of higher layer is illustrated in Fig. 8. Some robots (white circle) are drawn in the centers of the cells which have clusters, the orientation of every robot is the average orientation of each particle cluster. The probabilities of the clusters are calculated by the weight of particles, and shown in the Fig. 8 in gray color with different intensity level. The darker (brighter) intensity level corresponds to larger (smaller) probability.

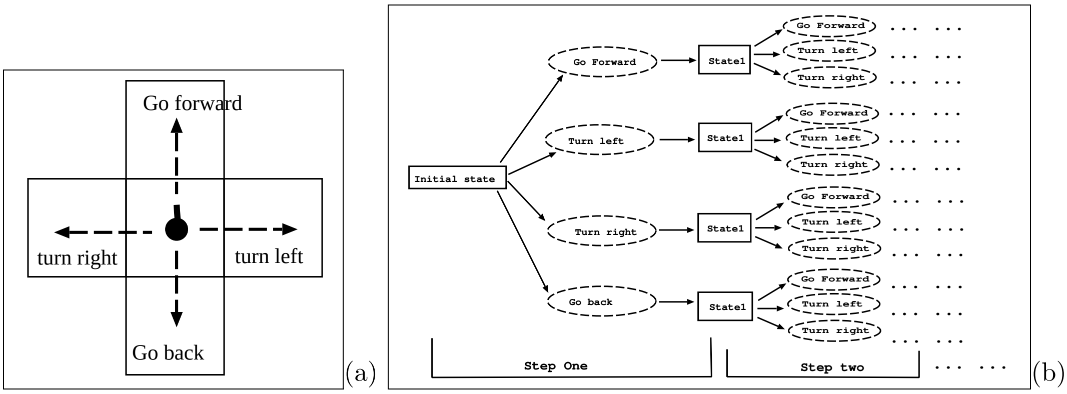

Fig. 9. (a) Four possible actions of the mobile robot in sensor planning for localization phase. (b) State transference during the sensor planning for mobile robot localization.

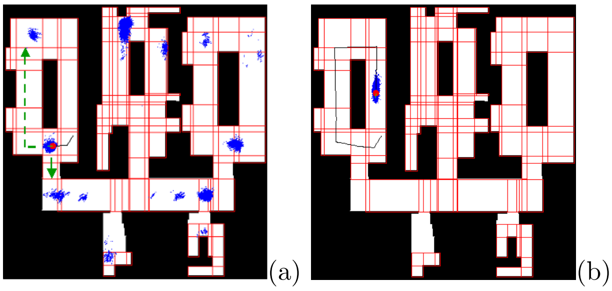

Fig. 10. (a) The mobile robot has two possible actions before the particles complete convergence. (b) If the mobile robot performs a forward action at the intersection, the particles needs a long sequence action for complete convergence.

B. Higher Layer Control for Sensor Planning

Since the particle filter just performs filtering, predicting and smoothing the sensing data, but cannot control the actions of the mobile robot. If we only implement the particle filter for the localization, and do not consider the action control, sometimes, the random sensing actions will influence the efficiency of particles convergence. For example, in Fig. 10 (a), the mobile robot faces an intersection before the particles become complete convergence. In Fig. 10 (b), if the mobile robot performs a forward action at the intersection, the robot should perform a long sequence action for complete particles convergence. In Fig. 5(d), the robot turn left at the intersection, the efficiency of localization is better than the sensing action of Fig. 10 (b). So the system aims at this problem to construct a higher layer to performs the sensor planning for the localization using the Bayesian network.

To reduce the planning computational cost, we assume there are four possible actions (e.g. Go forward, Turn left, Turn right, and Go back) of the robot during the sensor planning (see Fig. 9(a)). After particle clustering phase, the system find some particle clusters, and the sensor planning will be ran at this time. Since the system performs the sensor planning based on cells, the possible action sequences and state transference of every cluster during the sensor planning for mobile robot localization are shown in Fig. 9(b). In the Fig. 9(b), Initial state denotes the one pose of white circle (see Fig. 8), which is found during particle clustering phase. State1 means the pose of the robot, when the circle finished a possible action, we name this step in Step one. The circles will move from Initial state the neighbor cells after the Step one. If the circle performs two step action, the action sequence and state combination are illustrated in Step two of Fig. 9(b).

$$\begin{aligned} Utility &= (1-\theta) * AveBelief + \theta * Cost \qquad (1)\\ &if(AveBelief <= threshold),\\ &\qquad\qquad \theta = 0\\ else, &\qquad\qquad 0 < \theta < 1 \end{aligned}$$

The system assumes the white circles perform all of possible action sequences from its initial pose, and estimates the probabilities of the circles using Bayesian network after actions finished. We denote the average probability of all of the circles to be AveBelief after an action sequence finished, and Cost is defined by distance which the circle moved. Based on the Eq. 2, taking into account the balance between AveBelief and cost, the system calculates the Utility, and selects the optimal action sequence for the mobile robot localization using the Utility. In the Eq. 2, if the AveBelief is not larger than an threshold, the Utility will be AveBelief, in other cases, the Utility will be a value which is integrated by the AveBelief and cost.

For example, in Fig. 8, there are nine white circles which correspond to nine possible robot's poses, and each possible pose corresponds to an initial status (see Fig. 9(b)). Initially, the system assumes every circle performs step one. As shown in Fig. 9, in this step, the robot has four possible actions. The average probabilities of nine circles (AveBelief) is shown in Table. I when after the robot performed an action. We can find if the robot performs go back, the AveBelief will be largest. If the largest AveBelief is not larger than an threshold, the system has to perform the step two

Action	Average Probability (AveBelief)
Go forward	0.8010
Turn left	0.6754
Turn right	0.8630
Go back	0.7652

TABLE I

The average probabilities of all of particle clusters which 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

The average probabilities of all of particle clusters which inferred by the BN, when the system performed two actions.

or more, until find an AveBelief which is larger than the threshold. In this paper, we define the threshold is 0.9. If the system finds AveBeliefs which are larger than the threshold, it calculates the Utility by the Eq. 2. An optimal action sequence must be held a largest Utility. Since the largest AveBelief is not larger than the threshold (0.9), the system performed step two. In Table. II, the system calculates the AveBeliefs of nine circles when they finished “step one and step two”(see Fig. 9). We can find the AveBelief is largest when the nine circles performed turn left and go forward, and the largest AveBelief (0.9460) is larger than the threshold (0.9). In the Table II, we only find one AveBelief is larger than the threshold. So the action sequence, “turn left then go forward”, will be the optimal action sequence for localization. The simulations in Fig. 10 and Fig. 5 also validate the sensor planning results is effective for localization.

VIII. Conclusion

This paper presented the system which takes a hierarchical approach to solve a mobile robot sensor planning for global localization. The higher layer represents the contextual relation between the geometrical features of local environment, the robot sensing actions and the global localization beliefs using Bayesian network. The system allows sensor planning by taking into account the trade-off between global localization belief and the sensing cost to generate an optimal sensing action sequence. The lower layer was implemented in a particle filter which allows the robot obtain precise localization. Through the integration of two layers, the robot can precisely and efficiently localize itself using the optimal sequence of sensing action.

The simulation experiments show effectiveness of the proposed approach. Real robot experiments to validate the system in real environments are left for future work.

References

[1] S. Thrun, “Probabilistic Algorithms in Robotics,” AI Magazine, 21(4):93-109, 2000.

[2] S.Thrun, “Bayesian Landmark Learning for Mobile Robot Localization,” Machine Learning 33, pp.41-76, 1998.

[3] N. Roy, W. Burgard, D. Fox, and S. Thrun, “Coastal navigation: Robot navigation under uncertainty in dynamic environments,” Proc. of IEEE Int. Conf. on Robotics and Automation, 1999.

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

[5] F. V. Jensen, “Bayesian Networks and Decision Graphs,” Springer, 2001.

[6] H. Asoh, Y. Motomura, I. Hara, S. Akaho, S. Hayamizu, and T. Matsui, “Combining Probabilistic Map and Dialog for Robust Life-long Office Navigation,” Proc. of Int. Conf. on Intelligent Robots and Systems (IROS’96), pp.880-885, 1996.

[7] 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.

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

[9] P. Jensfelt and S. Kristensen, “Active Global Localization for a Mobile Robot Using Multiple Hypothesis Tracking,” Proc. of Workshop on Reasoning with Uncertainty in Robot Navigation (by IJCAI’99), pp.13-22, 1999.

[10] S. Kristensen, “Sensor Planning with Bayesian Decision Analysis,” PhD thesis, Faculty of Technology and Science, Aalborg University, Aalborg, Denmark, 1996.

[11] 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.

[12] 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.

[13] H. Zhou and S. Sakane, “Learning Bayesian Network Structure from Environment and Sensor Planning for Mobile Robot Localization,” Proc. 2003 IEEE Conference on Multisensor Fusion and Integration for Intelligent System (MFI2003).

[14] 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 International Conference on Robotics & Automation, pp.1111-1116, 2001.

[15] J.C. Latombe, “Robot Motion Planning,” Kluwer Academic Publishers, 1996.

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

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

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

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