Sensor Planning for Mobile Robot Localization Based on Probabilistic Inference Using Bayesian Network

Hongjun Zhou Shigeyuki Sakane

Department of Industrial and Systems Engineering, Chuo University
1-13-27 Kasuga, Bunkyo-ku, Tokyo, 112-8551 JAPAN
E-mail: {zhou, sakane}@indsys.chuo-u.ac.jp

Abstract

We propose a new method of sensor planning for mobile robot localization using Bayesian network inference. Since we can model causal relations between situations of the robot's behavior and sensing events as nodes of a Bayesian network, we can use the inference of the network for dealing with uncertainty in sensor planning and thus derive appropriate sensing actions.

In this system we employ a multi-layered-behavior architecture for navigation and localization. This architecture effectively combines mapping of local sensor information and the inference via a Bayesian network for sensor planning. The mobile robot recognizes the local sensor patterns for localization and navigation using a learned regression function. Since the environment may change during the navigation and the sensor capability has limitations in the real world, the mobile robot actively gathers sensor information to construct and reconstruct a Bayesian network, then derives an appropriate sensing action which maximizes a utility function based on inference of the reconstructed network. The utility function takes into account belief of the localization and the sensing cost. We have conducted experiments to validate the sensor planning system using a mobile robot simulator.

1 Introduction

In a complex environment, how to localize a mobile robot on its way and to navigate autonomously towards a goal is a very fascinating problem to many researchers. Until now, mobile robots have navigated mainly using a global map constructed from sensor information. A mobile robot localizes itself based on matching local or global sensor information to the map then decides its behavior subsequently based on the matching results. However, in the real world, since many uncertainty factors adversely affect navigation of robots, it is difficult to use map-based methods. Therefore, we need an approach to cope with such uncertainty factors. In this paper, we take Bayesian network approach. The field of Bayesian networks and graphical models has grown in recent years and much progress has been made in the theoretical analysis as well as its applications to real problems [1][2][3]. However, less progress has been made in its application to sensor planning of robots. Bayesian networks allow us to represent causal relations among situations of robot sensing and the obtained data or evidences in a natural manner and to quantitatively analyze beliefs about the situations. Consequently, the approach provides a sound basis for dealing with uncertainty in sensor planning.

2 Previous Studies

Tani [4] developed a mobile robot system which focuses on local sensor information and directly maps the information to motor command space. Although the method allows the robot to navigate along a previously determined path, it has no skill for recognizing and distinguishing two (or more) sets of patterns that hold the same sensor information. Thrun [5] proposed localization of a mobile robot using Bayesian analysis of the probabilistic belief. Asoh et al. [6] developed a mobile robot system which navigates using a prior-designed Bayesian network. The system reduces uncertainty in the localization by conversation with a human using a speech recognition subsystem. However, these methods have not implemented sensor planning mechanisms to efficiently gather information of the environment. As for sensor planning, Miura et al. [7] proposed a method for vision-motion planning of a mobile robot under vision uncertainty and limited computational resource though they did not use Bayesian networks. Rimey et al. [8] used Bayesian networks to recognize table setting, and plan the camera's movement based on maximum expected utility decision rules.

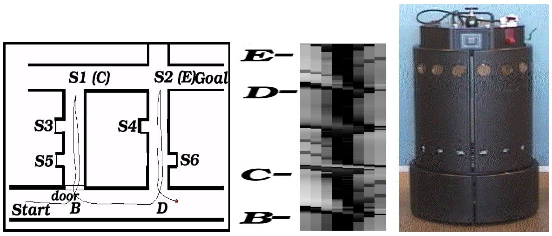

Figure 1: The trajectory and its associated sensor data flow of a mobile robot

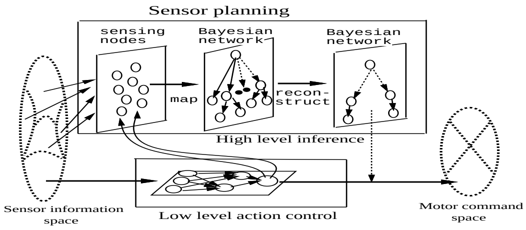

Figure 2: Multi-layered-behavior architecture for sensor planning

In this paper we propose a sensor planning system which avoids error of global measurement, maps limited sensor information to motor commands, and increases the belief of localization based on Bayesian network inference.

3 Task Setting

We would like to describe our main task setting of this paper. As shown in Fig 1, a mobile robot learns the local sensor information (**C, E, D** or **B**), so that it may navigate from a "*start*" point to a crossing **D** and arrive at a *goal* **E** while door (at a crossing **B**) is closed. However when door (at the crossing **B**) is open incidentally, the local sensing information at **B** and **D** will be identical. Therefore the mobile robot can not distinguish which crossing is correct to navigate itself to the *goal* **E** only based on the previously learned model of the local sensing. That is, if there are some crossings with the same local sensing information in a navigation path, how to recognize which is "**true D**", i.e., which crossing could guide the robot to the goal **E**? To solve this problem, we developed a system to infer the belief of the **D**.

4 Basic concept of the system

To cope with above problems, we propose an architecture of *multi-layered-behavior* to plan the sensor's action to localize a mobile robot. This architecture involves *low level action control* (**LLAC**) and *high level inference* (**HLI**) capabilities. Figure 2 shows the architecture of our system. The *low level action control* (**LLAC**) identifies local sensor patterns of a limited sensor information space and directly maps these patterns to the motor command space. However, since the sensor capability is limited in the real-world and the patterns may change depending on the environment, it is difficult to localize and navigate the robot correctly to the goal only by this control level. Therefore, the system employs high level inference **(HLI)** to estimate the robot's position based on causal relations of local sensor information nodes. Identified local sensor patterns are added into a group of sensing nodes, then the system constructs/reconstructs these sensing nodes into a Bayesian network.

Our method has the following key features:

- Our localization method differs from traditional methods in that we not only focus on local sensor information, but also perform sensor planning which takes into account causal relations of the local sensor information for the localization.

- In order to decrease uncertainty in localization caused by faulty sensor information, we attempt to actively gather information of the environment and to map these information nodes into a Bayesian network, then use them for probabilistic reasoning to correctly localize the robot.

- Initially the system does not have a complete prior-built Bayesian network. A robot gathers sensor information, creates nodes, and obtains the prior probabilities (conditional probabilities) automatically. Then the system compares the integrated utility of every sensing node in the Bayesian network. Finally, a configuration of the Bayesian network for efficient localization is obtained.

5 The Prototype System

We use a mobile robot (B14, Real World Interface) shown in Fig. 1. The mobile robot is equipped with a Pentium CPU, 16 sonar sensors, a color CCD camera, and other sensors. A desktop PC running Linux is used for the server of the Bayesian network inference **(HLI)**, and it transfers the calculated belief to the robot via a socket stream.

For the software in our prototype system, we implemented the Bayesian networks in C++ using the source code of Ref.[9]. The system calls the B14's software library (Bee Soft) to drive the mobile robot. We implemented a three-layered Back Propagation Neural Network (BPNN) to navigate the mobile robot by the *low level action control* (**LLAC**).

6 Implementation of LLAC

The mobile robot is basically driven by a potential method. Figure 1 (left) shows a trajectory of the robot in a workspace. Fig.1 (right) shows a time sequence of the corresponding sonar sensor data as a gray level image. The vertical axis represents the time and the eight pixels along the horizontal slice represent a set of sonar sensor data in which a darker (brighter) intensity level corresponds to a larger (smaller) sonar distance value, respectively. On a road with no crossings, a horizontal slice of the image has only one darkest point, the system searches for the maximum value in every glance of the sonar sensors, and tracks the angular direction of the largest distance value.

When a mobile robot comes to a crossing, the horizontal slice of the image will have two or more darkest points. We evaluate the distribution of every temporally sliced data to search the crossing. The robot's action is determined by low level action control at the crossing. We employ a three-layered Back Propagation Neural Network **(BPNN)** to model the filter function $\boldsymbol{\pi}$ and map the 8-direction sonar data of the front of the mobile robot into sensor feature space or action commands (translation and rotation) space at crossings (like $\perp$, $+$,$\top$) of the path.

7 Implementation of HLI

7.1 Active sensing for localization using Bayesian network inference

As shown in Fig. 1, the belief of position **D** at the crossings (**B** or **D**) can be obtained as the following formalization.

$$\boldsymbol{Bel(D)} = \boldsymbol{P}\ (\boldsymbol{D}\ |f) \tag{1}$$

where $\boldsymbol{Bel(D)}$ $\longrightarrow$ *the belief of position D at the crossings* **B** *or* **D**
$\boldsymbol{P}\ (\boldsymbol{D}\ |f)$ $\longrightarrow$ *the posterior probability supported by sensor feature f only.*

Since the local sensor information of **B** is identical with that of **D**, the mobile robot can not localize itself only by the local sensing pattern only by Eq.(1), while it runs from the "*Start*" point to the crossing **D** directly.

To overcome the difficulty and search the "**true D**", the mobile robot performs active sensing as shown by the solid line trajectory in Fig.1. This time we can obtain the belief of **D** at the crossings (**B** or **D**) from the following function:

$$\boldsymbol{Bel(D)} = \boldsymbol{P}\ (\boldsymbol{D}|f, s_1, ..., s_n) \tag{2}$$

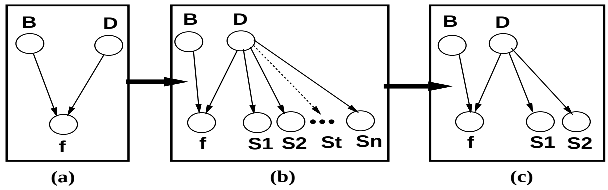

Figure 3: Construction and reconstruction of the Bayesian network for sensor planning

Note that $s_1, ..., s_n$ are the sensing nodes generated by active sensing. These sensing nodes are obtained from various sensors (for instance, range sensor, vision sensor, acoustic sensor, etc.) and difference in the position of feature along the path. We construct the Bayesian network as shown in Fig.3(b) to calculate the $\boldsymbol{Bel(D)}$ at the crossings (**B** or **D**). Sensing nodes propagate the evidences backward to the node **D**. $\boldsymbol{Bel(D)}$ of the crossing **D** is increased while $\boldsymbol{Bel(D)}$ of the crossing **B** is decreased.

7.2 Reconstruction of the Bayesian network for sensor planning

We can obtain the $\boldsymbol{Bel(D)}$ from Eq. (2), however we must note that we have not considered the sensing cost. By taking into account the balance between belief and the sensing cost, we propose an integrated utility function and a reconstruction algorithm of the Bayesian network for sensor planning.

7.2.1 Reconstruction Algorithm

We define an integrated utility (**IU**) function (Eq. 3) which we can adjust priority of the two criteria (belief and sensing cost). Depending on the balance between sensing cost and belief, we obtain different planning results of robot behavior for localization.

$$\mathbf{IU_i} = \mathbf{t} \times \mathbf{\Delta Bel_i} + (\mathbf{1} - \mathbf{t}) \times (\mathbf{1} - \frac{\mathbf{Cost_i}}{\sum_\mathbf{i} \mathbf{Cost_i}}) \tag{3}$$

where $$\mathbf{\Delta Bel_i} = |\mathbf{0.5} - \mathbf{Bel_i}| \tag{4}$$

$\mathbf{IU_i}$ denotes the integrated utility (**IU**) value of sensing node i, $\mathbf{Cost_i}$ denotes the sensing cost of sensing node i, $\mathbf{Bel_i}$ denotes the Bayesian network's belief while the mobile robot just obtains the evidence of active sensing i only, and $\mathbf{\Delta Bel_i}$ represents certainty of the belief of sensing node i which contributes to the Bayesian network. The maximum value of $\mathbf{\Delta Bel_i}$ is 0.5 when $\mathbf{Bel_i} = 0$ *or* 1 and the minimum is 0 when $\mathbf{Bel_i} = 0.5$. **IU** value will increase along with increasing belief and decrease along with increasing sensing

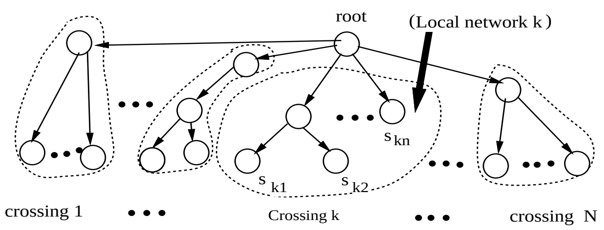

Figure 4: Local network of Bayesian network. Every local network is constructed by each crossing's active sensing nodes. Evidence of these sensing nodes will be propagated to root node, and using these posterior probability to decide if this crossing can guide the mobile robot to the goal.

cost. We use a parameter t ($0 \leq t \leq 1$) to balance sensing cost and belief.

Before presenting of our new *reconstruction algorithm*, we would like to describe a concept of *"local Bayesian network"*. Since the mobile robot must infer which crossing could guide itself to the goal based on the beliefs of sensing nodes (or *sensing node sets*) of the crossing, we associate the sensing nodes of each crossing to a *"local network"*. The mobile robot estimates the probability of every crossing using this *"local network"*. The reconstruction will be performed in every *"local network"*. The Figure 4 illustrates the concept.

The reconstruction algorithm has two steps, **STEP (1)** completes the refining process of each local network. In other words, Bayesian network will be reconstructed from every *local network* (active sensing nodes of every crossing) using **IU** function. **STEP (2)** combines local networks to the global Bayesian network.

Reconstruction Algorithm:

1. **Initialization of Bayesian network :**
   The mobile robot performs active sensing at every crossing, and constructs an original Bayesian network as Figure 4 using all of these sensing nodes.

2. **STEP (1)**: Refine the local network.
   For example, the system refine the local network **k** (the sensing nodes of a crossing **k**) of Fig.4 by the following algorithm:

   - Check the ΔBel_i of every terminal sensing node, remove the node which satisfies $\Delta Bel_i < \Theta$.
     (Θ ($0 < \Theta < 0.5$) is a threshold of $\Delta Bel_i < \Theta$. When $\Delta Bel_i < \Theta$, we regard the sensing node has no capability to localize the mobile robot.)
   - **IF** the number of survived nodes ($\Delta Bel_i > \Theta$) isn't zero,
     THEN sort the survived sensing nodes according to their **IU** values,
     $\mathbf{IU_{ki}} = max_{\Omega_k}\ \{IU\}$, ($\Omega_k$ denotes the sensing nodes group of crossing **k**.)
     Save this sensing node that has $\mathbf{IU_{ki}}$, and remove the other nodes.
   - **ELSE execute "combining process"** to combine the sensing nodes to improve belief until the sensing node set has enough ΔBel to distinguish the other crossings.

3. **STEP (2): Combine all of the local networks to construct the global Bayesian network :**

   (a) Refine the every local network (every crossing) based on **STEP (1)** algorithm.

   (b) Combine the local networks to reconstruct a new global Bayesian network.

   (c) Finally, compare the terminal nodes (or terminal sensing node sets combined by **"combining process"**), if they have exclusive relation,[1] then remove one side, and save the others.

4. **Combining Process of local network :**

   (a) Generate all combinations of sensing nodes in a local network,

   (b) Calculate the **IU** value of the combined sensing node sets which has $\Delta Bel_{(set)} > \Theta$, then sort these node sets based on **IU** value.

   (c) Leave the sensing node set j, which has $\mathbf{IU_{(set\ j)}} = max\ \{IU_{set}\}$, and remove the other node sets.

8 Experiments

We conducted experiments to validate the effectiveness of our system using a mobile robot simulator.

8.1 Assumptions of experiments

To simplify the calculation, our experiments have the following assumptions:

1. The parents-children relations are determined beforehand.

2. Prior probabilistic distribution (conditional probability table) of sensing nodes is acquired by measuring the frequencies of the events.

3. We omit the uncertainty of local moving distance of the mobile robot. The mobile robot may exactly estimate the local moving distance between each landmark, and compare the every landmark's local position and other sensing information to make **CPTs** (conditional probability tables) of the all of sensing nodes while it is moving in the workspace.

[1]We define the exclusive relation as $\overline{S_a} = S_b$. If robot obtained an evidence S_a, an evidence S_b will be ignored. For example the relation of S_5 and S_6 in Fig.1.

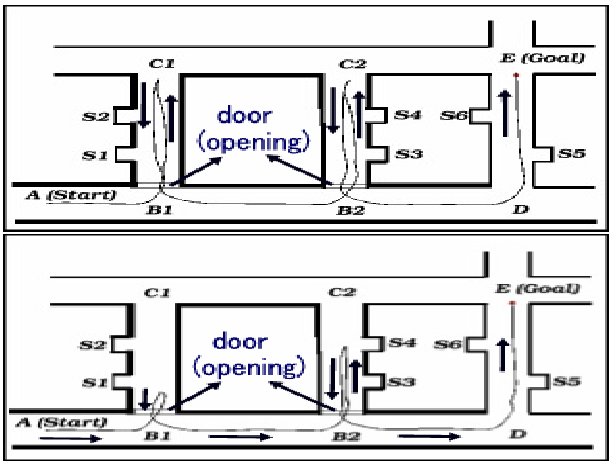

Figure 5: The mobile robot navigated following the solid line trajectory using inference of reconstructed Bayesain network. (up) $t = 1$; (down) $t = 0.33$.

8.2 Experiment 1

Firstly, we made an office environment (Figure 5) that has three crossings to validate our *reconstruction algorithm*. If the mobile robot has local sensing only, it can not recognize **D** which guides the robot to the goal **E**. The mobile robot will *turn left* at each crossing (**B1, B2** or **D**) to attempt to search the goal **E**. The search of each crossing will be finished while the mobile robot perceives the local environments is $\mathbf{C_1}$ or $\mathbf{C_2}$ ($\top$). Then the mobile robot turns back to gather the active sensing nodes by some tutorial commands given by human, and records all of sensing nodes (we can obtain sonar distance information only). To distinguish the **D** from **B1** (and **B2**) and construct the conditional probability table (CPT) of every sensing node, the mobile robot turns back at a goal **E** and records the sensing nodes. The original Bayesian network is constructed as Figure 6(a).

Consequently, we will reconstruct the original Bayesian network using the *reconstuction algorithm*. We can change the parameter t of **IU** function (Eq.3), the planed active sensing action will be different depending on the value of t. Figure 5 (up) shows the active sensing trajectory for localization of the mobile robot when the parameter $t = 1$. In this case, the mobile robot only focuses on the belief but does not consider sensing cost. Reconstruction process and every sensing nodes's **IU** value and belief is illustrated at Figure 6 (b) and (c). When $t = 0.33$, we obtain the results of **IU** value of sensing nodes as Figure 7 (c). After the reconstruction process based on the **IU**

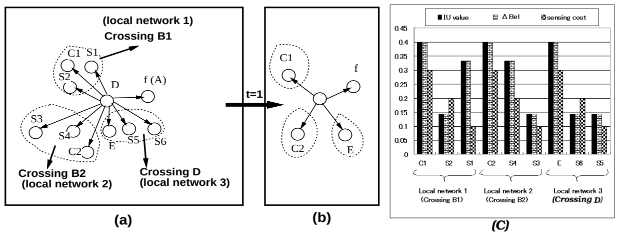

Figure 6: Reconstruction of the Bayesian network in the experiment 1 while $t = 1$.

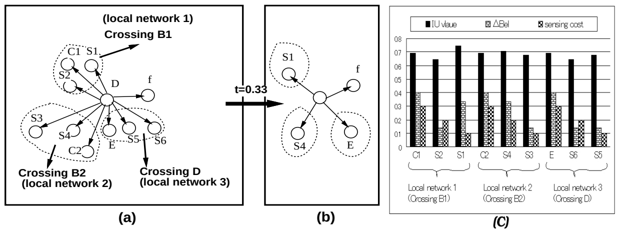

Figure 7: Reconstruction of the Bayesian network in the experiment 1 while $t = 0.33$.

value, we will acquire a new reconstructed Bayesian network (Figure 7 (b)). In this case, sensing action of the mobile robot will be planned as shown in Figure 5 (down).

As shown in the results, the proposed algorithm works successfully and the sensing behavior for localization varies depending on the parameter t.

8.3 Experiment 2

How should we construct and reconstruct a hierarchical Bayesian network which has hidden sensing nodes, states and multiple sensor information? Here, we build a more complex environment to describe the problem as shown in Figure 8. In the same way as the previous experiments, the mobile robot initially navigates by **LLAC**, and gathers information to make **CTP**s of the sensing nodes and an original Bayesian network (Figure 9 (a))

In Fig. 8, there are two hidden crossings ($\mathbf{F_2}$, $\mathbf{F_3}$) after passing crossings $\mathbf{B_2}$ and **D**, respectively. We assume some hidden states ($\mathbf{H_2}$ and $\mathbf{H_3}$) exist in the Bayesian network. $\mathbf{H_2}$ (or $\mathbf{H_3}$) denotes the sensing node sets of the hidden crossings $\mathbf{F_2}$ (or $\mathbf{F_3}$), we represent the causal relation between sensing nodes and hidden state as shown in Fig. 9 (a) ($\mathbf{C_3}$ and S_3's parent is $\mathbf{H_2}$; $\mathbf{C_4}$ and S_5's parent is $\mathbf{H_3}$). The sensed evidence will be propagated from terminal nodes to

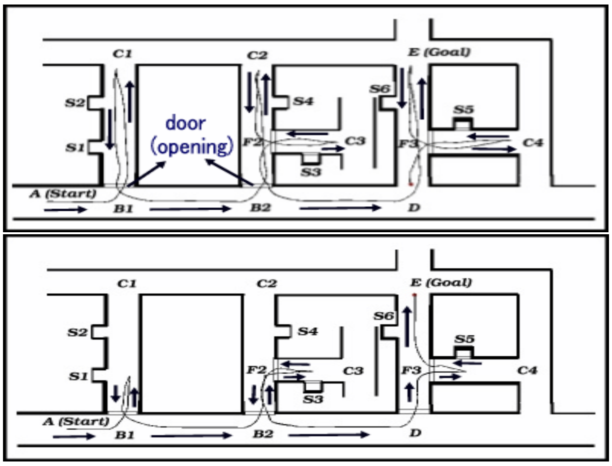

Figure 8: (up) The mobile robot navigates itself by **LLAC** and some tutorial commands to search the goal (**E**) and gathers the sensor information actively, then compares the difference of every crossing to construct the **CPT**s of every sensing node and original Bayesian network. (down) The mobile robot is navigated following the solid line trajectory using inference of reconstructed Bayesian network (t = *0.35*).

hidden state node ($\mathbf{H_2}$ or $\mathbf{H_3}$)), then **D**'s belief will be updated by propagation of hidden node's probability. When the t value (Fig. 9 (c)) of **IU** function is 0.35, the original Bayesian network (Fig. 9 (a)) is reconstructed as Fig. 9 (b). Fig. 8 (down) shows the planned path for localization of the mobile robot.

The results of the experiment show that our system effectively localize the mobile robot and allows to nagivate to the goal in the complex environments using the hierarchical Bayesian network.

9 Conclusions

We proposed a new method of sensor planning for mobile robot localization using Bayesian network inference. We can model causal relations between situations of a robot's behavior and sensing events as nodes of a Bayesian network and use the inference via the network for dealing with uncertainty in sensor planning. We employed a multi-layered-behavior architecture for navigation and localization. Since the environment may change during the navigation and sensor capability has limitations in the real world, the mobile robot actively gathers sensor information to construct and reconstruct a Bayesian network, then derives an appropriate sensing action which maximizes a utility function based on inference of the reconstructed network. The utility function takes into account the balance between belief of the localization and the sensing cost. The experimental results of the sensor planning for a mobile robot demonstrate the usefulness of the proposed system.

Our future plan includes the following: (1) validation of the system using a real robot, (2) attempt to learn structure of Bayesian network from **CPT**s (Conditional Probability Tables) of active sensing nodes. (3) validate our concepts using other applications.

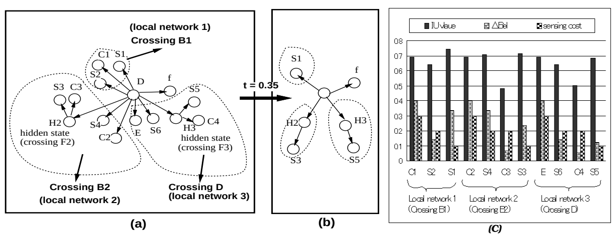

Figure 9: Reconstruction of the Bayesian network which has hidden states.

References

[1] J.Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[2] F.Jensen, *An Introduction to Bayesian Networks*, UCL Press, 1996.

[3] R.G.Cowell et al., *Probabilistic Networks and Expert Systems*, Springer-Verlag, 1999.

[4] J.Tani, "Model-based Learning for Mobile Robot Navigation from the Dynamic Systems Perspective," *IEEE Trans.on SMC*, Part B (Special Issue on Robot Learning), Vol.10, No.1, pp.153-159, 1977.

[5] S.Thrun, "Bayesian Landmark Learning for Mobile Robot Localization," *Machine Learning* 33, pp.41-76, 1998.

[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 the Int. Conf. on Intelligent Robots and Systems (IROS'96)*, pp.880-885, 1996.

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

[8] R.Rimey and C.Brown, "Control of Selective Perception using Bayes Nets and Decision Theory," *Int. Journal of Computer Vision*, Vol.12, pp.173-207, 1994.

[9] T.Dean et al., *Artificial Intelligence*, The Benjamin/Cummings, 1995.