

Sensor Planning for Mobile Robot Localization Using Bayesian Network Representation and Inference

Hongjun Zhou¹, Shigeyuki Sakane²

¹*Chuo University, Tokyo, Japan, zhou@indsys.chuo-u.ac.jp*

²*Chuo University, Tokyo, Japan, sakane@indsys.chuo-u.ac.jp*

Abstract

We propose a novel method to solve a kidnapped robot problem. A mobile robot plans its sensor actions to localize itself using Bayesian network inference. The system differs from traditional methods such as simple Bayesian decision or top-down action selection based on a decision tree. In contrast, we represent the contextual relation between the local sensing results and beliefs about the global localization using Bayesian networks. Inference of the Bayesian network allows us to classify ambiguous positions of the mobile robot when the local sensing evidences are obtained. By taking into account the trade-off between the global localization belief degree and local sensing cost, we define an integrated utility function to decide the local sensing range, and obtain an optimal sensing plan and an optimal Bayesian network structure based on this function. We have conducted simulation and real robot experiments to validate our planning concept.

1 Introduction

In a complex environment, how to localize a mobile robot as it moves and navigates autonomously towards a goal is a 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 by matching sensor information to the map then decides its behavior subsequently based on the matching results. However, since many uncertainty factors adversely affect the navigation of robots in the real world, it is often difficult to use map-based methods, particularly when only local information is available.

In this paper, we use a Bayesian network approach to cope with such uncertainty. The field of Bayesian networks and graphical models has grown in recent years and much progress has been made in the theoretical analysis [2] as well as its applications to real problems [5][6]. However, less progress has been made in its application to sensor planning of robots. Bayesian networks allow us to represent causal and contextual relations between sensing data

or evidences and beliefs about the global situation in a natural manner. Consequently, the approach provides a sound basis for dealing with uncertainty in sensor planning for mobile robot localization.

2 Previous Studies

Some Bayesian approaches to mobile robot navigation and localization have been proposed. Thrun et al.[3] 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 localization. However, the robot moves randomly rather than using sensor planning. Moreover, the system depends on distance information from the wheel encoder, and the environment information (to be prior knowledge or probability) is assumed to be static[4]. However, with wheelless robots it is very difficult to get accurate distance information using an odometer, and we have to consider how to cope with a dynamic environment. In this paper, we represent the causal and contextual relations of the sensing results and global localization in a Bayesian network[2], and propose a sensor planning approach based on Bayesian network inference to solve the dynamic environment in which we can not use accurate distance information.

Asoh et al.[6] developed a system to combine local information for localization via a previously designed Bayesian network, but the system could not actively plan how the mobile robot should gather sensing information. Basye et al.[9] built planning and control systems that integrated sensor fusion, prediction, and sequential decision-making using a temporal belief network. Kristensen[1] proposed a mobile robot sensor planning approach based on a top-down decision tree algorithm. Since the system has to compute the utility values of all possible sensor selections in every planning step, the planning process is very complex. Moreover, this system did not represent the contextual relations of local sensing results and the global localization. Miura et al.[7] also defined a utility value based on sensing cost and sensing un-

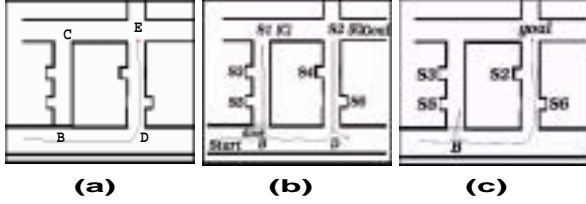


Figure 1: (a) When the door(B) is closed, the mobile robot navigates to the goal(E) only by the local sensing information. (b) While the door is open, the mobile robot may mistake B for D. (c) The mobile robot plans the sensing action to gather evidences to distinguish which is “true D” using Bayesian network inference.

certainly to plan the sensor action for mobile robot navigation, however, the system only applied a simple Bayesian rule. Outside of the mobile robot field, Bayesian networks have also been used in vision action planning for the next view problem[8].

In this paper we propose a sensor planning system for mobile robot localization. The system decreases the uncertainty of the global localization and classifies the ambiguous positions via the inference of the Bayesian network.

3 Uncertainty in localization

We describe the uncertainty in localization which we consider in this paper. As shown in Fig. 1(a), initially, a mobile robot recognizes the local sensor information (C, E, D or B), so that it may navigate from a “start” point to an intersection D and arrive at a goal E while the door (at an intersection B) is closed. However when the door (at the intersection B)(Fig 1(b)) is open, the local sensing information at B and D will be identical. Therefore, the mobile robot cannot distinguish which intersection is correct to navigate itself to the goal E based on only the previously learned model of the local sensing. That is, if there are some intersections with the same local sensing information in a navigation path, how can the system recognize which is “true D”, i.e., which intersection could guide the robot to the goal E? To solve this problem, we developed a system to infer the belief of the D(Fig 1(c)).

To resolve the above problems, we use a two-layered architecture to plan the sensor’s action to localize a mobile robot. The system consists of *low level action control (LLAC)* and *high level inference (HLI)* capabilities. 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 pat-

terns 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 contextual relations of local sensor information. We associate the local sensing information to a “node” of the Bayesian network, then the system constructs/reconstructs these sensing information nodes into a Bayesian network in which we can evaluate belief of the global localization.

4 Low level action control

In the simulation experiment, the mobile robot is basically driven by a potential method. When a mobile robot comes to an intersection (such as \perp , $+$, \top), the robot’s action is determined by a sensor mapping function. Here, we employ a three-layered back propagation neural network to map the sonar sensing information to a motor command space. The neural network is only used for recognition of intersections, but we also can use some distinct landmarks. For example, some color landmarks are employed in our real robot experiments.

The “*low level action control*” leads the mobile robot to search for a goal and gather the sensing event. The motor command space is defined by a human, in other words, the mobile robot’s search range is defined by a human. To simplify the problem, the system only allows the mobile robot to work in a partial environment.

5 High level inference

5.1 Active sensing for localization using Bayesian network inference

As shown in Fig. 1(b), the belief of position D at the intersections (B or D) can be obtained from the following function.

$$Bel(D) = P(D|f) \quad (1)$$

where $Bel(D) \rightarrow$ the belief of position D at the intersections B or D

$P(D|f) \rightarrow$ 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 reliably only by the local sensing pattern Eq.(1), while it travels from the “Start” point to the intersection D directly(Fig.1(a)).

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(b)(c). This time we can obtain the belief of D at the intersections (B or D) from the following function:

$$Bel(D) = P(D|f, s_1, \dots, s_n) \quad (2)$$

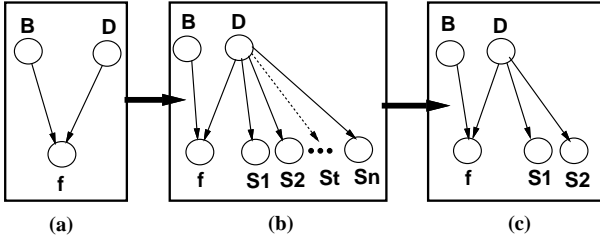


Figure 2: Construction and reconstruction of the Bayesian network for sensor planning

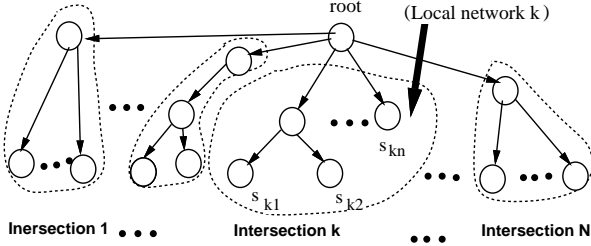


Figure 3: Local network of Bayesian network. Every local network is constructed by each intersection's active sensing nodes.

Note that s_1, \dots, 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 features along the path. We construct the Bayesian network as shown in Fig.2(b) to calculate the $Bel(D)$ at the intersections (**B** or **D**). Sensing nodes propagate the evidences backward to the node **D**. $Bel(D)$ of the intersection **D** is increased while $Bel(D)$ of the intersection **B** is decreased.

5.2 Local Bayesian network construction

We associate the sensing nodes of a corridor at each intersection to a “local network”(Fig. 3). If there are some sub-corridors in a certain corridor, some nodes named hidden state nodes (H) are defined in the “local network”. We associate a hidden state node with a sub-corridor. Sensing information of the sub-corridor corresponds to child nodes of the hidden state nodes (H) (see Fig.7 and Fig.8).

We assume the mobile robot can recognize a goal only by local sensing. When the mobile robot detects an intersection (for example, B_1, B_2, D of Fig.7) which **seems** to guide the mobile robot to a goal, the mobile robot begins to search the goal in its corridor. If there are some sub-corridors (or subsub-corridors), the mobile robot enters into every sub-corridor (or subsub-corridor) by following the wall to search the goal. The search action is finished when the mobile robot finds the goal (+) or local environment \top (for

example, C_1, C_2, C_3, C_4 of Fig.7). The mobile robot stops the search action, then turns back to the previous path, and records the sensing information of both sides of the sub-corridor (or subsub-corridor) until it comes back to the entry of the sub-corridor (or subsub-corridor), (for example, F_2, F_3 of Fig.7). In this searching and gathering information process, the mobile does not search all of the environment. We also can plan the robot to gather all of the environment information in some path planning methods, but this subject is not addressed in this paper.

5.3 Sensing nodes representation

The mobile robot can remember its orientation, actions (for example, “turn back” or “go forward”) and remember orders of sensing information in the sub-corridor (or subsub-corridor). Using this information the mobile robot can distinguish the sensing information (sensing nodes) in the same sub-corridor (or subsub-corridor). In the same way, the mobile robot can distinguish and label the sub-corridor (or subsub-corridor) of each corridor (for example, B_1 to C_1, B_2 to C_2, D to E of Fig.7). The mobile robot represents the position relation between the corridors and sub-corridors as a *position relation tree*, in which sub-corridors are child nodes of its upper layer corridor’s (or sub-corridor) node.

The mobile robot represents the sensing information (sensing node) as position p and sensing feature f of the environment. We label the intersections as counting number in the search order search. We also label the sensing nodes in *position relation tree*. We represent the position in a vector $(N_c, N_{sc}, N_{ssc}, \dots, order)$. $N_c, N_{sc}, N_{ssc}, \dots$ denote the corridor labels, for example, in Fig.7, we can represent the sensing node S_5 as vector $(D, F_3, order)$, the *order* corresponds to that of sensing information, when the mobile robot turns back to gather them in each corridor (or sub-corridor). We use the vectors and sensing feature (f) to categorize these sensing nodes of *local networks*.

5.4 Reconstruction of the Bayesian network for sensor planning

We can obtain $Bel(D)$ from Eq. (2), however 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.

We define an integrated utility (**IU**) function (Eq. 3) in which we can adjust the priorities 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.

$$\mathbf{IU}_i = t \times \Delta\mathbf{Bel}_i + (1 - t) \times \left(1 - \frac{\mathbf{Cost}_i}{\sum_i \mathbf{Cost}_i}\right) \quad (3)$$

where
$$\Delta\mathbf{Bel}_i = |0.5 - \mathbf{Bel}_i| \quad (4)$$

\mathbf{IU}_i denotes the integrated utility (\mathbf{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 $\Delta\mathbf{Bel}_i$ represents certainty of the belief of sensing node i which contributes to the Bayesian network. The maximum value of $\Delta\mathbf{Bel}_i$ is 0.5 when $\mathbf{Bel}_i = 0$ or 1, and the minimum is 0 when $\mathbf{Bel}_i = 0.5$. The \mathbf{IU} value will increase with increasing belief and will decrease with increasing sensing cost. We use a parameter t ($0 \leq t \leq 1$) to balance sensing cost and belief.

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

Reconstruction Algorithm:

1. Initialization of Bayesian network :

The mobile robot performs active sensing at every intersection, and constructs an original Bayesian network as shown in Fig. 3 using all of these sensing nodes.

2. STEP (1): Refine the local network.

For example, the system refines the local network \mathbf{k} (the sensing nodes of a intersection \mathbf{k}) of Fig.3 by the following algorithm:

- Check the $\Delta\mathbf{Bel}_i$ of every terminal sensing node, remove the node which satisfies $\Delta\mathbf{Bel}_i < \Theta$. (Θ ($0 < \Theta < 0.5$) is a threshold of $\Delta\mathbf{Bel}_i < \Theta$. When $\Delta\mathbf{Bel}_i < \Theta$, we consider that the sensing node has no capability to localize the mobile robot.)
- **IF** the number of survived nodes ($\Delta\mathbf{Bel}_i > \Theta$) is not zero, **THEN** sort the survived sensing nodes according to their \mathbf{IU} values, $\mathbf{IU}_{\mathbf{k}i} = \max_{\Omega_k} \{IU\}$, (Ω_k denotes the sensing nodes’ group of intersection \mathbf{k} .) Save this sensing node that has $\mathbf{IU}_{\mathbf{k}i}$, and remove the other nodes.
- **ELSE** execute “combining process” to combine the sensing nodes to improve the belief until the sensing node set has enough $\Delta\mathbf{Bel}$ to distinguish the other intersections.

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

- (a) Refine every local network (every intersection) based on the **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 \mathbf{IU} value of the combined sensing node sets which has $\Delta\mathbf{Bel}_{(\text{set})} > \Theta$, then sort these node sets based on \mathbf{IU} value.
- (c) Leave the sensing node set j , which has $\mathbf{IU}_{(\text{set } j)} = \max \{IU_{\text{set}}\}$, and remove the other node sets.

6 The Prototype System

The mobile robot (B14, Real World Interface) is equipped with a Pentium CPU, 16 sonar sensors, a color CCD camera. 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. 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 *low level action control*(**LLAC**).

7 Experiments

We conducted experiments to validate the effectiveness of our system using a simulator and a real robot (B14).

7.1 Assumptions of experiments

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

1. The structure of the Bayesian network is determined beforehand.
2. The prior probabilistic distribution (conditional probability table (**CPT**)) 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 local position of

¹We define the exclusive relation as $\overline{S_a} = S_b$. If the 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.

every landmark and other sensing information to construct **CPTs** (conditional probability tables) of all of the sensing nodes while it is moving in the workspace.

7.2 Experiment using a simulator

(1) Firstly, we made an office environment (Fig. 4) that has three intersections 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 intersection (**B1**, **B2** or **D**) to attempt to locate the goal **E**. The search of each intersection will be finished when the mobile robot perceives that the local environment is **C1** or **C2** (\top). Then the mobile robot turns back to gather the active sensing nodes upon receiving some instruction given by a human, and records all of the sensing nodes (we can obtain sonar distance information only). To distinguish **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 shown in Fig. 5 (a).

Consequently, we will reconstruct the original Bayesian network using the *reconstruction algorithm*. We can change the parameter t of the **IU** function (Eq.3), and the planned active sensing action will be different depending on the value of t . Fig. 4 (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 and does not consider sensing cost. The reconstruction process and the value and belief of every sensing node **IU** are illustrated in Fig. 5 (b) and (c). When $t = 0.3$, we obtain the results of the **IU** value of sensing nodes as shown in Fig. 6 (c). After the reconstruction process based on the **IU** value, we will acquire a new reconstructed Bayesian network (Figure 6 (b)); the planned action of the mobile robot is shown in Fig. 4 (down).

(2) We build a more complex environment to describe the problem as shown in Fig. 7. In the same way as in the previous experiments, the mobile robot initially navigates by **LLAC**, and gathers information to make **CTPs** of the sensing nodes and an original Bayesian network (Fig. 8 (a))

In Fig. 7, there are two hidden intersections (**F2**, **F3**) after passing intersections **B2** and **D**, respectively. We assume some hidden states (**H2** and **H3**) exist in the Bayesian network. **H2** (or **H3**) denotes the sensing node sets of the hidden intersections **F2** (or **F3**). We represent the causal relation between sensing nodes and hidden states as shown in Fig. 8 (a) (**C3** and **S3**'s parent is **H2**; **C4** and **S5**'s parent is **H3**). The sensed evidence will be propa-

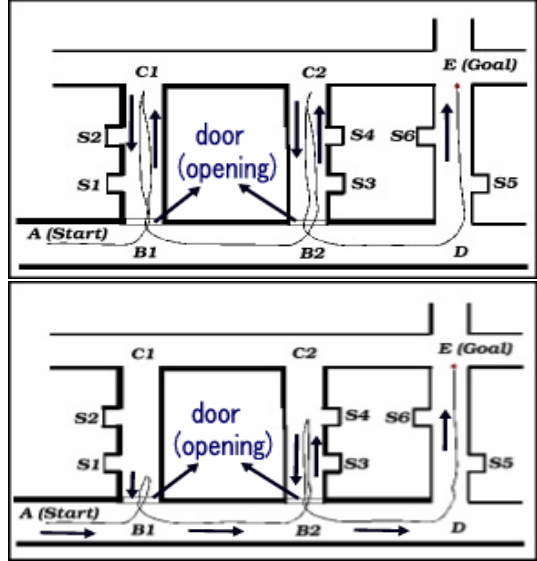


Figure 4: The mobile robot navigated following the solid line trajectory using inference of the reconstructed Bayesian network. (up) $t = 1$; (down) $t = 0.3$.

gated from terminal nodes to hidden states nodes (**H2** or **H3**), then **D**'s belief will be updated by propagation of the hidden node's probability. When the t value (Fig. 8 (c)) of **IU** function is 0.3, the original Bayesian network (Fig. 8 (a)) is reconstructed as Fig. 8 (b). Figure 7 (down) shows the planned path for localization of the mobile robot.

7.3 Real robot experiments

To validate our algorithm in a real environment, we built an experimental environment (Fig. 9), and the mobile robot performed wall-following using sonar sensors and local sensing using vision. A CCD camera is mounted on the robot to recognize the local environment (color landmark). Initially, when the door(B) is closed, the mobile robot recognizes the local sensor patterns for localization and navigation using low level sensor information processing (color landmark) (Fig.9(a)). While the door(B) is open, in the same way as in the previous simulation experiments, since the mobile robot cannot localize itself only by local sensing information, active sensing is performed using the sonar sensor (looking for some hollows on the walls). The mobile robot can observe the local sensor information (landmark) by vision to decide whether the position is the goal. The mobile robot performs active sensing using the sonar sensor while it senses the position **C** is not the goal (Fig.9(b)(left) and (c)(left)), and constructs the **CPTs** of every sensing node. The original Bayesian network is constructed following the robot's movement (Fig.9(b)(right) and Fig.9(c)(right)). Based

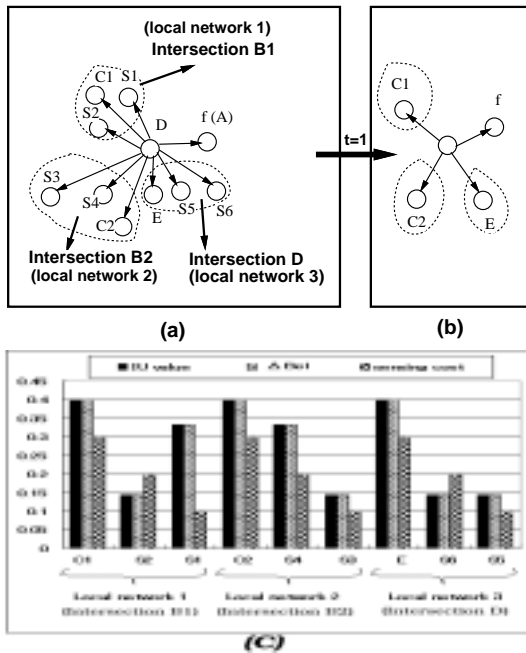


Figure 5: Reconstruction of the Bayesian network in experiment 1 while $t = 1$.

on our reconstruction algorithm, we can obtain a t value ($t = t_0$) to balance the localization belief and sensing cost (Fig.9(d)). Consequently, the mobile robot plans its action to obtain the active sensing event(Fig.9(e) and (f)) and infers the localization($\text{Bel}(\mathbf{D})$) of itself using the reconstructed Bayesian network(Fig.9(d)(right)).

8 Conclusions

We proposed a novel method of sensor planning for mobile robot localization using Bayesian network representation and inference. We can represent contextual relations between local sensing results and global localization of a mobile robot in a Bayesian network. Inference of the Bayesian networks allows us to classify ambiguous positions of the mobile robot when the local sensing evidences are obtained. By taking into account the sensing cost and localization belief degree, we define an integrated utility function to decide the sensing range. The function allows us to reconstruct the Bayesian network and obtain the optimal sensor plan for localization. The experimental results of the sensor planning for a mobile robot demonstrate the usefulness of the proposed system.

Our future plans include the following: (1)to learn the structure of the Bayesian network from probabilistic data, and (2)to validate our concepts using other applications.

References

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

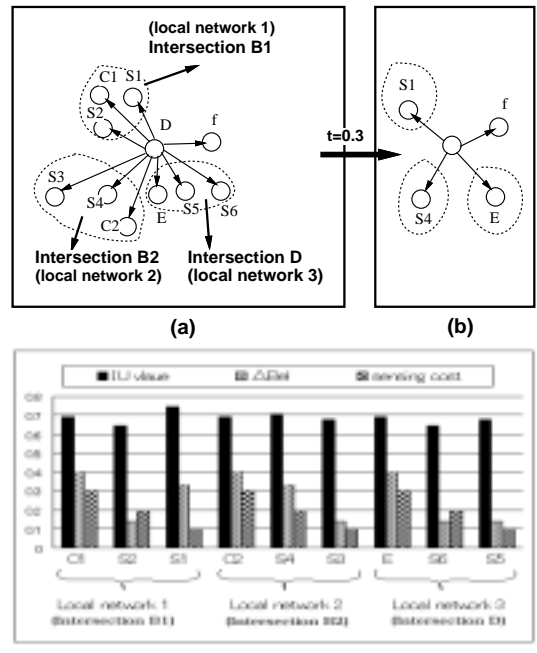


Figure 6: Reconstruction of the Bayesian network in experiment 1 while $t = 0.3$.

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

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

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

[5] E.Horvitz, J.Breese, D.Heckerman, D.Hovel, K.Rommelse, "The Lumiere Project: Bayesian User Modeling for Inferring the Goals the Needs of Software Users," *Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence*, pp.256-265, 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] Kenneth Basye, Thomas Dean, Jak Kirman, and Moises Lejter, "A Decision-Theoretic Approach to Planning, Perception, and Control", *IEEE Expert*, Vol.7, Number 4, Pages 58-65, 1992.

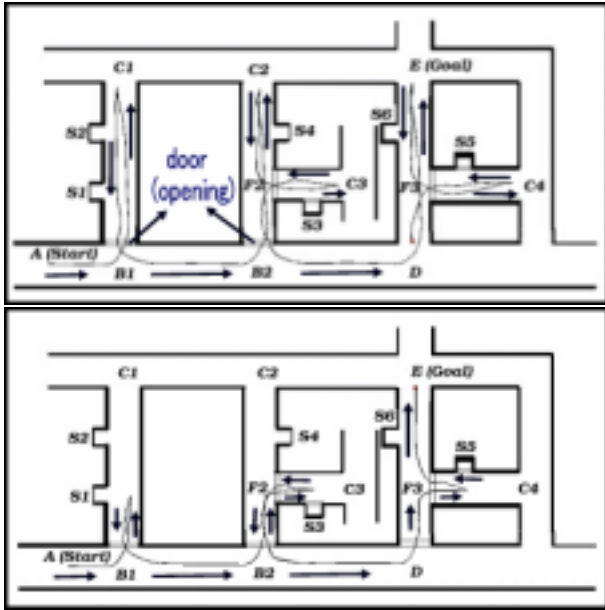


Figure 7: (up) The mobile robot navigates itself by LLAC and some tutorial commands to search the goal (E) and actively gathers the sensor information to construct the CPTs of every sensing node and original Bayesian network. (down) The mobile robot is navigated following the solid line trajectory using inference of the reconstructed Bayesian network ($t = 0.3$).

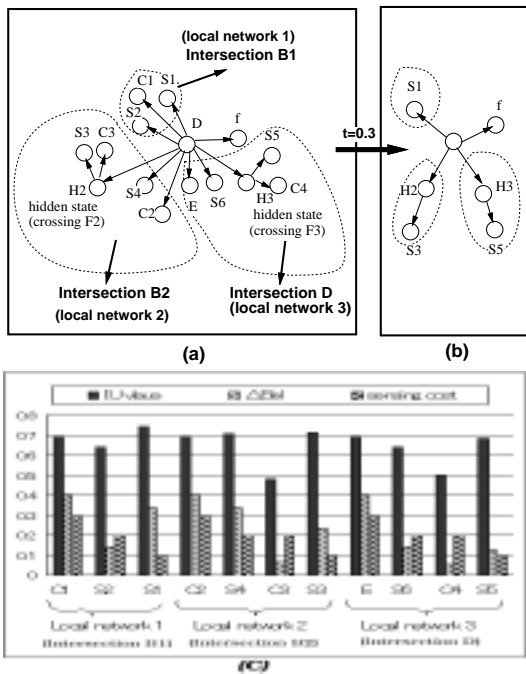


Figure 8: Reconstruction of the Bayesian network which has hidden states.

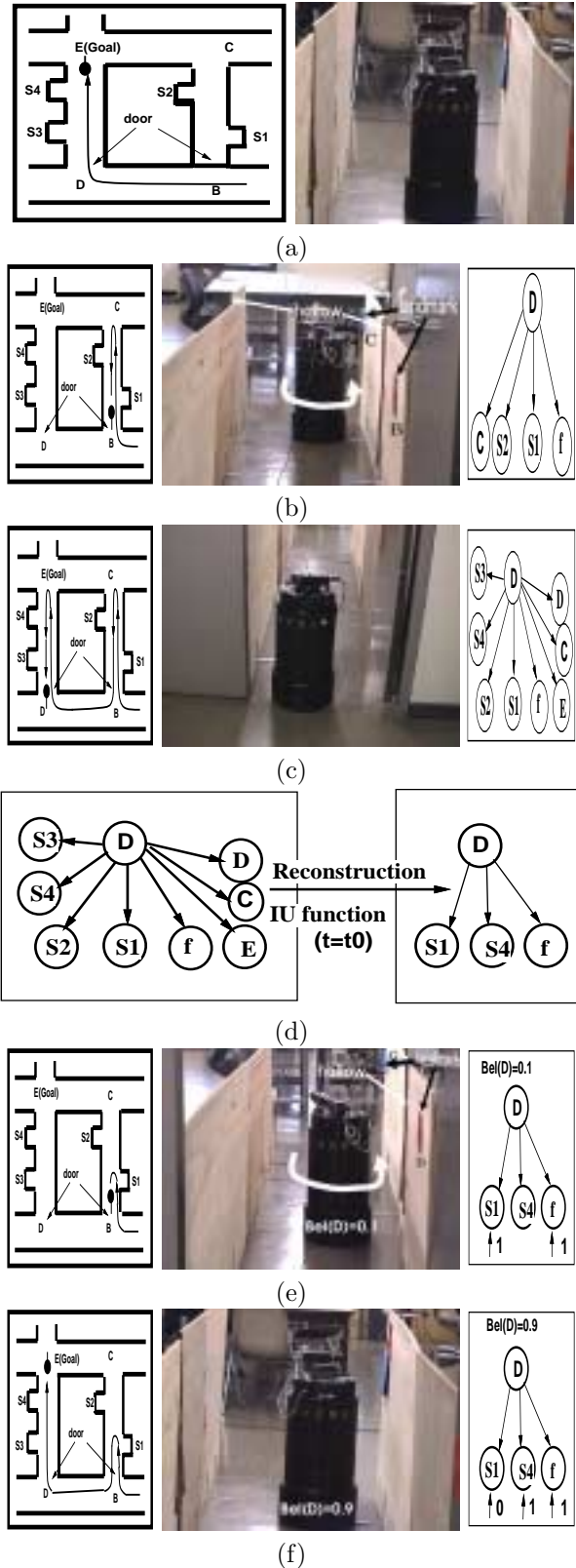


Figure 9: Real robot experiments of localization