

Sensor Planning for Mobile Robot Localization Using Bayesian Network Inference

Hongjun Zhou,^{1,*} Shigeyuki Sakane²

¹*Department of Industrial and Systems Engineering, Chuo University
1-13-27 Kasuga, Bunkyo-ku, Tokyo, 112-8551 JAPAN*

²*Department of Industrial and Systems Engineering, Chuo University
1-13-27 Kasuga, Bunkyo-ku, Tokyo, 112-8551 JAPAN*

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 some simulation and real robot experiments to validate the sensor planning system.

Key words: Sensor planning; mobile robot; localization; Bayesian network inference; uncertainty.

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 fac-

tors 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] [11] proposed localization of a mobile robot using Bayesian analysis of the probabilistic belief. The system models environment and actions in Markov model following the time axis, the environment must be static [11], moreover, the system depends on distance information from wheel encoder. However, wheelless robots are very difficult to get accurate distance information using odometer, and we have to consider how to cope with a dynamic environment. In this paper, we represent causal and contextual relation of the sensing results and global localization in a Bayesian network, 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 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. Inamura et al. [7] acquired a mobile robot's action based on a prior-built Bayesian network, and decreased uncertainty by "human teaching". These methods cope with uncertainty of local sensor information and localize the robots based on calculation of the posterior probability. However, these methods have not implemented sensor planning mechanisms to efficiently gather information of the environment. Moreover, since constructed Bayesian networks cannot be modified, the systems have to initially prepare a complete Bayesian network to have prior knowledge. Singhal et al. [8] presented an approach for multimodal sensor fusion using dynamic Bayesian networks and an occupancy grid. The robot updates the occupancy grid representation via dynamic Bayesian networks. However this method also has no sensor planning capability. As for sensor planning, Miura et al. [9] 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. [10] used Bayesian networks to recognize table setting, and plan the camera's movement based on maximum expected utility decision rules.

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. The robot actively gathers the sensor information and maps that information into Bayesian networks. To form an efficient sensing strategy, the system reconstructs the Bayesian network based on a composite criterion between localization belief and the sensing cost. This reconstructed Bayesian network is utilized to plan the sensing action and localization of the mobile robot.

3 Task Setting

We would like to describe our main task setting of this paper. As shown in Fig 3, a mobile robot learns 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 door (at an intersection **B**) is closed. However when door (at the intersection **B**) is open incidentally, the local sensing information at **B** and **D** will be identical. Therefore the mobile robot can not distinguish which intersection 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 intersections with the same local sensing information in a navigation path, how to 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**.

In our system, the mobile robot can change the start point, and the localization doesn’t depend on distance information obtained from wheel encoder. Moreover, since the distance information have lots of uncertainty, for example, in Figure 9, if we assume it is very near from B1 to B2, localization of B2 will be very difficult only by measuring distance from start point to B1 or B2. To cope with the uncertainty of measuring distance and dynamic environment, our system combine the low level navigation and high level inference in Bayesian network. Of course, we can also fixed the environment and combine the global position measuring or local distance measuring (for example, the distance from B1 to B2) into a Bayesian network [6], the localization will be trivial.

It is easy to measure the distance by a wheel encoder for instance, but if localization is applied to a humanoid rather than a wheeled mobile robot, the combination of pieces of low level navigation in Bayesian network is more available and general than measuring distance.

4 Basic concept of the system

Bayesian networks are directed acyclic graphs (DAGs) that represent dependencies between variables in a probabilistic models [1] [2] [3]. It is a graphical way to represent a particular factorization of a joint distribution. Each variable is represented by a node in the network. A directed arc is drawn from node A to node B if B is conditioned on A in the factorization of the joint distribution. For example, to represent the factorization (1) we would draw an arc from W to Y but not from W

to Z . The Bayesian network representing the factorization (1) is shown in Figure 1.

$$P(W, X, Y, Z) = P(W)P(X)P(Y|W)P(Z|X, Y) \quad (1)$$

If we would compute the conditional probability $P(Y|W, X, Z)$, firstly, we must obtain conditional probability of every pair of variables that connected by directed arc, and evidence of nodes will propagate the local conditional probability to the node Y following the arc. Some Bayesian network applications are implemented successfully in user modeling [12], mobile robot navigation ([6] [7]), and vision planning [10] in recent decade.

In this paper, we represent causal and contextual relation of the sensing results and global localization in a Bayesian network, but not in Bayesian rule, and fuse all of conditional probabilities between sensing results and global localization to compute the conditional probability (localization belief) $P(D|f, s_1, \dots, s_n)$ when some sensing events are happened.

In this paper we propose a concept of Bayesian network reconstruction for sensor planning. To summary, we use an extended *wet grass* example of Ref.[2] shown in Figure 2(a). Mr. Holmes infers which is the cause of Holmes's wet grass (node H), i.e., rain (node R) or sprinkler (node S). If Holmes knows Watson's grass is also wet, then the belief of R will increase and the belief of S will decrease. This is an "*explaining away*" example (Fig.2(a)). Now we consider the case in (Fig.2(b)) in which Holmes makes a choice in checking (or sensing) multiple evidences (W_1, W_2, W_3, \dots) which could increase belief of the node R . If the sensing condition such as belief of the parent node, sensing cost, etc. varies depending on W_i , Holmes should consider which W_i is appropriate based on some criterion taking into account the condition. This is basically a sensor planning problem. If we take the nodes R and S as beliefs of a mobile robot localization and the nodes W_i as the sensing actions, the sensor planning problem can be transformed into construction and evaluation of such Bayesian network.

For solving this problem, we propose a Bayesian network reconstruction based on an integrated utility function (a ratio of belief and the sensing cost). The system compares the integrated utility value of every node of *explaining away* step, then reconstructs the last configuration of the Bayesian network (Fig.2(c)). The reconstructed Bayesian network will help Holmes to determine the best sensing strategy, i.e., the most effective way to search for evidences supporting the node R and increase the belief. The reconstruction of the network serves to prune the useless search tree for evidences.

Based on the above concept, 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 4 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.
- Our method actively performs sensor planning and reconstruct the network structure taking into account both of sensing cost and localization belief, then forms a plan for the sensing action to 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 Figure 5. 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.[13]. 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 3 (left) shows a trajectory of the robot in a workspace. Fig.3 (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 pixel 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 intersections, 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 an intersection, 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 intersection. The robot's action is determined by *low level action control* at the intersection. Let us assume local sensor data \mathbf{S} are projected to a smaller feature space \mathbf{F} , and the robot is given a filter function π

$$\pi : \mathbf{S} \longrightarrow \mathbf{F} \quad (2)$$

$$f = \pi (s) \quad (3)$$

where $f \in \mathbf{F}$, $s \in \mathbf{S}$

We employ a three-layered Back Propagation Neural Network (**BPNN**) to model the filter function π 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 intersections (like \perp , $+$, \top) of the path.

7 Implementation of HLI

7.1 Active sensing for localization using Bayesian network inference

As shown in Fig. 6, the belief of position \mathbf{D} at the intersections (\mathbf{B} or \mathbf{D}) can be obtained as the following formalization.

$$\mathbf{Bel}(\mathbf{D}) = P(\mathbf{D} | f) \quad (4)$$

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

The trajectory indicated by the solid line in Figure 6 illustrates a result of this algorithm. Filtering the local sensing pattern of position \mathbf{B} and position \mathbf{D} , based on Eq.(3), we can acquire the $\mathbf{Bel}(\mathbf{D})$ of every intersection (\mathbf{B} or \mathbf{D}) using Eq.(4). Fig.6(a) shows the belief of position \mathbf{D} depends on the solid line trajectory. Since the local sensor information of \mathbf{B} is identical with that of \mathbf{D} , the mobile robot can not localize itself only by the local sensing pattern only by Eq.(4), while it runs from the "Start" point to the intersection \mathbf{D} directly.

To overcome the difficulty and search the "true \mathbf{D} ", the mobile robot performs active sensing as shown by the dotted line trajectory in Fig.6. This time we can obtain the belief of \mathbf{D} at the intersections (\mathbf{B} or \mathbf{D}) from the following function:

$$\mathbf{Bel}(\mathbf{D}) = P(\mathbf{D} | f, s_1, \dots, s_n) \quad (5)$$

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 feature along the path. We construct the Bayesian network as shown in Fig.7(b) to calculate the $Bel(\mathbf{D})$ at the intersections (\mathbf{B} or \mathbf{D}). Figure 6(b) shows the belief of the position \mathbf{D} which corresponds to the trajectory of a dotted line. Sensing nodes propagate the evidences backward to the node \mathbf{D} . $Bel(\mathbf{D})$ of the intersection \mathbf{D} is increased while $Bel(\mathbf{D})$ of the intersection \mathbf{B} is decreased.

7.2 Reconstruction of the Bayesian network for sensor planning

We can obtain the $Bel(\mathbf{D})$ from Eq. (5), 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 (\mathbf{IU}) function (Eq. 6) 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.

$$\mathbf{IU}_i = t \times \Delta\mathbf{Bel}_i + (1 - t) \times \left(1 - \frac{\mathbf{Cost}_i}{\sum_i \mathbf{Cost}_i}\right) \quad (6)$$

where

$$\Delta\mathbf{Bel}_i = |0.5 - \mathbf{Bel}_i| \quad (7)$$

\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 $\mathbf{Bel}_i = 0.5$. \mathbf{IU} value will increase along with increasing belief and decrease along with increasing sensing 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 intersection could guide itself to the goal based on the beliefs of sensing nodes (or *sensing node sets*) of the intersection, we associate the sensing nodes of the corridor of each intersection to a *local network*. If there is a sub-corridors in a certain corridor, a hidden state node (\mathbf{H}) will be defined in the *local network*, and sensing information of the sub-corridor will be the children nodes of the hidden state node (\mathbf{H}) (see Fig. 12 and Fig. 13). So a hierarchial Bayesian network can be constructed.

We associate the sensing nodes of a corridor at each intersection to a *local network*. If there are some sub-corridors in a certain corridor of each intersection, some nodes named hidden state nodes (\mathbf{H}) will be defined in the *local network*.

We associate a hidden state node with a sub-corridor. Sensing information of the sub-corridor corresponds to children nodes of the hidden state nodes (H) (see Fig. 12 and Fig. 13). So a hierarchical Bayesian network can be constructed.

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. 12) which it could guide the mobile robot to a goal using local environment sensing information, the mobile robot will begin 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 will be finished when the mobile robot finds the goal (+) or local environment \top (for example, C_1, C_2, C_3, C_4 of Fig. 12). The mobile robot stops search action, then turns back to the previous path, and records the sensing information of the 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. 12).

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 these 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. 12). The mobile robot represents the position relation between the corridors and sub-corridors as a *position relation tree*, in which sub-corridors are children nodes of its upper layer corridor’s (or sub-corridor) node.

The mobile robot can search and gather the sensing information in all of the corridors, sub-corridors, subsub-corridors (or more deep sub-corridor) following the wall, until it come back to the first intersection of corridor (for example, B_1, B_2, C of Fig. 12).

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* (Ref. Section 4). 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. 12, 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*.

Reconstruction Algorithm:

1. Initialization of Bayesian network :

The mobile robot performs active sensing at every intersection, and constructs an original Bayesian network as Figure 8 using all of these sensing nodes.

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

For example, the system refine the local network \mathbf{k} (the sensing nodes of an intersection \mathbf{k}) of Fig.8 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,
IU_{ki} = $max_{\Omega_k} \{IU\}$, (Ω_k denotes the sensing nodes group of intersection **k**.)
 Save this sensing node that has **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 intersections.

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

- (a) Refine the every local network (every intersection) 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 **IU_(set j)** = $max \{IU_{set}\}$, and remove the other node sets.

8 Experiments

We conducted some simulation and real robot experiments to validate the effectiveness of our system.

8.1 Assumptions of experiments

To simplify the calculation and construction of *local network*, our experiments have the following assumptions:

1. The parents-children relations are determined beforehand. The intersection is parents nodes, sensing nodes of its corridor (or sub-corridor) which starts from the intersection are children nodes (or grandchildren nodes).

¹We define the exclusive relation as $\bar{S}_a = S_b$. 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.3.

2. The mobile robot searches all corridors, sub-corridors (or subsub-corridors), and gathers the sensing information in following the wall.
3. The goal have unique local environment in the workspace The mobile robot will stop the search action in every corridor (or sub-corridor) when it recognizes that a certain intersection is (is not) the goal only by local sensing, and turns back to entry of this corridor (or sub-corridor, subsub-corridor), then gathers sensing information.
4. We use a position relation *tree* and search order to label the intersections and corridors (or sub-corridors).
5. The mobile robot can categorize sensing nodes using intersections and corridors (or sub-corridors) labels and sensing feature.
6. Prior probabilistic distribution (conditional probability table) of sensing nodes is acquired by measuring the frequencies of the events.
7. We have not used a global distance information (from start point to current point) and local distance information (between the sensing nodes) for localization. The system only records the number of sensing features in sensing order in every corridor (or sub-corridor). (Of course, we also can measure the accurate global distance using odometer, the localization will be trivial.)
8. Sensing cost is proportional to the number of sensing features.

8.2 Simulation experiment 1

Firstly, we made an office environment (Figure 9) 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 search the goal **E**. The search of each intersection will be finished while the mobile robot perceives the local environments is **C₁** or **C₂** (\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 10(a).

Consequently, we will reconstruct the original Bayesian network using the *reconstruction algorithm*. We can change the parameter t of **IU** function (Eq.6), the planned active sensing action will be different depending on the value of t . Figure 9 (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 node's **IU** value and belief is illustrated at Figure 10 (b) and (c). When $t = 0.33$, we obtain the results of **IU** value of sensing nodes as Figure 11 (c). After the reconstruction process based on the **IU** value, we will acquire a new reconstructed Bayesian network

(Figure 11 (b)). In this case, sensing action of the mobile robot will be planned as shown in Figure 9 (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 Simulation experiments 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 12. In the same way as the previous experiments, the mobile robot initially navigates by **LLAC**, and gathers information to make **CPTs** of the sensing nodes and an original Bayesian network (Figure 13 (a))

In Fig. 12, there are two hidden intersections (\mathbf{F}_2 , \mathbf{F}_3) after passing intersections \mathbf{B}_2 and \mathbf{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 intersections \mathbf{F}_2 (or \mathbf{F}_3), we represent the causal relation between sensing nodes and hidden state as shown in Fig. 13 (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 hidden state node (\mathbf{H}_2 or \mathbf{H}_3), then \mathbf{D} 's belief will be updated by propagation of hidden node's probability. When the t value (Fig. 13 (c)) of **IU** function is 0.35, the original Bayesian network (Fig. 13 (a)) is reconstructed as Fig. 13 (b). Fig. 12 (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 navigate to the goal in the complex environments using the hierarchical Bayesian network.

8.4 Real robot experiments

To validate our algorithm in a real environment, we built an experimental environment (Figure 14), and the mobile robot performed wall-following using sonar sensor 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.14(a)). While the door(B) is open, in the same way as in the previous simulation experiments, since the mobile 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 \mathbf{C} is not the goal (Fig.14(b)(left) and (c)(left)), and constructs the **CPTs** of every sensing node. The original Bayesian network is constructed following the robot's movement (Figure 14(b)(right) and Fig.14(c)(right)). Biased on our reconstruction algorithm, we can obtain a t value ($t = t_0$) to balance the localization belief and sensing cost (Fig. Fig.15(d)). Consequently, the mobile robot plans its action to obtain the active sensing event(Fig.15(e))

and (f)) and infer the localization($\mathbf{Bel}(\mathbf{D})$) of itself using the reconstructed Bayesian network(Fig.15(d)(right)).

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.

Although we have developed the prototype system and verified our approach, the current system has some limitations. For example, the parameter t of utility function has to be selected by several experimental results. However, how to balance the sensing cost and localization belief in a generally integrated method? We consider this is a very difficult problem. Since the parameter t decision very depends on sensing cost definition, moreover, it is difficult to represent the sensing cost objectively. Maybe a hieratical decision process is more general, in which the system calculates localization belief initially, then compares sensing cost of those sensing actions if they can give enough belief.

In this system, the searching goal and gathering sensing event actions are controlled by some sensor mapping functions (Eq.2,3) or some explicit landmarks. However, in some complex environments, we can not use this simple method to gather sensing information for Bayesian network construction. Some traditional path planning solutions (for example, Chinese postman problem) will help us to solve this problem.

Our future plan includes the following: (1) to construct a generally integrated method to balance the sensing cost and localization belief, (2) to learn structure of Bayesian network from probabilistic data of sensing information, (3) to validate our concepts using other applications.

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, 1997.
- [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] T.Inamura, M.Inaba, and H.Inoue,"Acquisition of Probabilistic Behavior Decision Model based on the Interactive Teaching Method," *Proc. of the 9th Int. Conf. on Advanced Robotics ('99ICAR)*. pp.523-528, 1999.
- [8] A.Singhal and C.Brown, "Dynamic Bayes Net Approach to Multimodal Sensor Fusion," *Proc. of the SPIE conference on Sensor Fusion and Decentralized Control*, Vol.3209, Pittsburgh, PA, October 1997.
- [9] 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.
- [10] 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.
- [11] S. Thrun, "Probabilistic Algorithms in Robotics", *AI Magazine*, 21(4):93-109, 2000.
- [12] 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.
- [13] T.Dean et al., *Artificial Intelligence*, The Benjamin/Cummings, 1995.

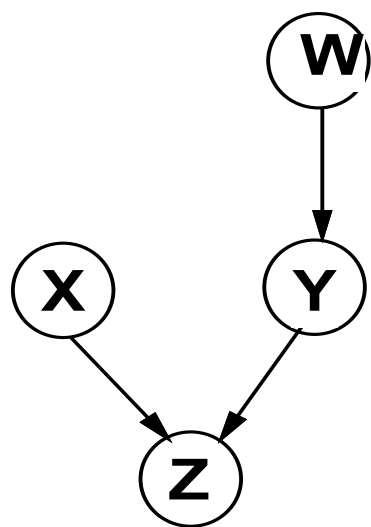


Figure 1: A directed acyclic graph (DAG) consistent with the conditional independence in $P(W, X, Y, Z)$.

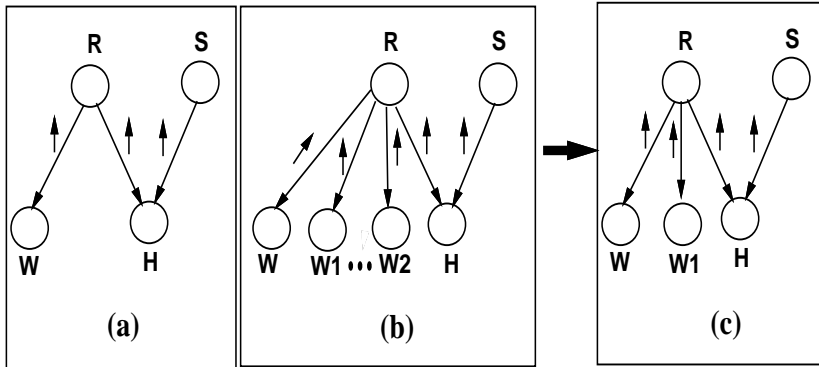


Figure 2: Bayesian network models for *wet grass* example (a)[10] and extended examples ((b),(c)). *Rain* (node **R**) and *sprinkler* (node **S**) are causes of *Holmes'* (node **H**) grass being wet, and *Watson's* grass (node **W**) supports belief of *Rain*.

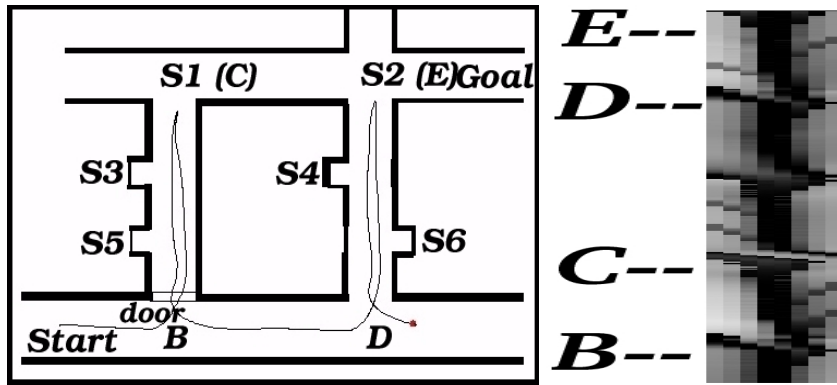


Figure 3: The trajectory and its associated sensor data flow of a mobile robot

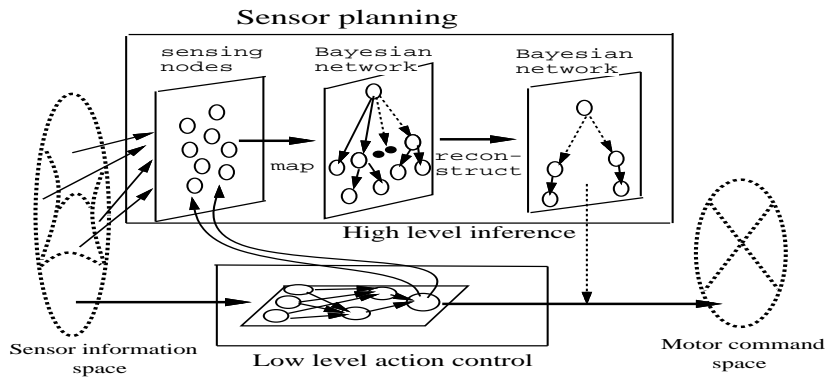


Figure 4: Multi-layered-behavior architecture for sensor planning



Figure 5: RWI B14 mobile robot of this paper

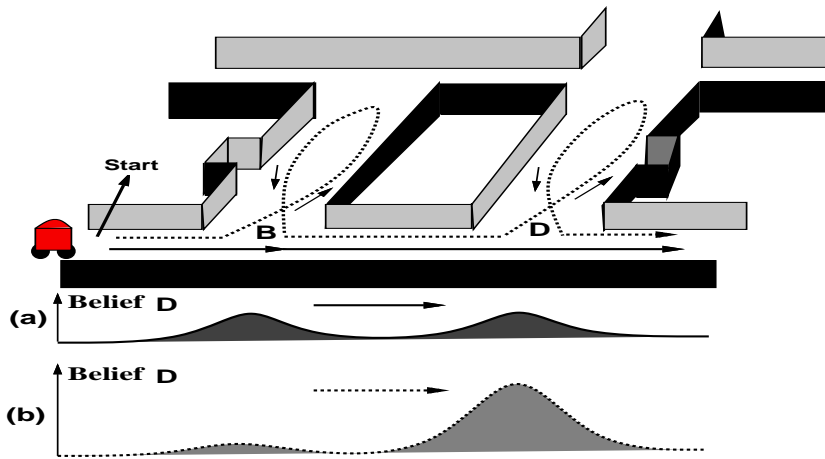


Figure 6: Active sensing for localization using Bayesian network inference

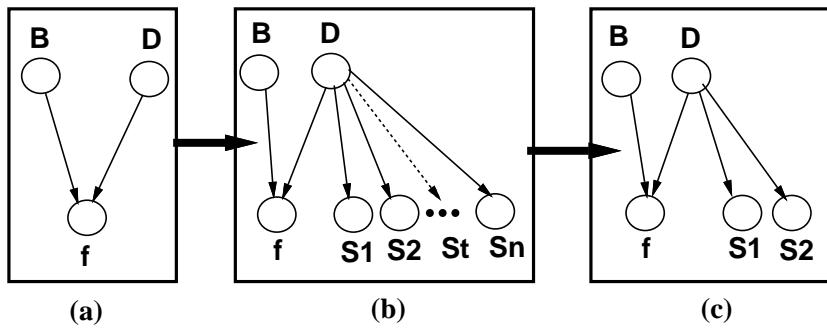


Figure 7: Construction and reconstruction of the Bayesian network for sensor planning

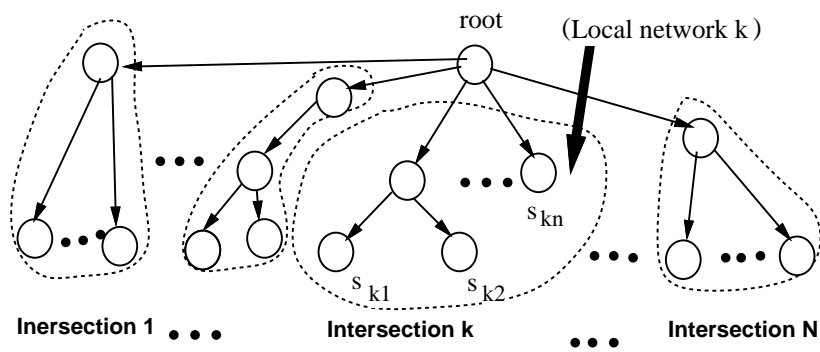


Figure 8: Local network of Bayesian network. Every local network is constructed by each intersection's active sensing nodes. Evidence of these sensing nodes will be propagated to root node, and using these posterior probability to decide if this intersection can guide the mobile robot to the goal.

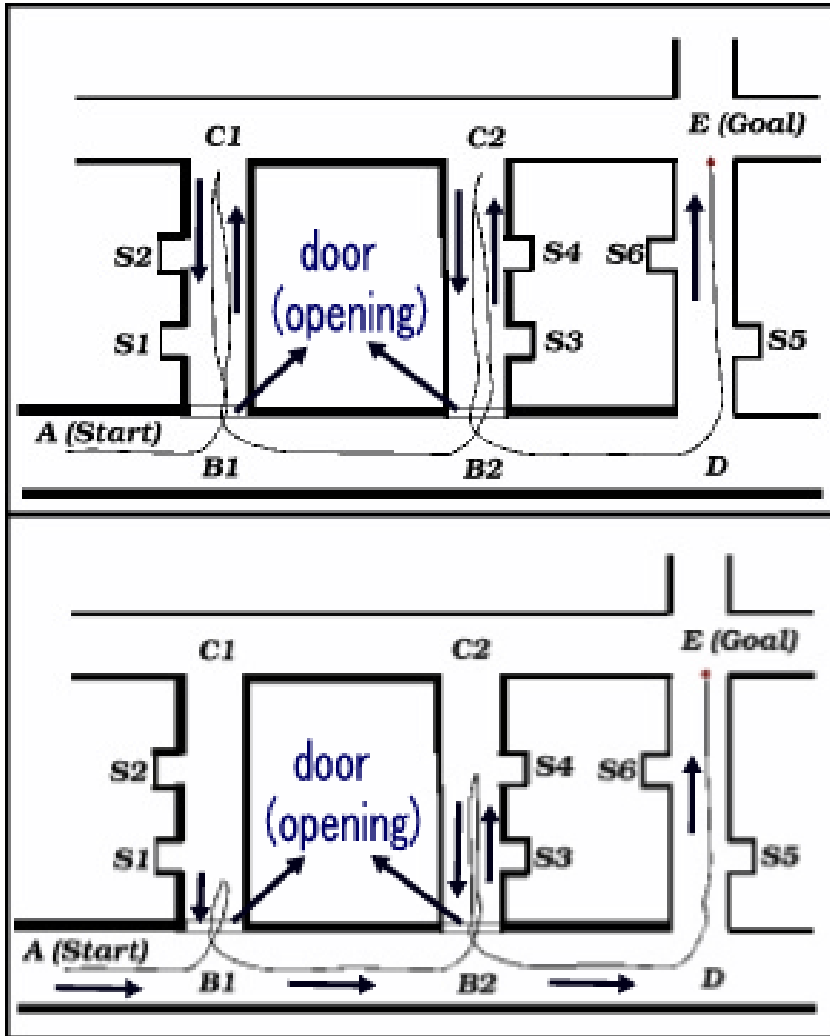


Figure 9: The mobile robot navigated following the solid line trajectory using inference of reconstructed Bayesian network. (up) $t = 1$; (down) $t = 0.33$.

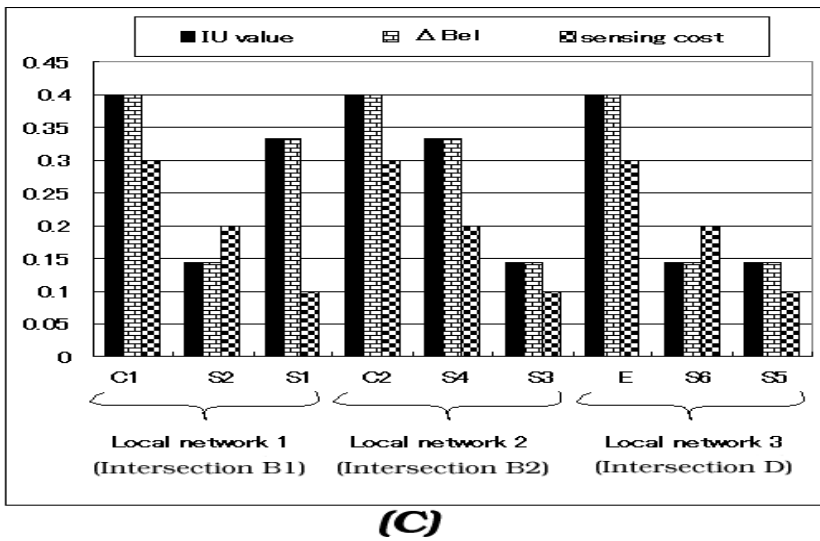
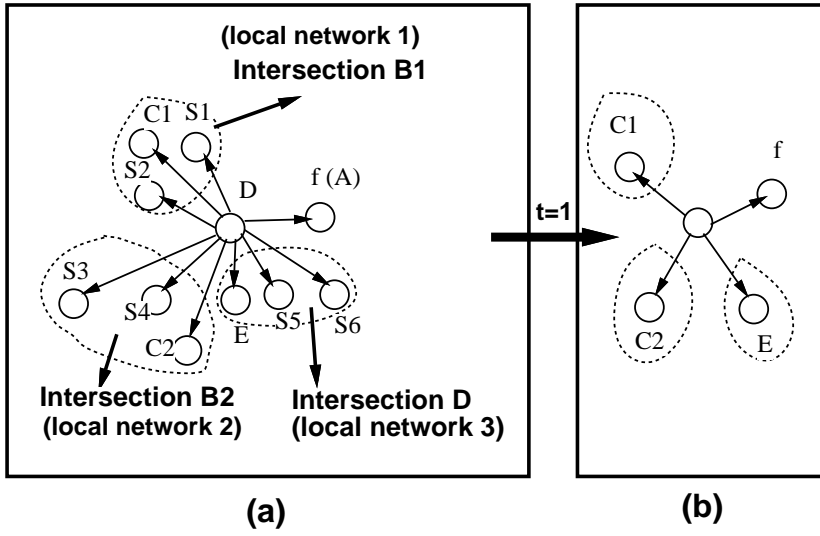


Figure 10: Reconstruction of the Bayesian network in the experiment 1 while $t = 1$.

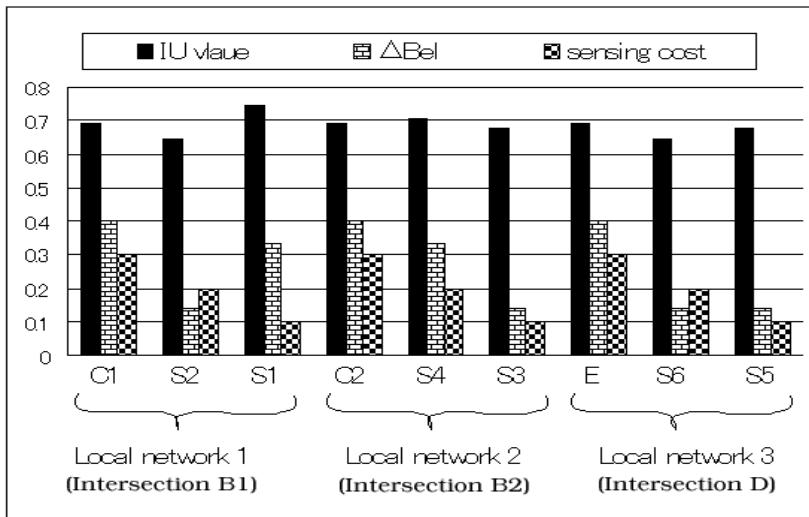
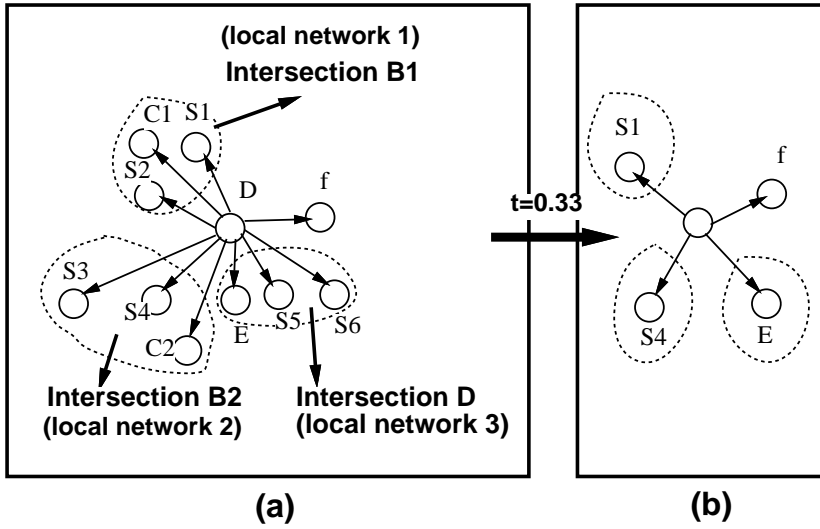


Figure 11: Reconstruction of the Bayesian network in the experiment 1 while $t = 0.33$.

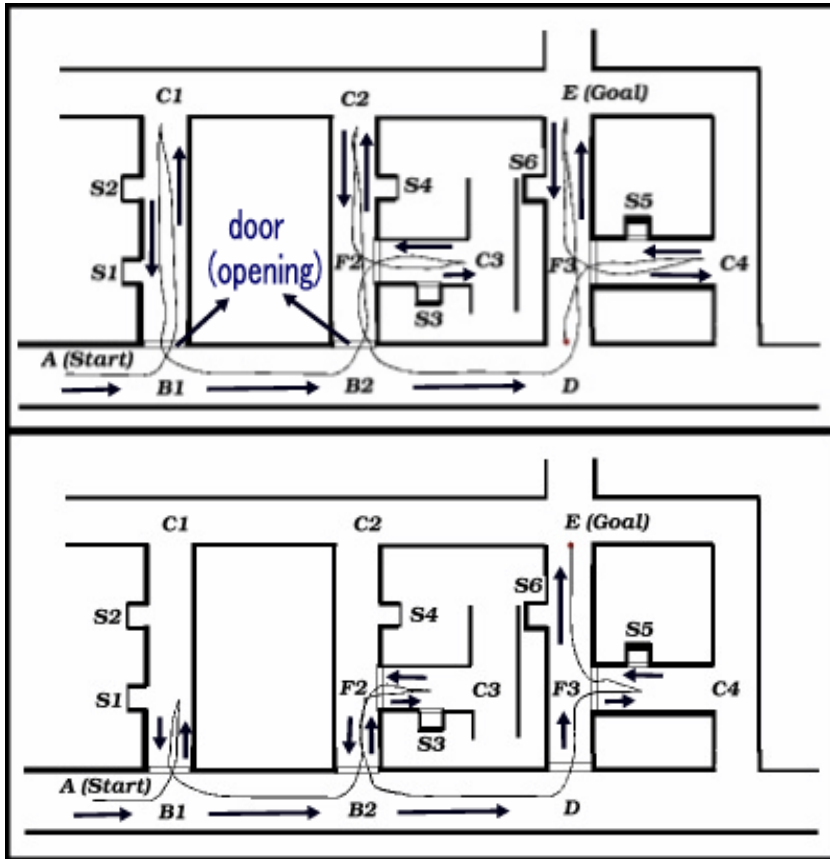
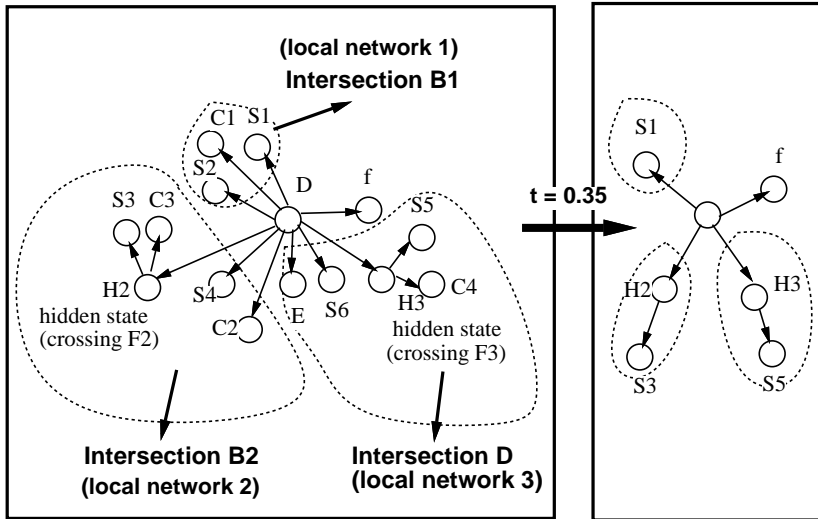
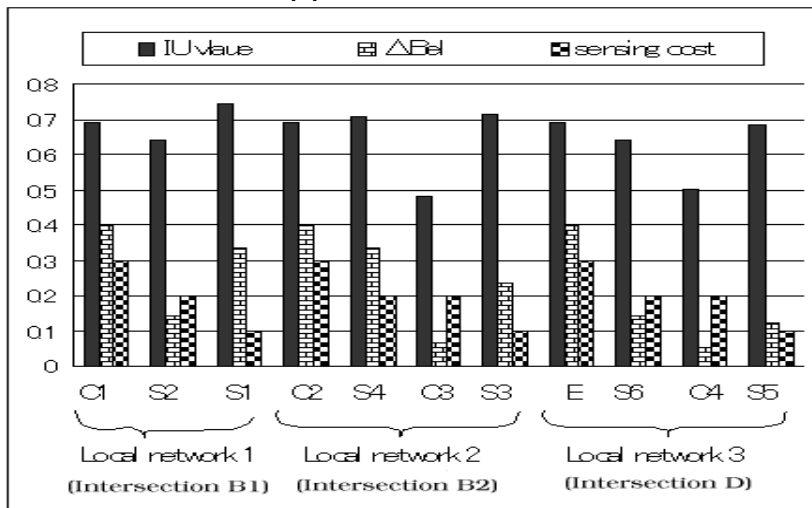


Figure 12: (up) The mobile robot navigates itself by **LLAC** and some tutorial commands to search the goal (**E**) and gathers the sensor information actively, then compares the difference of every intersection to construct the **CPTs** of every sensing node and original Bayesian network. (down) The mobile robot is navigated following the solid line trajectory using inference of reconstructed Bayesian network ($t = 0.35$).



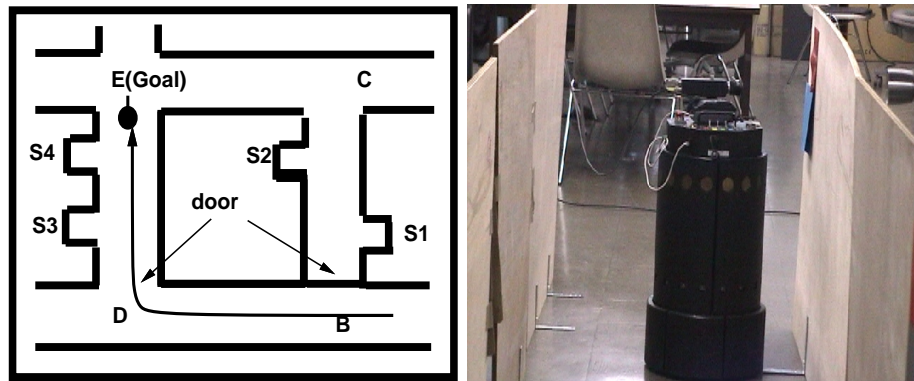
(a)

(b)

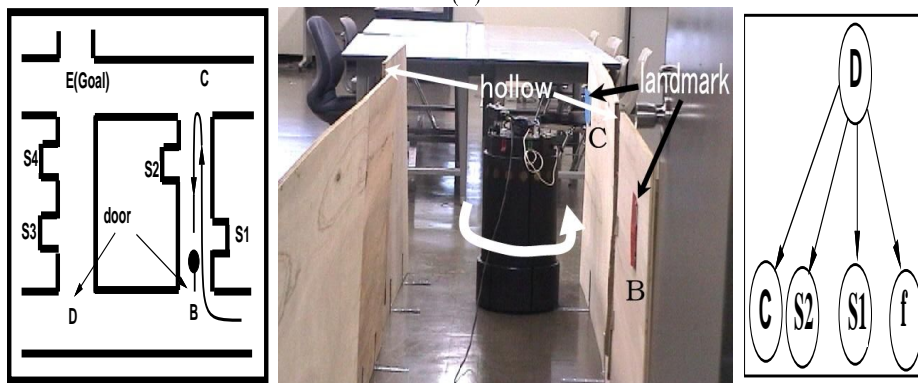


(c)

Figure 13: Reconstruction of the Bayesian network which has hidden states.



(a)

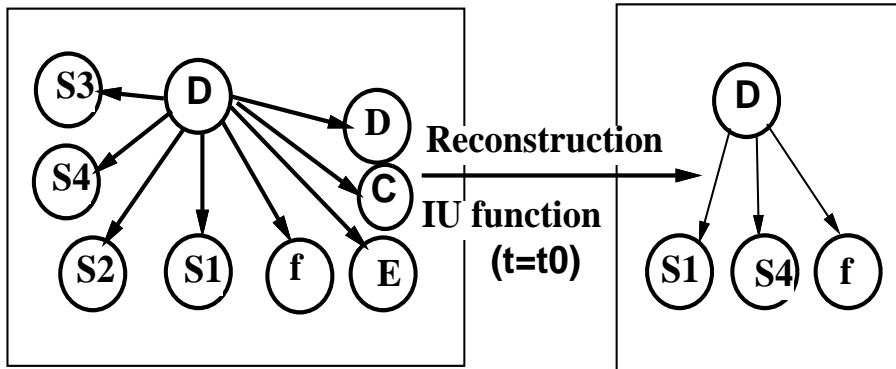


(b)

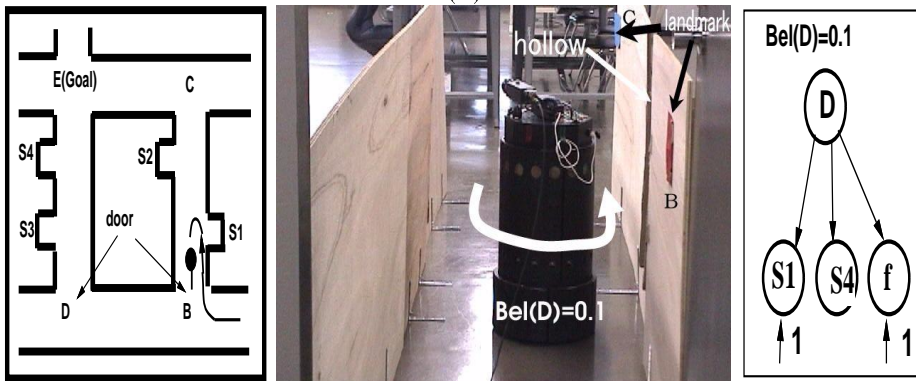


(c)

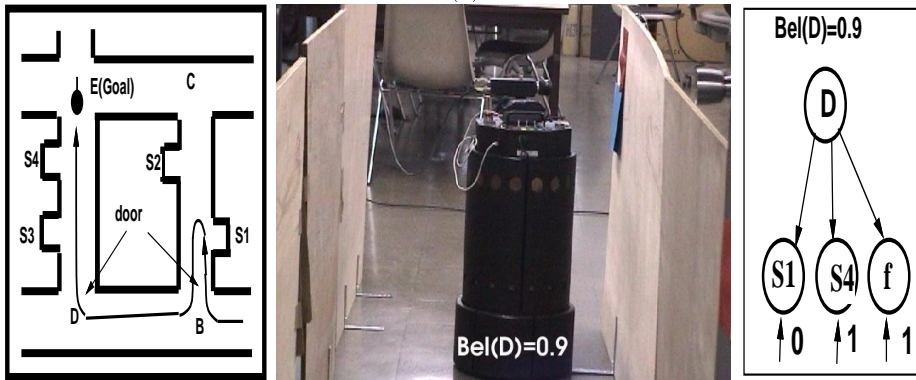
Figure 14: Real robot experiments of localization (1)



(d)



(e)



(f)

Figure 15: Real robot experiments of localization (2)