How a mobile robot selects landmarks to make a decision based on an information criterion

Noriaki Mitsunaga* (mitunaga@atr.jp) and Minoru Asada†+
(asada@ams.eng.osaka-u.ac.jp)
*ATR Intelligent Robotics and Communication Laboratories,
†Graduate School of Engineering, Osaka University,
+Handai Frontier Research Center

Abstract. Most current mobile robots are designed to determine their actions according to their positions. Before making a decision, they need to localize themselves. Thus, their observation strategies are mainly for self-localization. However, observation strategies should not only be for self-localization but also for decision making. We propose an observation strategy that enables a mobile robot to make a decision. It enables a robot equipped with a limited viewing angle camera to make decisions without self-localization. A robot can make a decision based on a decision tree and on prediction trees of observations constructed from its experiences. The trees are constructed based on an information criterion for the action decision, not for self-localization or state estimation. The experimental results with a four legged robot are shown and discussed.

Keywords: Decision making, Decision tree, Information criterion, Observation strategy, Active perception

Table of Contents

1. Introduction

Many active sensing methods for state space estimation have been proposed (Mihaylova et al., 2002). Most proposed methods for a mobile robot focus on or are based on estimation and tracking of positions of the robot itself. Moon et al. (Moon et al., 1999) have proposed a viewpoint planning method for a robot to reach a target position in less time by reducing the need for frequent observations for self-localization. Their method plans the viewpoints so that the estimation error of the robot position will be small enough to avoid hitting obstacles and walls. Kristensen (Kristensen, 1997) has proposed a planning method for sensing actions. A mobile robot selects the sensing actions that maximize an estimated utility, which will be given for (non-sensing) actions after the sensing. Jensfelt et al. (Jensfelt et al., 2000) have proposed a planning method based on Dynamic Programming. Based on predefined cost functions for sensing and resulted failures, such as hitting a wall, a robot makes the plan of the minimum cost.

Other than the use of the position error and utility estimation, the use of entropy of the state estimation has been proposed. The state or the position of the robot is represented with probabilities and the entropy is calculated for the probabilities. The entropy reflects the uncertainty of the state. Active sensing methods that selects sensors based on entropy are proposed for discrimination of an object (Hutchinson and Kak, 1989), for discrimation of a material of an object (Sakaguchi, 1994), and for target localization (Wang et al., 2004). Roy et al. (Roy et al., 1999) have proposed path planning for a navigation task of a mobile robot based on entropy with Markov localization using occupancy grids as a world model representation. Fox et al. (Fox et al., 1998) have proposed an active localization method. A mobile robot selects a motion for self-localization and navigation based on an utility function constructed from the entropy of the position itself and the utility of actions. They also used Markov localization using occupancy grids.

The above methods for a mobile robot rely on the self-localization to make decisions. However, is self-localization mandatory for decision making in a mobile robot? Humans do not seem to have adopted such a strategy. For example, when we cross a street, we do not seem to localize ourselves precisely in a crosswalk but seem to position ourselves within the boundaries of the crosswalk. In other words, we collect and use minimal information not for self-localization but for decision making. Mobile robots can also adopt an observation strategy that is independent of self-localization or state estimation but useful for action decision.

Tani et al. (Tani et al., 1997) have experimented with the task of watching two visual targets using a camera with a limited viewing angle. The robot predicts the position of the visual targets and switches its visual attention depending on the accuracy of the prediction. However, the robot's action (wall following) is fixed and the issue can be regarded as a view prediction problem on a route. Reinforcement learning, including sensing actions, has been proposed (Whitehead and Ballard, 1990; MaCallum, 1996; Cassandra et al., 1996; Miyazaki and Kobayashi, 1998; Busquets et al., 2002). Theoretically, actions directly related to both the task and sensing are optimally selected for the task completion. The most serious drawback of these the methods is the amount of time needed for the learning (Whitehead, 1991).

In this paper, we propose a method for a robot, which is equipped with a limited viewing angle camera that has panning and tilting facility, to make a decision without explicitly localizing itself. The premise of our observation strategy is not for self localization but for decision making, that is, to minimize observation as much as possible in decision making. A decision tree and prediction trees are constructed based on the entropy of the action decision or the information gain from training data in a view-based approach. That means the robot does not need to explicitly localize itself. By constructing a decision tree, the robot knows which landmark to observe first. By using prediction trees based on information criterion, less time for observation through decision making will be needed.

The rest of the paper is organized as follows: We first describe the task and assumptions, then we describe the proposed method. Next, we show the experimental results with a real robot. Finally we discuss the future issues and draw conclusions.

2. Task and assumptions

We assume the following for the robot, environment, and given data:

1) The viewing angle of the robot's camera is limited and the robot cannot always acquire sufficient information to decide immediately what to do.

2) There is a sufficient number of landmarks to be observed in order to decide what to do at any location, and the robot can acquire sufficient information by panning its camera.

3) The landmark to be observed for decision making depends on the situation.

Figure 1. A legged robot for the RoboCup 99 Sony legged robot league.

4) The action and the information are quantized to construct a decision tree, and sufficient training data to calculate information gain and to construct the tree is prepared. We use a direct teaching method to collect this kind of training data. Although the proposed method can generally handle other kinds of sensors including logical ones, in the following sections we only deal with landmark observations as sensor information for the simplicity of explanation.

We use a legged robot with a limited viewing angle camera which was originally used for the RoboCup 99 Sony legged robot league (Figure 1). The camera of the robot is embeded in its nose. The viewing angles of the camera are about 53 degrees in width and 41 degrees in height. The images are 88 pixels in width and 59 pixels in height. Each leg and the neck have three degrees of freedom. The robot stands still and only rotates the pan joints of the neck when it observes the landmarks in order to make a decision. Additionally it rotates its tilt joint for decision making when it observes the ball. The robot can rotate the pan joint from -90 degrees to 90 degrees and the tilt joint from -90 degrees to 10 degrees.

The experimental field is shown in Figure 2. In the field, there are eight landmarks and a ball. The landmarks are the target goal (TG), the robot's own goal (OG), the northwest pole (NW), the northeast pole (NE), the center west pole (CW), the center east pole (CE), the southwest pole (SW), and the southeast pole (SE). All of the landmarks and the ball are distinguished by their colors. The task is to push the ball into the target goal based on the visual information. The robot has to take an appropriate action according to its location and its position relative to the ball, for example, searching for, approaching, or turning around the ball.

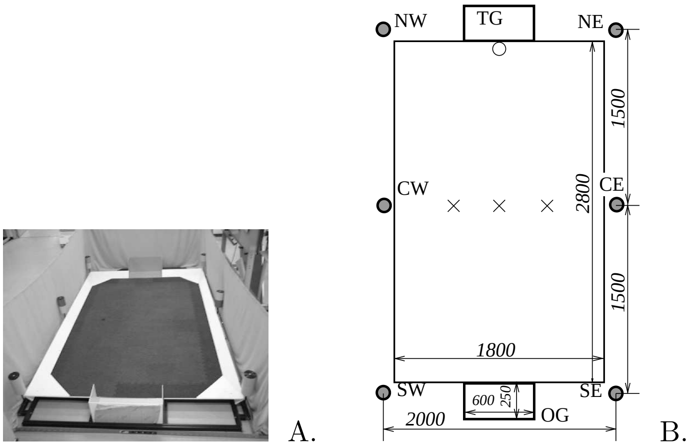

Figure 2. A) Photo of the experimented field and B) size of the field (same as the one for the RoboCup 99 Sony legged robot league). Cross and circle marks are for the first experiment.

3. The method

In the following section, we show that the uncertainty of deciding what to do is quickly reduced by using information gain to decide which landmark to observe. Next, we show that when the action decision tree is constructed by a classifier tree method based on the information gain criterion, the order of landmarks in the tree is the same as the order of observation. Then, we show how a robot makes decisions and calculates action and observation probabilities.

The observation strategy we propose here is to observe landmarks, starting from the landmark of highest information gain until one of the action probabilities exceeds a certain threshold. Here, the action probability is the probability that the action should be taken. The observation probability is the probability that the landmark is observed in one direction. We calculate observation probabilities based on the previous action and observations. For the calculation we use observation prediction trees constructed by the same method used for the construction of the action decision tree.

3.1. Information gain as an observation criterion

Suppose we have m landmarks. The observation of the landmark is quantized into q kinds of viewing categories including a non-visible situation, and r kinds of actions. A training datum consists of a set of observations of the landmarks at the current position and the action to accomplish the task, and we have n training data points. The occurrence probability of the k-th action p_k is given by

$$p_k = \frac{n_k}{n}, \tag{1}$$

where n_k denotes the number of actions k in the training data. Therefore, the entropy H_0 for the action probability is given by

$$H_0 = -\sum_k p_k \log_2 p_k. \tag{2}$$

The entropy increases when the action decision is more ambiguous. It becomes the highest when all p_ks are equal to each other, that is $p_k = 1/n$ and becomes zero when one of the p_k equals one. Next, we calculate the occurrence probabilities of actions after the observation of each landmark. We denote the number of times action k was taken as n_{ijk} when the landmark i is observed as the viewing category j. The occurrence probability p_{ijk} of action k becomes,

$$p_{ijk} = \frac{n_{ijk}}{\sum_k n_{ijk}}. \tag{3}$$

Here, the entropy of the action after observation is

$$h_{ij} = -\sum_k (p_{ijk} \log_2 p_{ijk}). \tag{4}$$

The h_{ij} decreases when the ambiguity becomes lower. The probability p_{ij} that the landmark i is observed as viewing category j is given by

$$p_{ij} = \frac{\sum_k n_{ijk}}{\sum_j \sum_k n_{ijk}}. \tag{5}$$

Then, the entropy H_i after the direction of the landmark i is known is

$$H_i = -\sum_j p_{ij} h_{ij}, \tag{6}$$

and the information gained by observing the landmark i is

$$I_i = H_0 - H_i. \tag{7}$$

The I_i is higher if the observation reduces ambiguity more. By observing the landmarks in decreasing order of information gain, we can quickly reduce the ambiguity and determine the action to take.

3.2. Construction of action decision and prediction trees

ID3 (Quinlan, 1979) is an algorithm to build a decision tree. To construct a decision tree, we prepare a training data set in which each datum has the discrete attribute values that are classified in advance. We calculate each information gain I_i to identify the class when the attribute i becomes known. Then, we divide the data set by the attribute with the largest information gain. This is a sort of branching in terms of the attribute values in the decision tree. We continue the division and branching by the information gain until each sub-data set includes one class.

Table I shows an example training data set. There are three attributes A, B, and C, and each takes one of three values α, β, or γ. The x, y, and z are the three classes to identify. Then, the entropy for the class identification is

$$\begin{aligned} H_0 &= -\sum_k p_k \log_2 p_k \\ &= -\frac{2}{4}\log_2\frac{2}{4} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} \\ &= \frac{3}{2}. \end{aligned} \tag{8}$$

Information gain for class identification by the attribute A is calculated by

$$\begin{aligned} H_A &= p_{\{A=\alpha\}}h_{\{A=\alpha\}} + p_{\{A=\beta\}}h_{\{A=\beta\}} \\ &= \frac{2}{4}\{-p_{\{x|A=\alpha\}}\log_2 p_{\{x|A=\alpha\}}\} \\ &\quad +\frac{2}{4}\{-p_{\{y|A=\beta\}}\log_2 p_{\{y|A=\beta\}} - p_{\{z|A=\beta\}}\log_2 p_{\{z|A=\beta\}} \\ &= \frac{1}{2}, \\ I_A &= H_A - H_0 = 1. \end{aligned} \tag{9}$$

Information gains for class identification by attribute B and C are 0.3 and 0.5, respectively. Then we divide the data set by the value of attribute A into a) data 1 and 2 and b) data 3 and 4. Both the data 1 and 2 have class x and the entropy is 0. The data subset with numbers 3 and 4 has entropy 1.0 for class identification and the information gains are 0.0 by the attributes A and B, and 1.0 by the attribute C. We divide the data set by the attribute C and the construction of the decision tree ends. Figure 3 shows the final decision tree for class identification.

We build an action decision tree regarding an action as a class and an observed direction of each landmark as an attribute. Table II shows

Table I. An example training data for ID3. The α, β, or γ indicates the value of the attribute for the datum. The x, y, or z indicates the class to be identified.

No.	attribute A	attribute B	attribute C	class
1	α	β	α	x
2	α	α	β	x
3	β	α	α	y
4	β	α	β	z

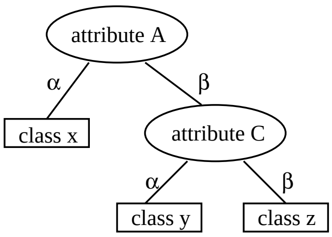

Figure 3. The example classification tree by ID3. Circles indicate branches by the attributes. Boxes indicate the identified classes.

an example data set. There are three landmarks A, B, and C. Each attribute is the direction α, β, or γ in which the landmark is observed. The x, y or z indicates the action to take for each observation. Since the probabilities before observation $p_x = 0.2$, $p_y = 0.4$, and $p_z = 0.4$, H_0 is 1.52. By calculating each p_{ijk}, we have expected information gain from observation of each landmark for the action decision, $I_A = 0.97$, $I_B = 0.72$, and $I_C = .52$. The ordering in the decision tree is landmark A, B, then C, and the tree is shown in Figure 4. At each branch, the data set is divided by the observed direction of the landmark. Although the original ID3 method iterates the calculation of the information gain at each branch and the orders of the landmark will be different in different sub-trees, we fixed the order in the tree for all the sub-trees to save memory consumption and to simplify the probability calculations.

In the action decision tree, the branches are located in decreasing order of information gain from the root to the leaves. Then, observation of the landmarks at the branches from the root to the leaves means that the observations are made in the order of importance based on information gain.

In an experiment with real robots, sometimes appropriate actions are different although the observations are the same because of the quantization of the observing directions. In such cases, for each action

Table II. An example of training data. α, β, or γ indicates the observed direction of each landmark. x, y, or z indicates the action to take for the observation.

Landmark A	Landmark B	Landmark C	Action
α	α	α	x
β	α	α	y
γ	β	α	y
α	γ	α	z
α	β	β	z

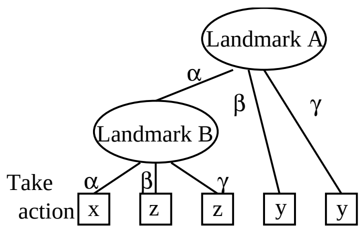

Figure 4. An example of an action decision tree. The ellipses indicate the observation of the landmark. α, β, or γ indicates the observed direction of each landmark. x, y, or z in a box indicates the action to take for the observation.

which has the same observation, we count the number of the training data, prepare a leaf, and record the ratio in the training data of the same observation. We regard the ratio as the probability that the action should be taken when the same observation is given.

We construct the landmark observation prediction trees in the same manner. To build a prediction tree for landmark i, we regard the next observed direction of landmark i (α, β, or γ for the example) as a class, and currently observed directions of landmarks and the action as attributes. Figure 5 shows the prediction tree for landmark A constructed from the training data set in Table II.

3.3. Decision making

The robot iterates observations and actions as shown in Figure 6. The robot determines its action based on the observation probabilities. The observation probabilities are calculated from the observation at the previous position (step 1) and observation at the current position (step 3). If one of the action probabilities exceeds a certain threshold, the robot takes that action (steps 5 and 8). Otherwise, until one of them exceeds the threshold, it continues to try to observe the landmark whose observation prediction probability distribution is uniform (large entropy in the observation probabilities). The landmarks are selected

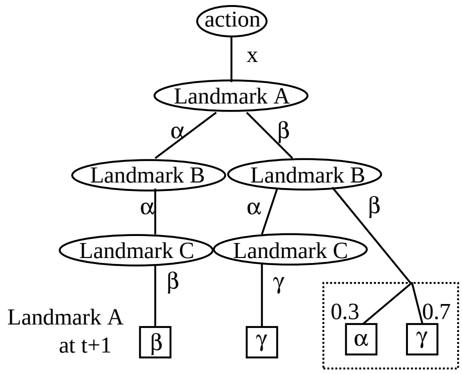

Figure 5. An example of a prediction tree for the landmark A. Ellipses indicate branches by observed directions of the landmarks. Boxes indicate the predicted direction of the landmark A at next time step.

in the order in which they were placed in the action decision tree; that is, the landmark is chosen by the information criterion. When the robot checks the landmark, observation of the peak direction of the probability profile may speed faster decision making. Otherwise it uses some fixed strategy. In the following experiments, we prepared the observation from left to right as the fixed strategy.

3.4. Calculation of the probability distribution

Here, we denote the probability that the landmark i was observed in the viewing direction j at time t as $p^L_{ij}(t)$ $(i = 1, ..., m, j = 1, ..., q)$, the probability that the action k was taken at time t as $p^a_k(t)$ $(k = 1, ..., r)$, and the probability that the action k should be taken by the training data at t as $\hat{p}^a_k(t)$ $(k = 1, ..., r)$.

Calculations of the probability distributions are as follows: If the landmark i is currently observed in the quantized viewing direction J, we assign the probability 1.0 to $p^L_{iJ}(t)$ and 0 to the others ($p^L_{ij}(t) = 0$ $(j \neq J)$). When the previously taken action was K, set $p^a_K(t-1) = 1$ and $p^a_k(t-1) = 0$ $(k \neq K)$. The probabilities of the landmarks, which are not currently observed, are calculated by the prediction trees using the probability distributions $p^L_{ij}(t-1)$ and $p^a_k(t-1)$. We assign the probability 1.0 to the quantized invisible direction and 0 to the others if the landmark cannot be observed even though the robot has looked for it.

Following a landmark prediction tree from the root to one of the leaves gives, 1) the condition of the landmark 's directions and the action at time $(t-1)$ which is given by logical product; and 2) the consequent landmark direction at time t. To calculate the probability to

1. Calculate the observation probabilities of landmarks by the prediction trees from the observation probabilities at the previous position.
2. Observe the landmarks which are seen in currently observing direction.
3. Update the observation probabilities.
4. Calculate the action probabilities $\hat{p}^a_k(t)$ from the observation probabilities.
5. If one of the action probabilities $\hat{p}^a_k(t)$ exceeds threshold, it is the determined action, then go to step 8.
6. Determine the next landmark to observe. The landmark,
   a) is not observed at current position, but
   b) has higher information gain (or near to the root), and
   c) its observation probabilities are equally large in two or more directions (large entropy in the observation probabilities).
7. Change the observing direction to search for the next landmark and go to step 2.
8. Take the determined action and change the observing direction to watch the landmark on the root of the decision tree (that is the ball in following experiments).
9. Stop locomotion.
10. Go to step 1.

Figure 6. The pseudo code for a robot to make decisions

reach each leaf, we change the logical product to an arithmetic one and conditions to probabilities at time $(t-1)$. We consider the summation of the probabilities of the leaves of the same direction as the probability of the direction at time t. For example, if the prediction tree of landmark A is the tree shown in Figure 5, then

$$\begin{aligned} p^L_{A\alpha}(t) &= p^a_x(t-1)p^L_{A\beta}(t-1)p^L_{B\beta}(t-1) \times 0.3, \\ p^L_{A\beta}(t) &= p^a_x(t-1)p^L_{A\alpha}(t-1)p^L_{B\alpha}(t-1)p^L_{C\beta}(t-1), \\ p^L_{A\gamma}(t) &= p^a_x(t-1)p^L_{A\beta}(t-1)p^L_{B\alpha}(t-1)p^L_{C\gamma}(t-1), \\ &\quad + p^a_x(t-1)p^L_{A\beta}(t-1)p^L_{B\beta}(t-1) \times 0.7. \end{aligned} \tag{10}$$

To calculate the action decision probability $\hat{p}^a_k(t)$, we use these probability distributions $p^L_{ij}(t)$, and follow the action decision tree in the same manner.

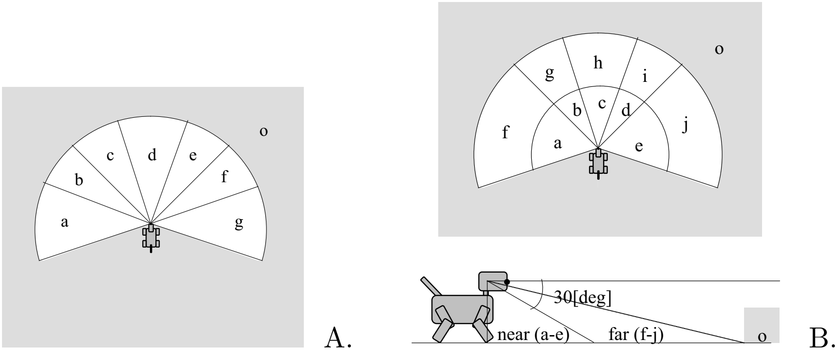

Figure 7. The quantization for A) landmarks and B) the ball. The shaded region means 'o' non-observable region.

4. Experimental results

Each landmark's observation is quantized into eight directions: $(-90°, -65°)$, $[-65°, -40°)$, $[-40°, -15°)$, $[-15°, 15°)$, $[15°, 40°)$, $[40°, 65°)$, $[65°, 90°)$, and one invisible situation which is denoted 'o' owing to the limited panning angle or because it is too far to observe (see Figure 7). The ball observation is quantized into eleven directions: the product of five pan angles, $(-90°, -45°)$, $[-45°, -12°)$, $[-12°, 12°)$, $[12°, 45°)$, $[45°, 90°)$, and two kinds of distances (near or far, divided at $30°$ below the level), and one invisible situation (denoted as 'o'). We treat the ball as a special landmark, which does not move unless the robot pushes it. Therefore, the observation of the ball can be predicted by its previous location in the image and the action of the robot. Note that for the ball prediction tree, we cannot use other landmarks because it may move in the field.

We have collected the training data by directly sending action commands through a PC connected to the robot. During the teaching phase, the robot was placed in the field and we repeated the following:

1. the robot looks around to observe the direction of the landmarks and the ball,
2. a human trainer orders the robot to take a specific action,
3. the robot performs that action and stop.

If the trainer made a mistake, the datum was marked and removed from the data set for construction of the action decision tree. The observation of the robot includes noise caused by the image processing. The maximum distance from which the robot is able to observe each landmark (both the ball and the landmarks except for the goals) is about 1.8[m], and it changes slightly depending on the lighting conditions. So there were cases in which a landmark had not been observed from almost the same position.

When it is running and making decisions with the trees, the robot searches for landmarks and the ball (if they are invisible) when the peak action probability is below 0.60. It watches or searches for the ball during both teaching and tree-based decision making except for landmark search periods.

Since some situations may not be represented in the training data, the sum of the probability distribution $\sum_{i=1}^{N} p_i$ might be less than one. To avoid this, we added $(1 - \sum_{i=1}^{N} p_i)/N$ to each p_i. N indicates the size of the distribution.

We conducted two experiments. One is a test with a small set of teaching data, to see how the robot makes decisions. The second is to see how the trees change with a larger set of teaching data. We used the trees in the games of the RoboCup 99.

4.1. Experiment 1

In the field (Figure 2), we experimented with the task of pushing a ball into the goal. The ball was placed in front of the goal (the circle before the target goal in the figure) and the robot was put at one of three cross marks in the middle of the field at the start of each trial. We prepared three kinds of actions: forward, left-forward, and right-forward walking based on the trot gait for Experiment 1. Each action consists of four walking cycles of 4.8 seconds. With the forward action, the robot walks about 0.45[m] in 4.8 seconds. We chose a distance of about 1.5 times the size of the robot's length (about 0.3[m]) for one action. Also the observation (discrete directions of the landmarks) changes at most places with one action.

For each starting position, we trained five times and obtained eighty data points to construct trees. The robot followed the same tracks for both the teaching phase and the phase when the robot makes decisions by the trees. We show part of the action decision tree in Figure 8. If the ball is observed in the left front direction (the direction b), the action changes according to the direction of the target goal (the direction c, d, or f). Table III shows the sizes of the action decision tree and prediction trees. The size of the action decision tree is as follows: the number of leaves is 43, the minimum depth is 1, the mean depth is 4.91, and the maximum depth is 8. The sizes of the prediction trees for landmarks SW and SE become small since most of them are behind the robot and not observed so much. Tables IV and V show the orders of the landmarks according to the information gain in the action decision

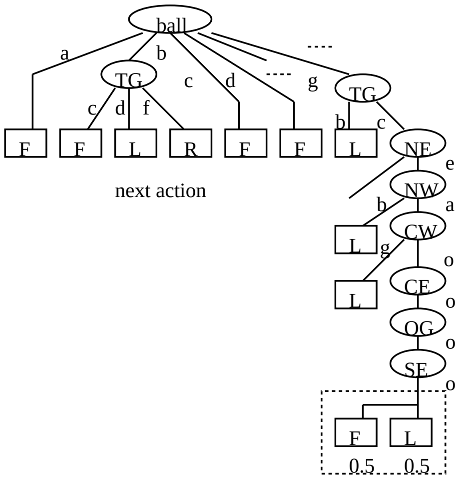

Figure 8. Part of the action decision tree (Experiment 1). F, L, and R mean forward, left forward, and right forward, respectively.

Table III. Depth of the prediction trees (Experiment 1).

tree for	# of leaves	min dep.	mean dep.	max dep.
ball	52	2	2	2
OG	13	1	4.23	8
TG	44	1	5.39	8
SE	6	1	2	3
SW	1	0	0	0
CE	28	2	4.69	8
CW	11	1	3.91	8
NE	51	1	5.96	8
NW	54	2	5.91	8

tree and the prediction trees. The number 1 indicates the root of a tree and the largest information gain, while 8 indicates the leaf of the tree and the least information gain. In Tables IV and V, "ball" and "act" indicate the previous direction of the ball and the previous action, TG, OG, NW, NE, CW, CE, SW, and SE indicate the previous direction of the landmarks, respectively. We see that the ball is at the root of the action decision tree and the previous action is near the root of the prediction trees. We show a sequence of the robot's actions with these trees in Figure 9.

Table IV. The order of information for the action decision tree (Experiment 1).

1	2	3	4	5	6	7	8
ball	TG	NE	NW	CW	CE	OG	SE

Table V. The order of information for prediction trees (Experiment 1).

tree for	1	2	3	4	5	6	7	8
ball	ball	act						
OG	act	NE	TG	NW	CW	CE	OG	SE
TG	TG	act	NE	NW	CE	OG	CW	SE
SE	act	CE	NE	OG	NW	TG	CW	SE
SW	-							
CE	act	NE	TG	CE	NW	CW	OG	SE
CW	TG	act	NE	NW	CE	CW	OG	SE
NE	NE	act	NW	TG	CE	CW	OG	SE
NW	act	NE	TG	NW	CE	OG	SE	CW

Next, we show examples of action sequences using these trees. From the starting position in the center of the field, the robot selected the forward action four times. In this experiment, the ball and the target goal were observed at every moment for decision making. The probabil-

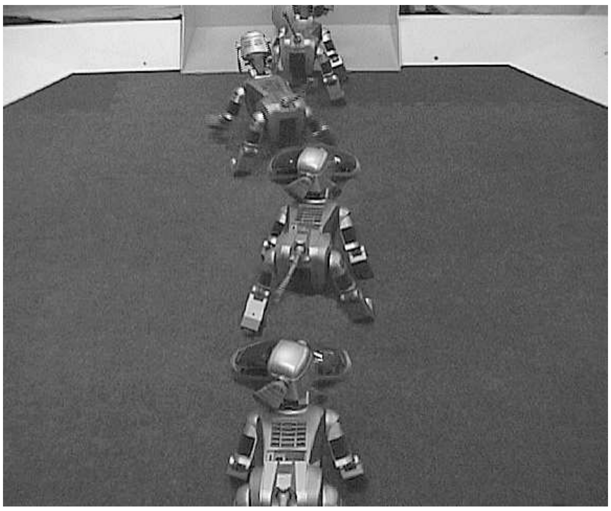

Figure 9. Robot actions with the action and decision trees.

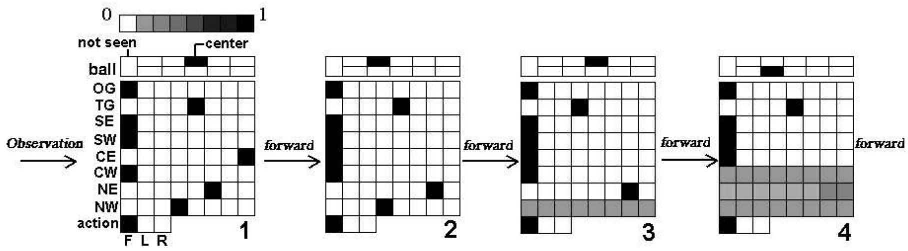

Figure 10. Probability distributions in Experiment 1-1 (The gray level of each box indicates the probability; black is 1 and white is 0) .

ity distributions of the landmark observation predictions and the action decision are shown in Figure 10. The intensity of each square indicates the probability. A white square means the probability is 0 while a black one means 1. The row of 'ball' in the figure indicates the observation probabilities of the ball directions by the observation or the prediction by the tree. The rows of OG, TG, SE, SW, CE, CW, NE, NW indicate the observation probabilities of the landmarks. The column of 'not seen' indicates the observation probability that the ball (or a landmark) can not be observed (direction 'o') even if the robot pans its camera. Other squares in the ball row indicate the probabilities of the directions, a, b,... , e (the lower row) and f, g,... , j (the upper row) from left to right. The squares in the landmark rows indicate the probabilities of directions a, b,... , g from left to right. The row of 'action' indicates the action probability and F, L, and R indicate forward, left forward, and right forward, respectively. The robot pans its camera at the beginning of each sequence. The first probability boxes are the ones of the first decision after the robot panned the camera. From Figure 10, we can see that at the time of the third and the fourth decisions, there was some ambiguity in the observation probabilities, but the robot could make decisions without panning its camera because one of the action probabilities exceeded the threshold.

The robot may take different actions from the same starting position. This is owing to the quantization of the observation, slight change of the lighting condition, and variance of the walking. In this example, the robot started from the center cross mark on the field as in the first example, but then took other actions instead i) forward; ii) forward; iii) landmark observation; iv) forward, v) landmark observation; vi) left forward; vii) forward; and viii) forward. In this experiment, the ball and the target goal were observed at every moment for decision making. The probability distributions of landmark observation predictions and the action decisions are shown in Figure 11. At the time of the third and the fifth decisions, action probabilities became uniform and the robot

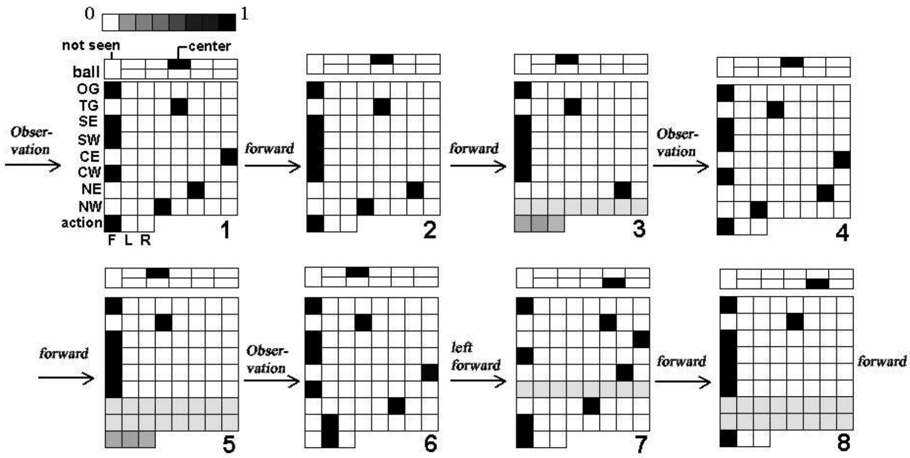

Figure 11. Probability distributions in Experiment 1-2.

had to pan its neck to observe landmarks. However, at the time of the seventh and the eighth decisions, there was some uncertainty in observation probabilities, but the robot could make decisions without additional observation.

Here is another example starting from the right cross mark on the field. In this example, the robot took the actions, i) left forward; ii) landmark observation; iii) forward; iv) forward; and v) left forward. The ball and the target goal were observed at every moment for decision making. The probability distributions of landmark observation prediction and the action decision are shown in Figure 12. At the time of the second decision, action probabilities became uniform, but at the time of the fourth and the fifth decisions, the robot could make decisions even though there was some ambiguity in observation probabilities.

We show the number of re-observations in Table VI. Each number indicates the number of trials, total number of steps, re-observations, and the rate of re-observations. We see that the number of re-observation is about half the number of total steps. This means that with the proposed method the robot does not need to stop-and-observe at every step but only half as often.

4.2. Experiment 2

We trained the robot on the same field in which the Robocop 99 games were held. In this experiment, we placed the robot and the ball at many more locations than we did in Experiment 1. To reduce the load of teaching, we prepared six actions: forward; left forward; right forward; left rotation; right rotation; and track the ball. Each action

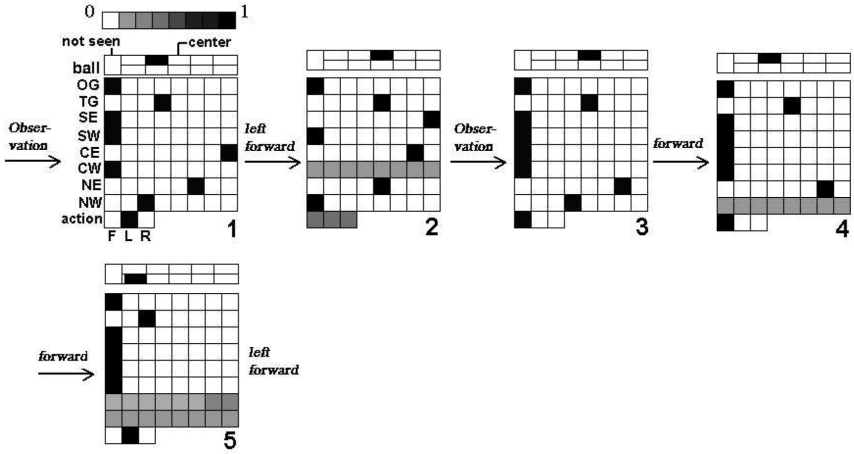

Figure 12. Probability distributions in Experiment 1-3.

Table VI. The number of re-observations (Experiment 1).

begin from	# of trials	# of total steps	# of re-obser-vations	rate of re-observation
center	12	35	18	.51
left	12	31	15	.48
right	16	64	38	.59

also consisted of four walking cycles, of 4.8-seconds. We also assumed that, in the teaching phase, the ball would not move unless the robot touched it. Although the ball will be moved by teammates or opponents, we assumed that the robot knows the ball direction without prediction since the robot is watching the ball.

We obtained 1364 data points by teaching. Deleting inappropriate data, we used 856 data points for the action decision tree and 1364 data points for the ball and landmark predictions. We show the sizes and the order of the trees based on the information criterion in Tables VII through X. We see that the difference in the depth among the prediction trees is reduced, the top two landmarks in the action decision tree remain as in Experiment 1, and most landmarks on top of the prediction trees are the predicted landmarks themselves.

We used these trees for the games in RoboCup 99. We prepared default actions for the cases when the robot cannot determine its action even it searched all directions for the landmarks. The robot showed actions that were expected from the teaching, though the robot looked for the landmarks more frequently as in Experiment 1-2 than expected

Table VII. Depth and size of the action decision tree (Experiment 2).

# of leaves	min dep.	mean dep.	max dep.
586	2	5.89	9

Table VIII. Depth and size of the prediction trees (Experiment 2).

	# of leaves	min dep.	mean dep.	max dep.
ball	403	2	2	2
OG	958	2	7.58	9
TG	1050	2	7.67	9
SE	845	2	7.35	9
SW	901	2	7.41	9
CE	901	2	7.13	9
CW	873	2	7.37	9
NE	1031	2	7.60	9
NW	980	2	7.55	9

from Experiment 1-1. The reason for this could be that the prediction trees did not work well. We discuss this in the next section.

5. Discussion

From the results of Experiment 1 (Figures 10-12), we see that the action probability distributions have either a sharp peak (nearly equal to 1.0) or a very uniform profile. When the action probability distribution is uniform, one or more of the landmark probability distributions are also uniform. Although the robot looked all around for the landmarks and the ball because of the uniform distribution of observation probabilities, the length of observation time might be reduced with the strategy to

Table IX. The order of information for the action decision tree (Experiment 2).

1	2	3	4	5	6	7	8	9
ball	TG	OG	SW	SE	NW	NE	CE	CW

Table X. The order of information for prediction trees (Experiment 2).

tree for	1	2	3	4	5	6	7	8	9
ball	ball	act							
OG	OG	SE	SW	TG	NW	CW	NE	CE	act
TG	TG	OG	SE	SW	NW	NE	CW	CE	act
SE	SE	OG	TG	SW	CE	NE	NW	CW	act
SW	SW	OG	CW	SE	TG	NW	NE	CE	act
CE	CE	SE	OG	TG	NE	SW	NW	act	CW
CW	CW	SW	OG	TG	NW	SE	NE	CE	act
NE	TG	NE	OG	SE	CE	NW	SW	CW	act
NW	NW	TG	OG	SW	CW	SE	NE	CE	act

observe the direction of a peak in the distribution if the distribution has one.

We have prepared simulated observations for the robot's and ball's positions at every 150[mm] in the field (area of 1500[mm]×2400[mm] in the center of the field as shown in Figure 13) and the robot's pose of every 15 degrees. Each observation is simulated as if the robot has observed all the directions. Excluding the data in which the robot and the ball share the same position, we obtained 834768 simulated observations. The robot could make decisions for 323054 observations (39%) with the action decision tree constructed in Experiment 2 with 856 teaching data points. The robot could not determine its actions for the other observations since the observations did not lead to any of the leaves in the tree. Then all of the probabilities go to zero and the same values are added for the compensation, which leads to the uniform distribution of the probabilities.

Figure 14 shows the proportion of the data points according to the number of the landmarks needed for the action decisions in 323054 observations. In the figure, the ball is also counted as a landmark. In the observations, the robot did not need 8 or 9 landmarks (including the ball) since the simulation did not include such observations. From the figure, we can see that more than half of the action decisions can be made with the directions of the ball and one or two landmarks. It suggests that the mobile robot does not need to track its exact position. This strongly supports our idea that the observation should be done for the action decision and an action decision does not always have to rely on the self-localization.

Figure 15 shows the accuracy of the predictions by the trees in 834768 observations. From this we can see that there was some bias in the teaching data. For example, we focused on shooting the ball

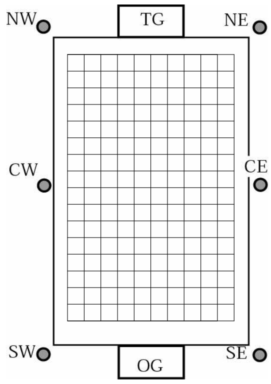

Figure 13. Simulated observations are prepared at each intersection shown in the figure for 12 postures.

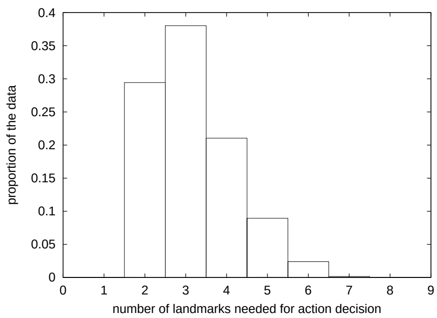

Figure 14. The proportion of the set of the position of the robot and the ball and the posture of the robot according to the number of the landmarks needed for the action decisions in 323054 observations. Note, the ball is also counted as a landmark.

to the target goal and not on defending. This is clearly seen in the differences between the prediction performances of TG and OG.

Comparing the order of action and landmarks in prediction trees between Experiments 1 and 2, we notice that the action has a higher priority in Experiment 1 than in Experiment 2. From Table V, we can see that the predictions of the landmark directions generally depend on the previous direction of the other landmarks in Experiment 1. On the other hand, Table X shows that the predictions depend on the previous direction of that landmark itself. This suggests that the number of the training data was too small to extract the fact that prediction of landmarks highly depends on their locations, as shown in Experiment 1. Note that although the order is different in both cases,

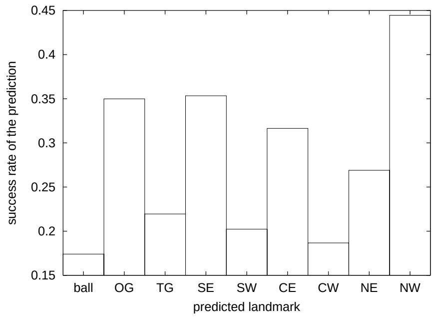

Figure 15. The success rate of the prediction by the trees in 834768 observations.

they are extracted from the training data by the information criterion. Therefore, they both are optimal in that sense and the landmark prediction can be made for the observations in the training data.

Since our method presumes the discrete observations, it does not consider the continuity of the observations. Therefore the constructed trees do not have enough generalization abilities even with the teaching data in Experiment 2. For the observations which are consecutively taken, the prediction performance can be better with some other methods like an Extended Kalman Filter method (Wan and Merwe, 2000) although we have not dealt with such cases in this paper.

We can expect generalization by the compression of the trees. However, with generalization, there may be an increase in the number of cases when the robot does not confirm the landmark directions even when it should confirm them. The order in the tree, which is fixed in the method for memory consumption and simplicity, might be changed to ID3 or C4.5 for further abstraction and observation efficiency. In this paper, we quantized the sensor space by hand. However, quantization may be self-organized during tree construction by using a method like C4.5 (Quinlan, 1993) which automatically divides data with continuous attributes according to the information gain by the division for construction of a decision tree.

6. Conclusions

We have proposed an observation strategy for a mobile robot to decide what to do based on information criterion. An action decision tree is constructed from the training data based on the information gain, and the observation strategy is involved in the tree representation. We have confirmed our method through experiments with a real robot. We have experimented with a robot equipped with a vision sensor to observe landmarks; however, the method can be successfully applied to robots with other types of sensors.

The method answered the question of whether to look or to move, and it can be one solution to decide what to look for. However, other questions remain: how to define the length of one action, and when to confirm the locations of landmarks. We have designed the system so that the robot watches the ball while walking, but the target to watch should be determined by the information gain. In our method the robot first observes and then determines its action. Since 1) the walking we used was not stable enough for the robot to use the landmark observation for the decision; and 2) the robot could not observe all directions simultaneously because of the limited viewing angle camera, then the robot was not able to collect observations while walking. Action decision using observation while walking is a future issue. Automatic determination of the action probability threshold and an efficient method to prepare the training data set are also our future work.

Acknowlegement

This research was supported by the Japan Science and Technology Corporation's Core Research for the Evolutional Science and Technology Program (CREST) titled Robot Brain Project. We thank Dr. Karl F. MacDorman for reading the manuscript and providing us with his useful suggestions.

References

D. Busquets, R. L. de Mantaras, C. Sierra, and T. G. Dietterich. Reinforcement learning for landmark-based robot navigation. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, 2002.

A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile robot navigation. In *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 963–972, 1996.

K. Miyazaki and S. Kobayashi. Learning deterministic policies in partially observable Markov decision processes. In Y. Kakazu, M. Wada, and T. Sato, editors, *In Proceedings of the Intelligent Autonomous Systems 5*, pp. 250–257, 1998.

D. Fox, W. Burgard, and S. Thrun. Active Markov localization for mobile robots. In *Robotics and Autonomous Systems*, Vol. 25, pp. 195–207, 1998.

S. Kristensen. Sensor planning with bayesian decision theory. In *Robotics and Autonomous Systems*, Vol. 19, pp. 273–286, 1997.

R. A. McCallum. Hidden state and reinforcement learning with instance-based state identification. *IEEE Transaction on System, Man and Cybernetics Part B*, 26(3):464–474, 1996.

L. Mihaylova, T. Lefebvre, H. Bruyninckx, K. Gadeyne, and J. D. Schutter. Active sensing for robotics – a survey. In *Proceedings of the 5the International Conference On Numerical Methods and Applications*, 2002.

I. H. Moon, J. Miura, and Y. Shirai. On-line viewpoint and motion planning for efficient visual navigation under uncertainty. *Robotics and Autonomous Systems*, 28(2–3):237–248, 1999.

N. Roy, W. Burgard, D. Fox, and S. Thrun. Coastal navigation: Robot navigation under uncertainty in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999.

Y. Sakaguchi. Haptic sensing system with active perception. *Advanced Robotics*, Vol. 8, No. 3, pp. 263–283, 1994.

H. Wang, K. Yao, G. Pottie, and D. Estrin. Entropy-based sensor selection heuristic for target localization, In *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, pp. 36–45, 2004.

E. A. Wan and R. van der Merwe. The unscented Kalman filter for nonlinear estimation, In *Proccedigns of the Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000*, 2000.

J. R. Quinlan. Discovering rules from large collections of examples: A case study. In D. Michie ed., *Expert Systems in the Microelectronic Age*. University Press, 1979.

J. R. Quinlan. *C4.5: Programs for machine learning.* Morgan Kaufmann Publishers, 1993.

J. Tani, J. Yamamoto, and H. Nishi. Dynamical interactions between learning, visual attention, and behavior: An experiment with a vision-based mobile robot. In P. Husbands and I. Harvey eds., *Fourth European Conference on Artificial Life*, pp. 309–317. The MIT Press, 1997.

S. D. Whitehead and D. H. Ballard. Active perception and reinforcement learning. In *Proceedings of the Seventh International Conference on Machine Learning*, pp. 179–188. Morgan Kaufmann, 1990.

S. D. Whitehead. A complexity analysis of cooperative mechanisms in reinforcement learning. In *Proceedings of AAAI-91*, pp. 607–613, 1991.

P. Jensfelt, D. Austin, and H. I. Christensen. Towards task oriented localization. In *Proceedings of the Intelligent Autonomous Systems 6*, pp. 612–619. IOS Press, 2000.

S. A. Hutchinson and A. C. Kak. Planning sensing strategies in a robot work cell with multi-sensor capabilities. *IEEE Transactions on Robotics and automation*, Vol. 5, No. 6, pp. 765–783, 1989.

Address for Offprints: Emergent Robotics Area, Department of Adaptive Machine Systems, Graduate School of Engineering, Osaka University, 2-1 Yamadaoka Suita, 565-0871, Japan