

# Observation strategy for decision making based on information criterion

Noriaki Mitsunaga and Minoru Asada

Dept. of Adaptive Machine Systems, Osaka University, Suita, Osaka, 565-0871, Japan

**Abstract.** Self localization seems necessary for mobile robot navigation. The conventional method such as geometric reconstruction from landmark observations is generally time-consuming and prone to errors. This paper proposes a method which constructs a decision tree and prediction trees of the landmark appearance that enable a mobile robot with a limited visual angle to observe efficiently and make decisions without global positioning in the environment. By constructing these trees based on information criterion, the robot can accomplish the given task efficiently. The validity of the method is shown with a four legged robot.

## 1 Introduction

When it makes a decision, a mobile robot highly depends on its own location. For self-localization, methods such as dead reckoning and global positioning by geometric reconstruction from image data are commonly used [1][2][3]. However, it seems difficult for a legged robot to use dead reckoning without special treatment since the moving distance on the walking plane is less accurate than that of a wheeled robot. For the latter, geometric reconstruction often highly depends on accurate motion information. Such information may include drastic errors in the case of legged robots due to causes such as slipping. Furthermore, it takes a lot of time not just for geometric reconstruction but also for observation to capture the image surrounding the robot in order to make geometric reconstruction stable and accurate. Human beings do not seem to take such a strategy to localize themselves. Rather, they use the minimum necessary information for their decision making when they do not have enough resources (time, etc.).

Moon et al. [4][5] have proposed a view point planning method to move efficiently by reducing the frequent observation for self-localization mostly based on dead reckoning. To apply their method, the map and the route to the goal should be given in advance.

Burgard et al. [6] have proposed an active localization method by Markov localization using occupancy grids as a world model representation. To apply their method, the map of the environment and the dead reckoning model need to be prepared. Also calculation and memorization costs for occupancy grids are very high. These methods depend on geometric reconstruction.

Tani et al. [7] have experimented with the task of watching two visual targets with a limited view angle camera. The robot switched the visual attention depending on prediction accuracy. Since the robot action (wall following) is fixed, the issue can be regarded as a view prediction problem on a route which is almost fixed.

In this paper, we propose a method for a robot, which has a limited view angle camera with panning facility, to make a decision by efficient observation without explicitly localizing itself. With a limited view camera, a robot can widen the angle by panning, but it takes time. Efficient observation is done by a decision tree and prediction trees constructed based on the information criterion. The basic idea of our observation strategy is not for self localization but for decision making, that is, to minimize observation as long as decision making is possible. By constructing a decision tree, one can know which landmark to observe first. Similarly by making and using prediction trees with information criterion, one can reduce the time for observation through decision making.

## 2 The method

### 2.1 Assumptions

We assume that the robot can make a decision for the given task at any position by panning its camera head. Before making the decision and prediction trees, sufficient example data are necessary. We used a teaching method to collect such data.

### 2.2 Making decision and prediction trees

Suppose we have  $m$  landmarks and  $q$  kinds of actions. Each appearance of the landmark is quantized into  $r$  kinds of viewing categories including a non-visible situation. A training datum consists of a set of the appearance of the landmarks at the current position and the action to accomplish the task, and we have  $n$  training data. During the training period, the robot pans its camera head from the left-most angle to the right most one, and observes as many landmarks as possible.

First, calculate the occurrence probabilities of actions  $p_k$  ( $k = 1, \dots, q$ ) as  $p_k = n_k/n$ , where  $n_k$  denotes the number of taken action  $k$ . Therefore, the information  $I_0$  for the action probability is given by

$$I_0 = - \sum_k p_k \log_2 p_k. \quad (1)$$

Next, calculate the occurrence probabilities of actions after each appearance of the landmark was known. We denote the number of times action  $k$  was taken as  $n_{ijk}$  when the landmark  $i$  is observed at the quantized direction  $j$ . Then, the occurrence probability becomes,

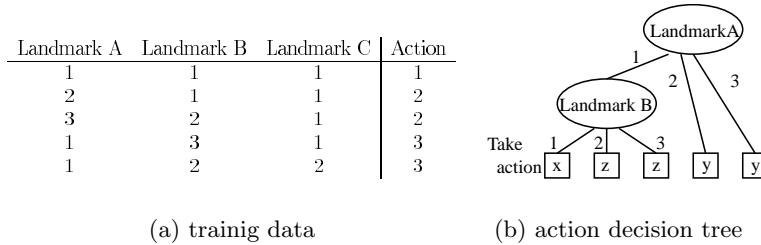
$$p_{ijk} = \frac{n_{ijk}}{\sum_k n_{ijk}}. \quad (2)$$

Next, calculate the expected information after one of these landmarks is found, as follows:

$$I_i = - \sum_j \left\{ \frac{\sum_k n_{ijk}}{\sum_j \sum_k n_{ijk}} \sum_k (p_{ijk} \log_2 p_{ijk}) \right\}. \quad (3)$$

The smaller  $I_i$  is, the smaller the uncertainty after the observation is. We put the landmarks into the tree in decreasing order of uncertainty after its observation. This information criterion is same as ID3 [8]. For the training data which take different actions for the same situation, we add a leaf for each action and record the probability that it was taken.

For example, suppose we have training data as shown in Fig.1(a). Since  $p_x = 0.2$ ,  $p_y = 0.4$ , and  $p_z = 0.4$ ,  $I_0$  is 1.52. By calculating each  $p_{ijk}$ , we have expected informations,  $I_A = 0.551$ ,  $I_B = 0.8$ , and  $I_C = 1.2$ . So the ordering in the decision tree is landmark A, B, and C, and the tree is shown in Fig.1(b). We calculate the landmark appearance prediction trees in the same manner.



**Fig. 1.** Examples of training data and an action decision tree.

### 2.3 Calculation of the probability distribution

Here, we denote the probability that the viewing direction of the landmark  $i$  is  $j$  at time  $t$  as  $p_{ij}^L(t)$  ( $i = 1, \dots, m, j = 1, \dots, r$ ), the probability that the action  $k$  was taken at time  $t$  as  $p_k^a(t)$  ( $k = 1, \dots, q$ ), and the probability that the action  $k$  should be taken by the training data at  $t$  as  $\hat{p}_k^a(t)$  ( $k = 1, \dots, q$ ).

Calculation of the probability distributions are as follows. If currently the landmark  $i$  is observed in the quantized viewing direction  $J$  we assign the probability 1.0 to  $p_{iJ}^L(t)$  and 0s to others ( $p_{ij}^L(t) = 0$  ( $j \neq J$ )). When the previous taken action was  $K$ , set  $p_K^a(t-1) = 1$  and  $p_k^a(t-1) = 0$  ( $k \neq K$ ). The probabilities of the landmarks, which are not currently observed, are predicted by the prediction trees using the probability distributions  $p_{ij}^L(t-1)$  and  $p_k^a(t-1)$ . We assign the probability 1.0 to the quantized invisible direction and 0s to others if the landmark cannot be observed while the robot looks around.

Following a landmark prediction tree from the root to one of the leaves, one can obtain 1) the condition of the appearances of the landmarks and the action at time  $(t-1)$  which is given by logical product, and 2) the consequence appearance of the landmark at time  $t$ . In order to calculate the probability of reaching 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 appearance as the probability of the appearance at time  $t$ .

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

## 2.4 Decision making

In order to make a decision on which action to take, the robot calculates the  $\hat{p}_k^a(t)$  as described above. If one of the action probabilities is very high, it takes that action. Otherwise, until one of them becomes high enough, it continues to try to observe the landmark whose probability distributions is flat in the order in which they were placed in the action decision tree (information criterion). When the robot checks the landmark, one may find a peak of the profile, and observation of that direction may help.

# 3 Experiments

## 3.1 Task and Assumptions

The task is to push a ball into a goal based on the visual information. We used a legged robot with a limited view angle for the RoboCup 99 SONY legged robot league (Fig.2). In the field, there are 8 landmarks, that is, target goal (TG), own goal (OG), north west pole (NW), north east pole (NE), center west pole (CW), center east pole (CE), south west pole (SW), and south east pole (SE). All the landmarks and the ball are distinguished by their colors. The view angle / number of image pixels of the robot's camera are about 53 degrees / 88 pixels in width, and about 41 degrees / 59 pixels in height. Each leg and the neck have three degrees of freedom. We fixed the joint angles of the legs and the neck except for the pan joint when it observes the landmarks and the ball to make its decision. The robot can rotate the pan joint from -90 degrees to 90 degrees.

The ball can be treated as a special landmark, which is static if the robot does not push it while it can change its position if the robot pushes it. Therefore, the appearance 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.

Each landmark's appearance is quantized into eight categories, that is, seven directions and one invisible situation (Fig.4). The ball appearance was quantized into eleven categories, that is, the product of five directions and two kinds of distances (near or far), and one invisible situation.



**Fig. 2.** The SONY legged robot for RoboCup 99 SONY legged robot league.

Since the training data may lack some situations, 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.

When it plays back the taught action based on the trees, the robot looks all around to search landmarks and the ball (if they are invisible) if the peak action probability was below 0.6 .

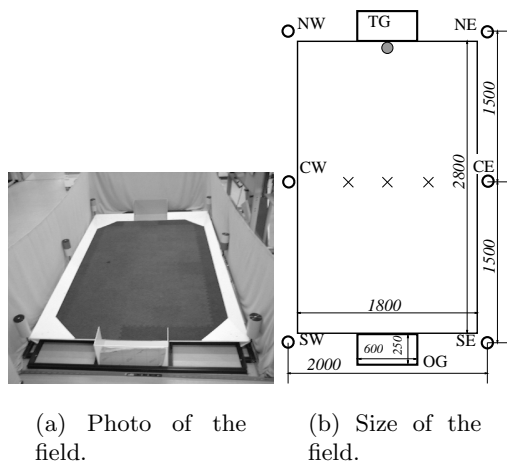
### 3.2 Experiment 1

In the field (Fig.3), we experimented with the task of guiding a ball into the goal. The ball was put in front of the goal (the circle before the target goal in the figure) and the robot was placed at middle of the field (one of three cross marks in the middle). As actions, we prepared three movements, forward, left forward and right forward for 4.8 seconds. The walking was programmed by us and a motion for 4.8 seconds corresponds to four walking periods. In 4.8 seconds, the robot walks about 0.45[m] in forward movement and at most places the appearance of the landmarks changes due to one of the motions.

We trained the robot starting from one of three positions in the middle of the field. For each starting position, we trained five times and obtained eighty data points to construct trees. We show the sizes (the number of leaves, minimum, mean, and maximum depth) of decision and prediction trees in Table 1. The orders in the trees by the information criterion are shown in Table 2.

Next, we show the examples of action sequences using these trees. From the starting position in the center of the field, the robot took the forward motions four times. 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 decision are shown in Fig.5.

The robot took different actions even if the starting position was the same. This is due to the quantization of the observation and variance of the walking. In this example, the robot started from the center of the field as in the first example, but then took other actions instead ( 1) forward, 2) forward, 3) landmark

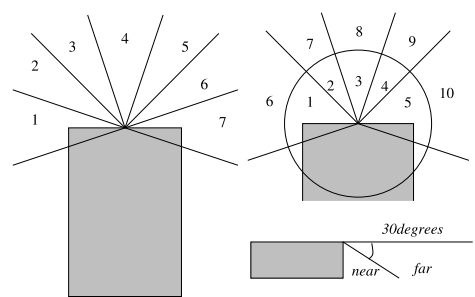


**Fig. 3.** Experimental field (same as the one for RoboCup SONY legged league). Cross marks indicate the initial positions of the robot and the ball for the first experiment.

observation, 4) forward, 5) landmark observation, 6) left forward, 7) forward, and 8) 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 decision are shown in Fig.6.

Here is another example starting from the center right cross mark. In this example, the robot took the actions, 1) left forward, 2) landmark observation, 3) forward, 4) forward, and 5) left forward. The ball and the target goal were observed at every moment for decision making. The probability distributions of landmark observation predictions and the action decision are shown in Fig.7.

We show the number of re-observations in Table 3.2. Each number indicates the number of trials, total steps, re-observations, and the rate of re-observations.



**Fig. 4.** Quantization for landmarks (left) and the ball (right).

**Table 1.** Depth of the action and prediction trees (Experiment 1).

	# of leaves	min depth	mean depth	max depth
action	43	1	4.91	8
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

**Table 2.** The order of information in trees (Experiment 1, ‘act’ means action, ‘1’ means root node).

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

We see that the number of re-observation is reduced to about half of the total steps.

### 3.3 Experiment 2

In the same field we trained the robot with regard to the games of the RoboCup 99. In this experiment, we placed the robot and the ball at many more locations than 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 is performed for 4.8 seconds.

By this training, we obtained 1364 data points. Deleting inappropriate data, we used 856 data points for the action decision tree and 1364 data points for the ball and landmarks predictions. We show the sizes and the order of the trees based on the information criterion in Tables 4 and 5. We used these data for the RoboCup 99 and the robot showed the movement that was expected though the robot did look for the landmarks more frequently than anticipated.

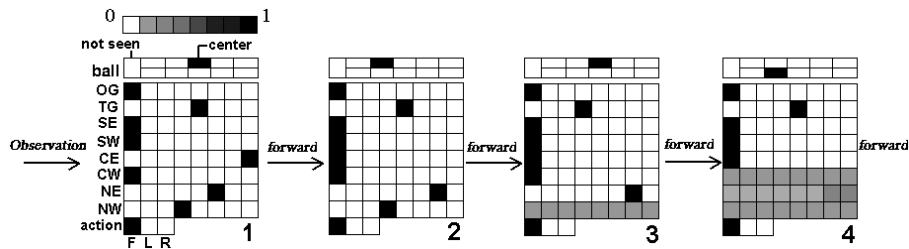


Fig. 5. Probability distribution in Experiment 1-1 (The gray scale of each box indicates the probability. Darkest 1 and lightest 0.).

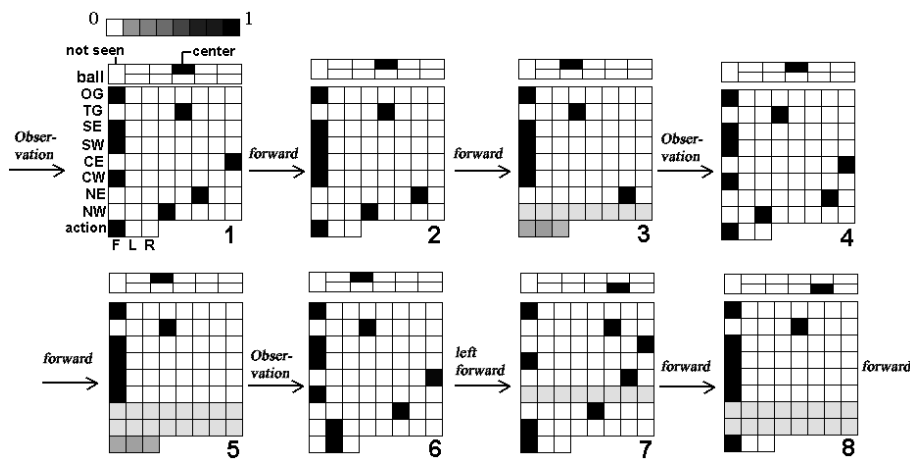


Fig. 6. Probability distribution in Experiment 1-2.

## 4 Discussions and conclusions

From Experiment 1, we see that the action probability distributions have either a sharp peak (nearly equal to 1.0) or a very flat profile. When the action probability profile is flat, the landmark probability profile, which is important for decision making, is also flat due to poor training data for making prediction trees. Then, the robot looked all around to search the landmarks and the ball.

Comparing the order of action and landmarks in prediction trees between Experiment 1 and Experiment 2, we notice that the action is higher priority in Experiment 1 than in Experiment 2. It seems that in Experiment 1 the training data was too few to extract the fact that prediction of landmarks highly depends on its location, which indicates prediction can be done mostly by landmark observation. Note that although the order is different in both cases, they are extracted from the training data by information criterion and are therefore



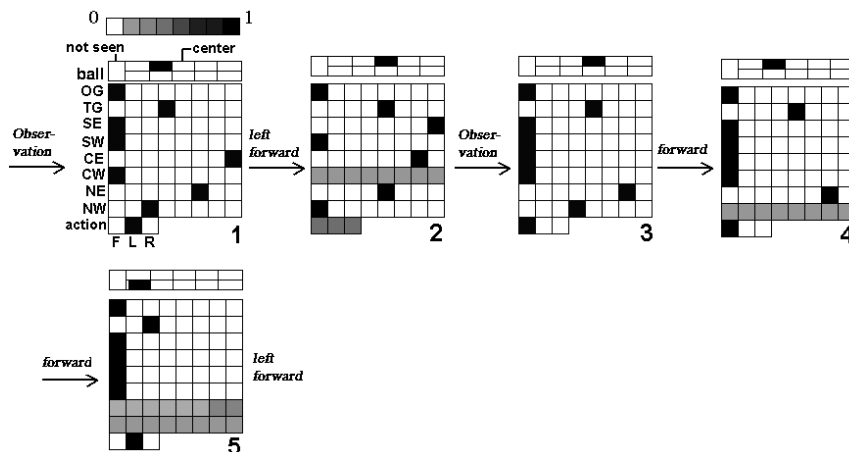


Fig. 7. Probability distribution in Experiment 1-3.

Table 3. The number of needed observation (Experiment 1).

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

optimal in that sense. Although the proposed method does not depend on it, the training data should be sufficient to accomplish the task.

In this paper, we quantized the appearance space by hand. But, by using a method like C4.5 [9] for making action decision trees, quantization may be self-organized. The order in the tree, which is fixed in this method for memory consumption and simplicity, might be changed as ID3 or C4.5 for further abstraction and observation efficiency.

## References

1. J. Wendler, S. Brüggert, H.-D. Burkhard, and H. Myritz. Fault-tolerant self localization by case-based reasoning. In *Proc. of The Fourth International Workshop on RoboCup*, pages 82–89, 2000.
2. S. Enderle, M. Ritter, D. Fox, S. Sablatnög, G. Kraetzschmar, and G. Palm. Vision-base localization in robocup environments. In *Proc. of The Fourth International Workshop on RoboCup*, pages 232–237, 2000.
3. S. Lenser and M. Veloso. Sensor resetting localization for poorly modeled mobile robots. In *Proceedings of ICRA-2000*, 2000.
4. I. H. Moon, J. Miura, Y. Yanagi, and Y. Shirai. Planning of vision-based navigation for mobile robot under uncertainty. In *Proceedings of the 1997 IEEE/RSJ Interna-*

**Table 4.** Depth and size of the trees (Experiment 2).

	# of leaves	min depth	mean depth	max depth
action	557	2	5.87	9
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

**Table 5.** The order of information in trees (Experiment 2).

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

*tional Conference on Intelligent Robots and Systems*, volume 1, pages 1202–1207, 1997.

5. I. Moon, J. Miura, and Y. Shirai. Dynamic motion planning for efficient visual navigation under uncertainty. In Y. Kakazu, M. Wada, and T. Sato, editors, *In Proc. of the Intelligent Autonomous Systems 5*, pages 172–179, 1998.
6. W. Burgard, D. Fox, and S. Thrun. Active mobile robot localization. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*. Morgan Kaufmann, San Mateo, CA, 1997.
7. 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, editors, *Fourth European Conference on Artificial Life*, pages 309–317. The MIT Press, 1997.
8. J. R. Quinlan. Discovering rules from large collections of examples: a case study. In D. Michie, editor, *Expert Systems in the Microelectronic Age*. University Press, Edinburgh, Scotland, 1979.
9. J. R. Quinlan. *C4.5: PROGRAMS FOR MACHINE LEARNING*. Morgan Kaufmann Publishers, 1993.