Visual Attention Control by Sensor Space Segmentation for a Small Quadruped Robot Based on Information Criterion

Noriaki Mitsunaga and Minoru Asada

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

Abstract. Since the vision sensors bring a huge amount of data, visual attention is one of the most important issues for a mobile robot to accomplish a given task in complicated environments. This paper proposes a method of sensor space segmentation for visual attention control that enables efficient observation by taking the time for observation into account. The efficiency is considered from a viewpoint of not geometrical reconstruction but unique action selection based on information criterion regardless of localization uncertainty. The method is applied to four legged robot that tries to shoot a ball into the goal. To build a decision tree, a training data set is given by the designer, and a kind of off-line learning is performed on the given data set. The visual attention control in the method and the future work are discussed.

1 Introduction

Mobile robots often have visual sensors that bring a huge amount of data for decision making. Therefore, attention control which extracts necessary and sufficient information for the given task is demanded for efficient decision making. We have proposed a method of efficient observation for decision making [1], which assumed that the sensor values were quantized in advance and observation cost (the time needed for the camera head motion and image acquisition) did not vary. For more adaptive and efficient observation, self-segmentation of the sensor space for attention control is necessary.

In the reinforcement learning area, the number of states should be minimum because the learning time is exponential to the size of the state space [2]. Then, several sensor space segmentation methods for state space construction have been proposed (Takahashi et al.[3], Yairi et al.[4], Kamiharako et al.[5], and Noda et al.[6]). However, they did not consider the actual time for observation nor used an active vision system. Kamiharako et al.[5] showed some results with the coarse to fine attention control but they adopted the omni-directional vision system by which the robot capture the image whole around of itself.

In this paper, we propose a method of sensor space segmentation for visual attention control that enables efficient observation taking the time needed for observation into account. The efficiency is considered from a viewpoint of not

© Springer-Verlag Berlin Heidelberg 2002

geometrical reconstruction but unique action selection based on information criterion regardless of localization uncertainty. The method is applied to four legged robot that tries to shoot a ball into the goal. To build a decision tree, a training data set is given by the designer, and a kind of off-line learning is performed on the given data set.

The rest of the paper is organized as follows. First, the method is introduced along with basic ideas related to information criterion, efficient observation, prediction model, and decision making. Then, the experimental results using RoboCup four-legged robot league platform (almost same as Sony AIBO) are shown. Finally, discussion on the visual attention control in the method is given and the future works are shown.

2 The Method

2.1 Assumptions

We assume, 1) the robot needs to pan and tilt its camera to acquire the necessary information for action selection, since the visual angle of the camera is limited. 2) The environment includes several landmarks, which provide the robot sufficient information to uniquely determine the action from their views. 3) Sufficient training data for decision making are given. We used a teaching method to collect such data. A training datum consists of a set of the appearance of the landmarks and the action to accomplish the task at the current position. 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.

There are methods which construct a classifier in the form of decision tree with information gain, such as ID3 and C4.5 [7]. To construct a decision tree, we need a training data set. Each datum consists of a class which it belongs to and attribute values by which we classify it to the class. If we apply these methods, a class and an attribute correspond to an action and a sensor, respectively. When we use ID3 which only handles quantized attribute values, 1) we calculate each information gain I_i in terms of action after observing sensor i, 2) divide data set according to the sensor values with the largest information gain. We iterate these process until the information gains for all sensors become zero or the action in the divided data becomes unique. In an action decision tree, a node, an arc, and a leaf indicate the sensor to divide data set, the sensor value, and the action to take, respectively. C4.5 handles continuous attribute values by dividing data set with a threshold. The threshold to divide is determined so that the information gain can be the largest after the division. Due to the limited view angle of its camera, the robot needs to change its gazes in order to know whether a landmark is observed on the left side of the threshold or not. However, it needs only one gaze to know whether a landmark is observed inside a limited area (attention window) or not. Therefore we use an attention window which maximize the information gain for dividing the training set into two sub-sets.

2.2 Information Gain by Observation

Suppose we have r kinds of actions and n training data. First, calculate the occurrence probabilities of actions p_j $(j = 1, ..., r)$ as $p_j = n_j/n$, where n_j denotes the number of taken action j. Therefore, the entropy H_0 for the action probability is given by

$$H_0 = -\sum_{j=1}^{r} p_j \log_2 p_j. \tag{1}$$

Next, calculate the occurrence probabilities of actions after observation. After the observation, it knows whether the landmark is inside the attention window $(\theta_{Lk}, \theta_{Uk}]$ or not. The lower and upper limits of a window θ_{Lk}, θ_{Uk} are a pair of middle points of adjacent sensor values of the training data. We denote the number of times action j was taken as n^I_{ijk} when the landmark i was observed in $(\theta_{Lk}, \theta_{Uk}]$ and n^O_{ijk} when not observed. Then, the occurrence probability becomes, $p^I_{ikj} = n^I_{ikj}/n^I_{ik}$, $p^O_{ikj} = n^O_{ikj}/n^O_{ik}$. Where $n^I_{ik} = \sum_j^r n^I_{ikj}$, and $n^O_{ik} = \sum_j^r n^O_{ikj}$. Next, calculate the entropy after the observation, as follows:

$$H_{ik} = -\sum_{x=\{I,O\}} \frac{n^x_{ik}}{n_{ik}} \sum_{j=1}^{r} (p^x_{ikj} \log_2 p^x_{ikj}). \tag{2}$$

The information gain by this observation is $I_{ik} = H_0 - H_{ik}$. The larger I_i is, the smaller the uncertainty is after the observation.

2.3 Actual Time for Observation

When the time for observation is constant, we can use information gain for making action decision tree. The tree becomes compact and the robot can determine its action at shortest observation time by following the tree. However, if the time for observations changes depending on the gaze directions, the time for action decision can be longer using the tree. Therefore, we use the information gain per time, in other words the velocity, rather than information gain.

We denote T as the time to get the observation after previous observation, and information gain per time i_{ik} as,

$$i_{ik} = \frac{I_{ik}}{T + a}. \tag{3}$$

Here a is a positive constant. When the direction is already observed $T = 0$.

2.4 Making an Action Decision Tree

We put the attention windows into the tree in decreasing order of uncertainty after its observation. Based on i_{ik} we divide training data into two subsets until the action in the subset becomes unique. 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 the left of Table 1. The numbers in the table indicates the direction in which the landmark was observed. The view angle is limited and it can gaze and observe three areas $[0, 15)$, $[15, 30)$, $[30, 45)$. It gazes in $[15, 30]$ at the beginning of action decision, and needs one period of time to change the direction to observe. Since $p_x = 2/4$, $p_y = 1/4$, and $p_z = 1/4$, H_0 is 1.5. The information gain by observation I_{ik} and the information gain per time i_{ik} are shown in the right of Table 1. Since i_{ik} of the observation which checks whether the landmark A is in $[27, 30)$ or not is the largest, we put this to the root of the tree. If the landmark is in $[27, 30)$ the action is unique and it is y. Else, the subset has three training data and the actions are not unique. The information gain per time of observation whether landmark B is in $[0, 15)$ or not and observation whether landmark A is in $[30, 40)$ or not is 0.05. We prefer left $[0, 15)$ to observe and the action decision tree is shown in Fig.1.

Table 1. Example training data (left) and calculated information and information per time (right). Lm means landmark.

Data #	Lm A	Lm B	action
1	5	5	x
2	25	15	x
3	30	10	y
4	40	30	z

Observation	Info. I_{ik}	Info. / time i_{ik}
$0 \leq (LmA) < 15$	.31	.15
$15 \leq (LmA) < 27$	.31	.31
$15 \leq (LmA) < 30$	.50	.50
$27 \leq (LmA) < 30$	1.4	1.4
$30 \leq (LmA) < 45$	1.4	.70
$0 \leq (LmB) < 7$	.31	.15
$0 \leq (LmB) < 12$	.5	.25
$0 \leq (LmB) < 15$	1.4	.70
$7 \leq (LmB) < 12$	1.4	.70
$7 \leq (LmB) < 15$	.5	.25
$30 \leq (LmB) < 40$	1.4	.70

```
Observe [15, 30), if 27<=(Landmark A)<30
then
        take action y
else
        Observe [0, 15) and if 0<=(Landmark B)<15
        then
                take action x
        else
                take action z
```

Fig. 1. Action decision tree of the example data

2.5 Making a Decision

In order to make a decision on which action to take, first, the robot calculates the observation probability $\boldsymbol{x}(t)$ from previous $\boldsymbol{x}(t-1)$ and action $a(t-1)$ if possible. If no prediction model is applicable, use the probability 1 or 0 if the direction of the window has been observed, otherwise the probability is 0.5. Then it updates $\boldsymbol{x}(t)$ by the observation and calculate action probabilities. An action probability is the sum of the probability to reach leaves to take that action in the action decision tree. If one of the action probabilities is very high, it takes that action. Otherwise, until one of them becomes high enough, it continues to check attention windows from the root of the action decision tree, update the observation probability, and the action probabilities.

3 Experiments

3.1 Task and Environment

The task is to push a ball into a goal based on the visual information. We used a legged robot for the RoboCup SONY legged robot league (Fig.2). The robot is equipped with a limited view angle camera. 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 role of the neck joint when it observes the landmarks and the ball to make its decision. The robot can rotate the pan joint from -88 to 88 degrees and the tilt joint from -80 to 43 degrees. We prepared five directions (every 44 degrees) in the pan joint and four directions (every 40 degrees) in the tilt joint to observe. The maximum angular velocity of the pan joint is $5.9[rad/s]$ and $3.9[rad/s]$ in the tilt joint. The robot waits at least for $0.16[s]$ after rotating pan or tilt joint. Since it needs at least 0.29[s] before action decision after changing observing directions, we prepared $a = 0.29[s]$.

As vision sensors, we used the coordinates of the image centers of the landmarks and the ball, the minimum x/y and maximum x/y (totally four) of the goals. We did not directly use the values in the images, but converted to the pan and tilt angles when the targets are viewed at center of an image. And, we used pair of the pan (x) and tilt (y) angles as a sensor value rather than one of them, or we divided training data-set by the observation to check whether a sensor value is in the rectangle of $(x_{\min}, y_{\min}) - (x_{\max}, y_{\max})$ (attention window). This is because if we divide the data-set by the observation to check whether the x is in $[10, 20)$ or not, we have to rotate the tilt angle to check whole y variation.

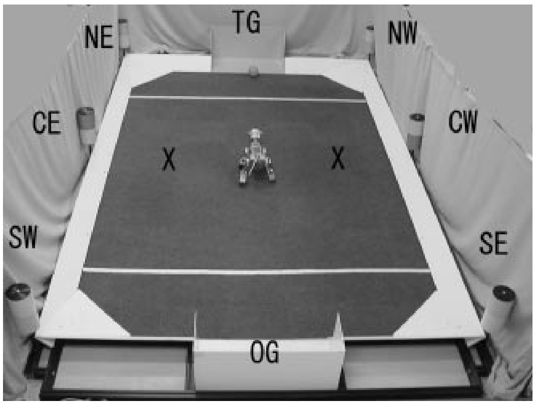


Fig. 2. The legged robot for RoboCup SONY legged robot league (left) and the field (right).

3.2 Experimental Results

We trained the robot starting from one of three positions in the middle of the field. We prepared seven actions, try to reach the ball, move forward, move left / right forward, turn left / right, and turn leftward / rightward. For each starting position, we trained five times and obtained 99 data points to construct trees. We show the part of the action decision tree in Fig.3. The unit of the figures are degrees. When the robot has no prediction and not observed yet, first attention window is rectangle of $(-19,-8)-(-8,18)$ for the ball and observe the direction $(3,2)$ (Fig.4(a)). Direction (X,Y) indicates the direction of observation. X is the panning direction $(1,...,5)$ and 3 if it is center. Y is the tilt direction $(1,...,4)$ and 2 if it is horizontal. Since the robot is facing $(3,2)$ at the beginning of action decision time, the direction $(3,2)$ is preferred by nodes near the root of the tree. If the center of the ball image is in the window, the next window is $(-19,-10)-(-18,-13)$ for minimum y of TG and the direction is same $(3,2)$ (Fig.4(b)). If it is in the window, try to reach the ball otherwise move forward. If the center of the ball image is not in the first window, the next windows to check is $(-19,-18)-(18,18)$ for minimum x of TG and the direction is $(3,2)$ (Fig.4(c)). If it is in the window check the next window $(-19,-10)-(4,4)$ (Fig.4(d)) and so on.

We show and compare the attention windows generated by three methods in Fig.5-7. Fig.5 shows the attention windows generated with pre-quantized sensor values in every 20 degrees without time consideration (we directly used information gain rather than velocity). Fig.6 shows the one generated by the proposed segmentation method without time consideration. Fig.7 shows the one generated with proposed segmentation method with time consideration. With these figures we can see that pre-quantized segmentation is not efficient for decision making

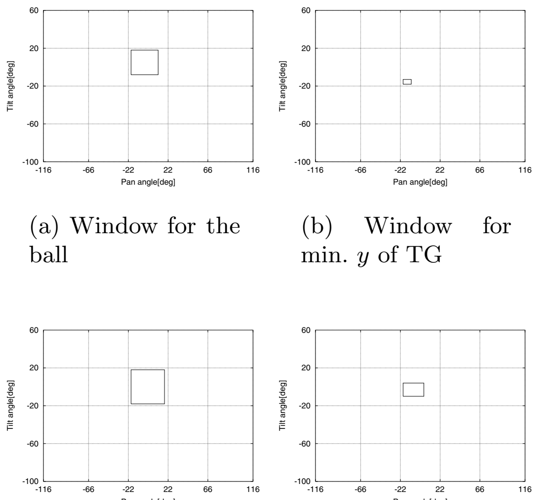
```
[ball] -19<x<11, -8<y<18 (Fig.5(a)) then
  [TG ymin] -19<x<-10, -18<y<-13 (Fig.5(b)) then
    Do TryReachBall
  else
    Do Forward
  [TG xmin] -19<x<18, -18<y<18 (Fig.5(c)) then
    [TG xmin] -19<x<4, -10<y<4 (Fig.5(d)) then
      Do Forward
    else
      [TG xmin] -19<x<11, -15<y<-3 then
        Do RightTurn
      [TG xmin] -6<x<18, 15<y<18 then
        [TG xmin] -12<x<11, -1<y<18 then
          [ball] 27<x<46, 26<y<32 then
            Do RightFoward
          else
            Do RightTurn
        else
          Do LeftRotate
          [TG xmin] 15<x<18, -8<y<4 then
            [CW] -52<x<-30, -15<y<5 then
              Do Forward
            else
              Do RightTurn
          else
            Do Forward
                  :
```

Fig. 3. Part of action decision tree generated by proposed method

and with time consideration the windows are preferred which bring information gain faster rather than the ones which bring higher information gain.

We show the comparison of quantization and consideration of time in Table 2. We compare the number of nodes (windows) in a tree (# of nodes), depth of the tree, the number of expected observing directions, the expected time for observation (time). The expected number of observing directions and time are the one when it does not use predictions. The proposed quantization shows smaller size of the tree and the number of expected observing directions. And it can reduce the time for observation to half by using information gain per time.

The mean time for observation without predictions in experiments with real robots were 6.6[s] with pre-quantization method, 5.2[s] with proposed quantization without time consideration and 2.5[s] with proposed method. Though the time is longer than expected, the time of proposed method is below half of other methods.



(a) Window for the ball

(b) Window for min. y of TG

(c) Window for min. x of TG

(d) Another window for min. x of TG

Fig. 4. Attention windows

Table 2. Comparison of sizes of the tree, expected number of observing directions, and expected time for observation

	# of nodes,	depth,	# of leaves,	directions,	time[s]
pre-quant.	61	18	31	7.6	2.8
quant. only	39	11	20	5.5	2.0
proposed	59	15	30	3.4	0.83

4 Discussions and Conclusions

We showed that a decision tree which is constructed with greedy for information gain or information gain per time. Efficient observation for decision making was achieved by greedy approach. However, decision making with tree constructed with greedy approach may prone to sensor noise, occlusions, and so on [4]. Occlusions are ignored if an action is determined without occluded windows, otherwise they may lead to wrong action selection. Currently, training data should contain the some variations with sensor values which cover noises or occlusions, so that the reliability is reflected to the information gain. The sufficiency of training data is measured with the actions determined by the action decision tree. If you can use some prediction model it will help to solve the problem of temporal occlusions.

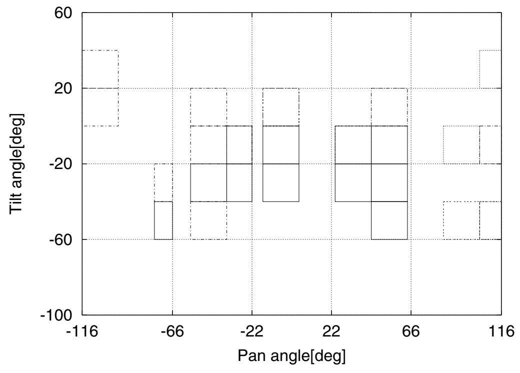


Fig. 5. Created attention windows by pre-quantization method without time consideration.

We showed only the results without prediction model. Since the data were not enough for simple least square method. A prediction model which can be consists from small data robustly is expected.

To conclude, we proposed a method to make a decision tree with an autonomous sensor value segmentation with consideration for variance in time interval to acquire observation. Attention control is done by observation following a decision tree which is constructed based on information criterion with sensor space segmentation. The validity of the method was shown with a four legged robot.

Acknowledgement

This research was supported by the Japan Science and Technology Corporation, in Research for the Core Research for the Evolutional Science and Technology Program (CREST) title Robot Brain Project in the research area "Creating a brain".

References

[1] Noriaki Mitsunaga and Minoru Asada. Observation strategy for decision making based on information criterion. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1038–1043. 2000.

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

[3] Yasutake Takahashi, Minoru Asada, and Koh Hosoda. Reasonable performance in less learning time by real robot based on incremental state space segmentation. In *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1518–1524, 1996.

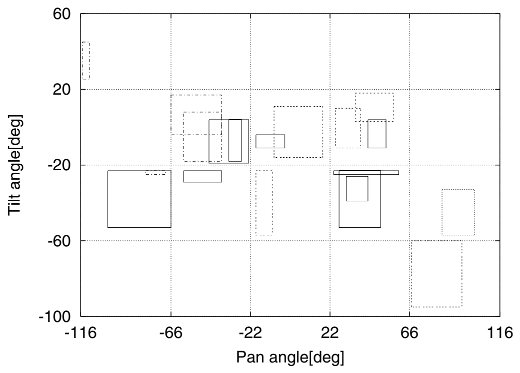


Fig. 6. Created attention windows by proposed segmentation without time consideration.

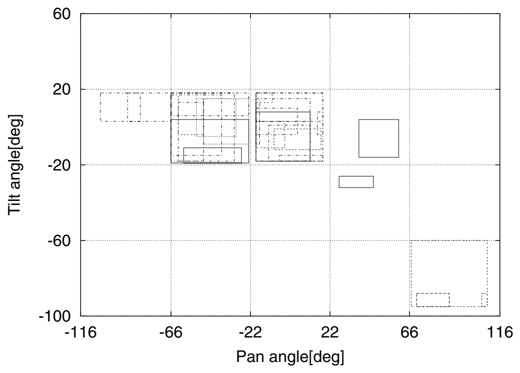


Fig. 7. Created attention windows by proposed method.

[4] Takehisa Yairi, Shinichi Nakasuka, and Koichi Hori. State abstraction from heterogeneous and redundant sensor information. In Y. Kakazu, M. Wada, and T. Sato, editors, *In Proc. of the Intelligent Autonomous Systems 5*, pages 234–241, 1998.

[5] Masatoshi KAMIHARAKO, Hiroshi ISHIGURO, and Toru ISHIDA. Attention control for state space construction. In Y. Kakazu, M. Wada, and T. Sato, editors, *In Proc. of the Intelligent Autonomous Systems 5*, pages 258–265, 1998.

[6] Minoru Asada, Shoichi Noda, and Koh Hosoda. Action based sensor space segmentation for soccer robot learning. *Applied Artificial Intelligence*, 12(2-3):149–164, 1998.

[7] J. Ross Quinlan. *C4.5: PROGRAMS FOR MACHINE LEARNING.* Morgan Kaufmann Publishers, 1993.