Sensor Space Segmentation for Visual Attention Control of a Mobile Robot based on Information Criterion

Noriaki Mitsunaga **Minoru Asada**

Dept. of Adaptive Machine Systems,
Graduate School of Engineering, Osaka University
Suita, Osaka, 565-0871, Japan
mitchy@er.ams.eng.osaka-u.ac.jp asada@ams.eng.osaka-u.ac.jp

Abstract

Visual attention is one of the most important issues for a vision guided mobile robot not simply because visual information bring a huge amount of data but also because the visual field is limited, therefore gaze control is necessary. This paper proposes a method of sensor space segmentation for visual attention control that enables mobile robots to realize efficient observation. 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 builds a decision tree based on the information criterion while taking the time needed for observation into account, and attention control is done by following the tree. The tree is rebuilt by introducing contextual information for more efficient attention control. The method is applied to a four legged robot that tries to shoot a ball into the goal. Discussion on the visual attention control in the method is given and the future issues are shown.

1 Introduction

Mobile robots are often equipped with visual sensors that bring a huge amount of data about the environment of which image processing takes time, and often the visual field is limited. Therefore gaze control is necessary before changing view directions. Thus, attention control which extracts necessary and sufficient information for the given task is needed 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 the time for gaze control and sensory data acquisition was fixed. Further, the contextual information was not directly involved into the system. Self-segmentation of the sensor space and the direct use of the contextual information seem necessary to realize more adaptive and efficient attention control.

McCallum[2] has proposed a learning algorithm for attention control. However, the number of the gaze directions is just four and the gaze control is rigidly linked to the moving direction, that is, the system cannot change its moving directions without changing gaze directions.

In the reinforcement learning area, the number of states should be minimized because the learning time is exponential to the size of state space [3]. Then, several sensor space segmentation methods for state space construction have been proposed (Takahashi et al .[4], Yairi et al. [5], Kamiharako et al. [6], and Noda et al. [7]). However, they have not considered the actual time for observation nor used an active vision system. Kamiharako et al. [6] showed some results using a coarse to fine attention control but they adopted the omni-directional vision system by which the robot does not need to change its view directions.

In this paper, we propose a method of sensor space segmentation for visual attention control that enables mobile robots to realize efficient observation. The efficiency is considered from a viewpoint of not geometrical reconstruction but unique action selection based on the information criterion regardless of localization uncertainty. The method builds a decision tree based on information criterion while taking the time needed for observation into account, and observation is done by following the tree. The tree is rebuilt by introducing the contextual information for more efficient attention control. The method is applied to a four legged robot that tries to shoot a ball into the goal. To build a decision tree, a training 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 the information criterion, the efficient observation, the contextual information, and the decision making. Then, the experimental results using the 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 issues are shown.

2 The method

2.1 Assumptions

In our experiments, the robot has to pan and tilt its camera to acquire the necessary information for action selection since the visual field of the camera is limited. The environment includes several landmarks of which appearances provide the robot with sufficient information to uniquely determines the action. Training data are given for making decisions and predictions.

We used a teaching method to collect such data. A training datum consists of a set of the views of the landmarks at the current position and the action to accomplish the task. During the training period, the robot pans and tilts its camera head to observe as many landmarks as possible.

2.2 Decision tree

There are methods which construct a classifier in the form of decision tree with information gain, such as ID3 and C4.5 [8]. To construct a decision tree, we need a training data set. Each datum consists of a class which it belongs to and its attribute values. In our experiments, a class and an attribute correspond to an action and a sensor, respectively. In case of ID3 (which only handles quantized attribute values), 1) we calculate each information gain I_i for action after observing sensor i, and 2) divide the data set according to the sensor values with the largest information gain. We iterate these processes until the information gains for all sensors become zero or the action in the subset of the training data becomes unique. In an action decision tree, a node, an arc, and a leaf indicate the sensor to divide the data set, the sensor value, and the action to be taken, respectively. C4.5 handles continuous attribute values by dividing data set. The threshold to divide is determined so that the information gain becomes the largest after the division.

Due to the limited view angle camera, the robot needs to change its gazes in order to know whether a landmark is observed in left of the threshold or not. However, it needs only one gaze control to know whether a landmark is observed inside a limited area (attention window) or not in order to divide the training set into two subsets and to calculate the information gain.

2.3 Information gain by observation

Suppose we have r kinds of actions and n training data. First, the occurrence probabilities of actions p_j ($j = 1, ..., r$) are calculated from $p_j = \frac{n_j}{n}$, where n_j denotes the number of taken action j. Therefore, the entropy H_0 for the action probability is given by

$$H_0 = -\sum_{j=1}^{r} p_j \log_2 p_j. \qquad (1)$$

Next, the occurrence probabilities of actions after observation are calculated. After the observation, it knows whether the landmark is inside the attention window $(\theta_{Lk}, \theta_{Uk}]$ or not. 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,

$$p^{I,O}_{ikj} = n^{I,O}_{ikj} / n^{I,O}_{ik}, \qquad (2)$$

where $n^I_{ik} = \sum_j^r n^I_{ikj}$, and $n^O_{ik} = \sum_j^r n^O_{ikj}$. Next, the entropy after the observation are calculated as follows:

$$H_{ik} = -\sum_{x=\{I,O\}} \frac{n^x_{ik}}{n_{ik}} \sum_{j=1}^{r} (p^x_{ikj} \log_2 p^x_{ikj}). \qquad (3)$$

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

2.4 Actual time for observation

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

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

$$i_{ik} = \frac{I_{ik}}{T + T_C}, \tag{4}$$

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

2.5 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 such training data that take different actions for the same situation, we add a leaf for each action and record its probability that it was taken. This tree enables action decision and observation without direct use of the contextual information.

For example, suppose we have a set of training data as shown in Table 1. The numbers in the table indicate the direction in which the landmark was observed. The view angle is limited and it can gaze and observe three areas $[0, 15]$, $[15, 30]$, and $[30, 45]$. It gazes in $[15, 30]$ at the beginning of action decision, and needs one period of time to change the direction to observe, then we set $T_C = 1$. Since $p_x = 2/4$, $p_y = 1/4$, and $p_z = 1/4$, H_0 is 1.5. The entropy after the observation to check whether the landmark A is in a window $(27, 30]$ or not is 0.68. Since after observation, the data is divided into two subsets, one has data number 3 and the entropy is 0, and the other has training data 1, 2, and 3 whose entropy is 0.91. The information gain I_{ik} is $1.5 - 0.68 = 0.82$ and the information gain per time i_{ik} is $0.82/(0 + 1) = 0.82$. Since this i_{ik} is the largest among all attention windows, we put this to the root of the tree. If the landmark is in $(27, 30]$, the action is unique or entropy is 0 and the action is y. Else, the subset has three training data and the actions are not unique. The information gain per time for observation whether landmark B is in $(0, 15]$ or not, and observation to check whether landmark A is in $(30, 40]$ or not is 0.46. We prefer left $(0, 15]$ to observe and the resultant action decision tree is shown in Fig.1.

Table 1: Example training data

Data #	Landmark A	Landmark B	action
1	5	5	x
2	25	15	x
3	27	10	y
4	40	30	z

```
Observe [15, 30], if 26<(Landmark A)<=30
then
        take action y (leaf a)
else
        Observe [0, 15] and if 0<(Landmark B)<=15
        then
                take action x (leaf b)
        else
                take action z (leaf c)
```

Figure 1: Action decision tree of the example data

2.6 Rebuilding the tree

The previous observation and action construct the current context. However, the previous observation is done for previous action decision and cannot be always used for current action decision. The previous action can be used but does not have rich information. Here, the knowledge which leaf has been previously reached includes previous observation and constructs the current context. Therefore we use the previous leaf as a logical sensor and rebuild the tree as follows; 1) build the action decision tree without contexts, 2) attach each training datum with leaf previously has been visited in the sequence and the current leaf, 3) calculate and extract the attention window which has largest information gain per time by regarding the leaf which it belongs to, 4) calculate and extract the partitioning of the data set by their previous leaves which has the largest information gain per time (since the time to gather this knowledge is zero, we just divide information gain with T_C), 5) compare the results of 3) and 4), then divide training data set into two data sets by 3) or 4) with larger information gain per time, 6) repeat from 3) until the all data in the data set belong to the same leaf.

We use the information gain regarding the leaf it belongs to, since the leaves will be different ones from previously created if we use information gain regarding action. We limited the partitioning of data set only to dividing into two subsets even by the knowledge of previous leaf so as not to cause combinatorial explosion.

2.7 Making a decision

In order to make a decision to take an action. first the robot sets the observation probability of attention windows to 1 or 0 for the direction it is observing and 0.5 to windows of other directions. Then using the observation probabilities and the probability which leaf previously visited, calculate action probabilities. An action probability is the sum of the probability to reach leaves in the action decision tree to take the action. 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 the direction of 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 with a limited view angle for the RoboCup 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 52 degrees (88 pixels) in width, and about 71 degrees (48 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 about $6[rad/s]$ and $4[rad/s]$ in the tilt joint. Since it needs at least 0.36[s] before action decision after changing observing directions, we prepared $T_C = 0.36[s]$.

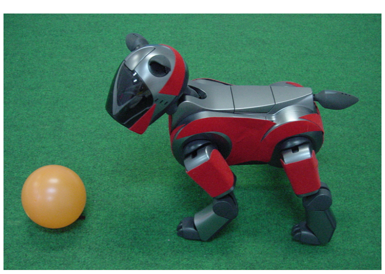

Figure 2: The SONY legged robot for the RoboCup SONY legged robot league.

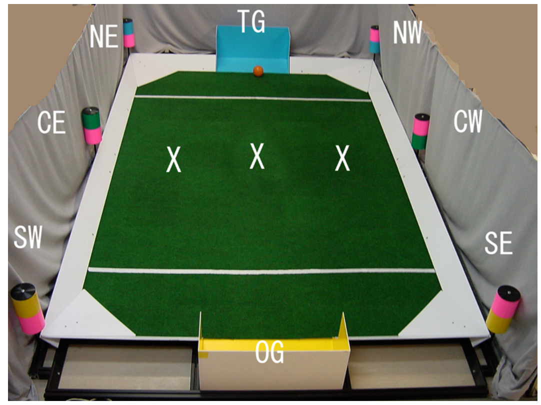

Figure 3: Experimental field (same as the one for the RoboCup SONY legged robot league).

As vision sensors, we used the coordinates of the image centers of the landmarks and the ball, the minimum and maximum x, y coordinates (totally four) of the goals. We did not directly use the values in the images, but used the pan and tilt angles when the targets are viewed at center of an image. Also, we used a pair of the pan (x) and the tilt (y) angles as a sensor value, or we divided training data set by the observation to check whether a sensor value is in the rectangle attention window $(x_{\min}, y_{\min}) - (x_{\max}, y_{\max})$ or not.

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 92 data points to construct trees. We show the part of the action decision tree in Fig.4. The numbers in the figure indicates the angles in degrees. When the robot has no prediction and not observed yet, first attention window is rectangle of $(-19, -7) - (15, 20)$ for the ball and observe the direction $(3, 2)$. 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. If the center of the ball image is in the window, the next window is $(-10, 10) - (7, 17)$ for the ball and the direction is same $(3, 2)$. Again the next window $(-8, -4) - (6, 11)$ is for the ball and the direction is $(3, 2)$. If it is in the window, try to take the left-forward action otherwise

```
O(3, 2) [ball] -19<x<=15, -7<y<=20 then
  O(3, 2) [ball] -10<x<=7, 10<y<=17 then
    O(3, 2) [ball] -8<x<=6, -4<y<=11 then
      Do LeftFoward
    else
      Do LeftTurn
  else
    O(3, 2) [ball] -20<x<=-12, 12<y<=19 then
      Do LeftFoward
    else
      Do LeftFoward
else
  O(3, 2) [TG xmin] -1<x<=20, -20<y<=20 then
    O(3, 2) [ball] -2<x<=20, 11<y<=18 then
      Do LeftTurn
    else
      Do LeftTurn
  else
    O(3, 2) [NE] -20<x<=21, -18<y<=-3 then
      O(3, 2) [NE] -18<x<=-4, -4<y<=9 then
        Do LeftFoward
      else
        Do RightFoward
    else
                 :
```

Figure 4: Part of action decision tree generated by proposed method

left-turn action and so on.

Fig.5 shows the attention windows generated with proposed segmentation method with time consideration (information gain per time). Fig.6 shows the attention windows generated with the knowledge which leaf the robot previously visited. We assumed that action sequences are started from the beginning as training data sequences and can use the knowledge of previous visited leaf in the situations of middle of the training sequences. When there are chances that the robot starts from middle of the sequences it has to use another tree (without context) at the begging of the sequence. The windows are slightly different from the ones without knowledge.

Fig.7 shows the attention windows generated with the knowledged which leaf the robot previously visited. We prepared data so that the robot can start from middle of the training sequences. When we mark each training datum with the previously visited leaf we doubled the datum, one is marked with previous leaf and the other marked that it does not know previous leaf. The created windows (Fig.7) resembles more to the ones without knowledge (Fig.5) than Fig.6.

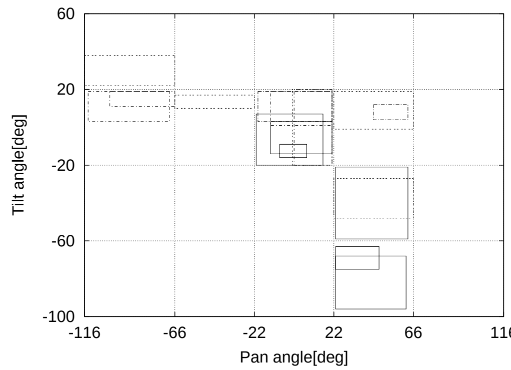

Figure 5: Created attention windows by proposed segmentation with time consideration.

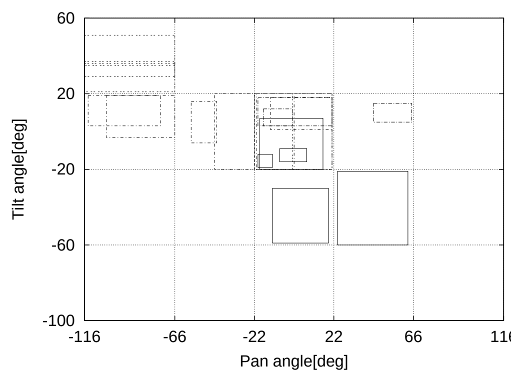

Figure 6: Created attention windows with knowledge of previous leaf (we used data which has knowledge of previous leaf only, context(1)).

Table 2 shows the comparison of the number of nodes (windows) in a tree (# of nodes), the depth of the tree, the number of expected observing directions (dirs), the expected time for observation (time). In the table, "pre-quant." means attention windows were generated with pre-quantized sensor values in every 20 degrees with information gain, "quant." only means attention windows were generated with proposed quantization with information gain, "proposed quant." means proposed quantization with information gain per time. The proposed quantization shows smaller size of the tree and the smaller number of expected observing directions than fixed quantization. By using information gain per time, it can reduce the time for observation.

It shows that by rebuilding and using contexts it

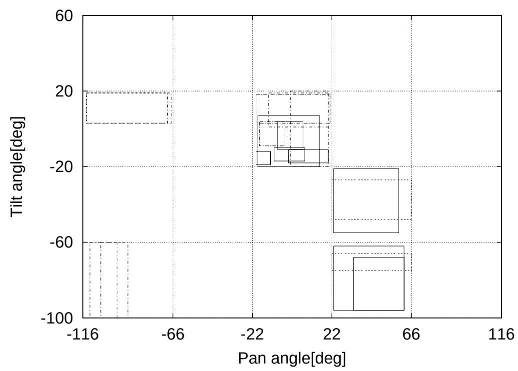

Figure 7: Created attention windows with knowledge of previous leaf (we used all data, context(2)).

can reduce much time. It is interesting that by assuming that it does not start from the middle of the training data sequence (context(1)), the time for observation is reduced, however the size of the tree is larger than when we do not assume (context(2)). Also we note that the knowledge of previous leaf was used only a few times.

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

	# of nodes	depth	# of leaves	dirs	time[s]
pre-quant.	47	17	24	5.4	3.5
quant. only	23	8	12	4.3	2.5
proposed quant.	35	13	15	3.4	2.0
with context(1)	49	9	25	2.0	1.1
with context(2)	37	9	19	2.6	1.6

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 might be prone to sensor noise, occlusions, and so on [5]. Currently, training data should cover the case of sensor noises or of occlusions in order to overcome this problem, then we may need large training data. A method which can find robustness of observation may be desired.

We used the same T_C for calculation of information gain per time by the partitioning of data-set by the knowledge of previous leaf. To use the knowledge much more we should use another T_C used for attention windows. This parameter should be studied.

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 and a kind of context. 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.

References

[1] N. Mitsunaga and M. 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] A. K. McCallum. Learning to use selective attention and short-term memory in sequential tasks. In *From animals to animates 4*, pages 315–324. 1995.

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

[4] Y. Takahashi, M. Asada, and K. 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.

[5] T. Yairi, S. Nakasuka, and K. 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.

[6] M. KAMIHARAKO, H. ISHIGURO, and T. 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.

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

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