State Space Construction by Attention Control

Hiroshi Ishiguro, Masatoshi Kamiharako, Toru Ishida
Department of Social Informatics
Kyoto University
Sakyo-ku, Kyoto 606-8501, Japan

Abstract

In order to understand cognitive aspects of autonomous robots, it is fruitful to develop a mechanism by which the robot autonomously analyzes physical sensor data and construct a state space. This paper proposes a coherent approach to constructing such a robot oriented state space by statistically analyzing sensor patterns and rewards given as the result of task executions. In the state space construction, the robot creates sensor pattern classifiers called *Empirically Obtained Perceivers* (EOPs) which, when combined, represent internal states of the robot. A novel feature of this method is that the EOP directs attention to select necessary information, and the state space is obtained with the attention control mechanism using EOPs. We have confirmed that the robot can effectively construct state spaces through its vision sensor and execute a navigation task with the obtained state spaces in a complicated simulated world.

1 Introduction

Acquisition and utilization of internal representation are key issues in realizing intelligent behaviors in computer systems. They are especially difficult for robots that process noisy sensory information and interact in the real world. There are two directions for the research. One requires the programmer to provide the internal representation [10; 3]. Another direction is to develop robots with cognitive interests [6]. The purpose is not to develop a robot that can execute complex tasks, but to stress the possibility that a robot may learn how to behave in the real world. Researchers in *cognitive robotics* [11] focus on cognitive aspects of physical agents – agents that learn their internal representation autonomously in order to perform tasks.

Our research direction basically takes the cognitive approach. In the cognitive approach, the key issue is how the robot efficiently obtains necessary information from complicated environments through sensors. This paper proposes an approach to construct a state space with an attention control mechanism that performs nonlinear classification of sensory data. The method proposed here is rather simple as a method of pattern recognition, but it works well in the state space construction by a robot in a complicated environment. The key points of this paper are as follows:

1. The proposed method enables the construction of a state space from redundant and noisy physical sensory data, such as visual data.
2. The proposed method efficiently and autonomously constructs the state space from physical sensory data by employing an attention control mechanism.

We consider, it is interesting and important to illuminate and examine the close relationship between state space construction and attention control. This paper discusses such a key issue in cognitive robotics.

The task of the robot discussed in this paper is to avoid obstacles. While moving, the robot collects data sets consisting of an action, sensor information, and a reward. Then, the robot develops *empirically obtained perceivers* (EOPs) [5]. EOPs statistically analyze the data sets to classify sensory patterns according to actions and rewards to obtain a proper state space for the task. In the state space construction, EOPs perform attention control. There are two types of attention control: feature selection from sensory data and continuous gaze control. Each EOP select features in the image consisting of a huge amount of pixel data. Further, the application of a combination of EOPs constitutes a sequence of changes in attention for gaze control. The acquisition of the sequence is called *visual sequentialization* [2]. We have confirmed that the robot can construct state spaces for avoiding obstacles in a complicated simulated world.

There are several research approaches that autonomously construct state spaces and reduce its size. Nakamura's work [6; 9] deals with physical sensory data but the feature extraction is based on human intuitions. Mahadevan's work based on Q-learning [13] basically has trouble dealing with physical sensory inputs that have huge dimensions. Although the G algorithm [4] and UDM [7] address certain problems, they do not use complex physical sensory data. While both methods use statistics, they assume that the sensors generate

information that can be directly used to represent the state space. In other words, they do not address the problem of how to construct the state space from raw sensory data. G algorithm's sensory data selection performs a kind of attention control. However, difficulties arise when the robot directly deals with physical sensory data since simple pattern matching methods cannot be applied to complicated sensory data. The purpose of this paper is to propose a coherent method for constructing a state space from physical sensory inputs and to propose a means of attention control for physical sensors in state space construction.

2 State Space Construction

2.1 Process Flow

Figure 1 shows the process flow for incrementally constructing a state space. In the beginning, the robot does not discriminate any sensory data and it has just one state. While moving the robot collects data sets, consisting of sensory data, an action, and a reward. These data sets are divided into two classes.

Here, the robot can move along plans if it has a state space whose size is sufficiently large for planning the robot motion. While constructing the state space, state transition probabilities can be computed from the data sets, and the robot can make plans for going toward goal states with standard planning methods by referring to the state transition probabilities and rewards assigned to states [1].

For discriminating the two classes, an EOP is obtained and the state is split in two. Then, if there are any identical states, they are coalesced. By iterating the process, a state space is incrementally constructed.

2.2 Data Collection

The data set consists of the following four components:

$$(State, Action, Sensory\ vector, Reward)$$

where *State* means a state of the robot when it executes an *Action*, acquires sensory information and receives a *Reward*. *Sensory vector* is one dimensional array of sensory inputs.

The method proposed in this paper can deal with any kind of physical sensory data. However, it is instructive to consider a vision-based robot, since visual data contains a very large of information and it is difficult to extract what is necessary for the robot to perform its task. For the analysis discussed below, 2-D vision data is simply converted into a one dimensional array, the *Sensory vector*.

[1]Several papers [12] concerning state space construction propose to apply Q-learning to assigning Q-values (which provide an action policy) during or after constructing the state space. However, such relations between states can be directly obtained by computing state transition probabilities from the data used for constructing the state space.

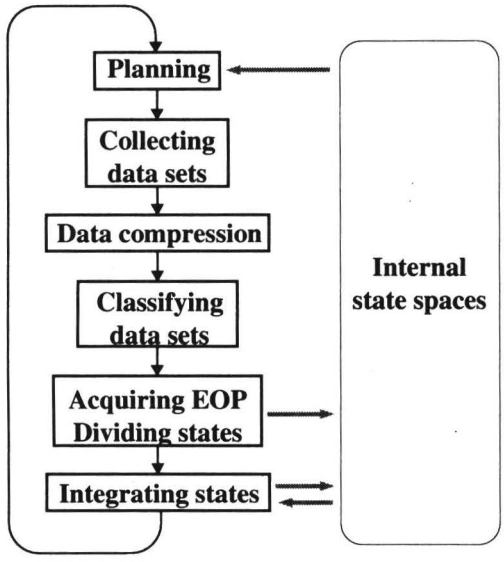

Figure 1: Process flow

In our model, humans cannot directly specify the robot's internal states. However, they can express their intentions to the robot through rewards. These rewards provide the robot's solely motive for learning to classify its sensory data. Thus, by referring to the rewards, the robot constructs its internal state space by itself. For reactive robotic tasks such as obstacle avoidance, the robot receives immediate rewards. For deliberative tasks such as moving toward goals, delayed rewards are used in place of immediate rewards.

2.3 Sensor Pattern Classification

Imagine that a robot with an incomplete state space moves about in a real environment in which rewards are embedded. If the robot receives different rewards when executing an action in a given state, the robot finds that the state should be appropriately divided into two states.

Let $g(r)$ be the frequency of $r \in \{r|\langle v, a, \mathbf{s}, r\rangle \in R(v_i, a_j)\}$, n be the number of samples in $R(v_i, a_j)$, and let K_a, K_b, K_c, and K_d be constant values, where v, a, $\mathbf{s}$, and r are *State*, *Action*, and *Sensory vector*, and *Reward*, respectively. $R(v_i, a_j)$ represent a set of data sets of which *State* and *Action* are v_i and a_j, respectively. The following equations give the conditions for dividing v_i (see Figure 2).

$$g(r) \leq K_a n \quad \forall r \in [\alpha, \beta] \tag{1}$$

$$\beta - \alpha \geq K_b(Max\ r - Min\ r) \tag{2}$$

$$n_0 \geq K_c n, \quad n_1 \geq K_c n \tag{3}$$

$$n_0 \geq K_d, \quad n_1 \geq K_d \tag{4}$$

where n_0 is the number of samples such that $r < \theta$, $\theta = (\alpha + \beta)/2$, and n_1 is the number of samples such that $r \geq \theta$. If several θs are found, the longest $[\alpha, \beta]$ is selected. In the conditions, (3) means that rare events

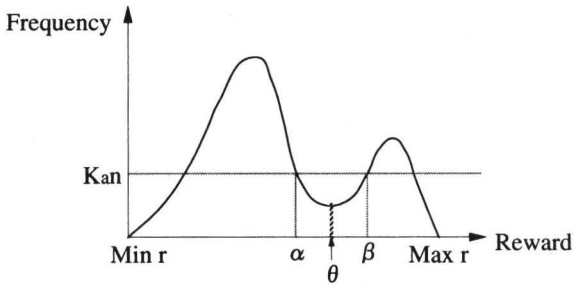

Figure 2: Divide condition

are neglected. (4) guarantees enough data to make an EOP. The method for clustering seems *ad hoc*. Of course, we can employ other standard clustering methods using variance. However, the clustering needed for this approach is simple and we consider the four parameters that we have employed characterize the robot behavior.

Data sets $R(v_i, a_j)$ are divided by θ through the following procedure for state division:

1. Take actions in the real world and collect quadruplet data $\langle v, \mathbf{s}, a, r\rangle$. Classify them into disparate sets

$$R(v_i, a_j) = \{\langle v, a, \mathbf{s}, r\rangle | v = v_i, a = a_j\}$$

2. For each $R(v_i, a_j)$ check whether v_i should be divided. Terminate if no state can be divided.
3. If $R(v_i, a_j)$ should be divided by the threshold θ as shown in Figure 2, define two classes:

$$Class\ 0 = \{v | \langle v, a, \mathbf{s}, r\rangle \in R(v_i, a_j), r < \theta\}$$

$$Class\ 1 = \{v | \langle v, a, \mathbf{s}, r\rangle \in R(v_i, a_j), r \geq \theta\}$$

4. Look for an EOP which can discriminate class 0 and class 1 in EOPs already acquired. If no proper EOP is found, then make a new EOP using class 0 and class 1 (see 2.4). If the EOP cannot be constructed because of the classification error, then abort to divide and go to 2.
5. Let us refer to the EOP selected at 4 as EOP_k. v_i is divided into new states v_{i0} and v_{i1} by EOP_k. Divide $R(v_i, a_j)$ into $R(v_{i0}, a_j)$ and $R(v_{i1}, a_j)$ for each a_j and go to 2.

$$R(v_{i0}, a_j) = \{\langle v, a, \mathbf{s}, r\rangle | v = v_i, a = a_j, EOP_k = 0\}$$

$$R(v_{i1}, a_j) = \{\langle v, a, \mathbf{s}, r\rangle | v = v_i, a = a_j, EOP_k = 1\}$$

2.4 Acquisition of EOPs

Assuming that sensor patterns classes 0 and 1 are given with respect to an action, a discriminant function can be obtained in the following manner [5]: The discriminant function is a function which takes as input a sensory vector $\mathbf{s}$, represented as a N-vector, and outputs a scalar value. It is represented as a sum of weighted components of a vector $\mathbf{x}$.

$$DF(\mathbf{s}) = \mathbf{w}^t \cdot \mathbf{s} + c; \quad \mathbf{s}, \mathbf{w} \in \mathbf{R}^N, c \in \mathbf{R}$$

where c is decided so that DF outputs a positive value for a sensor pattern in class 1 and a negative value for a sensor pattern in class 0. A value for $\mathbf{w}$ is determined to minimize the error rate [1] as follows:

$$\mathbf{w} = \mathbf{\Sigma}^{-1}(\mu 1 - \mu 0)$$

where $\mathbf{\Sigma}$ is a covariance matrix of class 0 and class 1. $\mu 1$ and $\mu 0$ are mean vectors of class 1 and class 0, respectively.

The error rates of DF are given by the following formulae:

$$e0 = m0/n0, \quad e1 = m1/n1$$

where $n0$ and $n1$ are the number of data sets in class 0 and 1, respectively. $m0$ is the number of data sets such that $DF \geq 0$ in class 0, and $m1$ is the number of data such that $DF < 0$ in class 1.

Here, the discriminant analysis generally takes much computational time for a large number of dimensions of the sensory data. Therefore, it is necessary to perform a kind of dimensionality reduction to reduce the size of the sensory data. However, our policy in this research approach is that the robot has no access to models based on human intuitions for segmenting features, such as lines, color regions, and so on. For this problem, our idea is to use principal component analysis (PCA) [1], which calculates orthogonal dimensions of maximal variation in the data. The Hotelling (aka Karhunen-Loeve) transform is used to rotate the axes, and only those dimensions that account for most of the variation are retained. The principle components of the visual data serve as low-level quantitative features [8] and represent spatial relations among pixels. Therefore, it is meaningful to refer to the principle components for analyzing vision data. Although we do not consider principle component analysis is ideal for the data compression, this approach is general insofar as it can be applied to any kind of sensory data. As better methods, we can consider to apply independent component analysis and Wavelet transform, which have interesting relations with human visual process.

The weight vector of a discriminant function represents the importance of each component of the sensory input for the classification. Thus, the weight vector defines attention control for pixel (or feature) selection. Several approaches can be considered for the pixel selection with referring to the weight vector. Our method employed in this paper is as follows.

Pick up the maximum and minimum values of absolute values of the weights. Then, we divide the range between the maximum and minimum values into ten (this number has been empirically determined) and acquire ten threshold values $(TH_1, TH_2, ..., TH_t, ..., TH_{10})$. With the threshold, the weight values are modified; if the value is less than the threshold, the weight is set as 0 and

the pixel is ignored. Thus, we obtain ten modified discriminant functions $MDF(t, \mathbf{s}), t = 1, ..., 10$.

Next, we check error rates of all modified discriminant functions. The modified discriminant function $MDF(t, \mathbf{s})$ that does not satisfy the following condition is discarded.

$$em0_t < 0.8 \ \& \ em1_t < 0.8$$

where $em0_t$ and $em1_t$ are error rates of the modified discriminant function and they are computed in the same way as $e0$ and $e1$.

For the remained discriminant functions, we evaluate performance of the threshold values by using the following evaluation function.

$$EF(t) = (em0_t + em1_t)/n$$

where n is the total number of data sets. Then, a threshold $t = l$ that gives the minimum value of the evaluation function $EF(t)$ is selected.

The original linear discriminant function divides the parameter space with a hyper plane based on distribution of data in the parameter space. In this case, outliers significantly influence the discrimination. On the other hand, performance of the modified discriminant function is evaluated based on only the number of misclassified data. That is, there is a possibility that the pixel selection using the modified discriminant function ignores the outliers.

By the pixel selection, the discriminant function can be represented with a small number of coefficients and focuses on important sensor components. Further, a combination of modified discriminant functions realizes nonlinear discrimination as discussed later. The efficiency of this attention control method is experimentally verified in Section 3. The idea of attention control proposed here is simple but general, and it can be applied to any kinds of sensory data.

The modified discriminant function $MDF(l, \mathbf{s})$ cut minor components off with a threshold value l and outputs a positive or negative value for an input data $\mathbf{s}$. With $MDF(l, \mathbf{s})$, we define a logical function EOP as follows:

$$EOP(l, \mathbf{s}) = \begin{cases} 1 & MDF(l, \mathbf{s}) \geq 0 \\ 0 & MDF(l, \mathbf{s}) < 0 \end{cases}$$

By iterating to collect data sets and obtain EOPs, the robot may generate a large number of states. In this case, identical states may exist. The states can be identified by their distribution of rewards across the possible actions. Suppose the robot has two actions and each state-action pair has a distribution of rewards that it has acquired so far. If, for two states, all corresponding actions have a similar reward value, the states can be treated as identical. The identified pair of states is coalesced into a single state. The new state is represented by combining the EOPs used to represent the previous two states. In other words, by iterating the processes of identifying and coalescing states, the robot is able to perform a complex, nonlinear discrimination of an n-dimensional parameter space.

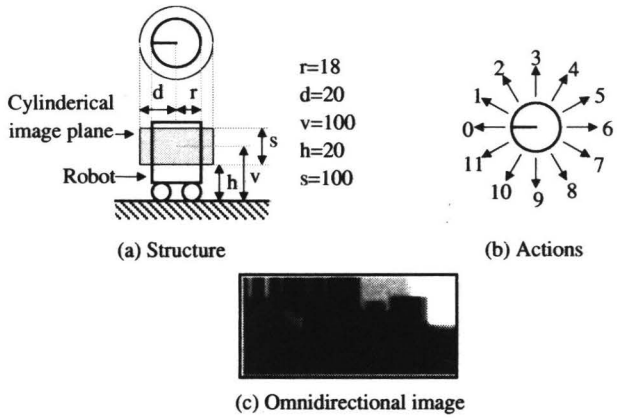

Figure 3: Robot in the simulator

3 Experimental Results

3.1 Robot and Environment

We have developed a robot simulator and used it to verify the proposed method. The robot in the simulator has an omnidirectional vision sensor. The robot can rotate in any of 12 directions and go forward, in other words, it has 12 actions. The configuration of the sensor and the actions are shown in Figure 3(a) and (b). An action a_i consists of a rotation of $30 \times i$ *degrees* and a forward movement of 15 *pixels* in the simulated environment shown in Figure 4 with some errors (The size of the simulated environment in Figure 4 is 640×480 *pixels*). The actual direction and distance of a_i follow the normal distribution whose means are $30 \times i$ *degrees* and 15 *pixels*, respectively. The sensor provides edge images of 43×18 *pixels* as shown in Figure 3(c) at each time step; and the robot acquires edge images for the state space construction by applying Sobel edge operators [2]. The acquired images are compressed by acquiring the principle components. In this experimentation, we have determined the number of the principle components with a accumulated proportion of 0.8.

3.2 Visual Sequentialization

The robot task in this experiment is obstacle avoidance. The robot receives a -1 reward when it collides with a wall and 0 otherwise. At the beginning, the robot knows nothing about the external world, it has no EOP, and its internal state space consists of only one state s_0. Figure 5 shows a state tree at a time when the robot created 9 EOPs. The combination of the EOPs represent the states. For example, state s_4 is represented as $EOP_0 EOP'_2 EOP'_4 EOP'_7$, and s_8 is represented as $EOP_0 EOP'_2 EOP'_4 EOP_7$. Those two states are discriminated by EOP_7. In the image of the EOP, bright and

[2] We have tried both of intensity images and edge images. As the results, we have found that the edge images are superior in efficient acquisition of EOPs to the intensity images in this simulated worlds.

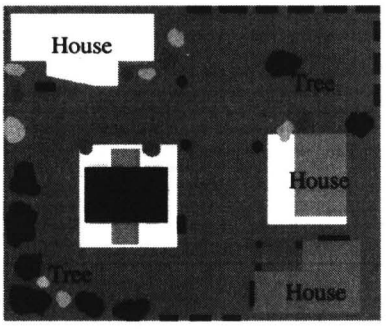

Figure 4: Simulated worlds

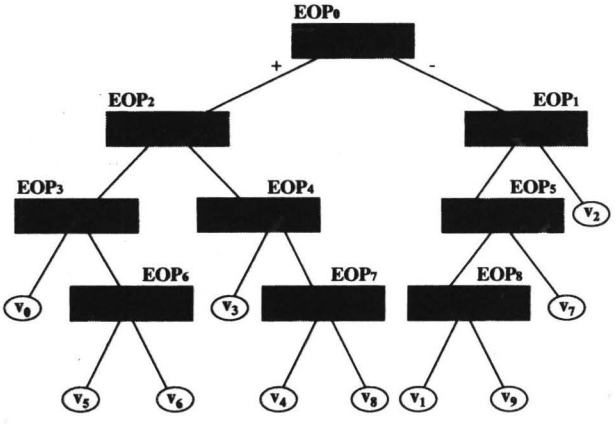

Figure 5: Visual sequentialization

dark gray areas indicate positive and negative components of the weight vector, respectively.

The tree structure of the state space represents how the robot continuously pays its attention in order to identify the states. In Figure 5, the robot coarsely observes with EOP_0, then changes its attention to more detailed regions with EOP_2, EOP_4, and EOP_7. This behavior called visual sequentialization is how our method constructs the state space.

3.3 Effectiveness of Attention control

To evaluate the effectiveness of our proposed method of attention control, we have experimented in the complex outdoor environment with the reactive robot behavior for avoiding obstacles. First, we have given a fixed number of data sets for verifying effectiveness of the attention control method. With 10000 data sets collected while the robot randomly moves, the robot has constructed a state space. In Figure 6, the vertical and horizontal axes indicate the error of pattern classification and the number of states, respectively. The error E is defined as:

$$E = 1 - \sum_{i=0}^{nv} \sum_{j=0}^{11} |np_{ij} - nn_{ij}|/n$$

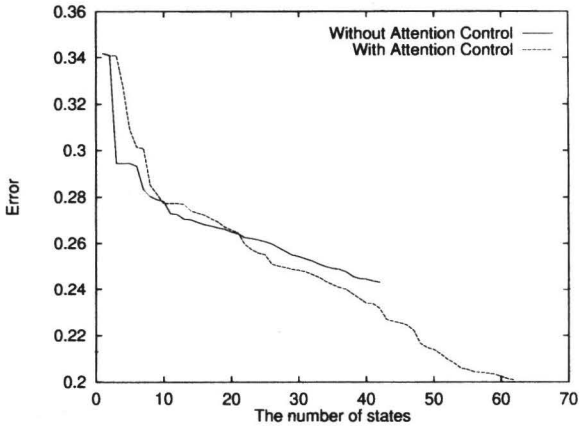

Figure 6: Error of pattern classification for 10000 data sets

where i, j are the IDs of *state* and *action*, respectively. nv and n are the number of acquired states and the number of all acquired data sets, respectively. For the obstacle avoidance, np_{ij} is the numbers of data sets that have a reward of 0 for an action a in a state s; and nn_{ij} is the number of data sets that have a reward of -1. That is, the error represents the normalized number of misclassified data sets. In the beginning, the error is large since the number of states is small. As the number of states increases, the error is reduced.

Figure 6 shows an interesting behavior of state space construction with the attention control. In the beginning, the error of the method using standard discriminant functions was smaller than the method using the attention control. Then the method using the attention control method became better at 20 states.

We analyze these phenomena as follows: In the beginning, the method using attention control suffers classification error in the attention control for each EOP. However, after acquiring some total amount of EOPs, the method performs effective nonlinear state division by combining the acquired EOPs.

Another characteristic is shown in Figure 7. The vertical axis indicates the number of nonzero components of weight vectors for EOPs. A small number of the nonzero components means that the robot refers a small amount of pixels in the image to distinguish the states. In other words, it saves computational time for sensory data processing. As shown in Figure 7, the method using standard discriminant functions needs more than three times of sensory data than the method using the attention control.

We have performed another experimentation in a more realistic situation. In this experimentation, the robot incrementally constructs a state space while collecting data sets and makes an EOP in a state when

1. 10 data sets for both classes, class 0 and class 1, are acquired,

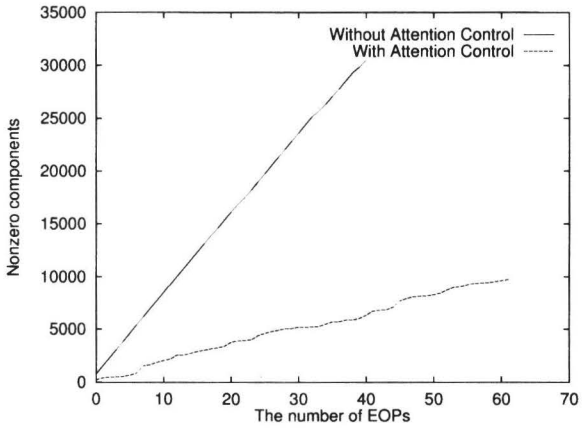

Figure 7: The number of nonzero components of the weight vectors

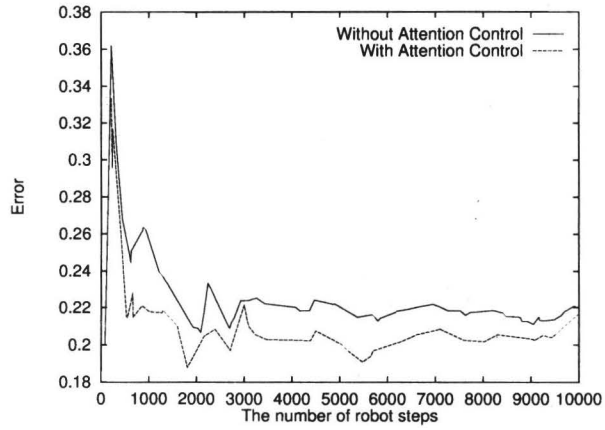

Figure 8: Error of pattern classification in incremental state space construction

2. or the number of data sets in the class that has a smaller number of data sets than the other becomes 10 % of the other.

The error of the attention control method was smaller than the method using standard discriminant functions as shown in Figure 8. In Figure 8, we could not find the similar phenomena to Figure 6, in which performance of the two methods is reversed. In the incremental construction, nonlinearity of the sensory data space is stronger since the robot needs to move by referring to the incomplete state space. We, therefore, consider that the method not using attention control could not represent good performance in the beginning. Figure 9 shows a more impressive result. The method using the attention control was better than seven times. We consider that, in the incremental state space construction, outliers are often acquired since the robot acquire data sets while randomly moving and the method of the pixel selection efficiently filters them out.

In this experimentation, we could verify the attention control method. Performance of the attention control method is better for the incremental state space construction than for the state space construction with a fixed number of data sets. In the incremental state space construction, nonlinearity of the sensory data space is strong; and our method is suitable to deal with such a complex classification problem. Further, our method significantly reduces pixel data needed for the state space construction. This means how the method works well as an attention control method.

4 Discussion and Conclusion

In this paper, we have proposed a coherent method for constructing a state space using attention control from physical sensory data. In addition to the approach's coherency, the proposed method has the following interesting point:

- It constructs a state space from a huge amount of physical sensory data.
- The EOP realizes a simple and general method of attention control.
- A combination of EOPs controls continuous changes in visual attention and represents a state.
- Attention control with the EOPs efficiently constructs a state space.

Finally, we discuss remaining problems and future directions for our research approach. One problem is how to determine the threshold for the attention control. In this paper, we have determined the threshold based on performance of each EOP. However, an EOP influences performance of other EOPs obtained after the EOP. That is, the threshold should be adjusted by taking all related EOPs in account. This threshold control is expected to construct a state space that has a more reasonable abstraction hierarchy.

In this paper, the EOPs have been made under the assumption that actions are already defined, but some tasks may require more precise actions and others can be done with coarser actions. In order to change the granularity, it is necessary to detect coarse actions or to identify similar actions. It maybe possible to segment actions with a method similar to that used above for sensory data segmentation.

This paper has discussed with a single simple robot task. However, a robot needs several tasks, for example avoiding obstacles, moving toward goals, following other robots and so on. A problem is how to implement the tasks in our approach. As one of the promising solutions, we consider to employ subsumption architecture [3]. The idea is to prepare an independent state space for each robot task and connect the modules with a hierarchical network. Another solution is to develop an abstract state space where we consider a robot behavior as an action. In other words, this idea is to construct a state space for representing relations between robot behaviors. The

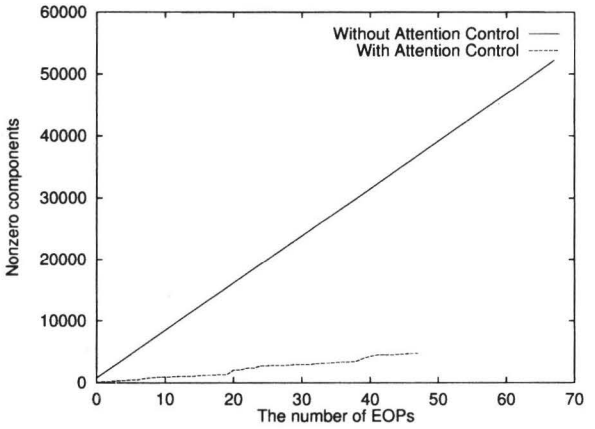

Figure 9: Nonzero components in incremental state space construction

later idea is more ideal, however it has a difficult problem of simultaneous construction of state spaces in different abstraction levels.

The method proposed in this paper obtains a state space for a given task and environment. To extend this method, we must consider how to obtain a general state space which can be used in other environments. Imagine if we place our robot in a different room and reconstruct the state space. It is expected that newly created EOPs will indicate the difference between the current room and the previous room, and EOPs that are commonly used indicate more general information. The final goal of our research approach is to obtain a general internal representation of a robot behaving in a class of environments.

References

[1] T. W. Anderson, An introduction to multivariate statistical analysis, John Wiley & Sons, Inc., 1958.

[2] D. H. Ballard and C. M. Brown, Principles of animate vision, CVGIP: Image Understanding, Vol. 56, No. 1, pp. 3-21, 1992.

[3] R. A. Brooks, Intelligence without representation, Int. J. Artificial Intelligence, Vol. 47, pp. 139-159, 1991.

[4] D. Chapman and L. P. Kaelbling, Input generalization in delayed reinforcement learning: An algorithm and performance comparisons, Proc. IJCAI, pp. 726-731, 1991.

[5] H. Ishiguro, R. Sato and T. Ishida, Robot oriented state space construction, Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 1496-1501, Nov. 1996.

[6] S. Mahadevan and J. Connel, Automatic programming of behavior-based robots using reinforcement learning, J. Artificial Intelligence, Vol. 55, pp. 311-365, 1992.

[7] R. A. McCallum, Overcoming incomplete perception with utile distinction memory, Machine Learning, pp. 190-196, 1993.

[8] H. Murase and S. K. Nayar, Visual learning and recognition of 3-D objects from appearance, Int. J. Computer Vision, Vol. 14, pp. 5-24, 1995.

[9] T. Nakamura and M. Asada, Motion sketch: Acquisition of visual motion guided behaviors, Proc. Int. Joint Conf. Artificial Intelligence, pp. 126-132, 1995.

[10] N. J. Nilsson (eds.), Shakey the robot, SRI Technical Note 323, 1984.

[11] M. Shanahan, Robotics and the common sense informatic situation, European Conf. Artificial Intelligence, pp. 684-688, 1994.

[12] E. Uchibe, M. Asada, and K. Hosoda, Vision based state space construction for learningmobile robots in multi agent environments, Proc. European Workshop on Learning Robots, pp.33-41, 1997.

[13] C.H. Watkins and P. Dayan, Technical note: Q-learning, Machine Learning 82, pp. 39-46, 1992.