Generated Image Feature based Selective Attention Mechanism by Visuo-Motor Learning

Takashi Minato **Minoru Asada**
Dept. of Adaptive Machine Systems, Graduate School of Engineering, Osaka University
2-1 Yamada-oka Suita Osaka 565-0871 Japan
Email: {minato,asada}@ams.eng.osaka-u.ac.jp
TEL&FAX: +81-6-6879-4180

Visual attention is an essential mechanism of an intelligent robot to avoid processing an enormous amount of data. Existing research typically specifies in advance the attention control scheme required for a given robot to perform a specific task. However, a robot should be able to adapt its own attention control for varied tasks and environments. In our previous work, we proposed a method for a mobile robot to generate a filter to extract an image feature by visuo-motor learning. The generated image feature extractor is considered to be generalized knowledge from which a kind of image feature should be extracted for the robot to accomplish a task of a certain class. In this paper, we propose an attention mechanism, by which the robot selects the generated feature extractors based on its task-oriented criterion. A subset of supervised data which gives the local information of the task makes the selective mechanism more effective. We discuss the results of applying the method to indoor navigation and soccer shooting tasks.

Key Words: Automation, Mobile Robot, Selective attention, Image feature generation, Image feature selection, Task-oriented

1. Introduction

Attention control is an essential mechanism for an intelligent robot to avoid processing an enormous amount of data. It is a data reduction process to facilitate decision making.With regard to visual attention control, it involves selection of viewpoint, focus, image features, and so on. Existing research typically specifies in advance the attention control scheme required for a given robot to perform a specific task. However, a robot should be able to adapt its own attention control for varied tasks and environments.

Human beings have very highly developed mechanisms of attention. Much research focused on early visual processing and proposed a computational model in which a bottom-up system computes low-level image features and saliency maps and a top-down system selects the salient parts (e.g., [1]). Some computer vision researchers proposed a viewpoint selection method to facilitate object recognition based on information gain (e.g., [2]). The mechanisms are intended to obtain a better observation for object recognition, but are not directly related to the physical actions needed to accomplish a given task.

Some robot researcher focused on the attention problem of robot vision. Vlassis et al. [3] extracted image features correlated with a mobile robot's self-localization from the observed images based on a probabilistic method. Kröse and Bunschoten [4] proposed a method to decide the robot's camera direction by maximizing information gain. Winters and Santos-Victor [5] proposed a method to extract pixels correlated with a robot's localization. These methods are considered to be task-relevant visual attention but are not related to any physical actions.

We have focused on visual attention control related to a robot actions to accomplish a given task and proposed a method in which a robot generates an image feature extractor (i.e., image filter) which is needed for the selection of actions through visuo-motor map learning [6]. The robot's learning depends on the experience gathered while performing a task. The robot's state is calculated in two stages. First the image feature is extracted from the local area of the observed image, and then the state is calculated from the entire area of the feature image. Consequently, a generalized feature extractor is generated because it works much like a bottleneck layer of neural network in the state calculation process. In this model, the robot uses only one feature extractor for a given task. It is, however, obvious that the robot needs to select and use multiple feature extractors properly to accomplish various tasks.

A number of connectionist models have been proposed that constitute systems that selectively respond to visual stimuli. Scheier and Egner [7] proposed a system that selectively connects image feature maps with robot's actions according to saliency. The connection is given a priori and has less adaptability. A simulated nervous system was proposed that learns the connection based on the co-occurrence between sensor activities of NOMAD [8]. In this system, the robot follows its innate preference and, therefore, cannot learn any given task.

Some research has addressed a method of feature selection based on task-relevant criteria. McCallum [9] proposed a method in which a robot learns not only its action but feature selection using reinforcement learning.

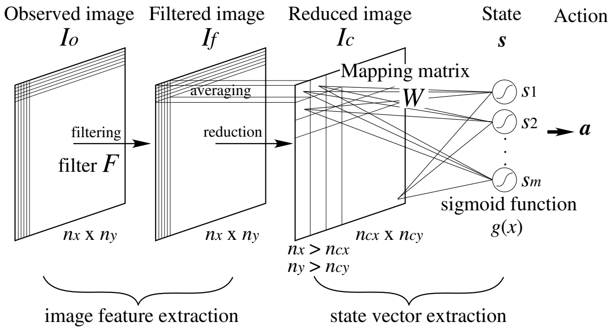


Fig. 1. Image feature generation model

Mitsunaga and Asada [10] proposed a method to select a landmark according to the information gain on action selection. In these methods, however, the image features to detect the landmarks from the observed image is given a priori. It is desirable that the image feature adapts to environmental changes.

This paper proposes a method in which a robot learns to select image feature extractors generated by itself according to a task-relevant criterion. The generated feature extractors are not always suitable for new tasks, though they are generalized. The robot must learn to select them to accomplish the task. The criterion of selection is the information gain calculated from given task instances (supervised data). Furthermore, a part of supervised data which gives the local information of the task makes the selective mechanism more effective. The method is applied to indoor navigation and soccer shooting tasks.

2. The basic idea

Image feature extraction is necessary in a robot's visual attention and as well as in human visual processing, consists of a low-level feature extraction process and a high-level recognition process. In the proposed method, a robot generates an image feature extractor that is necessary for the action selection through visuo-motor map learning [6] as shown in Fig. 1. The state calculation process is decomposed into feature extraction and state extraction. A robot learns the effective feature extractor and state mapping matrix for a given task through a mapping from observed images to supervised actions. During feature extraction, the interactions between raw data are limited to local areas, while the connections between the filtered image and the state spread over the entire space to represent non-local interactions. We, therefore, expect that the feature extractors are more general.

The robot calculates the filtered image I_f from the observed image I_o using the feature extractor F. To avoid the *curse of dimensionality*, the size of I_f is reduced to I_c. The state $\boldsymbol{s}$ is calculated from I_c by the sum of weighted pixel values (the weight matrix is W). The robot decides the appropriate action for the current state $\boldsymbol{s}$. The function model of the feature extractor is given, and the robot learns its parameters and the mapping matrix by maximizing the information gain of $\boldsymbol{s}$ with respect to action $\boldsymbol{a}$.

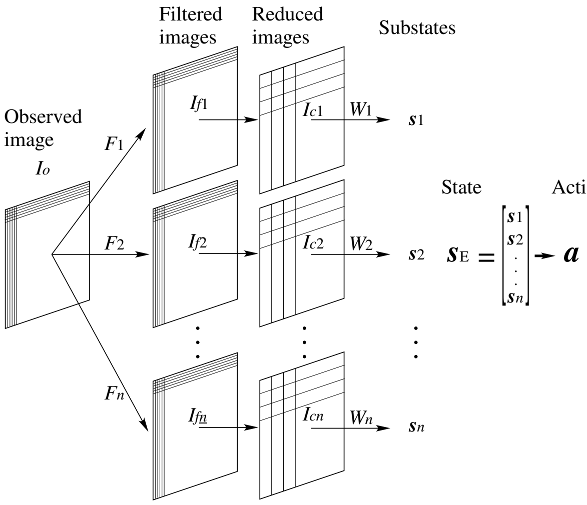


Fig. 2. Image feature selection model

The robot, which generates one feature extractor for a given task, obviously needs multiple feature extractors for more complex tasks. It is unnecessary to learn a feature extractor for every given task. The generated feature extractor must be generalized to make the robot more adaptable.

In the proposed method, the robot reuses a number of generated feature extractors from past experiences and selects effective ones for action decision. The system is shown in Fig. 2. The robot is given a number of different feature extractors, but must select which extractors are effective for given task. The robot, therefore, learns the state mapping matrix using the supervised data and evaluates which feature extractor is appropriate from the distribution of supervised data on the learned state space. If the robot uses all of the supervised data in the evaluation, optimality in a local part of the task is lost. To evaluate the effectiveness in the local task, the robot estimates which local task it is performing from the history of observations and selects the feature extractor using a portion of the supervised data corresponding to the local task.

3. Selective attention mechanism based on generated image feature extractors

3.1 The system overview

The robot is given n different feature extractors $(F_i, i = 1 \dots n)$ and calculates the substate $\boldsymbol{s}_i$ using the mapping matrix W_i corresponding to F_i. Each mapping matrix is learned by maximizing the information gain of $\boldsymbol{s}_E$ (direct product of $\boldsymbol{s}_1 \dots \boldsymbol{s}_n$) with respect to the supervised action $\boldsymbol{a}$.

The robot selects the feature extractor which has a maximum expected information gain and decides the appropriate action for the substate calculated using the selected feature extractor. It cannot always decide the appropriate action using one feature extractor. It, therefore, estimates the reliability of selected feature extractors and selects repeatedly until the reliability exceeds a given threshold.

For evaluation in the local task, the supervised data is segmented in temporal order. The robot selects a sub-supervised data according to the history of observation and selects feature extractors to decide an action using the selected one.

3.2 The model of feature extractor

In this paper, three feature extractor models are used.

- 3×3 spatial filter F_s, the parameter $\boldsymbol{f}_s \in \Re^9$:

$$\begin{aligned}\bar{I}_{x,y} = \; & f_{s1} I_{x-1,y-1} + f_{s2} I_{x,y-1} + f_{s3} I_{x+1,y-1} \\ & + f_{s4} I_{x-1,y} \;\; + f_{s5} I_{x,y} \;\; + f_{s6} I_{x+1,y} \\ & + f_{s7} I_{x-1,y+1} + f_{s8} I_{x,y+1} + f_{s9} I_{x+1,y+1},\end{aligned} \tag{1}$$

$$I_{f x,y} = g\left(\bar{I}_{x,y}\right). \tag{2}$$

- Color filter F_c, the parameter $\boldsymbol{f}_c \in \Re^3$:

$$\bar{I}_{x,y} = f_{c1} I_{r\; x,y} + f_{c2} I_{g\; x,y} + f_{c3} I_{b\; x,y}, \tag{3}$$

$$I_{f x,y} = g\left(\bar{I}_{x,y}\right), \tag{4}$$

- Temporal filter F_m, the parameter $\boldsymbol{f}_m \in \Re^5$:

$$\bar{I}_{x,y} = \sum_{i=1}^{5} f_{mi} I_{t-i+1\; x,y}, \tag{5}$$

$$I_{f x,y} = g\left(\bar{I}_{x,y}\right), \tag{6}$$

where x and y denote the position of the pixel, I, I_r, I_g, I_b; the gray, red, green and blue components of the observed image, respectively, I_t, the gray component of the observed image at time t, and $g(\cdot)$, a sigmoid function.

3.3 State learning

First the robot collects supervised successful instances of the given task for T episodes. An episode ends when the robot accomplishes the task. An instance consists of the observed image I_o and a given action $\boldsymbol{a} \in \Re^l$. An instance of ith episode at time t is shown as following:

$$u_t^i =< I_{ot}^i, \boldsymbol{a}_t^i > . \tag{7}$$

Next the robot learns the mapping matrices. The substate $\boldsymbol{s}_i \in \Re^m$ is calculated as following:

$$\begin{aligned}\boldsymbol{s}_j &= \boldsymbol{g}\left(W_j \boldsymbol{i}_{cj}\right), \\ \boldsymbol{g}(\boldsymbol{x}) &= (g(x_1), \ldots, g(x_m))^T,\end{aligned} \tag{8}$$

where $\boldsymbol{i}_{cj} \in \Re^{m_{cx} n_{cy}}$ denotes the vector of the jth I_{cj}, and $W_j \in \Re^{m \times n_{cx} n_{cy}}$ is the jth mapping matrix.

The evaluation function used to learn W_j is to maximize the information gain of $\boldsymbol{s}_E$ with respect to $\boldsymbol{a}$. It is equivalent to minimizing the following risk function R (e.g., [3]).

$$R = -\frac{1}{N} \sum_{i}^{N} \log p(\boldsymbol{a}_i | \boldsymbol{s}_{Ei}), \tag{9}$$

where N denotes the number of instances. The probability density functions are represented by kernel smoothing.

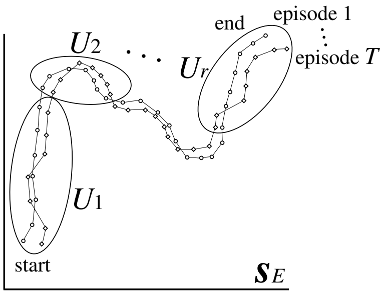


Fig. 3. Segmentation of supervised data

Using the gradient method, the mapping matrices W_j, which minimize R, are obtained.

$$W_j \leftarrow W_j - \alpha \frac{\partial R}{\partial W_j}, \tag{10}$$

where α is a step size parameter.

3.4 Feature extractor selection

All instances of U are divided into r subsets $U_j, j = 1 \ldots r$ before performing the task (Fig. 3). The subsets are arranged in time order. The choice of r includes a trade-off between a locality of the evaluation and a reliability of the action decision. To evaluate it, U is divided so that instances of similar state and action are included in a subset. The following vector $\boldsymbol{c}_t^i$ is defined from the instance u_t^i, and U is divided by applying the ISODATA algorithm for the set $\{\boldsymbol{c}_t^i\}$.

$$\boldsymbol{c}_t^i = \left(\boldsymbol{s}_{Et}^i, \boldsymbol{a}_t^i, \frac{t}{L_i}\right), \tag{11}$$

where L_i is the time taken to accomplish the task. The value of each component is normalized to [0,1]. To avoid an aliasing problem, the robot always uses the two neighboring subsets to evaluate the effectiveness of a feature extractor.

The robot executes the following process at every interval.

1) Selecting subsets of instance;
   Select subsets of instance $\mathcal{U}$ according to a procedure shown in the next section. $k = 0$.
2) Calculating a reliability of action decision;
   Calculate substates $\boldsymbol{s}_{o1}, \ldots, \boldsymbol{s}_{ok}$ corresponding to the selected $k (\geq 0)$ feature extractors and the entropy of the action $H_{\mathcal{U}}(\boldsymbol{a}|S_o)$ using the instances in $\mathcal{U}$.

   $$H_{\mathcal{U}}(\boldsymbol{a}|S_o) = -\sum P_{\mathcal{U}}(\boldsymbol{a}|S_o) \log P_{\mathcal{U}}(\boldsymbol{a}|S_o), \tag{12}$$

   where $S_o = \{\boldsymbol{s}_{o1}, \ldots, \boldsymbol{s}_{ok}\}$. $H_{\mathcal{U}}(\boldsymbol{a}|S_o)$ means an uncertainty of the action decision. Evaluate the uncertainty using a threshold H_{th}.
   - If $H_{\mathcal{U}}(\boldsymbol{a}|S_o) \leq H_{th}$, then go to 4.
   - Otherwise, if $k = n$ and $\mathcal{U} = U$, then go to 4.
   - Otherwise, if $k = n$ and $\mathcal{U} \neq U$, then go to 1 with $\mathcal{U} = U$.
   - Otherwise, go to 3.
3) Selecting a feature extractor;

Calculate an expected entropy of the action for each unselected feature extractor F_u. The expected entropy is:

$$\sum_{\mathcal{U}} P_{\mathcal{U}}(\boldsymbol{s}_u) H_{\mathcal{U}}(\boldsymbol{a}|S_o, \boldsymbol{s}_u), \tag{13}$$

where $\boldsymbol{s}_u$ is a substate corresponding to F_u. Select the feature extractor which has the minimum entropy, that is, has the maximum information gain. $k \leftarrow k+1$. go to 2.

4) Deciding an action;
Execute the following action $\boldsymbol{a}$:

$$\boldsymbol{a} = \arg\max_{\boldsymbol{a}'} P_{\mathcal{U}}(\boldsymbol{a}'|S_o). \tag{14}$$

3.5 Selecting subsets of instance

The robot selects subsets of instance $\mathcal{U}$ which are used to calculate a probability and an entropy according to the states $S_{ot-1}, \ldots, S_{ot-h}$ observed in the past h steps. Each S_{ot} consists of a number of substates. The size of S_{ot} is differs depending on t, because the number of the selected feature extractors is different.

For each subset U_i the robot calculates a matching ratio, P_{bi}, that S_{ot} satisfies Eq. 15 in h substates. If the ratio is greater than a threshold P_{bth}, U_i and U_{i+1} are added to $\mathcal{U}$. U_{i+1} is the successor of U_i. If there is no P_{bi} which is greater than P_{bth}, the robot uses all instances ($\mathcal{U} = U$).

$$P_{U_i}(S_{ot}) > 0. \tag{15}$$

4. Experiment

4.1 Experimental setting

We use a small mobile robot which is about 40 cm high and has a camera with a fixed orientation to look ahead at the floor. The task is to move along a given path to a destination. The size of I_o and I_f in pixels is 64×54 and that of I_c is 8×6. The robot is controlled at the rate of 15 Hz. Each pixel value of I_c is the average value of the corresponding region in I_f. We defined the dimension of substate as $m = 1$.

The robot can move at a translational speed v and a steering speed ω independently, so the action vector is represented as follows:

$$\boldsymbol{a} = (v, \omega)^T. \tag{16}$$

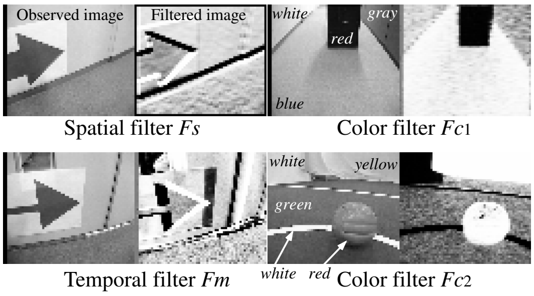
To reduce the computation cost, we discretized the state and action space and calculated the probabilities. The thresholds are set as $H_{th} = 0.4$ and $P_{bth} = 0.8$.

4.2 Feature extractors

We prepared four feature extractors shown in Fig. 4. F_s, F_{c1}, F_{c2}, F_m are generated in tasks A, B, C, and D (Fig. 5), respectively. The environment of task A, B, and C is a corridor in a laboratory and that of task D is a field of robot soccer.

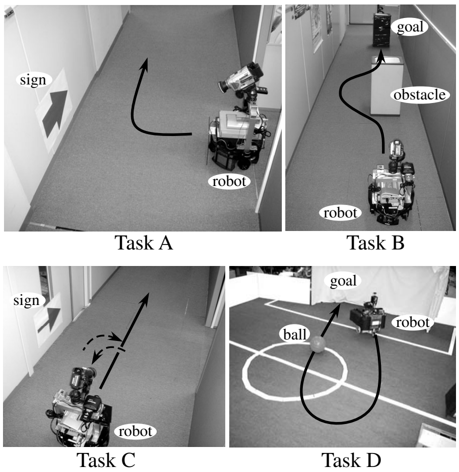
The feature extractors have the following characteristics.

- F_s: Emphasizing and inhibiting horizontal edge
- F_{c1}: Inhibiting red
- F_{c2}: Emphasizing red and yellow, and inhibiting white
- F_m: Emphasizing current image and inhibiting past image



Spatial filter *Fs*
Color filter *Fc*1
Temporal filter *Fm*
Color filter *Fc*2

Fig. 4. Feature extractors



Task A
Task B
Task C
Task D

Fig. 5. Tasks in which the robot learned feature extractors

4.3 Feature extractor selection

The robot was given F_s, F_{c1}, F_m and learned task 1 shown in Fig. 6. The robot moves to the front of the door and waits for it to open. It moves to the destination after the door opens. The environment is same as that of tasks A, B, and C. We gave three episodes of successful instances ($L = 234, 254, 233$). After learning, the robot divided all instances into 13 subsets. We set the history length $h = 10$.

Fig. 7 shows the learned behavior and Fig. 8 shows the selected feature extractors at each time step. The selected feature extractors, their number, and their order change according to the situation. The average number per step is 1.57.

Fig. 9 shows the selected subsets of instances at each step. When the robot cannot choose an action from

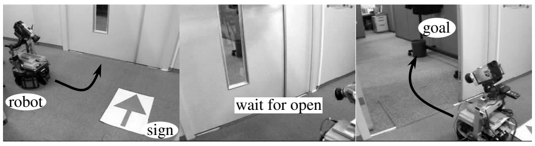


Fig. 6. Task 1

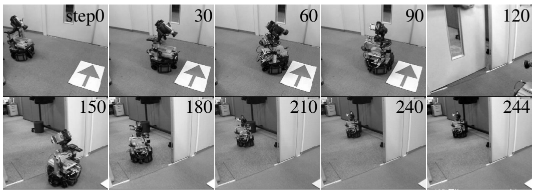


Fig. 7. Resultant behavior (Task 1)

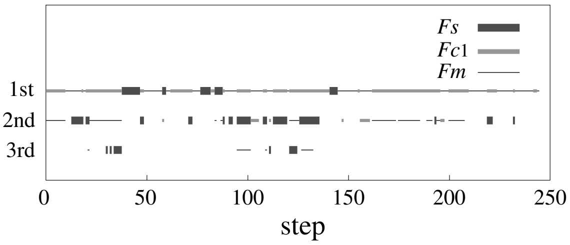


Fig. 8. Selected feature extractors with subsets of instance (Task 1)

the selected subsets because of low reliability, it uses all instances to decide again. $\mathcal{U}$ in the figure shows the step when the robot can choose an action from the selected subsets. If the robot chooses an action with few feature extractors in the past, many subsets of instance are used because the number of subset satisfies Eq.15 increases. We verify that the robot accomplishes the task selecting effective feature extractors.

4.4 Verification of subset of instance

To verify the subset of instances, we performed an experiment which was the same as Sec. 4.3 except the procedure to select the subsets. In this experiment, the robot always selects all instances. Fig. 10 shows the selected feature extractors at each step. The feature extractor that the robot selects first does not change. In this result, F_{c1} is always selected at first. The average number of the selected feature extractors per step is 1.97, which is larger than the result of Sec.4.3. This means that using all instances brings about inefficient selection for the action decision. Hence, the robot effectively decides the action using part of the instances.

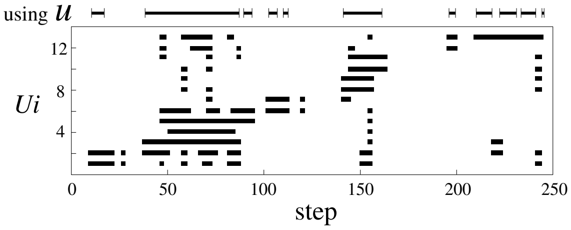


Fig. 9. Selected subsets of instance (Task 1)

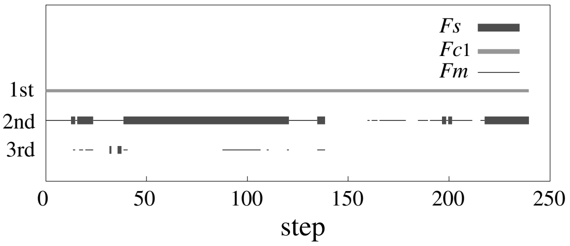
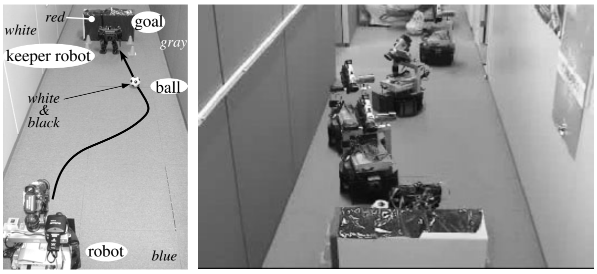


Fig. 10. Selected feature extractors with all instances (Task 1)



(a) Task 2

(b) Resultant behavior

Fig. 11. Task 2 and resultant behavior

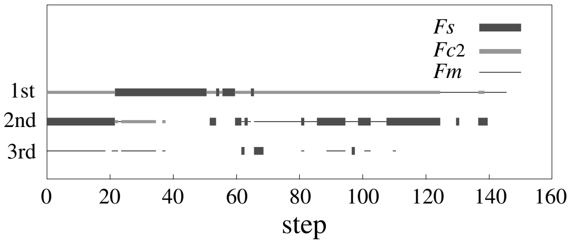


Fig. 12. Selected feature extractors (Task 2)

4.5 Reuse of feature extractors

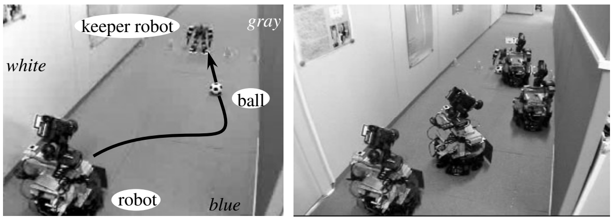
In the above, it has been shown that the robot can accomplish new tasks with feature extractors generated in the past tasks. This section shows that a generated feature extractor can be reused in new environment.

The robot was given F_s, F_{c2}, F_m, and learned task 2 shown in Fig. 11 (a). The environment was same as that of task A, B, and C, however, F_{c2} was generated in the different environment. We gave three episodes of successful instances ($L = 132, 131, 134$). The number of the subset was 7 and the history length, h, was 15.

Fig.11 (b) shows the learned behavior and Fig. 12 shows the selected feature extractors at each time step. The robot selected F_{c2} at first in some situations. This indicates that F_{c2} is effective for this task. Hence, the generated feature extractor can be used in different environment and task.

4.6 Irrelevant feature extractor

Contrary to the previous section, this section shows that the robot can neglect the irrelevant feature extractor to the task. The robot was given F_s, F_{c1}, F_{c2}, and learned



(a) Task 3

(b) Resultant behavior

Fig. 13. Task 3 and resultant behavior

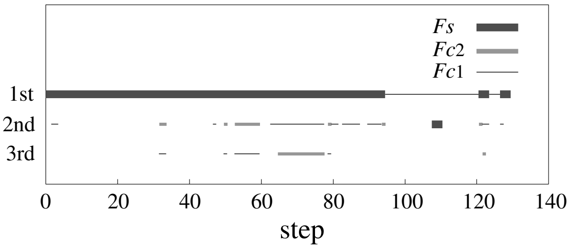
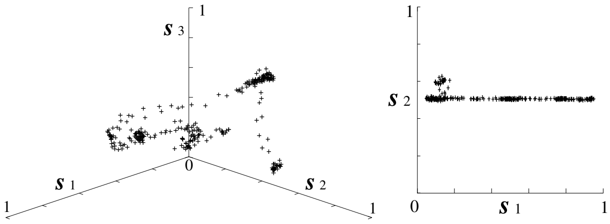


Fig. 14. Selected feature extractors (Task 3)



(a) Perspective view

(b) Projection onto $\boldsymbol{s}_1$–$\boldsymbol{s}_2$ plane

Fig. 15. Distribution of supervised data

task 3 shown in Fig. 13 (a). The environment was same as that of task 3 except the goal. We gave two episodes of successful instances ($L = 143, 153$). The number of subset was 6 and history length, h, was 15.

Fig. 13 (b) shows the learned behavior and Fig. 14 shows the selected feature extractors at each time step. The importance of F_{c2} is low because the robot rarely selected it. It can also be seen from the distribution of the instances. Fig. 15 (a) shows the distribution of all instances on the learned state space. $\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3$ are the substates of F_s, F_{c2}, F_{c1}, respectively. Fig.15 (b) shows the $\boldsymbol{s}_1$–$\boldsymbol{s}_2$ plane. The value of state $\boldsymbol{s}_2$ is irrelevant to identifying the state because the instances do not distribute along the axis of $\boldsymbol{s}_2$. This means that F_{c2} is irrelevant to the action decision. F_{c2} emphasizes red and yellow, and inhibits white. It is, however, useless to identify the state in the environment. Hence, the robot can neglect an irrelevant feature extractor.

5. Conclusion

This paper has proposed a method in which a robot learns to select image feature extractors generated by itself according to task-relevant criterion. A portion of supervised data which gives the local information of the task makes the selection of feature extractors more effective.

In the proposed method, a robot can accomplish more complicated tasks using multiple feature extractors. This paper, however, does not mention a method to generalize a feature extractor. This must be considered in order to increase the robot's adaptability.

References

[1] J. M. Wolfe, K. R. Cave, and S. L. Franzel. Guided search: An alternative to the feature integration model. *Journal of Experimental Psychology: Human Perception and Performance,* 15(3):419–433, 1989.

[2] T. Arbel and F. P. Ferrie. Viewpoint selection by navigation through entropy maps. In *Proceedings of the Seventh International Conference on Computer Vision*, pages 248–254, 1999.

[3] N. Vlassis, R. Bunschoten, and B. Kröse. Learning task-relevant features from robot data. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pages 499–504, 2001.

[4] B. J. A. Kröse and R. Bunschoten. Probabilistic localization by appearance models and active vision. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pages 2255–2260, 1999.

[5] N. Winters and J. Santos-Victor. Visual attention-based robot navigation using information sampling. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1670–1675, 2001.

[6] T. Minato and M. Asada. Towards selective attention: Generating image features by learning a visuo-motor map. *Robotics and Autonomous Systems*, 45:211–221, 2003.

[7] C. Scheier and S. Egner. visual attention in a mobile robot. In *Proceedings of the International Symposium on Industrial Electronics*, pages 48–53, 1997.

[8] J. L. Krichmar and J. A. Snook. A neural approach to adaptive behavior and multi-sensor action selection in a mobile device. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3864–3869, 2002.

[9] A. K. McCallum. Learning to use selective attention and short-term memory in sequential tasks. In *Proceedings of the foruth International Conference on Simulation of Adaptive Behaivior: From Animals to Animats 4*, pages 315–324, 1996.

[10] 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.