Systems and Computers in Japan, Vol. 33, No. 14, 2002
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J84-D-II, No. 9, September 2001, pp. 2121–2130

Active Perception and Recognition Learning System Based on Actor-Q Architecture*

Katsunari Shibata,[1] Tetsuo Nishino,[2,†] and Yoichi Okabe[2,‡]

[1]Department of Electrical and Electronic Engineering, Oita University, Oita, 870-1192 Japan

[2]Research Center for Advanced Science and Technology, The University of Tokyo, Tokyo, 153-0041 Japan

SUMMARY

This paper proposes the Actor-Q architecture, which is a combination of Q-Learning and Actor-Critic architecture, as well as the active perception and recognition learning system based on that architecture. In Actor-Q architecture, the system output is divided into the "action," which is a discrete value as intention, and the "motion," which is a continuous-valued vector. As the first step, the "action" is determined from Q-values. If the "action" is accompanied with a "motion," the "motion" is executed according to the corresponding Actor output. Q-value is learned by Q-learning, and Actor is trained with the Q-value corresponding to that "action" on behalf of the Critic output. In this study, the action is defined as the decision of the sensor motion or the recognition of the respective pattern. Q-value is assigned to each of those. When the sensor motion is selected, the sensor is moved according to the Actor output. When recognition is selected, the recognition result that the presented pattern is the one corresponding to the selected Q-value is output. The Q-value is learned, using the reinforcement signal representing the true/false of the result. Both Q-value computing module and Actor are composed of neural networks, with the visual sensor signals as input. By this architecture, the following three problems of the conventional active perception and recognition learning system are dissolved. (1) The sensor can be trapped in a local maximum of the recognition evaluation. (2) It is necessary that the recognition output should be evaluated for each time-step, and the reinforcement signal with a continuous value should be provided. (3) The system cannot decide by itself the timing to output the recognition result. The above effect was verified by some simulations, using the visual sensor with nonuniform sensor cells.
 Syst Comp Jpn, 33(14): 12–22, 2002; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/scj.10207

Key words: reinforcement learning; neural network; active perception and recognition; Actor-Q architecture; decision making.

1. Introduction

Biological species including humans acquire external information, which is necessary in their activities, through various sensory organs. The amount of the external information, however, is tremendous, which makes it difficult and less efficient to acquire all information as to details. In

*The basic idea for this study was presented at the Technical Group Meeting, IEICEJ, March 1997.

†Presently with Japan Oracle Co.

‡Presently Professor, Graduate School of Engineering and Director, Information Technology Center at the University of Tokyo.

Contract grant sponsor: Supported in part by a Grant-in-aid for Scientific Research on Priority Area from the Ministry of Education Japan ["Emergent systems" (No. 264)].

order to cope with this situation, the biological subject has the active perception function, whereby it actively moves the sensory organ and efficiently acquires the necessary information.

The visual sensor (eye) acquires the largest amount of information among the human sensory organs. In the eye, the distribution of the sensor cells on the retina is nonuniform. It is generally considered that correct recognition is realized efficiently by the division of roles as follows. The whole sensor is used to acquire the global external situation, and the details are acquired by moving the central part which has a higher sensor cell distribution density.

A human learns, whenever necessary, to view and recognize correctly an unknown object. At birth there is no knowledge concerning an object of recognition as to what the object is, or how to view the object in order to recognize it. Such knowledge is considered to be acquired by learning after birth.

In the engineering pattern recognition considered at present, the usual method is that the pattern is excised by segmentation, and the feature is extracted from the excised pattern as preprocessing. Then, recognition is tried based on the features [1, 2]. Fukushima realized a pattern recognition process, which is robust against the position translation, in a form close to the biological function, where the absorption of the position translation and the feature extraction are iterated using the neural network [3].

Recently, the active behavior in recognition is considered to be interesting both from the viewpoint of "brain" as well as from the viewpoint of "robot" [4, 5]. In the former, the model for the selective attention [6] is used as the basis, and the pattern segmentation is emphasized mostly in the form of "attention."

The following property is also shown. Using the visual sensor signal as the input to the recurrent neural network and training the network for recognition, the function is realized whereby the context is extracted from the past visual sensor signal to be retained, providing the selective attention to the next recognition process. It is also shown that the associative memory function is realized in storing the context [7].

In studies on robots, on the other hand, the system that moves the visual sensor is already constructed [5], where, however, the detection and tracking of the moving object are mostly emphasized. It is not intended to learn the point to view for recognition.

Recently, in addition, "reinforcement learning" is considered to be interesting because of its autonomous, adaptive, and purposive learning ability. In the past, reinforcement learning has been considered as training for action planning. There is an approach, on the other hand, where the sensor signal is directly input to the neural network, and the motor is driven by the output signal. Then, by applying the reinforcement learning, the process from the sensors to the motors, including recognition and attention, is comprehensively learned with harmonization [8].

Among those studies, Whitehead and Ballard considered the block restacking problem, and the selection function of the block to focus on is acquired by Q-learning [9], a type of reinforcement learning [10]. In their system, however, the recognition is not explicitly handled. As another point, even if the block to focus on is selected, the action to move the sensor to that position is not considered.

In contrast to those studies, the authors are trying to construct the engineering system by applying reinforcement learning where the sensor motion, which is as adequate as the human action, is acquired by learning, so that efficient pattern identification is realized. It is also intended to demonstrate the possibility that reinforcement learning is used in the biological acquisition of sensor motion and recognition.

A method was proposed whereby the sensor signals are directly input to the neural network, and the sensor motion is learned as a continuous value, based only on the evaluation of the recognition result [11]. The system in this method, however, is not one that can respond to the delayed reward, making it necessary to output the recognition result in each unit time, and the difference between the supervisor pattern and the output pattern must be used as the scalar evaluation with a continuous value. Because of this, there arose a problem that the recognition output pattern can be trapped in a pattern which is locally close to the supervisor pattern, making it impossible to arrive at the correct solution.

As another point, the timing for the system to decide on the final recognition result is set as a time after a certain period has elapsed. Consequently, it may happen that, even if the sensor moved to the position to make the recognition, the recognition result is not produced until the time comes. Conversely, it may happen that the time limit arrives before the sensor moves to the position to make the recognition, and the correct recognition result cannot be output. In order to set this time limit, it is necessary beforehand to know the recognition problem to be given; this is contrary to the expected autonomous and flexible ability of the reinforcement learning.

This paper proposes Actor-Q architecture, where the system output is divided into "action," which is a discrete value, and "motion," which is a continuous vector, and the output and learning are executed. The proposed architecture is applied to the active perception and recognition learning system. It is proposed to add a function that determines whether the sensor should be moved or the recognition result should be output, so that the function is acquired by the reinforcement learning as well as the sensor movement. Using the visual sensor with a nonuniform sensor cell density, the behavior of the system is examined. We examine whether or not it is possible to move the sensor to match

the part with the higher sensor cell density to the point in the pattern to focus on, and recognize the pattern correctly by deciding by itself the timing to output the recognition result.

2. Actor-Q Architecture

This section describes the Actor-Q architecture proposed in this paper, as well as the active perception and recognition learning system based on that architecture. First, the system output is divided into "action," which is the discrete intention, and "motion," which is the vector with a continuous value. "Action" is determined first. If the action required "motion," "motion" is further determined. Consider, as an example, that "action" is composed of "run," "walk," and "stop." When "walk" is selected as the "action," "motion" is determined to specify the command value to the foot muscle.

Q-value is used to determine the "action," and Q-learning is used for learning. Actor of Actor-Critic [12] is applied for determining "motion." Usually, Critic output is used for the learning of Actor. It is noted, however, that both Critic learning and Q-learning are based on TD (temporal difference) learning [13]. Consequently, Critic is not provided in particular, and the Q-value corresponding to the "action" is used as Critic output. This structure is called Actor-Q architecture.

The idea is implemented by two neural nets shown in Fig. 1—Q-net and Actor-net. Each output of Q-net corresponds to the action, and each output of Actor-net corresponds to the output destination of the motion signal. If the input signals are the same, it is possible to combine the two networks into one, by sharing the input and the hidden layers. When more than one "action" requires the "motion," it is possible to prepare multiple Actor-nets, and the output of the Actor-net corresponding to the selected action is output as "motion" by gating Actor-nets.

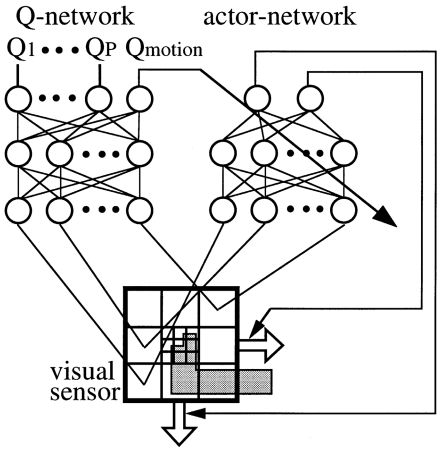


Fig. 1. The active perception learning system based on Actor-Q architecture proposed in this paper.

Actor-Q architecture is applied to the active perception and recognition learning system as follows. In this case, to "output the recognition result that the pattern is *p*" and to "move the sensor" are considered as actions. When there exist *P* patterns to be presented, there can be $(P + 1)$ actions, as shown in Fig. 1, which are *P* actions to give the recognition result for the respective pattern, as well as an action to move the sensor. *Q*-value is assigned to each of those, and an action is selected based on the *Q*-values.

If "recognition" is selected as the action, it is examined whether the result is correct or incorrect. A reward is given if the result is correct, but nothing is given if the result is incorrect. This completes a trial. *Q*-value after completing a trial is set as 0.0. Then, the expression for Q-learning for "recognition" action "*Recog*(*p*)" to decide that the pattern is *p*, is given as

$$Q(s(t),'' Recog(p)'') = (1-\alpha)Q(s(t),'' Recog(p)'') + \alpha r \quad (1)$$

$s(t)$ is the sensor input (state) at time t. α is the learning rate. r is the reward, which is set as 1.0 for the correct result, and 0.0 for the incorrect result. The above is similar to the case of training a monkey for recognition, where a reward is given for the correct result.

In this study, the neural network is trained, and the training signal is given as

$$Q_{train}(s(t),'' Recog(p)'') = r \quad (2)$$

The training by this supervisor signal is not iterated but is executed only once, to train only for the corresponding *Q*-value.

When the action "sensor motion" is selected, the sensor is moved according to the Actor-net output. In this case, the trial is not completed. New input signals are given after the sensor movement, and the next action is selected. Only the corresponding *Q*-value is learned with the following training signal only once, according to the usual Q-learning with zero reward.

$$Q_{train}(s(t),'' motion'') = \gamma \max_a Q(s(t+1), a) \quad (3)$$

γ is the discount factor, and a is the possible action.

The sensor movement **m** specifies the sensor velocity in *x*- and *y*-axis directions. It is given as follows by adding a random vector **rnd** to Actor-net output $\mathbf{o}_m$:

$$\mathbf{m} = \mathbf{A}(\mathbf{o}_m + \mathbf{rnd}) \tag{4}$$

A is a constant matrix composed only of diagonal elements.

Actor-net is trained by the following training signal:

$$\mathbf{O}_{m,train} = \mathbf{O}_m + (\gamma \max_a Q(s(t+1), a) - \max_a Q(s(t), a))\mathbf{rnd} \tag{5}$$

This expression can be derived as follows. Consider Actor-Critic executing the usual TD learning, with zero reinforcement signal. The learning equation for Actor is given as

$$\mathbf{O}_{m,train} = \mathbf{O}_m + (\gamma P(x(t+1)) - P(x(t)))\mathbf{rnd} \tag{6}$$

$P(x(t))$ is the Critic output (state evaluation) at time t. Equation (5) is derived by replacing $P(x(t))$ by $\max_a Q(s(t), a)$. In other words, the best evaluation value for the actions in that state is defined as the Critic output (evaluation for the state). If the selection of the action is greedy, that is, the action with the largest Q-value is selected, the last $\max_a Q(s(t), a)$ in Eq. (5) becomes $Q_{motion}(s(t))$.

The training is executed for both Q-net and Actor-net by BP (back propagation) [14]. Figure 2 shows the flowchart for the process.

The architecture combining Q-learning and Actor-Critic has also been proposed by Morimoto and Doya [15]. In their approach, however, a subgoal is set in the learning in the high-dimensional space. According to Q-value, which of the subgoals is to be selected is determined at the

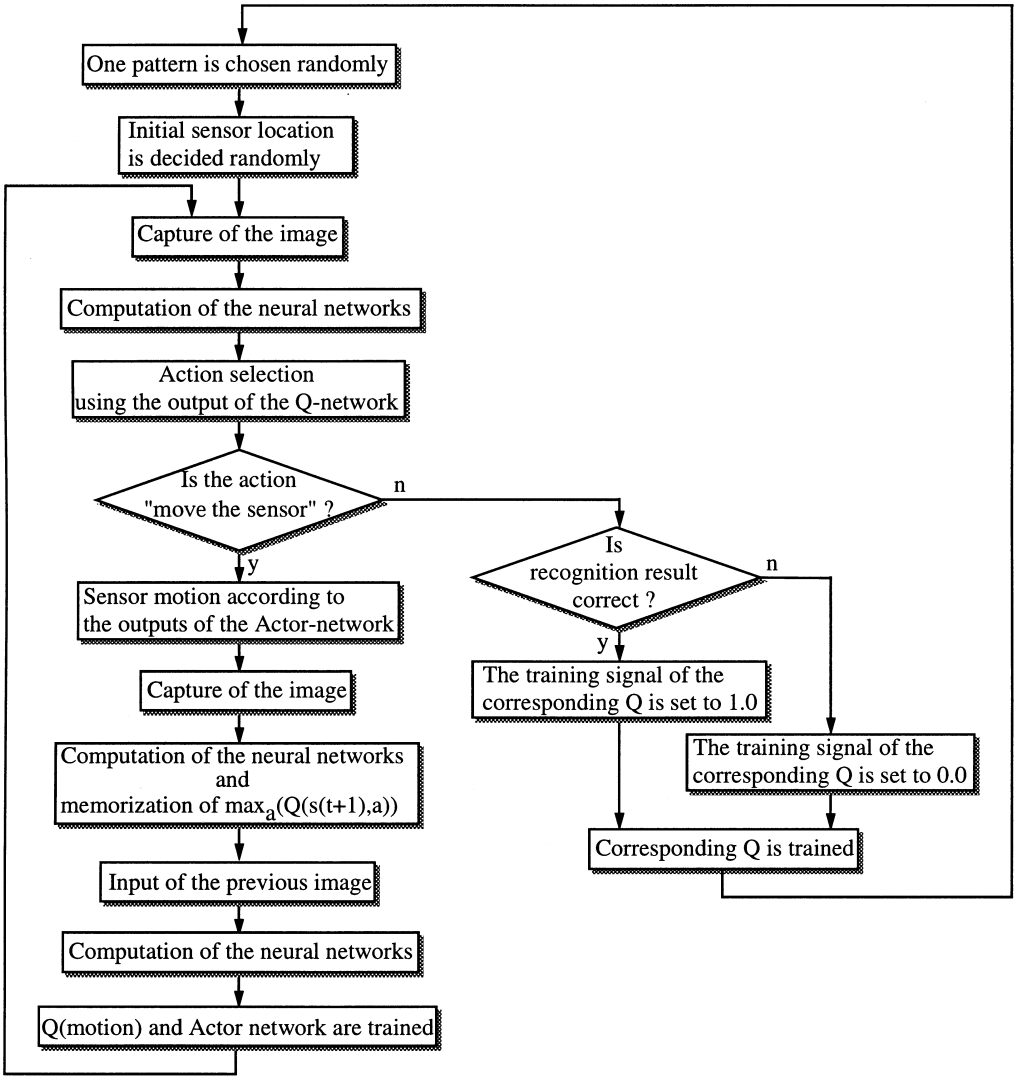


Fig. 2. Flowchart of the proposed learning.

present state, and Actor-Critic is used in the training to achieve that subgoal. In other words, their approach differs from the approach in this paper in that the partitioning of the higher-dimensional space is intended. It also differs in that Q-value is not replaced by the Critic signal, and Critic exists in each Actor-Critic.

3. Simulation of Viewpoint Motion

3.1. Problem formulation and learning

This paper considers a problem where multiple patterns are prepared, and it is required to identify which of the patterns is presented. The two-dimensional visual sensor shown in Fig. 3 is used. The nonuniformity is included where the central part of the sensor has a high resolution and the peripheral part has a low resolution.

The pattern to be recognized is presented to this sensor. The position of the pattern presentation differs in each trial, and there is no information to indicate the position of the pattern presentation. The initial position of pattern presentation can be such that only a part of the pattern is contained in the visual field as in Fig. 4(a), although the case where the pattern is totally excluded from the visual field is not considered. It may also happen, as in Fig. 4(b), even if the whole pattern is contained in the visual sensor, that the pattern cannot be identified. In such cases, the identification is impossible using the sensor input.

Then, the sensor must be moved to an adequate position. In a case as in Fig. 4(a), it suffices to move the visual sensor to the direction of the center of gravity of the pattern. In a case as in Fig. 4(b), on the other hand, the direction of the adequate movement depends on the pattern to be identified. The system moves the sensor, and finally decides which of the patterns is presented. Then, the reinforcement signal is given according to whether or not the decision is correct. Thus, the adequate sensor motion and the timing to conclude the recognition result are acquired by learning.

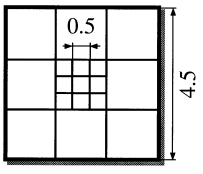
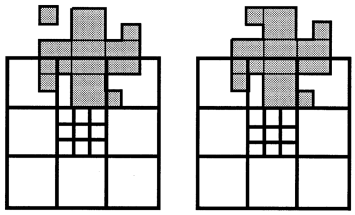
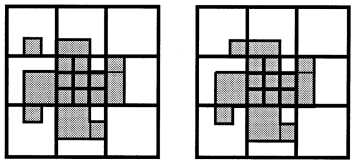


Fig. 3. Visual sensor with nonuniform sensory cells employed in this paper.



(a) A part of the presented pattern is caught



(b) All of the presented pattern is caught

Fig. 4. Two cases in which the presented pattern cannot be identified.

3.2. Definition of task

Figure 5 shows the two kinds of pattern sets used in this study. As already shown in Fig. 3, the sensor is a 4.5×4.5 two-dimensional visual sensor, being composed of 17 sensor cells. The central part is composed of 9 small cells with an edge length of 0.5. The peripheral part is composed of 8 large cells with an edge length of 1.5. The individual sensor signal is determined as the ratio of the area occupied by the projected pattern to the area of the sensor cell. As the input to the neural net, the signal is linearly converted to a value between –1.0 and 1.0.

The initial position of the sensor is determined at random for each trial, in the range where the total sum of the sensor cell signals is not less than 0.5. When a pattern in set 1 is presented, the visual sensor must be moved to the upper-left region. When a pattern in set 2 is presented, the

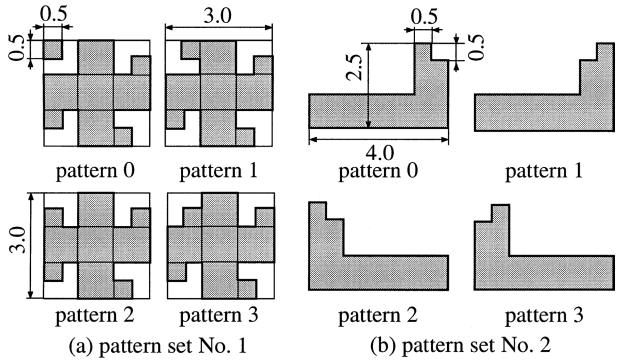


Fig. 5. The presented pattern sets.

system must decide whether the pattern is one of 0 and 1, or one of 2 and 3. In order to discriminate further, the sensor must be moved to the right in the former case, and to the left in the latter case.

The neural net is composed of three layers. The hidden layer of Q-net is composed of 30 neurons, and that of Actor-net is composed of 10 neurons. The output function of each neuron is the sigmoid function with the range from –0.5 to 0.5, that is, $(1/(1+\exp(-x)) - 0.5)$. In the actual system, the training signal for Q-value in Eq. (3) is used after converting as

$$O_{train} = \alpha(Q_{train} - 0.5) \tag{7}$$

where α is a constant.

Conversely, the output O of the neuron is converted to a new Q-value by

$$Q = O/\alpha + 0.5. \tag{8}$$

In order to avoid the saturation region of the neuron output function, α is set as 0.8, so that the output remains between –0.4 and 0.4. If Q-value is less than 0.0, the new value is set as 0.0.

Before training, all initial weights from the hidden layer to the output layer are set as 0.0, so that the output is a constant independently of the input. The bias of the output neuron is fixed as 0.0 in Actor-net, and –2.2 in Q-net, in order to avoid the instability of learning and to avoid the sensor velocity becoming uniform easily. By setting the bias as –2.2 in Q-net, the output is nearly –0.4. In other words, each initial Q-value is almost 0.0 at the beginning.

The total number of trials in training is limited to 100,000. The discount factor γ in the learning of Q-value by Eq. (3) is set as 0.99. The constant β in the conversion from Actor output to sensor motion in Eq. (4) is set as 0.4. Consequently, the maximum movement in a unit time is 0.2 in both x- and y-axis directions. The random variable **rnd** is derived by multiplying three random numbers in the range from –1.0 to 1.0.

In the selection of the action, Boltzmann selection is used during learning, where the temperature is gradually decreased from 1.0 to 0.01, as shown in Fig. 6. After the learning, the action with the maximum Q-value is selected (greedy policy), and the random variable is not added to the sensor motion.

The trial is continued, including the case where the pattern disappeared from the visual field of the sensor, until the system gives the recognition result. At the initial stage of learning, however, the temperature in action selection is high and the behavior is almost random. Then, it never was the case that the sensor motion is continuously selected. When the pattern disappeared from the visual field of the sensor in the stage where the learning is well progressed, if the sensor motion is continuously selected, Q-value continues to decrease due to the learning of Q-value in Eq. (3). Consequently, the sensor motion is never selected indefinitely, even if the upper bound is not posed on the number of sensor motion selections.

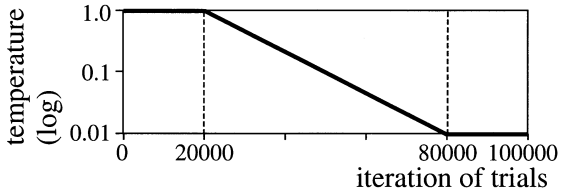


Fig. 6. Temperature cooling schedule used in the action selection.

3.3. Result

The recognition was actually tested after learning. For either pattern set, "recognition" action is selected after the sensor motion, and the correct result of recognition is observed. Figure 7 shows the sensor movement from 132 initial positions with 0.25 intervals, after learning pattern set 1. It is seen that the center of the sensor moves to the upper left of the pattern, for any presented pattern and for any initial position of the sensor.

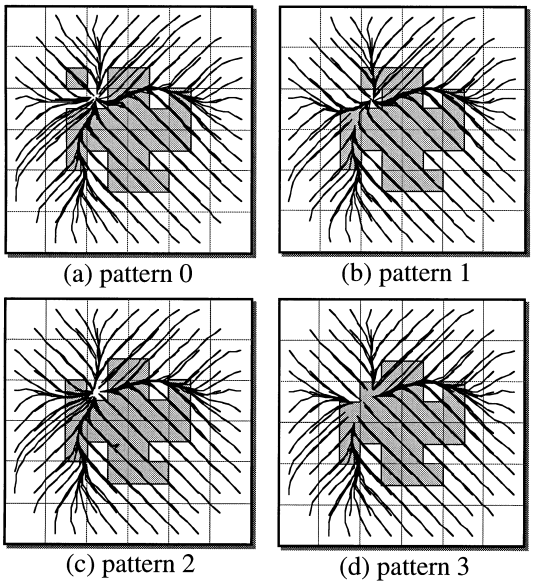


Fig. 7. Trajectories of the visual sensor when pattern set No. 1 was presented.

It is also seen that the sensor position finally converges almost to a point for patterns 0, 1, and 2. At the final state, one of the small sensor cells in the central part catches the small square, which defines the difference between patterns. In the case of pattern 3, the area of convergence is wider than in other cases. The above tendency is the same even if the initial weight of the neural net is varied.

Figure 8 shows the trajectory of the visual sensors for pattern set 2. The sensor moves to the upper right for patterns 0 and 1, and to the upper left for patterns 2 and 3, before outputting the recognition result. In other words, it is seen that the direction of the sensor motion depends on the presented pattern.

Figure 9 shows the distribution of Q-value for the presented pattern when the sensor position is varied. It is seen also in this result that Q-value is larger when the pattern is 0 or 1 and the sensor is upper right, and when the pattern is 2 or 3 and the sensor is upper left.

Figure 10 shows the distribution of Q-value for the sensor motion and that for pattern 1 when pattern 0 is presented. It is seen that Q-value for the sensor motion is generally large, independently of the sensor position. It is also seen that Q-value for pattern 1 takes a large value when the sensor is located at the upper right of the pattern, as in the case of Q-value for pattern 0 as in Fig. 9(a).

It is difficult to see the magnitude relation among Q-values of the actions. Figure 11 is the cross section of the Q-value distribution on the horizontal line in Figs. 9(a), 10(a), and 10(b). It is seen that all the Q-values are large at the recognizable part indicated by the arrow. Among those, Q-value is especially large for the recognition of pattern 0. In the other part, Q-value for sensor motion takes the largest

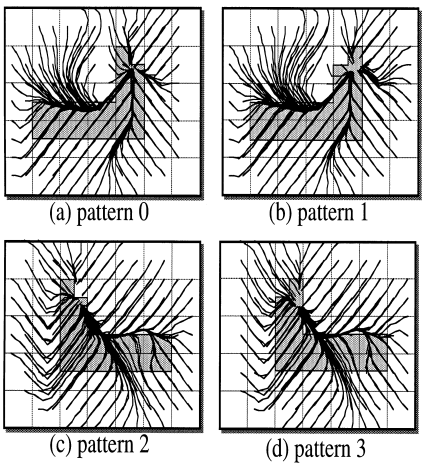


Fig. 8. Trajectories of the visual sensor when pattern set No. 2 was presented.

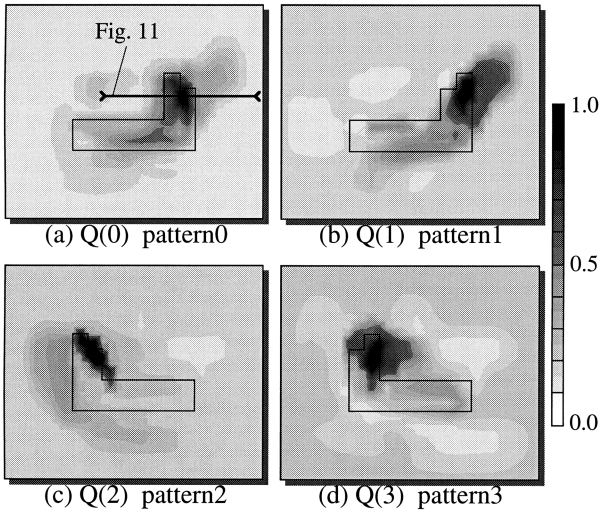


Fig. 9. Distribution of the Q-values corresponding to the presented pattern.

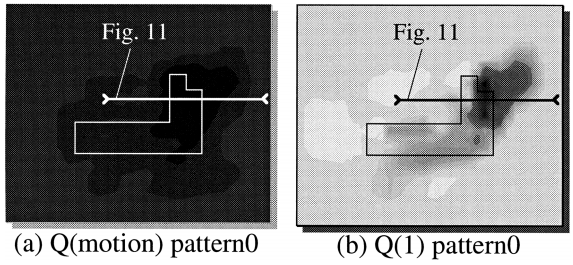


Fig. 10. Distribution of the Q-values for the sensor motion and the Q-value for pattern 1 when pattern 0 was presented.

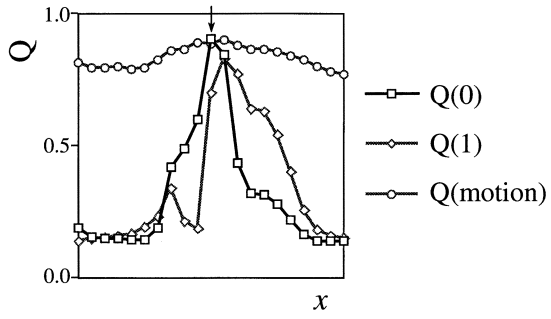


Fig. 11. One dimension of the Q-value distribution when the sections of the Q-value surfaces were observed.

value, and there is no misrecognition due to the Q-value for pattern 1 being the largest.

Q-value for sensor motion gradually decreases, where the slope depends on the discount factor γ in the learning of Q-value by Eq. (3). Making γ close to 1, the slope of Q-value in the sensor motion is reduced. Then, the sensor motion is selected without being trapped, even if there exists a local peak in Q-value for recognition.

Simulations were iterated five times by varying the initial weights of the neural net, and the change of the recognition rate with the course of learning was examined. Figure 12 shows the result. The vertical axis is the probability of correct recognition. In any of 5 trials, the recognition rate increases rapidly around 50,000 trials, exhibiting a learning curve of similar shape.

Figure 13 shows the sensor trajectories after learning by 50,000 trials. It is seen that, although the sensor can move to some extent, Q-value for recognition increases in a region where the recognition is still impossible, resulting in the decision. In other words, the system then makes an incorrect decision that the pattern is 0, although pattern 1 is presented.

As the next step, the sensor motion was not selected, and the system was trained using only Q-values for recognition. Figure 14 shows the Q-value distribution. Comparing this figure with Fig. 9(a), it is seen that the peak at the upper right of the pattern is low, and the peak at the upper left is high. This implies that the training for Q-value is not satisfactory unless the sensor is moved. The reason for this is as follows.

When the presented pattern cannot be identified, different supervisor signals are given, while the input signals are the same. If the sensor is not moved, the training must proceed almost at such positions, which has the effect that the training does not progress satisfactorily, even at the position where the correct recognition is essentially possible. If the sensor can be moved, on the other hand, Q-value for recognition is not learned while the sensor is moving, and the part to be learned is restricted. This further increases

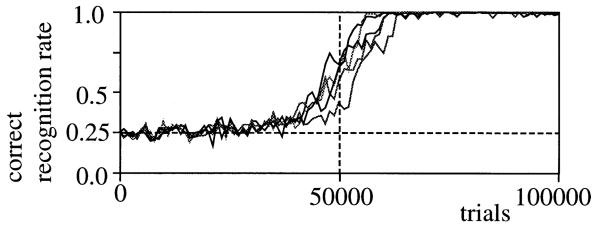


Fig. 12. Learning curve when pattern set No. 2 was presented. The y axis indicates the probability of successful recognition.

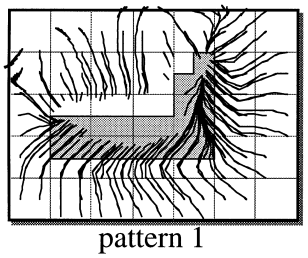


Fig. 13. Trajectories of the visual sensor when pattern 1 was presented after 50,000 trials of learning.

Q-value for the corresponding recognition in the region, where the correct recognition is possible. This again enhances the training for the sensor motion. In other words, the recognition is improved by the above interaction.

3.4. Sensor motion by context

When humans recognize a character or a pattern, the efficiency is greatly improved if the next to come is predicted from the context. A problem in realizing this function is how to extract and utilize the context. As a preliminary study in this direction, it is assumed in this study that the context is already extracted, and the sensor motion based on the extracted context is examined.

A pattern set is prepared as in Fig. 15, where any of the patterns cannot be identified when the center of the sensor is placed at a point in the pattern. In this case, patterns 0 and 1, as well as patterns 2 and 3 cannot be discriminated even if the sensor is moved to the upper left. Patterns 0 and 2, as well as patterns 1 and 3, can neither be discriminated even if the sensor is moved to the lower right.

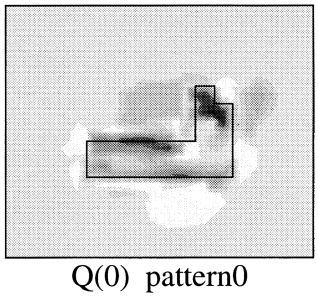


Fig. 14. Distribution of Q-value when the sensor did not move and only the Q-values for recognition were trained.

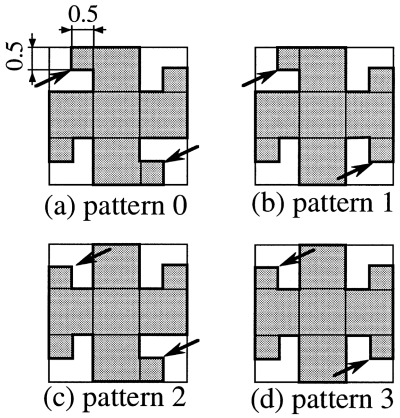


Fig. 15. The pattern set in which the system requires the context inputs to identify each presented pattern.

In order to cope with this problem, four inputs, each corresponding to the pattern, are prepared. Signal 1 is given to the input for the pattern, which has the possibility of being presented, and 0 is given to others. In this case, it suffices to move the sensor to the lower right if the pattern is known to be 0 or 1, and to the upper left if the pattern is known to be 0 or 2. In this study, the number of 1's as the context input is always assumed as 2, and there can arise four cases: 0 or 1, 0 or 2, 1 or 3, and 2 or 3.

The training in this case is very difficult compared to the previous training. The hidden layer of Q-net is composed of 50 neurons, and that of Actor-net is composed of 20 neurons. Two million trials were executed. γ in Eq. (3) is set as 0.96. The same temperature change in selecting the action as in Fig. 6 is applied, but the x-axis is expanded.

Figure 16 shows the sensor motion. It is seen that the sensor trajectory changes, depending on the context. Figure 17 shows the Q-value distribution. It is seen that the position

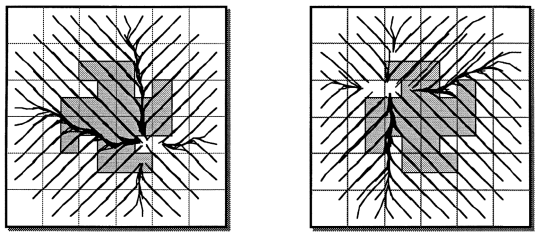


Fig. 16. Difference in sensor trajectories depending on context inputs.

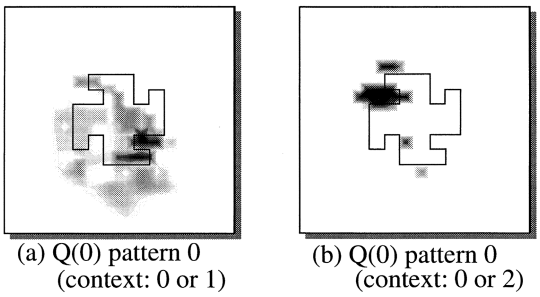


Fig. 17. Difference in Q-value distribution depending on context inputs.

of the Q-value peak depends greatly on the context information.

In this task, the direction of the sensor motion must be adjusted according to the context information, even if the same pattern is presented. The following problem then arose when γ was set as 0.99. When pattern 0 was presented, and the sensor motion for the case, where the context information indicates pattern 0 or 1, was learned first, the sensor moved to the lower right also when the context information indicates pattern 0 or 2. Then the Q-value for sensor motion took large values over a wide range, reducing the opportunity for the system to "recognize" at the upper left of the pattern and to learn the Q-value for recognition, and thus preventing satisfactory training. Those problems of the difficulty in parameter setting and the slow speed of learning are left for future studies.

3.5. Discussion

This section considers the validity of the proposed system, as an engineering pattern recognition system, or as a model for biological vision. It is in general desirable in the pattern recognition system that the system can cope with the expansion, contraction, and rotation of the pattern, not only the translation. In this system, however, only the translation is considered, and the expansion, contraction, or rotation is not considered.

In order to train the system for the expansion, contraction, or rotation, the basic algorithm, which is essentially the same as in the proposed system, is expected to be applied. In this case, however, the speed of learning is a problem, since the degree of freedom in motion is increased. In order to improve the learning speed, the hardware implementation of the system or the introduction of the a priori knowledge is required. That is also necessary from the viewpoint of application to more practical problems.

The proposed system may also be characterized as a model for biological vision. Examining the trajectories in Figs. 7, 8, and 16, however, the sensor does not move straight-forward from the start to the end, but often moves with the same velocity in the x- or y-direction. This is obviously unnatural as the eye movement. In this simulation, the sensor motion is specified by the velocities in the x- and y-directions, respectively. The unnatural movement seems due to the fact that the dynamics is not considered, and there is a maximum limit for the velocity.

It should be noted that the learning succeeds even if the dynamics is introduced in the application of the reinforcement learning for the reaching motion of the manipulator [16, 17]. Then, it is estimated that the above insufficiency is a problem in the visual system, and is not due to the architecture. It should be noted in the proposed system that the motion of the visual sensor is learned using only the reward. This seems to indicate the possibility that the eye movement is acquired by reinforcement learning also in the biological visual system.

Lastly, Actor-Q architecture proposed in this paper is considered as useful, not only in the learning of the active perception and recognition, but also in the problem in general, where the decision making is required. Consider the problem, as an example, where a mobile robot finds an obstacle and has to decide whether it should go to the right side or the left side of the obstacle. If the robot is trained by a simple Actor-Critic architecture, a problem arises that the robot sometimes stops in front of the obstacle, or strikes against the obstacle [8]. If Actor-Q architecture is used so that the right or the left pathway is already fixed, it is expected that such a problem can be avoided. It will also be possible, when there exist multiple action goals, to determine which of the goals should be aimed at.

In Q-learning, possible actions must be prespecified. In order to utilize the autonomy and the flexibility of reinforcement learning, there must be a mechanism to specify additionally the action whenever necessary.

4. Conclusion

This paper proposed Actor-Q architecture, which operates as follows. The system output is divided into a discrete action and a continuous motion vector. Q-value is assigned to each action, which is learned by Q-learning. The motion is the output of Actor in Actor-Critic, and is learned using Q-value of the corresponding action instead of Critic.

By applying Actor-Q architecture to the active perception and recognition learning system, the following function can be obtained.

(1) The timing to decide the final recognition output is determined by the system itself.

(2) The system is trained using only the reinforcement signal after giving the recognition result, which indicates whether the result is correct or incorrect.

(3) The sensor is moved to the position where the correct recognition is realized, so that the correct recognition is realized without being trapped in a local peak.

A task to recognize the pattern was tried, using a visual sensor with nonuniform sensor cells, and the performance was verified. It was also shown by providing the context input that the sensor motion depending on the context can be realized, even if the sensor inputs are the same.

Acknowledgment. Part of this study was supported by a Grant-in-aid for Scientific Research on Priority Area from the Ministry of Education Japan ["Emergent systems" (No. 264)], for which the authors are grateful.

REFERENCES

1. Toriwaki J. Basis of pattern information processing. Asakura Shoten; 1998. (in Japanese)
2. Ohtsu T, Kurita T, Sekita I. Pattern recognition—Theory and applications. Asakura Shoten; 1996. (in Japanese)
3. Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift position. Biol Cybern 1980;36:193–202.
4. Fukushima K. A study based on active visual information processing neural net model. Sci Grant Ministry of Education, "Systems approach to higher-order brain function," Rep. Year 1998, 1999. (in Japanese)
5. Rougeaux S, Kuniyoshi Y. Robust real-time tracking on an active vision head. Proc IEEE-RSJ Int Conf on Intelligent Robots and Systems (IROS)'97, Grenoble.
6. Fukushima K. A neural network for visual pattern recognition. IEEE Comput 1988;21:65–75.
7. Shibata K, Sugisaka M. Dynamics of a recurrent neural network acquired through the learning of a context-based attention task. Proc Int Symp on Artificial Life and Robotics, 7th, p 152–155, 2002. (in Japanese)
8. Shibata K, Okabe Y, Ito K. Direct-vision-based reinforcement learning in going to a target task with an obstacle and with a variety of target sites. Proc NEURAP'98, p 95–102.

9. Watkins CJCH, Dayan P. Q-learning. Machine Learning 1992;8:279–292.
10. Whitehead SD, Ballard DH. Learning to perceive and act by trial and error. Machine Learning 1991;7:45–83.
11. Shibata K, Nishino T, Okabe Y. Active perception based on reinforcement learning. Proc WCNN'95, Vol. II, p 170–173.
12. Barto AG. Adaptive critics and the basal ganglia. In: Models of information processing in basal ganglia. MIT Press; 1995. p 215–232.
13. Sutton RS. Learning to predict by the methods of temporal differences. Machine Learning 1988;3:9–44.
14. Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature 1986;323:533–536.
15. Morimoto J, Doya K. Hierarchical reinforcement learning of low-dimensional subgoals and high-dimensional trajectories. Proc ICONIP'98, Vol. 2, p 850–853.
16. Shibata K, Sugisaka M, Ito K. Hand reaching movement acquired through reinforcement learning. Proc 2000 KACC (Korea Automatic Control Conference), 90th (CD-ROM, 4 pages).
17. Shibata K, Sugisaka M, Ito K. Acquisition of reaching motion by reinforcement learning. Tech Rep IEICE 2001;NC2000-170. (in Japanese)

AUTHORS (from left to right)



Katsunari Shibata (member) completed the master's program (mechanical engineering) at the University of Tokyo in 1989. He was affiliated with Hitachi Ltd. from 1989 to 1992. He retired from the doctoral program at the University of Tokyo in 1993. He is presently an associate professor at Oita University. His major research interests are reinforcement learning and autonomous learning using neural networks. He holds a D.Eng. degree.

Tetsuo Nishino completed the master's program (information engineering) at the University of Tokyo in 1997. He is now with the Japan Oracle Co.

Yoichi Okabe (member) completed the doctoral program (electronic engineering) at the University of Tokyo in 1972. He was a Visiting Researcher at IBM San Jose Research Laboratory, a professor at the Research Center for Advanced Science and Technology, and is presently a professor at the Graduate School of Engineering and Director, Information Technology Center at the University of Tokyo. He has largely been engaged in research on high-speed high-function devices, especially superconductivity, measurement of magnetoencephalography, and neural networks. He holds a D.Eng. degree.