Hand-Eye Coordination in Robot Arm Reaching Task by Reinforcement Learning Using a Neural Network

Katsunari Shibata & Koji Ito

Dept. of Computational Intelligence and Systems Science, Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama 226-8502, JAPAN
shibata@ito.dis.titech.ac.jp, ito@dis.titech.ac.jp

ABSTRACT

This paper shows that the robot with hand-eye system can learn the hand-reaching task without apparent calculations of the target, obstacle, and hand locations. This system consists of a neural network whose inputs are raw visual sensory signals, joint angles of the arm, and the existence of the obstacle. It is trained by reinforcement learning, and the reward is given only when the hand reaches the target that can be found only on the visual sensor. In order to show the effectiveness of this learning, the following three assumptions are introduced. (1) The target, obstacle, and hand cannot be distinguished on the vision. (2) The hand disappears out of the visual field. (3) The obstacle appears randomly but the location is always the same. The initial hand location and the target location are chosen randomly at each trial. After the learning, the robot could become to reach its hand to the target. By the analysis of the hidden neurons' representation after the reinforcement learning, it was known that the target location was not represented independently from the hand location, either on the work (visual sensory) space or on the joint space. Furthermore, the representation of the hand location is acquired by mixing the joint angles and visual signals.

Key Words : Direct-Vision-Based Reinforcement Learning, Neural Network, Hand-Eye Coordination, Reaching Task, Hidden Representation

1 INTRODUCTION

Reinforcement learning is focused recently by its autonomous and adaptive learning property. When we utilize it in robot control, it is the general way that we first design the state space that is calculated from its sensory signals, and then the appropriate mapping from the state space to the motion space is trained[1]. On the other hand, it has been proposed that the neural network is utilized in reinforcement learning to realize the continuous and non-linear mapping [2].

The authors have proposed that the sensory signals, those are often the outputs of the sensory cells with a local receptive field, are put into the neural network directly. It has been called direct-vision-based reinforcement learning. This results in the stability of the learning[3] and acquisition of the continuous and adaptive state space as the internal representation in the hidden layer[4].

Here the hand-reaching task is taken up as shown in Fig. 1, in which the acquisition of the hand-eye coordination is required. When we reach our hand to some object, we are rarely conscious of the hand even if it exists in our visual field. However, it was reported that the monkey, who had been restricted to see its hand for 34 days after its birth, was absorbed in looking its hand and could not reach its hand to some small object when it saw its hand for the first time[5]. This experiment shows that the hand-eye coordination can be acquired by learning after its birth. The reduction of the consciousness through experiences may suggest us to deny that the monkey calculates its hand position apparently and excludes its image from the visual sensory signals. There may be more direct mapping from vision to motion.

There are many works concerning to the hand-eye coordination to realize the visual feedback control. Jägersand et al. have claimed the advantages of their work as (1)no prior models of the transfer function from visual perceptions to the robot angles, (2)estimation of the model (visual motor Jacobian) through the iteration of the reaching tasks with no extra learning steps or movements[6]. Even in their work, the clipping out the hand and target locations from the visual image are dealt with as a premise. Although the task adopted here is toy problem with no redundant degree of freedom, but we propose the method for the hand-eye system to learn the reaching without apparent calculations of the target, obstacle and hand location. In order to emphasize the advantage of the method, we adopt the three assumptions that may not be natural in actual. The target, obstacle and hand cannot be distinguished on the vision. The hand can disappear out of the visual field depending on the joint angles. The obstacle appears randomly, but the location is always the same.

2 Task Setting

Here the setting of the task as shown in Fig. 1 is described. The visual sensor has $5 \times 5 = 25$ cells and the output of each cell is the area ratio occupied by the target, obstacle or the robot's hand against its receptive field. Below here, the left-bottom corner of the visual sensor is supposed to be the origin. The size of each visual cell is 1×1, and the size of the target, obstacle and hand is also the same. So these three

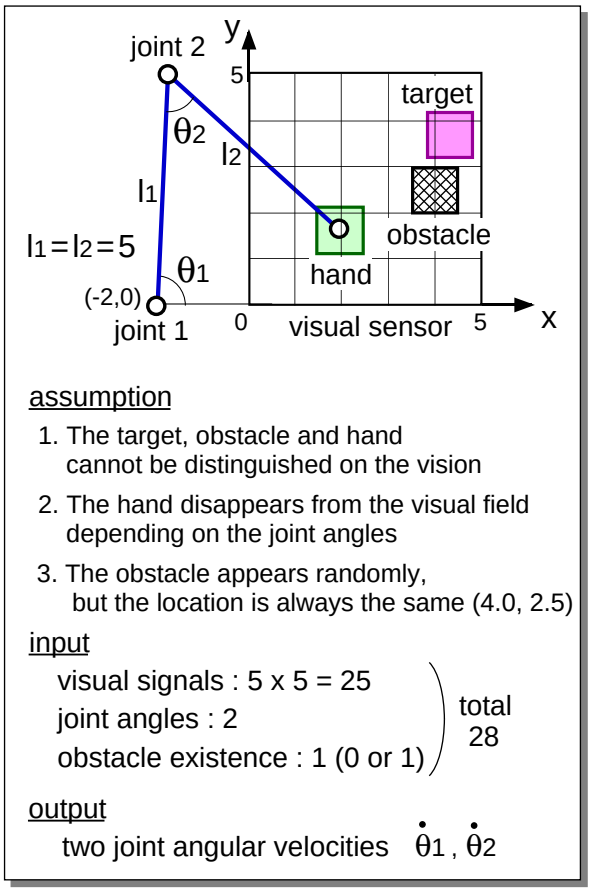

Figure 1: The robot hand-reaching task employed in this paper

cannot be distinguished with each other on the vision. The target is located randomly in the range where the whole target can be caught in the visual field. In other words, the center of the target is chosen randomly in the range of $0.5 \leq x \leq 4.5$, $0.5 \leq y \leq 4.5$. The base of arm is fixed at (-2.0, 0.0), and the initial hand location is chosen randomly in the range of $-1.99 \leq x \leq 6.01$, $-1.99 \leq y \leq 6.01$ except for the range where the arm cannot reach. The obstacle appears randomly at the constant location with the probability 0.5, but when the obstacle does not exist, the hand or target can be located at the location. The target and obstacle are not moved during one trial. The length of each link of the arm is 5. Each joint angle is limited from 0.0 to π. So there is one-to-one corresponding between the joint angle vector to hand location vector except for the hand location (-2.0, 0.0). These two joint angles are supposed to be observed directly. The inputs of the robot are visual signals, joint angles, and the binary value that indicates the existence of the obstacle (0: not exist, 1: exist). The total number of inputs is 28. There are two outputs which represent the joint angular velocities, and two joints are moved according to them. When the hand touches the target, the reward is given, and when the hand touches the obstacle or the joint angle go over its limit, the small penalty is given. More details of the task are described in the section 5.

There are two technically interesting points. The first one is whether the coordination between visual sensory signals and joint angles of the arm can be acquired only by the reinforcement learning, especially in the case of raw visual sensory inputs. The second one is how the target information is clipped out and represented on the hidden layer. In order to know the target location from the visual sensory signals, it is required in this task that the information of the hand and obstacle is removed and only the information of the target is clipped out from the whole visual sensory signals. In this task, no prior information about the target, obstacle and hand is supposed to be obtained. When we are going to pre-process the visual sensory signals before the reinforcement learning, we have to know some information about this task in advance. That is because even if the locations of three objects are calculated from the visual sensory signals, the robot cannot know which is the target location. Of course, if the relation between the hand location and joint angles is known and it is known that the obstacle location is always the same, the target location can be known easily. However, the reinforcement learning is useful when the information about the given task is not enough. This indicates that the pre-processing does not fit for the autonomous and adaptive ability of the reinforcement learning. On the other hand, the direct-vision-based reinforcement learning promotes the flexibility of the system.

3 LEARNING OF CLIPPING

In order to show the basic ability of the neural network dealing with both the visual sensory signals and the other signals, the supervised learning which realizes the clipping of the target object information is executed here. The setting is almost the same as described in the last section. But the two outputs are trained to output the target location x, y respectively by Back Propagation, and the hand is located randomly at each time step. The number of hidden units of the neural network is 20. The overlap of the target and hand is allowed, but the obstacle is not allowed to be overlapped with the target or hand.

Fig. 2 shows the outputs as function of the target location x, y respectively. In each graph, 100 outputs chosen randomly are plotted. Even if the hand location and the existence of the obstacle were varied, and sometimes the hand disappeared, and sometimes the target overlapped with the hand, the outputs became close to the training signals which are drawn as the straight lines. It is known that the neural network can learn this clipping, even if it seems difficult at a glance.

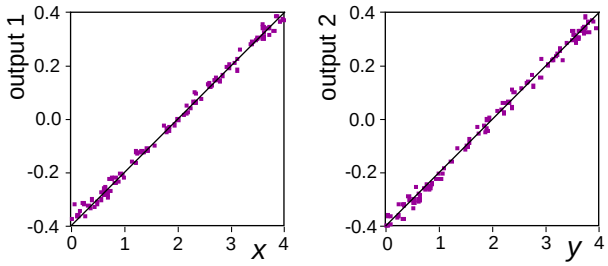

Figure 2: The results of clipping the target information from the visual signals by supervised learning.

4 REINFORCEMENT LEARNING

In this section, the reinforcement learning architecture and algorithm employed in this paper is introduced. The basic architecture is the Actor-Critic architecture[7], but there is only one layered neural network. The outputs are divided into two types, a value function output and some motion signal outputs (see Fig. 3 for reference). The reason of only one network is that the hidden units can be shared adaptively between the value function and the motion signals according to their necessity, and if their necessary knowledge has a common part, it can be shared by utilizing the same hidden units.

The algorithm is Temporal-Smoothing (TS) based reinforcement learning. This is very similar to Temporal Difference (TD) based reinforcement learning[7], and only the difference is the curve of value function along time axis becomes straight line or exponential curve. In TS based learning, the value function shows the approximated necessary time steps to achieve the task linearly. Here, adaptive slope method is also employed, in which the slope of the value function along time axis is changed adaptively by the maximum time steps for achieving the given task. The slope corresponds to the discount factor in TD based learning. The ideal slope ΔV_{ideal} is calculated as

$$\Delta V_{ideal} = V_{amp}/N_{max} \tag{1}$$

where V_{amp} : ideal amplitude of the value function, here 0.4-(-0.4)=0.8, and for adaptability N_{max} is calculated as

$$N_{max}(t) = \begin{cases} N(t) & \text{if } N(t) > \beta N_{max}(t-1) \\ \beta N_{max}(t-1) & \text{otherwise} \end{cases} \tag{2}$$

where $N_[i]$: necessary time at the i-th trial, β : a attenuation factor ($0.0 < \beta < 1.0$). Here the value range of the neuron output is from -0.5 to 0.5. Then by comparing the change of the actual value to this ideal one, the value at previous time $V(t-1)$ is trained by the training signal as

$$V_s(t-1) = V(t-1) - \eta(\Delta V_{ideal} - \Delta V(t)) \tag{3}$$

where V_s : training signal for the value function, $\Delta V(t) = V(t) - V(t-1)$, and η : a training constant.

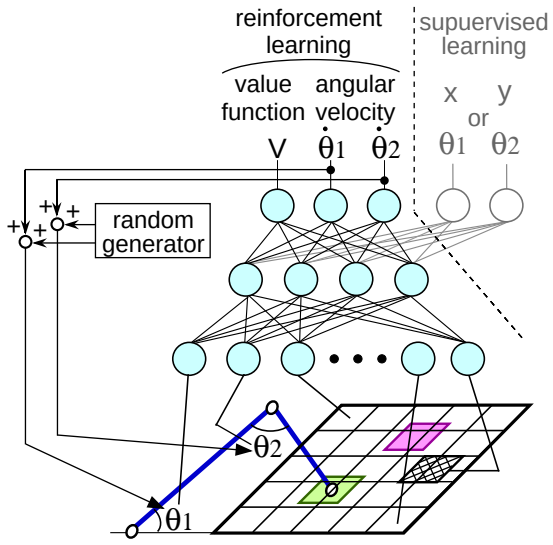

Figure 3: The inputs and outputs of the neural network in this simulation. The 4th and 5th outputs are utilized in 5.2 to examine the hidden layer representation.

By this learning, value function curve along time becomes smooth and the slope of the curve becomes constant. When the system arrives at the target state, the value is trained to be 0.4.

The system generates its motions according to the sum of the motion signals $\mathbf{m}$, and random numbers $\mathbf{rnd}$ as trial and error factors. The motion signals $\mathbf{m}$ are trained by the training signals as

$$\mathbf{m}_s = \mathbf{m} + \zeta \mathbf{rnd} \Delta V \tag{4}$$

where ζ : a training constant. By this learning, motion is trained to make more change of the value function. This learning is processed in parallel with the value function learning.

5 SIMULATION

5.1 Reaching

Fig. 3 shows the neural network utilized for this simulation. Note that the two of the outputs of the neural network in the right is utilized in the simulation in the next subsection. The neural network receives the continuous visual sensory signals, the continuous joint angles of the arm, and the existence of the obstacle, that is binary, as inputs. The number of hidden layer is one, and the number of the hidden units is 40. The number of motion outputs is two. The first one decides the angular velocity of the joint 1, and the second one decides that of the joint 2. Each velocity is decided by adding the small random number (uniform random number powered by 3) to the corresponding output and then 0.1 is multiplied. The amplitude of the random number is adjusted according to the relative value function gain as $\Delta V/\Delta V_{ideal}$. The value

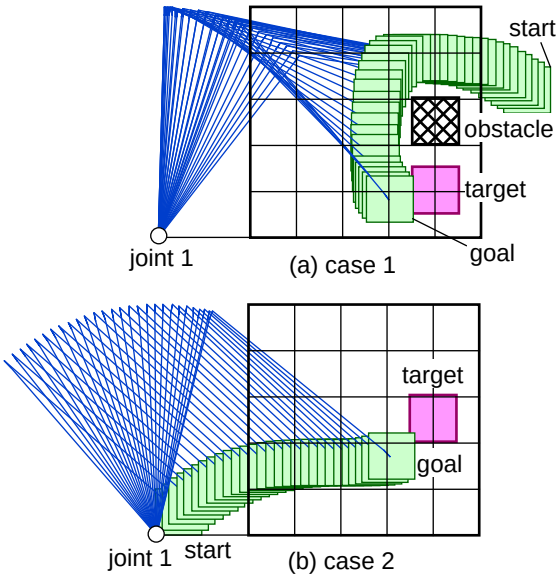

Figure 4: Two examples of the hand motion after learning for the different target location.

range of the output of the neural network is from -0.5 to 0.5, but the training signal calculated in Eq. 3 and 4 is limited from -0.4 to 0.4 for the appropriate learning of the neural network. So the maximum joint angular velocity except for the random factor is 0.04, and the minimum time steps to rotate $\pi/2$ is 32. If the robot cannot reach its arm to the target in some time steps, the target is moved gradually to the hand in some following trials. When each joint angle becomes less than 0.0 or larger than π, the joint is fixed at the angle 0.0 or π and small penalty 0.1 is given. If the robot selects the motion for its hand to crash the obstacle, the hand does not move and the small penalty 0.1 is also given.

Fig. 4 shows two examples of the hand loci for different target locations after 400000 trials of learning. It can be seen that the robot moves its arm while avoiding the obstacle, and finally it reaches its hand to the target successfully. It looks that the robot can distinguish its hand and the target. When the hand was not projected on the visual sensor in the initial state and then the hand appeared in the visual field, the arm motion was still smooth.

Fig. 5 show the value function, hand motion vector as a function of the hand location, and the loci of the hand after learning for the cases of the previous examples. It is known that when the obstacle existed, the value function changed especially around the right top corner of the obstacle. The right neighbor of the Fig. 5(b) shows the minimum time steps for the hand to reach the target for the case 2. It can be seen that the value function is similar to the minimum time step landscape and the value function approximates the necessary time through reinforcement learning. However, when the obstacle did not exist and the hand goes close to the obstacle location, the hand

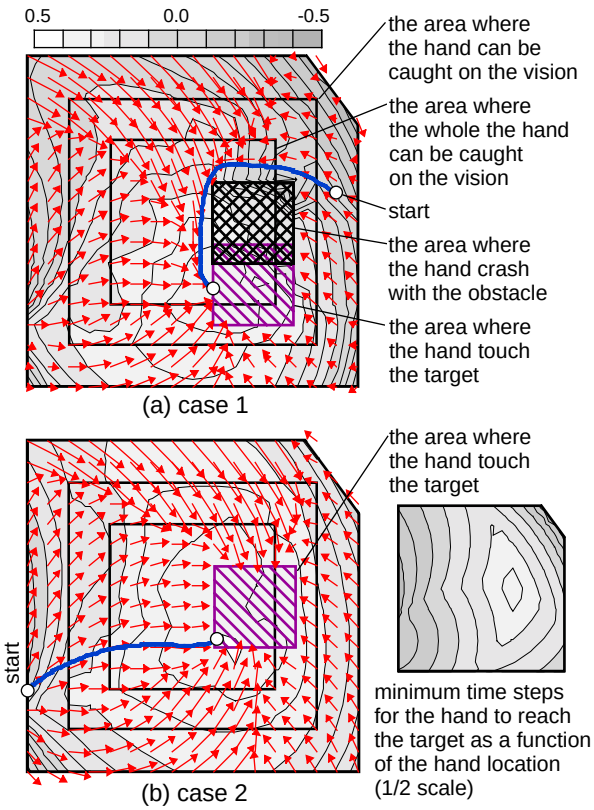

Figure 5: The value function and the hand velocity.

had a tendency to avoid the illusional obstacle, but the loci is different from those when the obstacle existed. Furthermore, when the hand located initially at the edge of the initial location range, the hand sometimes crashes the obstacle or goes over its joint angle limit, but by the random factor, the hand can escape in a small numbers of time steps in almost all the cases. That is because the hand rarely went through such states, and the learning was not enough. But it is rational that the robot utilizes its limited resources mainly for the well-experienced states. When TD based reinforcement learning is applied, the gap of the contour line becomes small when the hand is close to the target and becomes large when far from the target.

5.2 Examination of the Representation in Hidden Layer

Here the representation in the hidden layer is examined. The main interest about the representation is whether the representation of the target location in the hidden layer is independent to the hand and obstacle information. In this simulation, the neural network which has been already trained by reinforcement learning as the above is used, but the different output units are trained by supervised learning as shown in Fig. 3. The connection weights from the hidden layer to the output units are all 0.0 initially. This means that only the input-hidden connection weights trained

by the reinforcement learning is used. The target is located sequentially at the four corners of the visual sensor, while the hand location is chosen randomly in the range where the whole hand image is projected on the visual sensor as long as the hand does not overlap with the target. The training signals are the normalized x, y factors of the target location on the work (visual sensory) space or θ_1, θ_2 on the joint space. After the learning, the relation between the two-dimensional target location and the two outputs is observed. The result is compared with the case that the reinforcement learning was not applied beforehand.

Fig. 6 shows the relation between θ_2 and the output 2 for the case of joint space coordinates, and the relation between x and the output 1 for the case of work space coordinates. In these graphs, the output as a function of the target location is plotted for 100 target and hand location sets chosen randomly. The relation between the output and the target location without applying the reinforcement learning is closer to one-to-one correspondence than after the reinforcement learning. That was perfectly different from our expectation. At the early stage of the learning, the relation when the reinforcement learning was applied is far closer to one-to-one correspondence than the case of no reinforcement learning. That was as we had expected. In the supervised learning, the target was not located around the center of the visual field, and the excited input signals are perfectly different from those when the target is located at each corner. Therefore, the output is randomly varied when the object is located around the center. However, through the supervised learning, the outputs become to represent the target location. The reason may be as follows. The hand location is varied randomly in the learning and the influence of only the hand location has to be removed from the visual sensory signals. So the whole visual sensory signals are utilized and integrated to exclude the influence of only the hand location. The smooth representation of the hand location results in the smooth representation of the target location because the hand and target cannot be distinguished in the visual sensor. On the other hand, when the reinforcement learning was applied beforehand, only the target location information could not be extracted. That may be because the representation, in which both the hand location information and target location information are mixed, was acquired by the preceding reinforcement learning.

Next, the representation of the hand location is examined. Two outputs are trained by the supervised learning. The training signals are the normalized x and y factors of the hand location on the work (visual sensory) space. Here it is supposed that the target or obstacle does not appear in the visual field, and the hand is located randomly. Fig. 7 shows the two outputs when the hand location is varied for the following four cases to see the influence of applying reinforcement learning. In the first case, the full inputs are

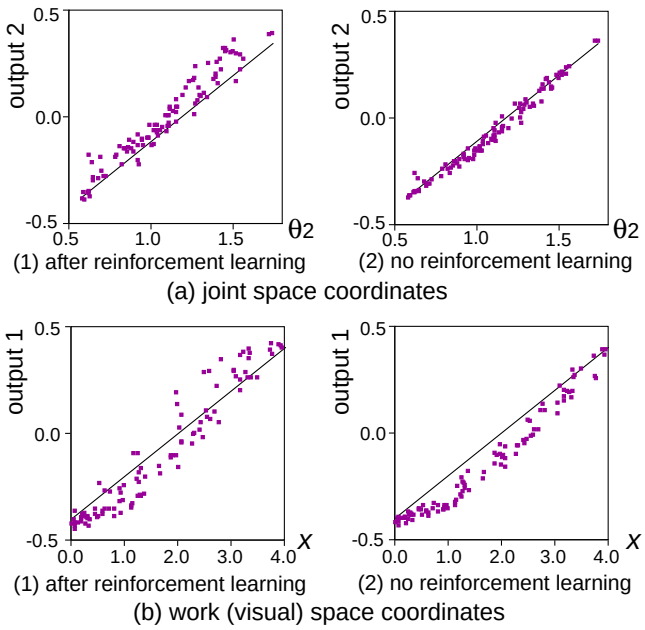

Figure 6: The comparison of the supervised learning result between after reinforcement learning and no reinforcement learning about the hidden representation of the target location.

given, and in the second case, the object location on the visual sensor is fixed at the center and only the joint angles are varied. In the third case, the object location is varied on the visual sensor but the joint angles are fixed as the hand is located at the center of the visual field. In these three cases, the hand location range is limited within the visual field, but in the fourth case, the range is wider out of the visual field. As well as the previous simulation, two cases are compared. It can be seen that the joint angle inputs mainly represent the x factor, while the visual inputs mainly represent the y factor when no reinforcement learning was applied beforehand. On the other hand, the joint angle inputs becomes to represent the both factors more, and the representation of the visual signals is difficult to be interpreted after reinforcement learning. According to the result of wide hand location range, it is easily known that the influence of visual signals becomes large after reinforcement learning. This shows that the hidden representation is mixed information of joint angles and visual signals. When another simulation is done with the different initial connection weight values, the above tendency is still valid. From these results, it is known that the representation on the hidden layer changes adaptively according to the given task through the reinforcement learning.

6 CONCLUSION

The hand-reaching task was achieved by the combination of reinforcement learning and neural network. The

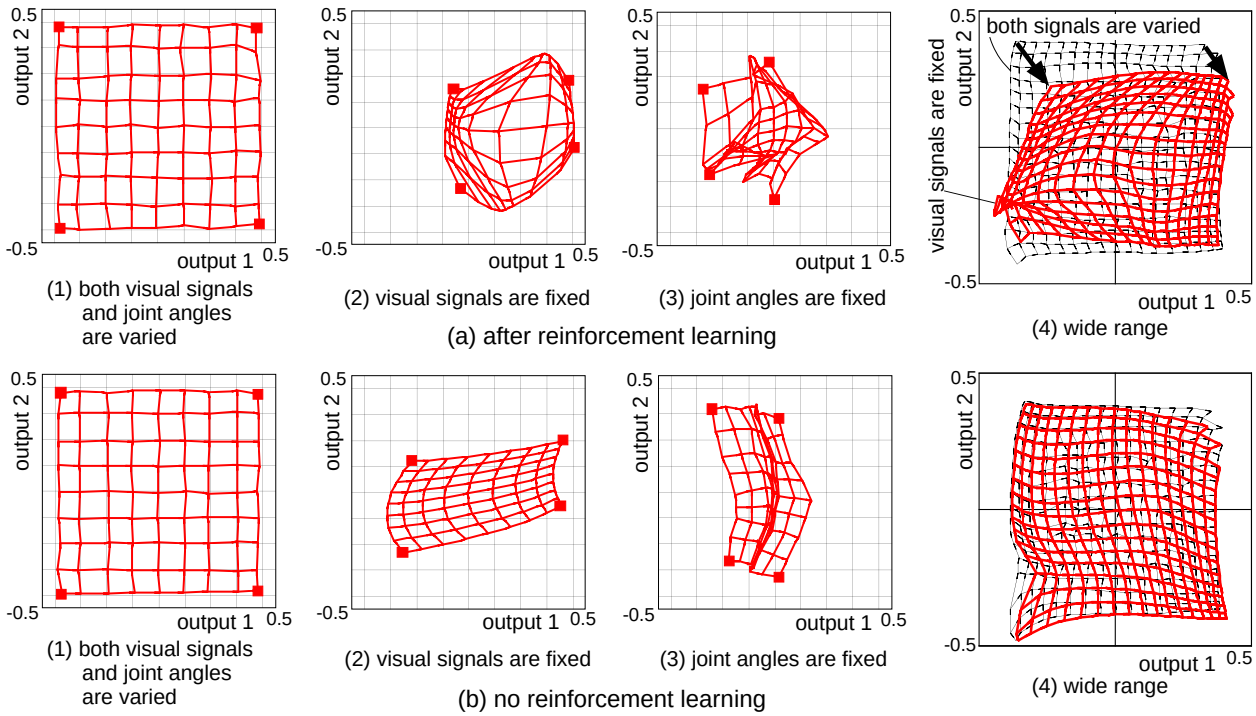

Figure 7: The comparison of the supervised learning result to see the contribution change of the joint angles and visual signals to the hand location representation on the hidden layer through reinforcement learning.

inputs are raw visual signals, two joint angles, and the binary value indicating the existence of the obstacle. The target, hand, and obstacle could not be distinguished on the vision. The hand loci after learning was smooth at the boundary whether the hand could be caught on the visual sensor or not. By the analysis of the hidden neurons' representation after the reinforcement learning by applying an additional supervised learning, the following was known. The hidden neurons did not represent the target location either on the work space coordinates or on the joint space coordinates. The hand location is represented by mixing the visual sensory signals and joint angles on the hidden layer. Such complicated representation is acquired through the reinforcement learning. Our future work is to introduce the dynamics of the arm at first. Next is to investigate the physiological results about the coding of the postcentral somatosensory cortex where the somatosensory and visual information are integrated, and then to suppose the mechanism for the acquisition of the hand-eye coordination in our living creatures.

Acknowledgments A part of this research was supported by the Scientific Research Foundation of the Ministry of Education, Science, Sports and Culture of Japan (#10450165) and by "The Japan Society for the Promotion of Science" as "Biologically Inspired Adaptive Systems" (JSPS-RFTF96I00105) in "Research for the Future Program".

REFERENCES

[1] Asada, M., Noda, S., Tawaratsumida, S. and Hosoda, K., "Purposive Behavior Acquisition for a Real Robot by Vision-Based Reinforcement Learning," *Machine Learning*, **24**, 279–303, 1996.

[2] Anderson, C. W., "Learning to Control an Inverted Pendulum Using Neural Networks", *IEEE Control System Magazine*, **9**, 31-37, 1989.

[3] not published yet.

[4] Shibata, K., Okabe, Y. & Ito, K, "Direct-Vision-Based Reinforcement Learning in "Going to an Target" Task with an Obstacle and with a Variety of Target Sizes", *Proc. of Int. Conf. on Neural Networks and Their Applications (NEURAP) '98*, pp. 95-102, 1998.

[5] R. Held & J. A. Bauer, "Visually guided reaching in infant monkeys after restricted rearing", *SIENCE*, **155**, pp. 718-720, 1967.

[6] M. Jägersand & R. Nelson, "Adaptive Differential Visual Feedback for uncalibrated hand-eye coordination and motor control", TR 579, Univ. of Rochester, 1994.

[7] Barto, A. G., Sutton, R. S. & Anderson C. W., "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," *IEEE Trans. of SMC*, **13**, 1983, pp. 835–846.