

Acquisition of Box Pushing by Direct-Vision-Based Reinforcement Learning

Katsunari Shibata and Masaru Iida

Dept. of Electrical & Electronic Eng., Oita Univ., 870-1192, Japan
shibata@cc.oita-u.ac.jp

Abstract: In this paper, it was confirmed that a real mobile robot with a CCD camera could learn appropriate actions to reach and push a lying box only by Direct-Vision-Based reinforcement learning (RL). In Direct-Vision-Based RL, raw visual sensor signals are the inputs of a layered neural network; the neural network is trained by Back Propagation using the training signal that is generated based on reinforcement learning. In other words, no image processing, no control methods, and no task information are given at premise even if as many as 1536 monochrome visual signals and 4 infrared signals are the inputs. The box pushing task is rather difficult than reaching task for the reason that not only the center of gravity, but also the direction, weight and sliding character of the box should be considered. Nevertheless, the robot could learn appropriate actions even if the reward was given only when the robot was pushing the box. It was also observed that the neural network obtained global representation of the box location through the learning.

Keywords: Direct-Vision-Based reinforcement learning, box pushing, neural network

1. Introduction

Many of modern robots are utilizing visual sensors to get plenty of information about environment. The visual sensor provides us a huge number of sensor signals. Even for the robot in which learning is a special feature, Applying image processing to the visual signals is taken for granted generally to extract some useful pieces of information and to assign the present visual signals to one state in state space. However, useful knowledge to solve a given task is often included in the image processing or other pre-processings. For example, in the work of Asada et al., when the soccer robot learned shoot action, the ball position and size, the goal position, size, and orientation were extracted from the image captured by the robot¹⁾. In that case, it is also a very intelligent process that the robot notices such information is important to solve the task, and that it finds how the such information can be extracted from the image.

These are based on the traditional idea that in order to make up high intelligence, the process from sensors to motors should be divided into some functional modules such as image processing, action planning, and control at first, then each module should be sophisticated, and finally they should be integrated into one intelligent process. This tendency can be seen in the brain research as well. However, reinforcement learning is an autonomous learning based on the sense-and-action loop as well as the learning of our living things. When we see the knowledge from the brain research, it is noticed that the boundary between functional areas is not so clear. The authors think that a variety levels of abstracted information existing between sensors and motors is the origin of our intelligence.

Direct-Vision-Based Reinforcement Learning(RL)²⁾ is one of the ways to utilize RL in robot-like system

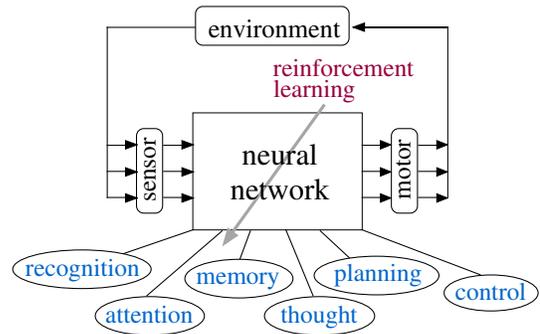


Figure 1: Direct-Vision-Based Reinforcement Learning.

with sensors and motors on the basis that given knowledge is reduced as much as possible. Concretely, a layered neural network is employed; the raw sensor signals are the input and motor commands are the output of the network as shown in Fig. 1. The main advantage is that RL does not remain only as the learning of action planning, but also can be extended as the learning for the whole process from sensors to motors including recognition, memory, and so on. The abstracted state representation in line with its purpose is formed in the neural network; that can be expected to lead to the emergence of high-order functions.

By simulation, it has been confirmed that a mobile robot with a linear monochrome visual sensor can reach a black target object²⁾³⁾. The neural network formed global representation of the target location, such as whether the target is located at the right hand side or left hand side. Each raw visual sensor signal represents only a local information about the object. This means that the neural network could integrate the local sensor signals into global representation only through the learning.

It was also shown that when the asymmetrical motion character was employed in the robot, the robot can learn appropriate motions, and the representation of the hidden neurons changes adaptively and reasonably²⁾³⁾. Furthermore, in the simulation of obstacle avoidance, the state that the target object is just behind the obstacle not depending on the object location was represented in the hidden layer of the neural network²⁾⁴⁾. This information can be considered as a higher order representation than the information of the object location. Moreover, in Direct-Vision-Based RL, the learning is fast and stable due to the local representation of the input signals²⁾⁵⁾.

It has been confirmed that a real mobile robot named Khepera with a CCD camera could learn to reach a target object by Direct-Vision-Based RL even though reward was given only when the robot reached the target, and no image processing was given beforehand for $64 \times 24 = 1536$ visual sensor signals⁶⁾. However, the task itself is easy in the meaning that a designer can write a program to realize such motions easily.

In this paper, a rather difficult task, “Box Pushing” is employed. It is examined whether the robot can learn to reach and push a lying rectangular parallelepiped box without any advance knowledge about the task. In this task, the robot should vary its motion according to not only the location, but also the direction of the box. It should also know the degree of sliding and the weight of the box; those cannot be obtained from the image, but from its experiences.

2. Reinforcement Learning

In this paper, actor-critic architecture⁷⁾ is employed, and actor (action command generator) and critic (state value generator) are composed of one layered neural network. This means that the hidden layer is used commonly by both actor and critic. TD (Temporal Difference) is applied for the learning of the critic. TD error is defined as

$$\hat{r}_t = r_t + \gamma P(\mathbf{s}_t) - P(\mathbf{s}_{t-1}), \quad (1)$$

where γ is a discount factor, r_t is a reward, \mathbf{s}_t is a state vector (sensor signals), and $P(\mathbf{s}_t)$ is a state value. The state value at the previous time $P(\mathbf{s}_{t-1})$ is trained by the training signal as

$$P_s(\mathbf{s}_{t-1}) = P(\mathbf{s}_{t-1}) + \hat{r}_t = r_t + \gamma P(\mathbf{s}_t), \quad (2)$$

where $P_s(\mathbf{s}_{t-1})$ is the training signal for the state value. On the other hand, the motion commands of the robot is proportional to the sum of the outputs of $\mathbf{a}(\mathbf{s}_t)$ and random numbers \mathbf{rnd}_t as trial and error factors. The actor output vector $\mathbf{a}(\mathbf{s}_{t-1})$ is trained by the training signal as

$$\mathbf{a}_s(\mathbf{s}_{t-1}) = \mathbf{a}(\mathbf{s}_{t-1}) + \hat{r}_t \mathbf{rnd}_{t-1}. \quad (3)$$

The neural network is trained by Back Propagation according to Eq. (2) and (3). By this learning, motion commands are trained to gain more state value.

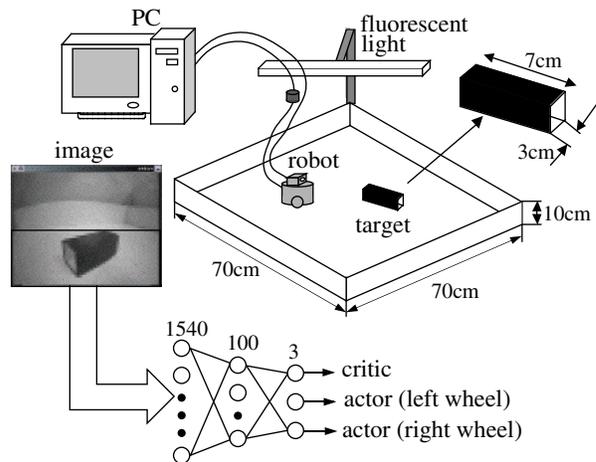


Figure 2: Experimental environment.

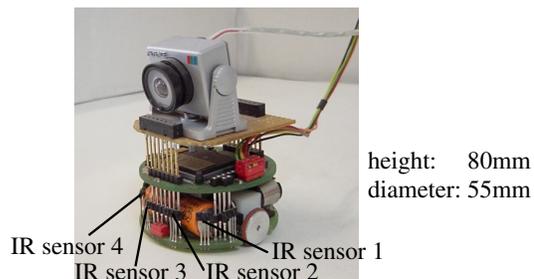


Figure 3: A mobile robot named “Khepera” with a CCD camera.

3. Experiment

3.1 Experimental Setup

The experimental environment is as shown in Fig. 2. The action area is 70×70 cm which is surrounded by a height of 10cm white paper wall, and a fluorescent light is set to keep stable brightness. As shown in Fig. 3, a small mobile robot (AAL, Khepera) has one CCD camera (KEYENCE, CK-200) with 114 degree of visual field by a wide angle lens. By the property of the camera, the central part of the image is brighter than the peripheral part. Furthermore, due to distortion, a straight line becomes curved around the right or left edge of the image. The visual sensor image is captured by a capture card on a PC. The number of pixels is 320×240 originally, but by the limitation of memory, only the lower half of the image was used after transforming into a monochrome image and averaging 5×5 area. Then $64 \times 24 = 1536$ visual signals are the input of the neural network after normalizing into a real number between 0.0 and 1.0. Here, the value for the darkest pixel is 1.0, and that for the brightest one is 0.0. Four infrared(IR) sensor signals are also added to the input. All of them are located at the front of the robot as shown in Fig. 3. Each of these sensors is used like a touch sensor such that the input signal from the sensor to the neural network is a binary value; it is 1.0 when the box is located just in front of the IR sensor and the sensor takes the maximum value.

The target object is a lying rectangular parallelepiped box made of paper. The size is $30\text{mm} \times 70\text{mm} \times 30\text{mm}$. Since the contents are empty, it is very light. The outer color is black, while the inner color is white. Since the box has a pipe-like shape, and the smaller sides are covered with no paper, the white inside is seen through the smaller sides.

The neural network has three layers; the number of neurons in each layer is 1540 in input layer, 100 in hidden layer, and 3 in output layer. The initial hidden-output connection weights are all 0.0, while input-hidden weights chosen randomly from -0.1 to 0.1. One of the outputs is used as critic after adding 0.5. A small reward 0.018 is given when two IR sensors (No.2 and 3 in Fig. 3) take the maximum value and the both motor commands are positive. When the robot misses the box out of its visual field, critic is trained to be 0.1. This corresponds to -0.4 for the training signal of the neural network. When the robot continues to get the reward for 10 time steps, the robot misses the box, or 50 time steps passes, one trial finishes.

Two of the three outputs are used as actor outputs. Each of them is used to generate a motor command for the right or left wheel. The random number added to each actor output as a trial factor is a uniform random number powered by 3.0 whose value range is -0.1 to 0.1. The actor output after added by the random number is multiplied by 8.0, and one of the integer number from -3 to 3 is chosen by rounding off. The number is sent to the robot as a motor command for each wheel through RS232C. At the beginning of the learning, the random number that is less than 0.05 was not used, because the motor command becomes 0 when a small random number is rounded off. If the training signal for each of three output neurons is less than -0.4 or more than 0.4, the training signal is set to be -0.4 or 0.4 respectively.

Next, it is explained how to decide the initial location of the robot at the beginning of each trial. At the first stage of the learning, a target center of gravity is chosen randomly in a trapezoid area in the image, the robot is controlled in order that the center of gravity of the black area of the binarized image comes close to the chosen target center according to the given program. When the difference between the center of gravity and the target center is within 1 pixel, the learning begins. At the beginning of the learning, the trapezoid area is very small and is located at the lower part of the image so as that the robot is located just in front of the center of the long side of the box. The trapezoid area becomes to spread wider to the upper area of the image gradually according to the progress of learning. In the most cases, the robot faces the long side of the box, and the angle between the moving direction and the long side of the box was not different so much from 90 degree. At the second stage of the learning, at the half of the trials, the angle between the moving direction and the long side of the box begins to vary by rotating on the box after reaching the target location of the stage 1. The angle becomes larger as the learning progresses. When the

box comes close to the white wall, the box was moved by the authors just after the trial.

All the learning is performed on-line using the real mobile robot. No learning on simulation was done. Fig. 4 shows the timing chart when the robot is learning its action. One time step corresponds to 320msec. The necessary time to execute each process is approximately as shown in Fig. 4. The video signal transmission includes the transformation into monochrome image and averaging operation of 5×5 pixels. The learning includes not only backward computation of the neural network, but also two sets of forward computation. That is because in the learning phase, the input signals at one time step before have to be entered, and after the backward computation, forward computation was done for the present input signal to reflect the weight change by the preceding learning in the critic output. As shown in Fig. 4, action commands are transmitted at the halfway between two successive capturing times. In other words, the TD error is influenced by both the present and previous action commands actually.

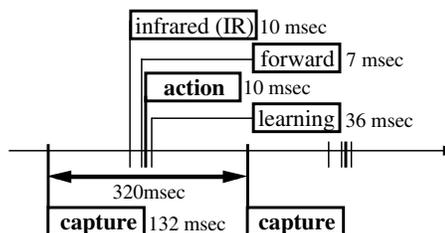


Figure 4: Timing chart of each main event.

3.2 Result

The robot could learn to go forward soon after beginning, and the rotation depending on the box location could be observed after 300 trials of learning. Figure 5 shows two samples of robot's behaviors after 5000 trials. Although no knowledge about image processing, control and task was given to the robot, it is seen that the robot could reach the target box and continue to push it.

The robot's motion depended not only on the location of the box, but also on the direction of the box. Then the box was located as one of two ways as shown in Fig. 6. Fig. 7 shows the robot's loci and sequences of the captured images, and Fig. 8 shows the change of the center of gravity in binarized image. It is seen that even though the location of the box is the same, the loci are different when the direction of the box is different.

When the box was put as (a) in Fig. 6, the robot went straight at first, then rotated anti-clockwise, and reached almost the center of the long side of the box as shown in Fig. 7(a). While, when the box was put as (b) in Fig. 6, the robot rotated anti-clockwise at first, then went straight as shown in Fig. 7(b). It is seen that the robot rotated clockwise slightly in the latter half of the trial, and finally, it reached the right edge of long side

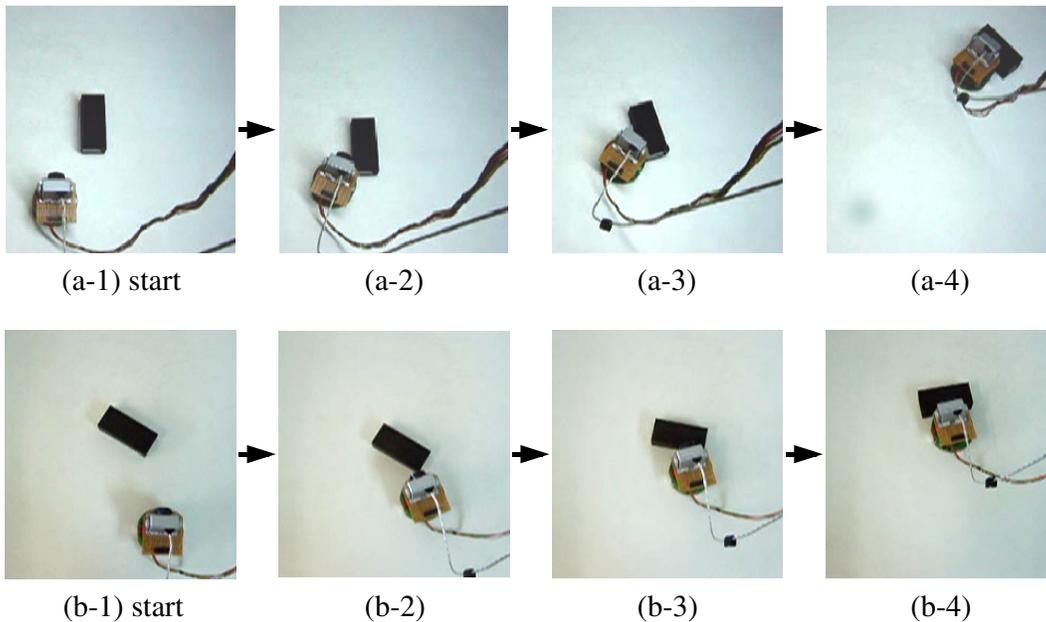


Figure 5: Two examples of the robot behaviors after learning.

of the box. A sequence of photos from top view of this trial also can be seen in Fig. 5(b).

The reason of the behavioral difference is suggested as follows. In the case of (b), if the robot makes a frontal approach toward the center of the long side, it has to go a long way round and takes a long time to reach. On the other hand, if it goes forward, the edge of the box is caught by one of the IR sensors; the robot cannot rotate relatively to the box. Furthermore, if the approaching angle is small, both IR sensors can not take the maximum value soon. Accordingly, the robot rotated at first, and then approaching the box while keeping the approaching angle in some degree. After the robot touched the box, since the box rotated by robot pushing, the robot moved from in front of the right edge of the long side to in front of the center. Since the task required the location and direction of the box as above, it can be said that the location and direction could be extracted from many visual signals through learning.

Fig. 9 shows the connection weights with the input units for each of three hidden neurons. Each of those has the maximum connection weight with one of the output neurons, ignoring sign. The weight value looks just a random number at a glance. However, by careful looking, it is noticed that the connection weight that is projected on the upper area (y is large) has a larger value in the hidden neuron No. 32 that mainly contributes the critic output. The shape of the area where the weight value is small (black) is similar to the image that the box is just in front of the robot as shown in the figures in the lowest row in Fig. 7. In the other two neurons that contribute the actor outputs, the connection weight that is projected on the right area (x is large) has a larger value. From the shape of the area where the absolute weight value is large, it is thought that these neurons detect lateral shift of the box from

the situation that the box is just in front of the robot. Furthermore, the irregularity of the weight value distribution was originated from the initial weight value.

Table 1 shows the change of the correlation between x or y and weight value through learning where (x, y) is the corresponding pixel location in the image. In the hidden neuron No. 32, the absolute correlation between y and weight becomes larger. While, in the hidden neuron No. 70 and No. 34, the correlation between x and weight becomes larger. This means that these neurons represent global information through learning, keeping the information of the initial connection weights. This is the same as observed in the paper⁸⁾ in which global information is given as the training signal of a neural network whose input is local sensor signals.

Then, some actual images are captured by locating the box in order. In one series of the box location, the forward distance y from the robot was constant and the lateral distance x was varied. In the other series, the lateral distance x was constant and the forward distance y was varied. In both cases, the long side of the box is perpendicular to the moving direction of the robot.

Fig. 10(a) shows the hidden neurons' outputs as a function of the lateral distance x in the former case, while Fig. (b) shows the hidden outputs as a function of the forward distance y in the latter case. x and y coordinates are the same as shown in Fig. 6. Totally, it is seen that the irregularity of the output curve becomes smaller through learning. The hidden neuron No. 32 represents mainly whether the box is just in front of the robot or not. The hidden neuron No. 34 represents mainly whether the box is located in the right hand side or left hand side, while the hidden neuron No. 70 does not represent clearly like No. 32 and No. 34. It can be thought that the concept of close or far, and the concept of right or left can be obtained through learning.

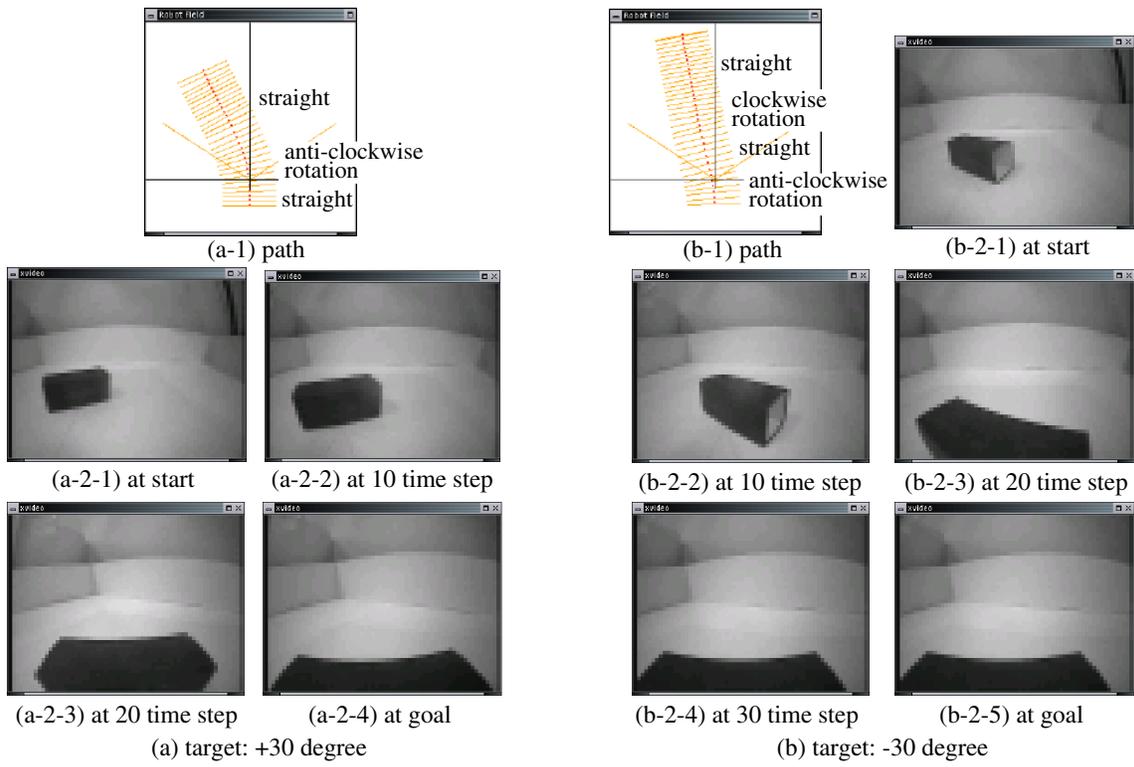


Figure 7: The robot locus and a series of images after learning for each of the two box directions at the same place.

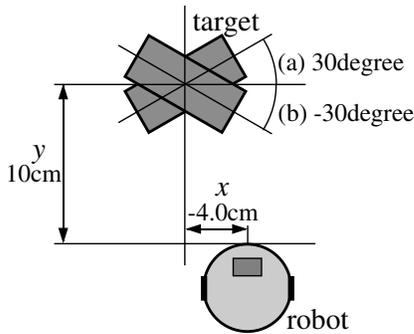


Figure 6: The location and direction of the box in the following experiment.

4. Discussion

In this experiment, the robot began to go forward just after the learning starts. This is because the reward is given only when the motor commands for two wheels are both positive, and the reward can be obtained often because the robot is located just in front of the long side of the box initially at each trial. The reason why the learning performed well may deeply depend on the small random trial and the small learning rate. If they are large, the robot may learn going backward. Once the robot learns going backward, it hardly gets the reward. On the other hand, in the above experiment, the robot could not obtain the action to rotate at the same place even if the action seems optimal. One possible reason is that the random number and the learning rate is too

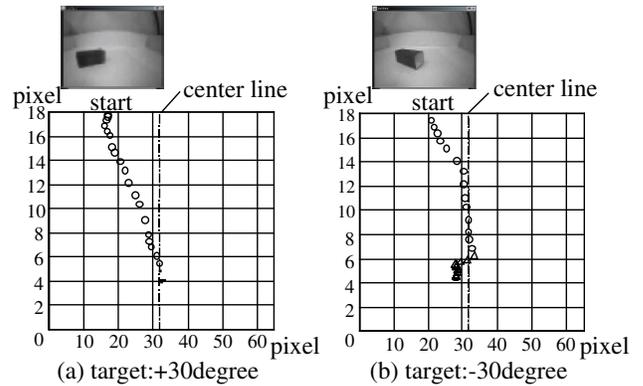


Figure 8: The change of the center of gravity of the box. Plot symbol indicates the number of IR sensors that take the maximum value as \circ : 0, \triangle : 1, and \bullet : 2.

small to learn in 5000 trials inversely.

From the above discussion, the reward, the initial location at each trial, and the random factor at each time step are very important factors for the learning. Strictly, it can be said that they are some given knowledge to the robot. However, it does not constrain the learning, it is better to say that they are not some direct knowledge but some hints for the robot.

5. Conclusion

A real mobile robot could learn to go and push a box without giving any image processing, control methods,

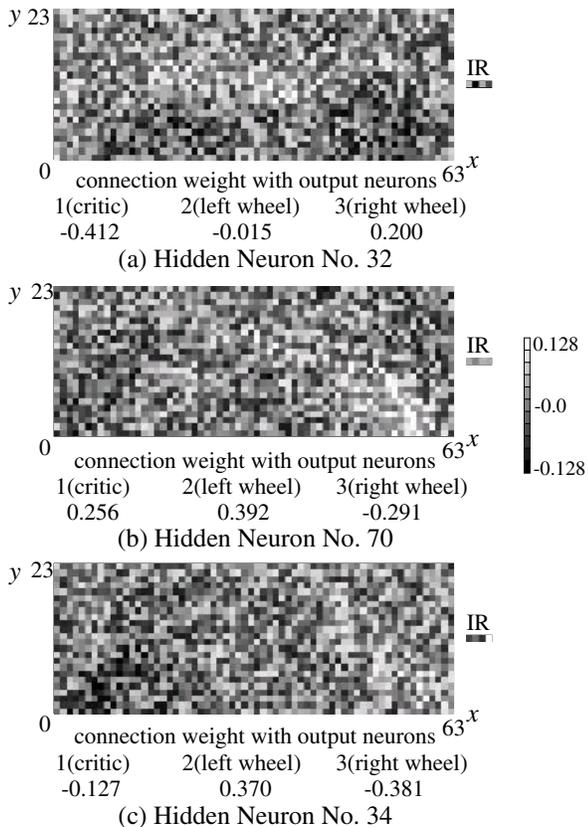


Figure 9: Three hidden neurons' connection weights with the inputs. Each hidden neuron has the maximum connection weights with one of the output neurons.

and task knowledge directly. It can be said that the neural network extracts the location and direction from the image with 1536 pixels to generate a series of appropriate motions only by reinforcement learning. It was also observed that the neural network extracted some global information. However, since the environment is very ideal, application to more real world is one of the most important problems. The way of efficient trial is also a big problem to be solved.

Acknowledgment

This research was supported by the Grants-in-Aid for scientific Research of the Ministry of Education, Culture, Sports, Science and Technology of Japan (#13780295, #14350227, #15300064).

References

- [1] Asada, M., Noda, S., Tawaratsumida, S. and Hosoda, K. (1996) Purposive Behavior Acquisition for a Real Robot by Vision-Based Reinforcement Learning, *Machine Learning*, **24**, 279–303.
- [2] Shibata, K., Ito, K. & Okabe, Y. (2001) Direct-Vision-Based Reinforcement Learning Using a Layered Neural Network - For the Whole Process from Sensors to Motors -, *Trans. of SICE*, **37**(2): 168–177 (in Japanese).
- [3] Shibata, K. & Okabe, Y. (1997) Reinforcement Learning When Visual Sensory Signals are Directly Given as Inputs, *Proc. of ICNN'97*, **3**, 1716–1720.

Table 1: Change of the correlation between the connection weight value with the input neurons and x or y coordinate of the corresponding visual cell in the image.

		before learning	after learning
(a) hidden neuron No. 32	x	-0.219 $\times 10^{-3}$	\Rightarrow -0.307 $\times 10^{-3}$
	y	0.085	\Rightarrow -1.311
(b) hidden neuron No. 70	x	0.025 $\times 10^{-3}$	\Rightarrow 0.353 $\times 10^{-3}$
	y	-0.029	\Rightarrow 0.147
(c) hidden neuron No. 34	x	0.168 $\times 10^{-3}$	\Rightarrow 0.607 $\times 10^{-3}$
	y	-0.178	\Rightarrow -0.482

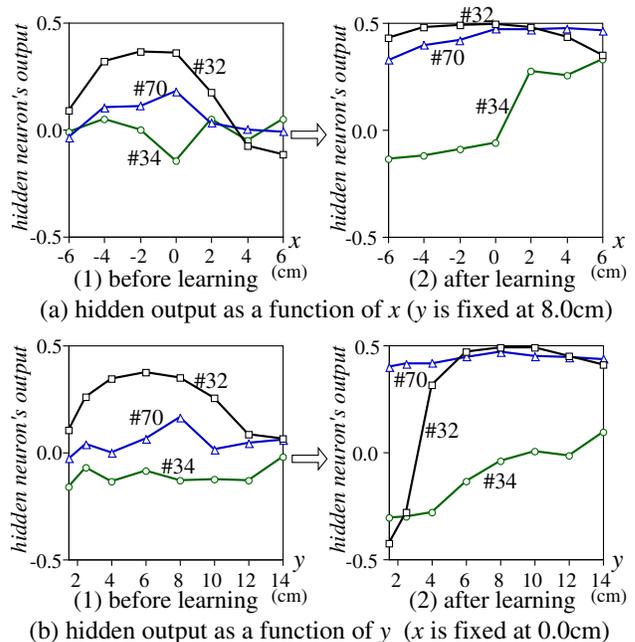


Figure 10: Hidden neuron's output as a function of the box location. Here, real captured image is put into the neural network.

- [4] Shibata, K., Ito, K. & Okabe, Y. (1998) Direct-Vision-Based Reinforcement Learning in "Going to an Target" Task with an Obstacle and with a Variety of Target Sizes, *Proc. of Int'l. Conf. on Neu. Net. & Their Appli. (NEURAP) '98*: 95–102.
- [5] Shibata, K., Sugisaka, M. & Ito, K. (2001) Fast and Stable Learning in Direct-Vision-Based Reinforcement Learning, *Proc. of the 6th AROB (Int'l Symp. on Artificial Life & Robotics)*, **1**: 200–203.
- [6] Iida, M., Sugisaka, M. & Shibata, K. (2003) Application of Direct-Vision-Based Reinforcement Learning to a Real Mobile Robot with a CCD Camera, *Proc. of the 8th AROB (Int'l Symp. on Artificial Life & Robotics)*, **1**: 86–89.
- [7] Barto, A. G., Sutton, R. S. and Anderson, C. W. (1983) Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems, *IEEE Trans. SMC-13*: 835–846
- [8] Shibata, K. & Ito, K. (2002) Adaptive Space Reconstruction and Generalization on Hidden Layer in Neural Networks with Local Inputs, *Technical Report of IEICE*, NC2001-153, 151–158