Establishing Interaction between Machine and Medaka using Deep Q-Network

Ryo Nishimura
Department of Engineering
Hokkaido University
North14 West9, Sapporo,
060-0814, Japan

Hiroyuki Iizuka
Graduate School of
Information Science and
Technology
Hokkaido University
North14 West9, Sapporo,
060-0814, Japan

Masahito Yamamoto
Graduate School of
Information Science and
Technology
Hokkaido University
North14 West9, Sapporo,
060-0814, Japan

{nishimura,iizuka,masahito}@complex.ist.hokudai.ac.jp

ABSTRACT

Social interaction is the basic ability for animals to survive. It is difficult for a machine to interact with human or other animals because it is not clear how the machine should interact. This paper examines whether an artificial dot controlled by a machine can interact with a medaka and induce a desired behavior. The dot is displayed on a monitor. We use deep Q network (DQN) to learn how to move the dot. As a result, the DQN could learn some basic elements to interact with the medaka and the desired behavior could be induced.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms

Keywords

machine animal interaction, deep Q network, real time, medaka

1. INTRODUCTION

Social interaction is the basic ability to communicate with other individuals to survive a world for any animals including human. For us, it is easy to understand what happens when interacting with others because we are the actual entities that perform communication. The social interaction can be cooperative or uncooperative. It becomes very difficult to understand what happens and what they communicate when the entities that perform communication are not human but animals. We can speculate it to some extent but it is usually limited to static information like the relation of two or their emotional states, i.e. anger or happy. The best way for this is that we develop an interface somehow to communicate with animals. The simplest interface is a common language. However, there is no such a language invented so far. Another approach is to develop a machine to communicate with animals instead of humans. In other words, it is to develop an animal model of the entity that can communicate with others as we develop the artificial intelligence for human. To build a model of animal or human intelligence is equivalent to understanding the animal itself and the social interaction of them.

Recently, many engineers are working on developing robots or software agents that establishes smooth communication or interaction with human. However, it is still far from human communication because it requires high human intelligence.

On the other hand, there are studies to attempt to build an internal model of lower animals, that is small fish. The advantage to use fish is that it is easy to take care of them and to build an experimental environment. Matsunaga and Watanabe investigated what kind of models can generate life-like behaviors, especially plankton-like behaviors that attracts medaka [1]. In their experiments, small white dots are displayed to medaka by a monitor attached on a tank and a variety of motions that have different power spectrums are shown. Their results show that pink noise motion (a density proportional to the inverse of its frequency) induce more predatory behaviors. It means that the white dots moving in a pink noise manner looks food for medaka and the pink noise movements are somehow more related to the life-like behaviors than the others.

Takayasu and Watanabe examined if medaka tries to form a shoaling behaviors with a biological motion of medaka in which the body structure from head to tail of medaka is represented by only isolated small number of dots [2]. The dots moves in a coordinated manner as if they are attached on a real medaka but only dots are visible. It is known that human can tell even gender correctly from human biological motion. The dots are recognized as just points without moving however when the dots move in a coordinated manner, e.g. walking, the artificial points can induce the feeling of the life presence for human. To investigate if the feeling

BICT 2015, December 03-05, New York City, United States

DOI 10.4108/eai.3-12-2015.2262508

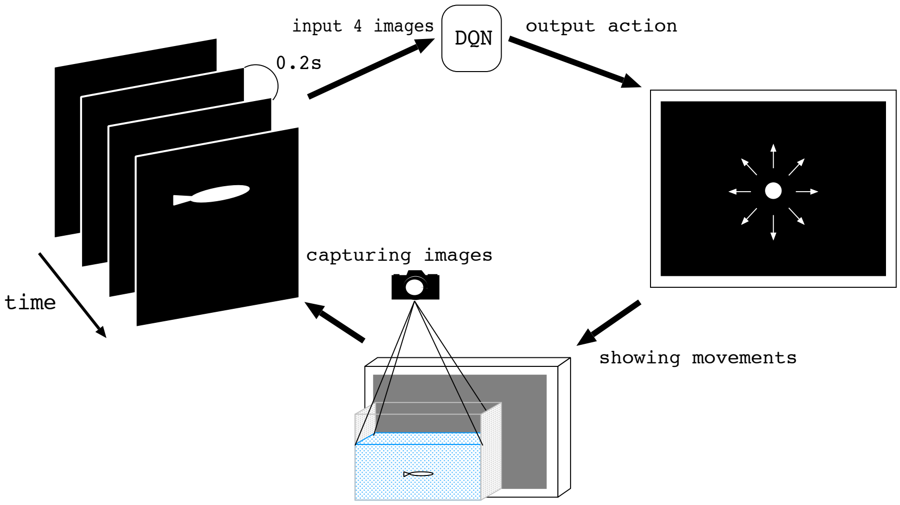


Figure 1: Cycle of our experiment.

of life can also be induced to medaka by medaka biological motion, they see whether medaka forms a shoaling behaviors with the biological motion. Their results show that the artificial points can induce more shoaling behaviors than other control stimulus of dots.

These two studies show that there is a certain mechanism in medaka to feel life from artificial dots movements and the mechanism must work when medaka recognizes the other presence, i.e. friend or food. It is successful to establish a certain form of interaction between a real life and a machine. However, those approaches are rather passive in a sense that the real-time medaka motion is not taken into account. Just predefined motions calculated from a model or recorded biological motions are shown to medaka and they moves regardless of real-time medaka motions. The recorded motions are not same as the motions for the immediate response to the partner's motion. The recorded motions can break down the social interaction in human and animals [3–5]. What we tried in this paper is more active in a sense that the artificial movements, i.e. white dots movements, shown to medaka can respond the realtime medaka motions and we investigate whether it is possible to control or induce their motions as what we desire through the interaction with them or to see the establishment of communication between a real life, medaka here and a machine.

2. METHOD

In order to obtain the artificial movements that can respond to medaka and control or induce medaka behaviors autonomously, we use deep Q network (DQN) as one of the successful reinforcement learning method. DQN is proposed by [6] and they show that DQN can learn sequential behaviors that can get good scores in computer games [7]. Those behaviors were gradually obtained through try and errors as human game player does. The advantages of reinforcement learning is that they can learn the good behaviors without any teaching signals. When a good behavior is generated, e.g. eating food or achieving a subgoal, the state and action are rewarded and connected states and behaviors with them are indirectly evaluated by propagating the rewards.

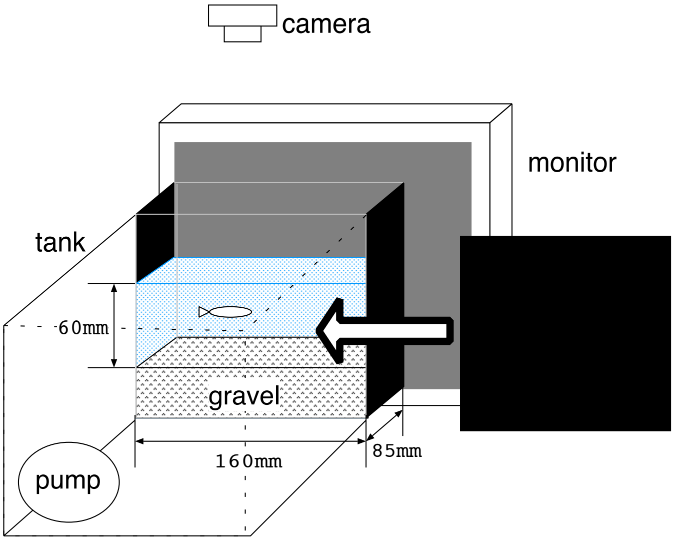


Figure 2: Experimental set-up

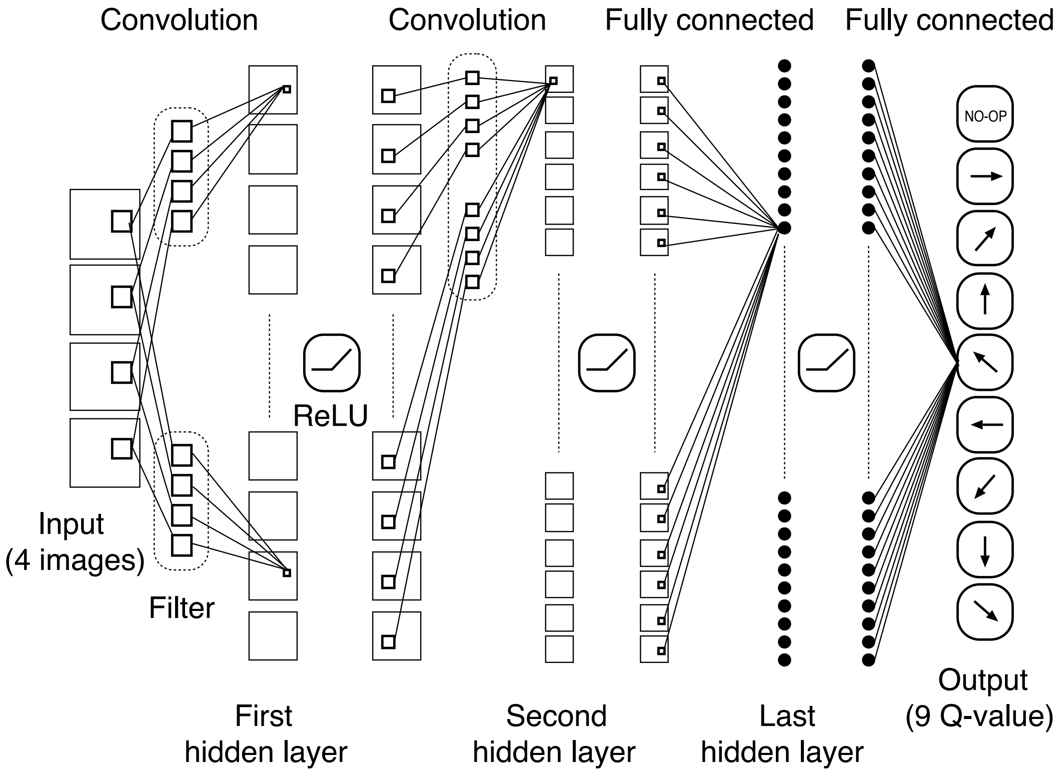


Figure 3: Structure of deep Q network.

The reinforcement learning is combined with a deep neural network in DQN. Therefore, DQN can receive the high dimension input and can decide which action should be taken. We use DQN to create a motion controller of an artificial dot that decides how to move while receiving the visual images of real-time medaka movements. The whole cycle of our experiment is shown in Fig. 1.

2.1 Experimental Conditions

A single medaka participated in the experiment. The medaka is released in a tank. The experimental environment is shown in Fig. 2. The size of the tank is 160×350×240 mm but the space where the medaka can swim is restricted 160×85×60 mm. The experimental space is covered by black boards on the sides of tank is to shut out the external environment except for the single side where a monitor is attached to show an artificial movement of a white dot. The bottom of the space is cover with white gravel. There is no ceiling on the tank and a camera is placed over the tank to capture the real-time images of the medaka movements. An air pump is placed in the tank but not in the experimental area.

The camera captures the images of the medaka from the top. The images are converted to binary 84×84 images and then the images are shifted in response to the position of the white dot in order to give the relative position of the medaka from the dot to DQN. Therefore, DQN can learn how to move the white dot depending on the relation to the medaka from the dot point of view.

A small white dot whose diameter is 1.47 mm on the monitor is shown to the medaka. The movement of the white dot is decided by DQN outputs every 200 ms. One of 9 movements, i.e., 8 directions and no-op, is chosen. The position is updated to move with a constant velocity, 4.9mm/s, to the chosen direction every 20 ms along with the monitor frequency. The area where the white dot can move is 176.4 mm × 64.68 mm. The space form a torus. When the white dot moves out of the space, it appears from the opposite side. Every 1,000 learning steps, the white dot position is reset to the center of monitor area.

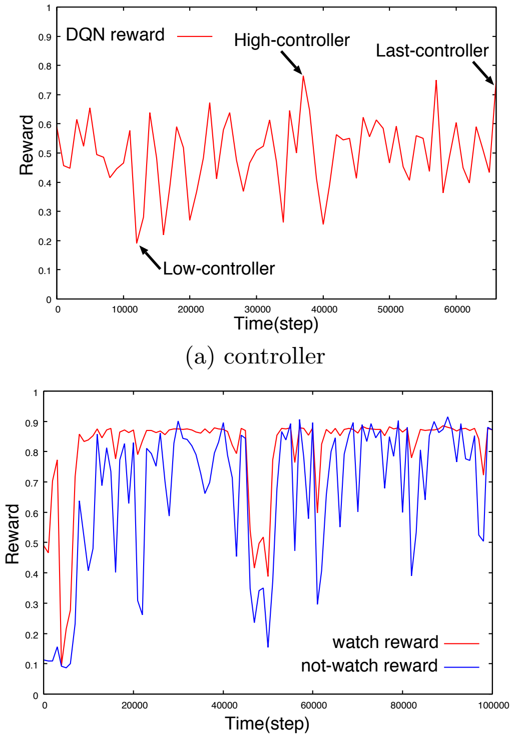
In another condition experiment, the facing direction of medaka is considered. The direction dicision needs the regression line of white pexels in the image and the vector from the center of the bounding box to the average point of white pixels. The closer to the vector of the regression line direction is regarded as the facing direction of medaka. The learning mechanism works only when medaka is facings to the monitor, i.e. the facing direction is in the range from -90 to +90 degree, in the all four input images. In this condition, the tank bottom is covered by black boards, and DQN output every 100 ms.

2.2 Deep Q Network

Here, we explain how to learn DQN as a controller through try and error. Firstly, DQN decides an action followed by a current policy, which is formed by output Q-values of DQN, at the current input state and execute the action. Secondly, DQN receives a reward and the next input state and stores the transition in a replay memory in which the past transi-

Algorithm 1 Deep Q-learning with Experience Replay [6]

```
Initialize replay memory $D$ to capacity $N$
Initialize action-value function $Q$ with random weights $\theta$
for episode $= 1, M$ do
    input sequence $s_1 = \{x_1\}$
    Preprocessed sequenced $\phi_1 = \phi(s_1)$
    for $t = 1, T$ do
        With probability $\epsilon$ select a random action $a_t$
        otherwise select $a_t = \arg\max_a Q^*(\phi(s_t), a; \theta)$
        Execute action $a_t$
        Observe reward $r_t$ and image $x_{t+1}$
        Set $s_{t+1} = s_t, a_t, x_{t+1}$
        Preprocess $\phi_{t+1} = \phi(s_{t+1})$
        Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $D$
        Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $D$
        Set $y_j = \begin{cases} r_j & \text{if terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q'(\phi_{j+1}, a'; \theta') & \text{else} \end{cases}$
        Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ with respect to the network parameter $\theta$
    end for
end for
```



(a) controller

(b) considered the facing direction

Figure 4: Reward during learning

tions are stored. Lastly, DQN weights are updated to reduce errors of output Q-values in terms of the transitions in the replay memory. A single step consists of these processes. A good behavior is gradually obtained by repeating the steps. The detail procedures are shown in Algorithm 1 and described in [6].

In this work, the DQN receives the last four images captured by a camera as the inputs. The camera sends the pictures to the DQN every 0.2 sec. It means the four input images are taken 0, 0.2, 0.4, and 0.6 seconds ago. The input images are binary 84 × 84 images shifted to make the dot be at the center. The input images are sent to DQN. The deep network of DQN consists of a convolutional neural network [6,8] whose architecture is shown in Fig. 3. The first convolution layer has 16 filters whose size is 8 × 8. The 84 × 84 input images are convolved with the filters. The second convolution layer has 32 filters whose size is 4 × 4. The outputs are converted to 9 outputs through 256 hidden units by the last fully-connected layer. The rectified linear unit (ReLU) is used as the activation functions. The DQN outputs are 9 Q-values for 8 directions and no move. The action is chosen with ϵ-greedy. The initial value of ϵ is set to 1 and then minus 0.1 every 1,000 steps. After 9,000 steps, it is kept 0.1.

For the sake of simplicity, the task of this paper is set to attracting the medaka and moving it to the monitor, which means that the best state is that the medaka always stay close to the monitor. To evaluate such a state, the reward r_t at step t is gven every step as follows.

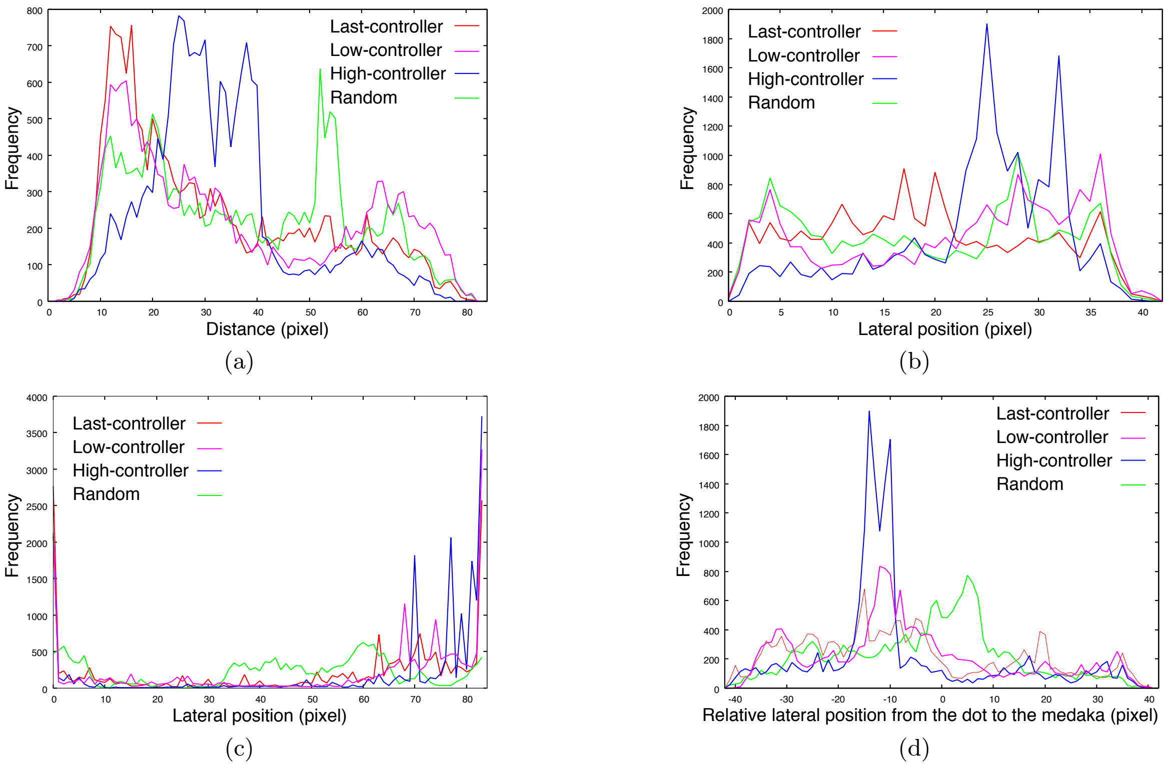
$$r_t = 1 - \frac{z_t}{z_{max}} \tag{1}$$

where, z_t is a distance from the center of the medaka to the monitor at step t. The lateral posiiton is ignored. z_{max} indicates the maximum distance from the monitor. Our DQN is implemented with Caffe, deep learning library [9]. DQN learning parameters are set as follows, discount rate = 0.95, batch size = 32, base learning rate = 0.2, gamma = 0.1, stepsize=10000, learning rate is updated by Ada delta, momentum = 0.95.

3. RESULT

DQN is trained for 3 hours and 36 minutes which corresponds to 65,000 steps, at each of which action decision and an update of DQN weights are performed. In addition, DQN is trained for 2 hours and 46 minutes which corresponds to 100,000 steps in the facing direction considered condition.

Figure 4(a) shows the average rewards every 1000 steps (= 200 s) during training. The improvements of the rewards is not clear but there are periods of high and low rewards. Therefore, we pick up the DQN weights obtained during high and low reward periods and the DQN weights at the end of experiment (= at 65,000 steps), whose reward is the third best actually. Then, we test how the medaka behaves towards the white dot controlled by the DQNs for 18,000 steps (= an hour). We call DQN weights during high and low reward periods high- and low-controller, respectively. The DQN weights at 65,000 steps is called last-controller. Those three controllers are also compared with a random controller as a control condition. It should be noted that



(a)

(b)

(c)

(d)

Figure 5: (a) distance from the medaka to the monitor. (b) lateral position of the medaka. (c) lateral position of the white dot. (d) relative lateral position from the white dot to the medaka.

the DQN weights of all controllers do not change during the test.

Figure 5(a) shows the histogram of the distances from the monitor to the medaka during the test. The lateral positions are ignored. Basically, the medaka spends a lot of time near the monitor in all cases, which is consistent to the known fact that the white dot can attract the medaka [1]. The last-controller can attract the medaka the most in all controllers. The high-controller cannot attract the medaka very close to the monitor however the medaka does not move far away, which means that it is less frequent that the medaka lose interest completely. The low-controller can also attract the medaka more than the random-controller. Therefore, the improvement of the rewards during training is not clear but the obtained DQN must have learned something that attracts the medaka.

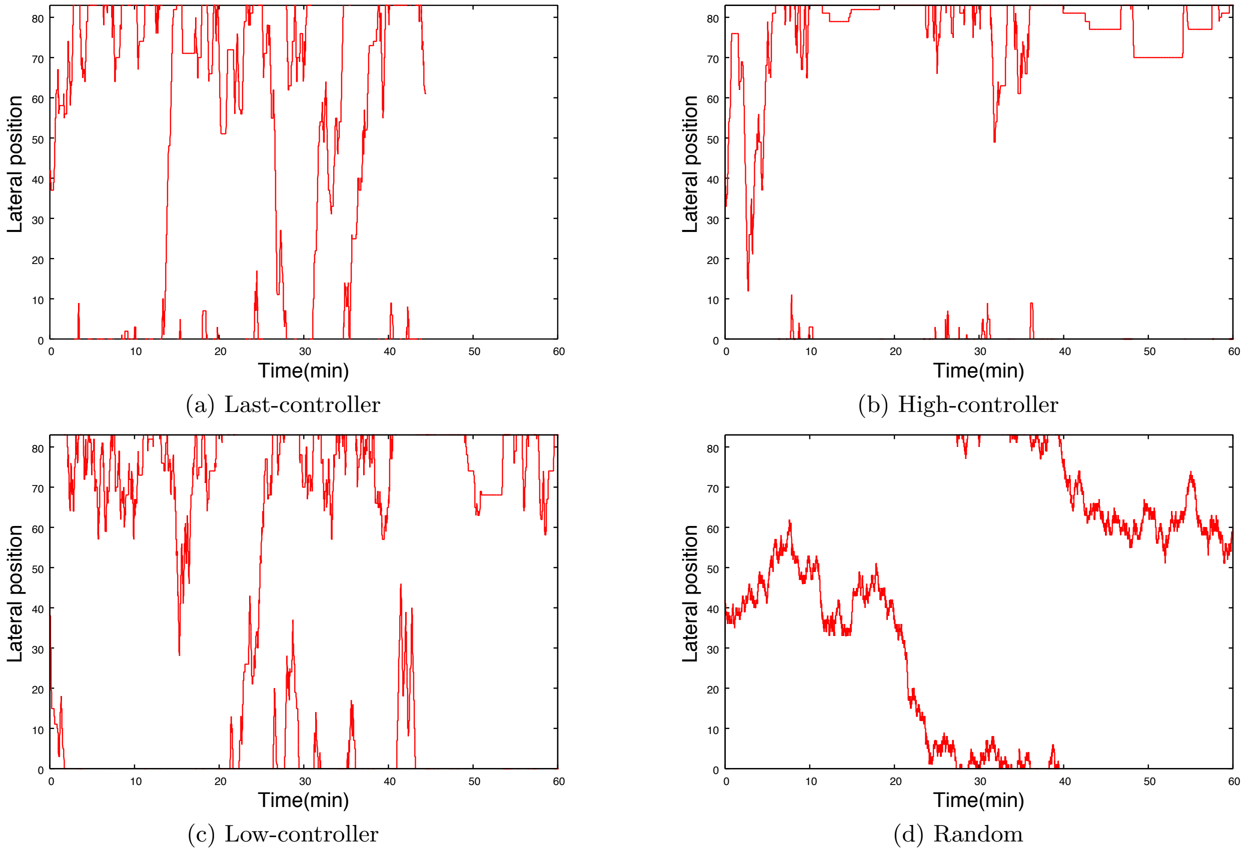
Figure 5(b) and 5(c) shows the histograms of absolute lateral positions of the medaka and the white dot, respectively. The tendencies of the medaka positions are almost same except for the high-controller, which attracts the medaka at certain position. The white dot controlled by a trained DQN often stays at the edge of the monitor. It is not clear why they prefer the edge but it might be curious for the medaka because the white dots sometimes disappear because it moves out of the monitor (it appears from the opposite edge). Figure 5(d) shows the histogram of the relative position of the medaka from the white dot. Medaka tends to be in front of the white dots in the case of the random controller. However, interestingly, the medaka stays in about 3 cm away from the dot. It is salient in all 3 controllers except for the random controller. The distance 3 cm is not far but not close either. The distance might be attractive for the medaka.

Figure 6 show the white dot movement in each case during the test. Trained DQN show crossing the boundary over and over. The movements of the white dot and the medaka in a short timescale are shown in Fig. 7. The white dot and the medaka movements looks synchronized sometimes. The white dot cannot control the speed continuously, however, it moves together with the medaka using 9 different motions.

Figure 4(b) shows the average rewards every 1,000 steps (= 100 s) during training in the facing direction considered condition. This result seems DQN could learn successfully around 10,000 steps. When the medaka is regarded to the medaka is watching the white dot, the rewards are high and stable. Therefore, the facing direction is efficient to learn, because it often occurs that the medaka does not watch the white dot. Even if the white dots moves in a very attractive manner, it does not affect the medaka behavior at all when the dot is not invisible.

4. DISCUSSION

As already described, it is known that medaka is attracted to the white dot displayed on a monitor. In our experiment, the white dot controlled by the trained DQN attracted the medaka more than the random controller. It means that the white dots by DQN were somehow more attractive to the medaka. The characteristics of the white dot movements are staying at the edge and moving to a same lateral direction.



(a) Last-controller

(b) High-controller

(c) Low-controller

(d) Random

Figure 6: White dot movements in the lateral direction.

The advantage of staying at the edge is to avoid or decrease the situations where the medaka loses visual contacts. The space forms a torus. Therefore if the dot stays near the edge of the monitor, it can appear immediately at both sides. If the medaka is far away, it can cross the edge. The strategy can increase the chances of visual contacts. The movements to a same lateral direction might be related to more communication, not for simple visual contacts but for contents of the interactions. The further analysis of movements of the medaka and the dot can clarify which one leads the interaction, e.g. the medaka chases the dot first and the dot runs away, or the dot attracts the medaka and drags.

5. CONCLUSIONS

In this paper, we could induce the movements of a real life, medaka. The task is simple, but a machine could learn some basic elements of communication through the interaction with a real life. The deep neural network technology combined with Q-learning was used and it can receive the visual live images captured by a camera. Therefore, this method can be applied to any tasks if the reward function is defined properly. Our further work would be applying this method to more complex communication task to understand communication by animals.

6. REFERENCES

[1] Wataru Matsunaga and Eiji Watanabe. Visual motion with pink noise induces predation behaviour. *Scientific reports*, Vol. 2, , 2012.

[2] Tomohiro Nakayasu and Eiji Watanabe. Biological motion stimuli are attractive to medaka fish. *Animal cognition*, Vol. 17, No. 3, pp. 559–575, 2014.

[3] Jacqueline Nadel, Isabelle Carchon, Claude Kervella, Daniel Marcelli, and Denis Réserbat-Plantey. Expectancies for social contingency in 2-month-olds. *Developmental science*, Vol. 2, No. 2, pp. 164–173, 1999.

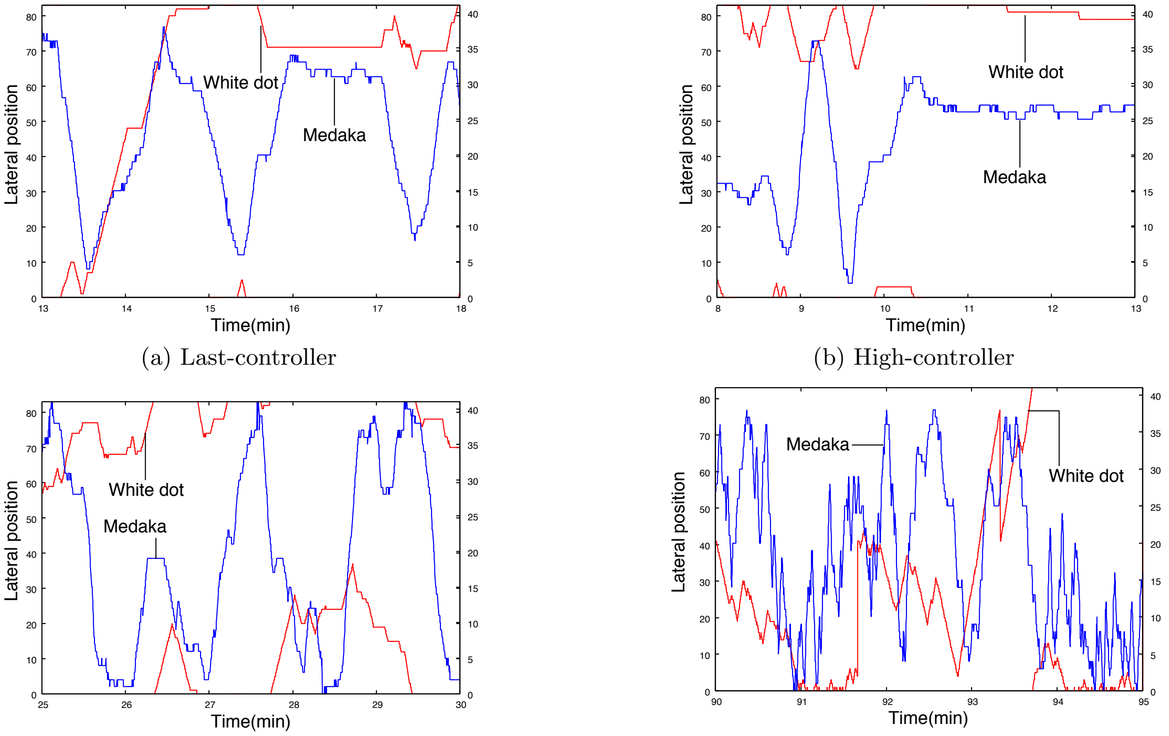
[4] C. Trevarthen L. Murray. Emotional regulations of interactions between two month-olds and their mothers. *Social perception in infants*, pp. 177–197, 1985.

[5] E.L.R. Ware. Interactive behaviour pigeons: visual display interactions as a model for visual communication. *PhD thesis, Queen's University*, 2011.

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, Vol. 518, No. 7540, pp. 529–533, 2015.

[8] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *Neural Networks, IEEE*



(a) Last-controller

(b) High-controller

(c) Low-controller

(d) during learning considered the facing direction

Figure 7: The white dot movement and the medaka movement in the lateral direction

Transactions on, Vol. 8, No. 1, pp. 98–113, 1997.

[9] *Caffe | Deep learning framework.* http://caffe.berkeleyvision.org/.