# Visuo-motor Learning for Behavior Generation of Humanoids

Masaaki Kikuchi[1], Masaki Ogino[1] and Minoru Asada[1,2]
[1]Dept. of Adaptive Machine Systems,
[2]HANDAI Frontier Research Center,
Graduate School of Engineering, Osaka University,
Osaka, Japan
Email: {kikuchi, ogino}@er.ams.eng.osaka-u.ac.jp,
asada@ams.eng.osaka-u.ac.jp

***Abstract*— This paper proposes a method of bahavior generation for humanoids in which a robot learns sensorimotor maps in each motion primitive as the forward and inverse relationship between optic flow in the robot's view and motion parameters. Robots use these maps to decide appropliate motion parameters that generate a desired flow given by a planner. Robots have encapsulated modules each of which consists of a planner and maps of primitives. Each module can accomplish a simple task, and using multiple modules as the situation demands, robots can accomplish complex task. Passing a ball (face-to-face pass) between two humanoids which have different camera lens and body parameters is realized as an example task.**

## I. Introduction

Recognizing the causal relationship between a motion and sensor change is inevitable for a robot to move in the world or recognize an object. Because a robot knows the results of its motion by the changes of its own sensor values, mapping between motor commands and corresponding sensor values (sensorimotor mapping) is crucial. Visual information is one of the most important sensors to know the outside world. The methods for mapping between visual information and motion commands are classified broadly into two categories. One is a top-down approach in which a designer constructs vision and motion system separately and gives mapping between them. In this approach, each system is calibrated to the global axis [4] [5]. Such a method does not consider changes of an environment or an embodiment of the robot. Therefore, if the environment or the embodiment is changed, the designer need to give another sensorimotor mapping or coordinate conversion model. Furthermore, it is difficult for designer to give a sensorimotor mapping which is complex causal relationship as interaction with an object in advance.

On the other hand, the other approach does not calibrate each system separately, but directly correlate visual information and motor commands. In this approach, optic flow has been used for mapping, because optic flow caused by the motion contains the information of causal relationship between the environmental change and the motion. The mapping between optic flow and motion commands are learned for obstacle avoidance planned by the learned forward model [2] or finding obstacles that show different flows from the environments using reinforcement learning [3] in wheel type robot. Also, it is used for object recognition by active touching in humanoid fixed in the ground [1]. In these studies, the robots have much fewer DoFs than humanoids, therefore it seems difficult to apply their methods to realize various kinds of humanoid behaviors. One solution is to decompose a humanoid action into basic motion primitives and to acquire the sensorimotor mapping in each motion primitive. This makes it possible for a robot to learn the sensorimotor mapping on line because in each motion primitive the relationship between the motion parameters and the sensor values is usually much simpler than in the general motion.

In this paper, as an example task, passing a ball between two humanoids (face-to-face pass) is realized based on the sensorimotor mappings of motion primitives. The task is decomposed into three basic motion modules: trapping a ball, approaching to a ball and kicking a ball to the opponent. Each motion module can be further decomposed into several motion primitives, each of which has motion parameters to control the motion trajectory. The sensorimotor mapping is learned as the forward and inverse relationship between these motion parameters and optic flow information in each motion. The acquired sensorimotor maps are used to select the appropriate motion primitive and its parameters to realize the desired pathway or destination in the robot's view given by the planner.

The rest of the paper is organized as follows. Section II introduces an overview of our proposed system. Section III provides the details of each module for ”passin a ball” task. Section IV shows the experimental result of the task to use integrated modules. Finally conclusion remarks are given.

## II. Task, Robot, and Environment

### A. Robot platforms

Fig. 1 shows biped robots used in the experiments, HOAP-1, HOAP-2, and their on-board views. HOAP-1 is 480 [mm] in height and about 6 [kg] in weight. It has a one-link torso, two four-link arms, and two six-link legs. The other robot, HOAP-2, is a successor of HOAP-1. It is 510 [mm] in height and about 7 [kg] in weight. It has two more joints in neck and one more joint at waist. Both robots have four force sensing registors (FSRs) in their foots to detect reaction force from the floor and a CCD camera with a fish-eye lens or semi-fish-eye lens.

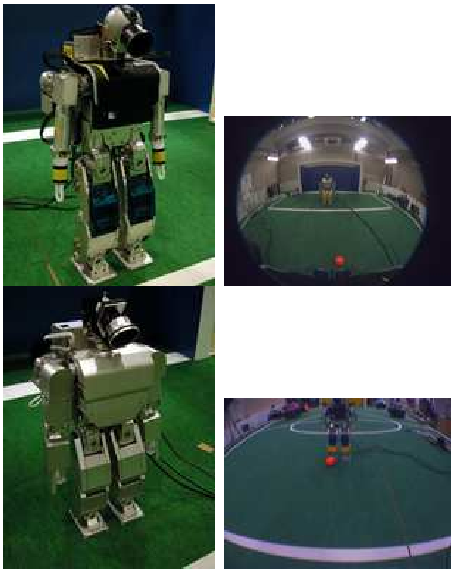


Fig. 1. HOAP-1 with fish-eye lens and HOAP-2 with semi-fish-eye lens

These robots detect objects in the environments by colors. In this experiment, a ball is colored orange, and the knees of the opponent robot are colored yellow. The centers of these colored regions in the images are recorded as the detected position.

### B. Visuo-motor learning

Let the motion flow vector $\Delta \boldsymbol{r}$ at the position $\boldsymbol{r}$ in the robot's view when a robot takes a motion, $a$. The relationships between them can be written,

$$\Delta \boldsymbol{r} = f(\boldsymbol{r}, a), \tag{1}$$
$$a = g(\boldsymbol{r}, \Delta \boldsymbol{r}). \tag{2}$$

The latter is useful to determine the motion parameters after planning the motion path way in the image. However, it is difficult to determine one motion to realize a certain motion flow because different motion primitives can produce the same image flow by adjusting motion parameters. So, we separate the description of the relationship between the motion parameters in each primitive and the image flow as follows.

$$\boldsymbol{a}^i = (p_1^i, \ldots, p_n^i)^T = g_p^i(\boldsymbol{r}, \Delta \boldsymbol{r}) \tag{3}$$
$$\Delta \boldsymbol{r} = f^i(\boldsymbol{r}, \boldsymbol{a}^i), \tag{4}$$

$\boldsymbol{a}^i$ are the motion parameter vector of the $i$-th motion primitive. We use neural networks to learn these relationships.

### C. Task and Assumptions

"Face-to-face pass" can be decomposed into following three modules:

- approaching to a ball to kick,
- kicking a ball to the opponent, and
- trapping a ball which is coming to the player

All these basic modules need the appropriate relationship between motion parameters and the environment changes. For example, to trap a ball appropriately, the robots must estimate the arrival time and position of the coming ball. To approach to a kicking position, the robot should know the causal relationship between the walking parameters and the positional change of the objects in its image. Further, to kick a ball to the opponent, the robot must know the causal relationship between the kicking parameters and the direction the kicked ball will go.

Moreover, basic modules to realize these behaviors should be activated at the appropriate situations. Here, the designer determines these situations to switch the behaviors, and we focus on the module learning based on optic flow information. Fig. 2 shows an overview of our proposed system.

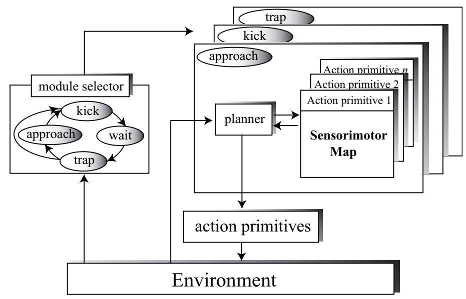



Fig. 2. A system overview

## III. Module Learning Based on Optic Flow Information

### A. Ball Approaching

Approaching to a ball is the most difficult task among the three modules because this task involves several motion primitives each of which has parameters to be determined. These motions yields various types of image flows depending on the values of the parameters which change continuously. We make use of environmental image flow pattern during various motions to approach to the ball.

We separate the description of the relationship between the motion and the image flow into the relationship between the motion primitive and the image flow, and the relationship between the motion parameters in each primitive and the image flow (Fig. 3), as follows.

$$m_i = g_m(\boldsymbol{r}, \Delta \boldsymbol{r}), \tag{5}$$
$$\boldsymbol{a}^i = (p_1^i, p_2^i)^T = g_p^i(\boldsymbol{r}, \Delta \boldsymbol{r}) \tag{6}$$
$$\Delta \boldsymbol{r} = f^i(\boldsymbol{r}, \boldsymbol{a}^i), \tag{7}$$

where $m_i$ is the index of the $i$-th motion primitive and $\boldsymbol{a}^i = (p_1^i, p_2^i)^T$ are the motion parameter vector of the $i$-th motion primitive. In this study, the motion primitives related to this module consists of 6 primitives; *forward walk* (left and right), *curve walk* (left and right), and *side step* (left and right). Each of the primitives has two parameters which have real values, as shown in Fig. 4.

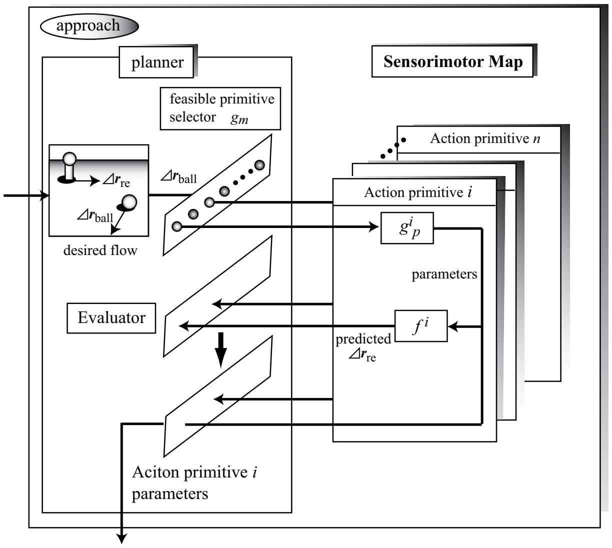



Fig. 3. An overview of the approaching module

Given the desired motion pathway in the robot's view, we can select appropriate primitive by $g_m$, and determine the motion parameters of the selected motion primitive by $g^i_p$ based on the learned relationships among the primitives, their parameters, and flows. If the desired image flow yields several motion primitives, the preferred motion primitive is determined by value function.

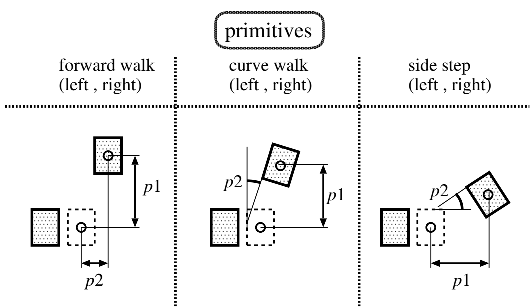



Fig. 4. Motion primitives and parameters for approaching

Images are recorded every step and the image flow is calculated by block matching between the current image and the previous one. The templates for calculating flows are 24 blocks in one image as shown in Fig. 5.

$g_m$: All of the data sets of the flow and its positional vector in the image, $(\boldsymbol{r}, \Delta\boldsymbol{r})$, are classified by the self organizing map (SOM), which consists of 225 (15×15) representational vectors. And after organizing, the indices of motion primitives are attributed to each representational vector. Fig. 6 shows the classified image vector (the figure at the left side) and the distribution of each primitive in SOM. This SOM outputs the index of appropriate motion primitive so that the desired flow vector in the image is realized.

$f^i, g^i_p$: The forward and inverse functions that correlates the relationship between the motion parameters in each

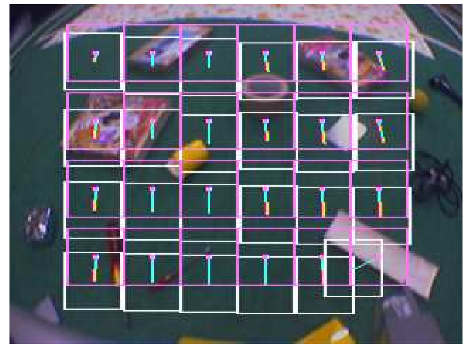


Fig. 5. An example of an optic flow in the robot's view

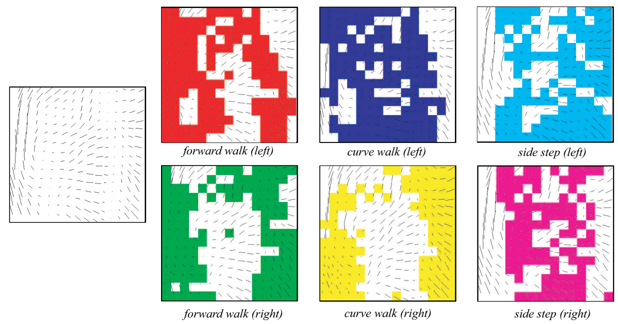



Fig. 6. Distribution of motion primitives on the SOM of optic flows

primitive and the image flow, $f^i, g^i_p$, are realized by a simple neural network. The neural network in each primitive is trained so that it outputs the motion parameters when the flow vector and the positional vector in the image are input.

*plannning and evaluation function:* In this study, the desired optic flow in the robot's view for the ball and the receiver, $\boldsymbol{s}_{ball}, \boldsymbol{s}_{re}$, are determined as a vector from the current position of a ball to the desired position (kicking position) in the robot's view, and as the horizontal vector from the current position to the vertical center line, respectively. The next desired optic flow of a ball to be realized, $\tilde{\boldsymbol{s}}_{ball}$, is calculated based on these desired optic flows,

$$n_{step} = \|\boldsymbol{s}_{ball}\|/\Delta r_{max}, \quad (8)$$

$$\tilde{\boldsymbol{s}}_{ball} = \boldsymbol{s}_{ball}/n_{step}, \quad (9)$$

where $\Delta r_{max}$ is the maximum length of the experienced optic flow. This reference vector is input to the primitive selector, $g_m$, and the candidate primitives which can output the reference vector are activated. The motion parameters of the selected primitive are determined by the function $g^i_p$,

$$\boldsymbol{a}^i = g^i_p(\boldsymbol{r}_{ball}, \tilde{\boldsymbol{s}}_{ball}), \quad (10)$$

where $\boldsymbol{r}_{ball}$ is the current ball position in the robot's view. When the primitive selector outputs several candidates of primitives, the evaluation function depending on the task, $V(m_i)$, determines the preferred primitive. In this study, our robots have to not only approach to a ball but also take an appropriate position to kick a ball to the other. For that, we set the evaluation function as follows,

$$\begin{aligned} V(m_i) &= \|\tilde{\mathbf{s}}_{ball} - f^i(\boldsymbol{r}_{ball}, \boldsymbol{a}^i)\| \\ &\quad + k\|\mathbf{s}_{re} - n_{step} f^i(\boldsymbol{r}_{re}, \boldsymbol{a}^i)\|, \qquad (11) \\ P &= \underset{i \in primitives}{\arg\min} \; V(m_i) \end{aligned}$$

where $k$ is the constant value, $\boldsymbol{r}_{re}$ is the current position of the receiver in the robot's view, and $P$ is the selected primitive.

Fig. 7 shows experimental results of approaching to a ball. A robot successfully approach to a ball so that the hypothetical opponent (a poll) comes in front of it.

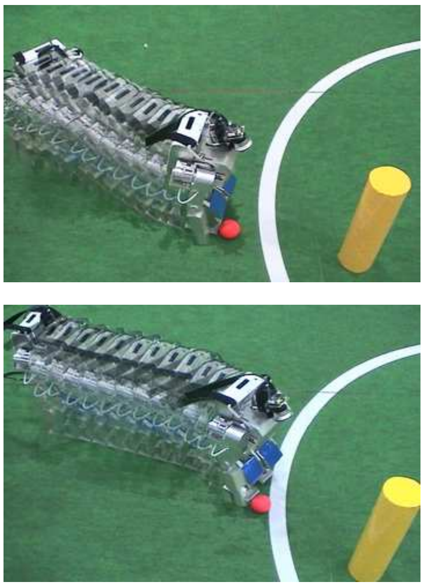


Fig. 7. Experimental results of approaching to a ball

### B. Ball Kicking to the Opponent

It is necessary for our robots to kick a ball to the receiver very precisely because they cannot sidestep quickly. We correlate the parameter of kicking motion with the trace of the kicked ball in the robot's view so that they can kick to each other precisely. Fig. 8 shows a proposed controller for kicking.

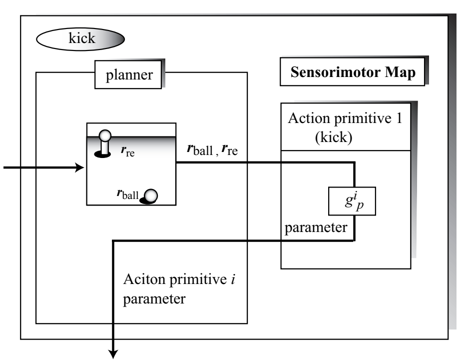
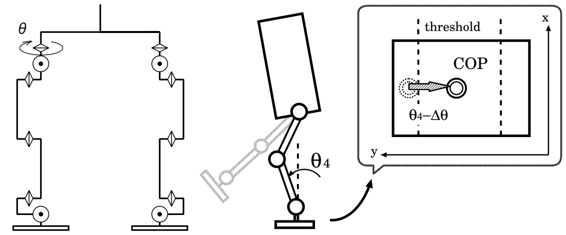


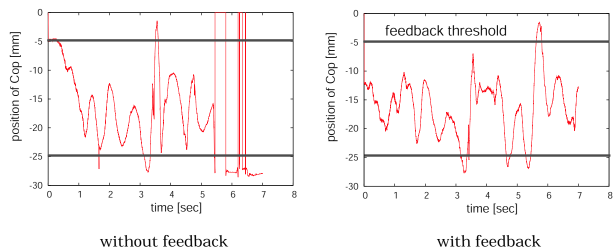

Fig. 8. The system for kicking module

The kicking parameter is the hip joint angle shown in Fig. 11(a). The quick motion like kicking changes its dynamics depending on its motion parameter. The sensor feedback from the floor reaction force sensors is used for stabilizing the kicking motion. The displacement of the position of the center of pressure (CoP) in the support leg is used as feedback to the angle of the ankle joint of the support leg (see Fig. 11(b)). ,Fig. 11(c) shows the effectiveness of the stabilization of the kicking motion.




(a) Kick parameter

(b) An overview of stabilization of kick motion




(c) The trajectories of CoP of the support leg during kicking motion

Fig. 9. The parameter and the stabilization of kicking

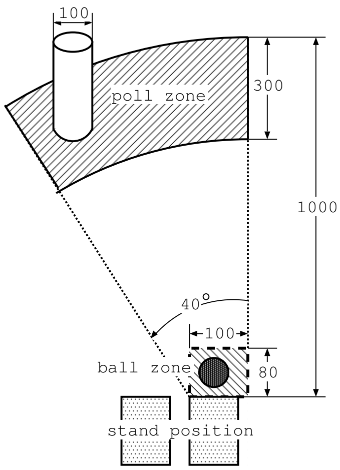
The initial ball position and the parameter of the kicking motion affects sensitively the ball trace in the robot's view. To describe the relationship among them, we use a neural network, which is trained in the environment where the poll (10 [cm]) is put about 1 [m] in front of the robot (Fig. 13(a)). The trace of the ball (the effects of the self motion is subtracted) is recorded every 100 [msec], and the weights in the neural network are updated every one trial. Fig. 13(b) shows the time course of error distance between target poll position and kicked ball in the robot's view. It shows that the error is reduced rapidly within 20 [pixel], which is the same size of the width of the target poll. Fig. 11 shows the kicking performance of the robot.

### C. Ball Trapping

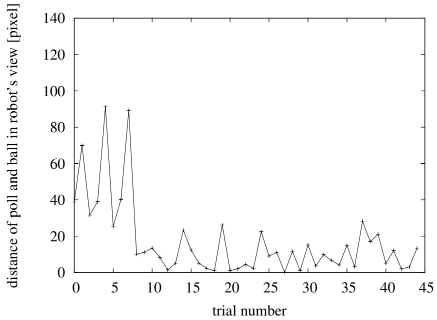
Fig. 12 shows an overview of trapping module. Robots learn the relationship between the position of the foot in robot's view and the trap parameter which affects the position of the foot, to acquire the skill to trap a coming ball.

Fig. 14 shows the trapping motion by HOAP-2 acquired by the method described below. In order to realized such a

Fig. 11. An experimental result of kicking a ball to the poll




(a) The environmental setting




(b) The error of learning kicking

Fig. 10. The environmental setting and the learning curve for kicking

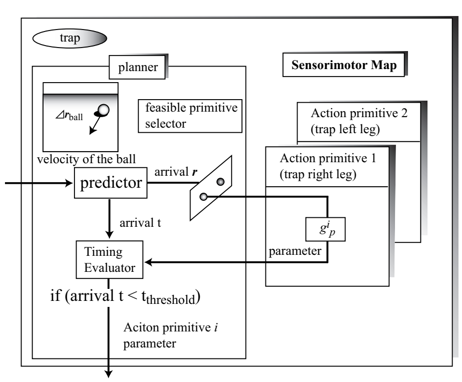



Fig. 12. An overview of trapping module

motion, the robot has to predict the position and the arrival time of a ball from its optic flow captured in the robot view. For that purpose, we use a neural network which learns the causal relationship between the position and optic flow of the ball in visual image of a robot and the arrival position and time of the coming ball. This neural network is trained by the data in which a ball is thrown to a robot from the various positions. Fig. 13 shows several prediction results of the neural network after learning. $\Delta x$ [pixel] and $\Delta t$ [sec] indicates the errors of the arrival position and the time predicted at each point (every 0.3[sec]) in the robot's view. $T$ means a duration of the ball rolling. Based on this neural network, the robots can activate the trapping motion primitive with the appropriate leg (right or left) at the appropriate timing (Fig. 14).

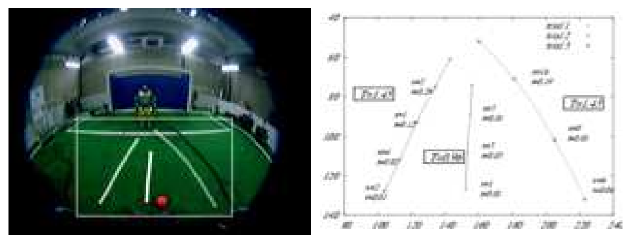


Fig. 13. The prediction of the position and time of a coming ball

## IV. Integration of the Modules for Face-to-face Pass

To realize passing a ball between two humanoids, the basic modules described in the previous chapter are integrated by the simple rule as shown in Fig. 15.

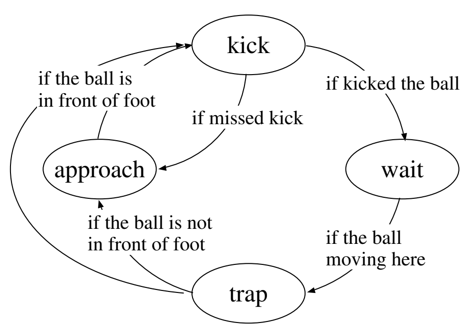



Fig. 15. The rule for integrating motion modules

Fig. 16 shows the experimental result. Two humanoids with different body and different camera lens realize the appropriate motions for passing a ball to each other based

Fig. 14. An experimantal result of a trapping module

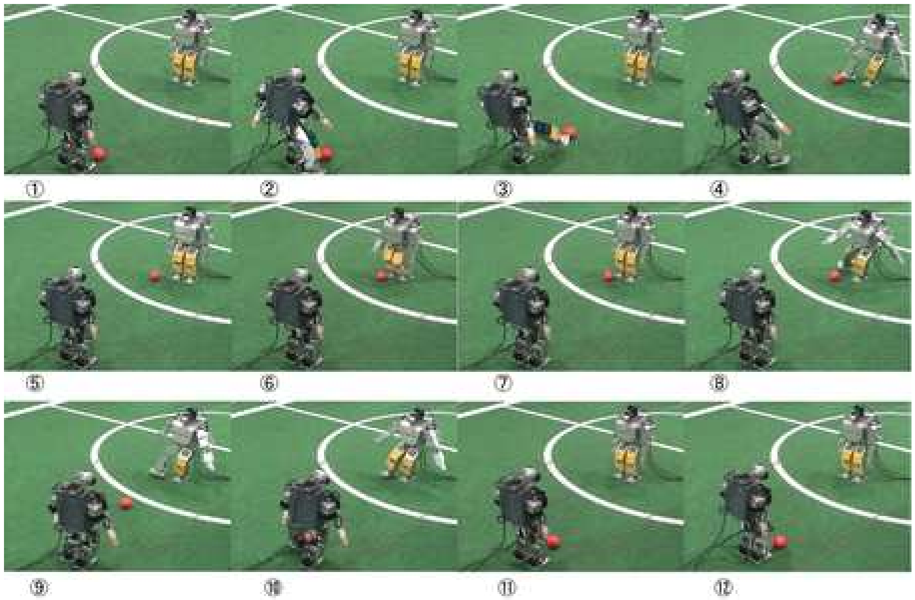


Fig. 16. An experimental result of passes between two humanoids

on their own sensorimotor mapping. The passing lasts more than 3 times.

## V. Conclusions

In this paper, the robots learn the sensorimotor mapping between optic flow information and motion parameters. Acquiring basic modules for passing a ball is achieved using the sensorimotor mapping. In each module, optic flow information is correlated with the motion parameters. Through this correlation, a humanoid robot can obtain the sensorimotor mapping to realize the desired modules. The experimental results show that a simple neural network quickly learns and models well the relationship between optic flow information and motion parameters of each motion primitive. However, there remain the harder problems we skip in this paper. First is module decomposition problem, that is how to determine what are the basic modules for the given task. Second is planning, that is how to organize each motion primitive to achieve the given task. In this paper, we assume module decomposition and planning are given in advance. Combining the learning in each module level with that in higher level is the next problem for us.

## References


[1] P. Fitzpatrick. First Contact: an Active Vision Approach to Segmentation, In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2161–2166, 2003.

[2] K. F. MacDorman, K. Tatani, Y. Miyazaki, M. Koeda and Y. Nakamura. Protosymbol emergence based on embodiment: Robot experiments, In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1968–1974, 2001.

[3] T. Nakamura and M. Asada. Motion Sketch: Acquisition of Visual Motion Guided Behaviors. In *Proc. of Int. Joint Conf. on Artificial Intelligence*, pp. 126–132, 1995.

[4] K. Nishiwaki, S. Kagami, J. Kuffner, M. Inaba and H. Inoue, Walking Control System of a Humanoid for Tracking a Moving Object with Estimate of the Target Motion, In *Proc. of 3rd IEEE Int. Conf. on Humanoid Robots*, 2003.

[5] J. F. Seara, K. Strobl, and G. Schmidt. Information Management for Gaze Control in Vision Guided Biped Walking, In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 31–36, Lausanne, Switzerland, October 2002.