Interactive Design for Behaviors of Robot Partners through Multi-modal Communication

Naoyuki Kubota, Koh Nishimura
Tokyo Metropolitan University
6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan
Email: kubota@tmu.ac.jp, nishimura.koh@sd.tmu.ac.jp

Beom Hee Lee
Seoul National University
San 56-1, Shillim-Dong, GwanAk-Ku, Seoul 151-742, Korea
Email: bhee@snu.ac.kr

Abstract — This paper deals interactive learning for behaviors of robot partners through multi-modal communication based on touch interface, accelerometers, and others. A robot partner is controlled based on multi-objective behavior coordination using collision avoidance, target tracing, wall following, and formation with other robots. A specific behavior can be trained through the gesture navigation using touch interface. We propose an interactive behavior learning method based on human gesture navigation. A robot extracts a trajectory pattern by spiking neural network, and performs unsupervised clustering by self-organizing map. Finally, we show several experimental results of the proposed method, and discuss the effectiveness of the proposed method.

I. INTRODUCTION

Recently, various types of robots such as amusement robots, robot pets, and robot partners have been developed so far [1-8]. A robot partner will be a concierge at the information service in daily life [6]. We will have three different types of robot partners from the interactive point of view (Fig.1). One is a physical robot partner. We can interact with the physical robot partner by using multi-modal communication like a human. The next one is a pocket robot partner without mobile mechanisms. This can be easily brought everywhere and interacted with the robot partner by touch and physical interface. The last one is a virtual robot partner. The virtual robot partner is in the virtual space within the computer, but we can interact with it through the virtual person or robot in the virtual space. The interaction style of these three robots is different, but they share the same personal database and interaction logs, and can interact with the person based on the same interaction rules independent from the style of interfaces. However, it is very difficult to store and share all of huge data in real time. Furthermore, some features should be extracted from the gathered data to obtain the required information. Therefore, the environment surrounding people and robots should have a structured platform for gathering, storing, transforming, and providing information. Such an environment is called informationally structured space [9,10] (Fig.2). We discuss informationally structured space from three different points of view: (1) the measurement for personalization, (2) the user interface, and (3) the structuralization of environmental model. The user-friendly interface based on multi-modal natural communication is important to reduce the load of users. Touch interface and gesture interface based on accelerometer are useful to input meta-level commands to a pocket robot partner. Furthermore, the structuralization of informationally structured space realizes the quick update and access of valuable and useful information for users. The integration of information technology, network technology, and robot technology [11-14] is a key concept to realize the informationally structured space. Furthermore, if the robot can share the environmental information with people, the interaction and communication with people might become very smooth and natural.

Human interaction based on gestures is very important to realize the natural communication [15,16]. Basically, imitative learning is composed of model observation and model reproduction. Furthermore, model learning is required to memorize and generalize motion patterns as gestures. In addition, the model clustering is required to distinguish a specific gesture from others, and model selection as a result of the human interaction is also performed. In this way, the imitative learning requires various learning capabilities of model observation, model clustering, model selection, model reproduction, and model learning simultaneously. We proposed a method for imitative learning of partner robots based on visual perception [5].

In this paper, we discuss how to make a robot partner learn various behaviors through interaction with a person. We use iPad as a pocket robot partner to reduce the load to a person in the behavior learning. A person can show the trajectory of behavior pattern by the gesture navigation using touch interface. The model observation is simply done by extracting human touch gesture navigation, and a gesture navigation pattern is extracted by a spiking neural network. Next, a self-organizing map is used for clustering human gesture navigation patterns. In this way, the robot can use the gesture navigation patterns for the interaction with a person. Finally, we show several experimental results and discuss the effectiveness of the proposed method.

This paper is organized as follows. Section II explains the detail of robot partners, and Section III explains the gesture navigation by tough interface, and a learning method of behaviors through gesture navigation. Section IV shows a several simulation results. Section IV summarizes the paper and discusses the future vision on the robot partners.
II. ROBOT PARTNERS

A. Interaction with Physical Robot Partners

We developed two types of partner robots; a mobile PC called MOBiMac (Fig.3) and a human-like robot called Hubot in order to realize the social communication with a human [10]. Each robot has two CPUs and many sensors such as CCD camera, microphone, and ultrasonic sensors. Therefore, the robots can conduct image processing, voice recognition, target tracing, collision avoidance, map building, imitative learning, and others. The basic behavior modes of these robots are human tracking, human communication, behavioral learning, and behavioral interaction. The communication with a person is performed by utterance as the result of voice recognition and gestures as the result of human motion recognition. The robot integrates the above behaviors according to the environmental conditions based on the multi-objective behavior coordination [17]. The multi-objective behavior coordination integrates outputs of several behaviors according to the time-series of perceptual information.

The basic capabilities common to physical, pocket, and virtual robot partners are human recognition, object recognition, mining of personal data, and learning of interaction patterns based on image processing and voice recognition. The data used for these capabilities are stored in the informationally structured space, and a robot partner can access and update the data through the wireless network.

B. Interaction with Pocket Robot Partners

We used apple Newton message pad as a pocket robot partner [4], and recently we have been capable to use sophisticated and advanced devices such as iPhone, iPod touch, and iPad as pocket robot partners [9]. Basically, robots and people are interacting each other based on the interaction rules and control rules in the virtual environment (Fig.4 (a)). The behavior modes are wandering behavior, circular formation behavior, gathering behavior, dancing behavior, people tracking behavior, and others. A person can control the behavior mode of the robots by touching buttons on the display or shuffling the device itself (Fig.4 (b)). Furthermore, the person directly accesses robots, people, and environmental information through the touch information and accelerometer information. For example, when the person touches a robot on the display, the robot stops at the point and starts the interaction with the person. Figure 4 (c) shows a navigation of the robot where the white line indicates the target trajectory that the person draws by the touch interface. The robot traces the target trajectory soon after the person touches off the display.

Basically, the teaching of behaviors to robots in virtual environments is easier than that in real environments, because the real environment includes much background noise. Therefore, the person should use the touch interface to teach behaviors and performances like pet dogs. Furthermore, the learned behaviors can be transferred to the interaction of physical robot partners.

Pocket robot partners need the intelligent capabilities of the extraction of human intention through the touch interface and accelerometers, and flexible display of environmental information according to the touch information and accelerometer information. For example, the person quickly shuffles the device in order to inform the robot of dangerous situation, and inclines the device in order to update the view angle of the environment displayed on the display. In this way, pocket robot partners can play the important role of human interface connecting between the real environment and virtual environment seamless.
A. Behavior Control of Robot Partner

Various intelligent methods such as production rules, Bayesian networks, neural networks, fuzzy inference systems, and classifier systems have been proposed for mobile robots, robot manipulators, and robot partners [18-28]. We have applied fuzzy inference systems to represent behavior rules of mobile robots, because the behavioral rules can be designed easily and intuitively by human linguistic representations. A behavior of the robot can be represented using fuzzy rules based on simplified fuzzy inference [26]. In general, a fuzzy if-then rule is described as follows,

**If** $x_1$ is $A_{i,1}$ and ... and $x_M$ is $A_{i,M}$  
**Then** $y_1$ is $w_{1,i}$ and ... and $y_N$ is $w_{N,i}$

where $A_{i,j}$ and $w_{j,k}$ are the Gaussian membership function for the $j$th input and the singleton for the $k$th output of the $i$th rule; $M$ and $N$ are the numbers of inputs and outputs, respectively. Fuzzy inference is performed by,

$$
\mu_{A_i}(x) = \exp\left(-\frac{(x-a_{i,j})^2}{b_{i,j}^2}\right)
$$

$$
\mu_i = \prod_{j=1}^{M} \mu_{A_i}(x_j)
$$

$$
y_k = \frac{\sum_{i=1}^{K} \mu_{A_i} w_{j,k}}{\sum_{i=1}^{K} \mu_{i}}
$$

where $a_{i,j}$ and $b_{i,j}$ are the central value and the width of the membership function $A_{i,j}$; $R$ is the number of rules. Outputs of the robot are output levels of the left and right motors ($N=2$).

Fuzzy controller is used for collision avoidance and target tracing behaviors. The inputs to the fuzzy controller for collision avoidance are the measured distance to the obstacle by ultrasonic sensors ($M_i=8$). The inputs to the fuzzy controller for target tracing are the estimated distance to the target point and the relative angle to the target point from the moving direction ($M_i=2$).

In general, a mobile robot has a set of behaviors for achieving various objectives, and must integrate these behaviors according to the environmental conditions. Therefore, we proposed the method for multi-objective behavior coordination [17]. The multi-objective behavior coordination can integrate outputs of several behaviors according to the time-series of perceptual information, while the original subsumption architecture selects one behavior. This multi-objective behavior coordination is composed of a sensory network, behavior coordinator, and behavior weight updater. The sensory network extracts perceptual information based on sensing data and updates the parameters of sensors recursively according to the perceptual information. A behavior weight is assigned to each behavior. Based on eq.(3), the output is calculated by

$$
y_k = \frac{\sum_{j=1}^{K} \mu_{A_i} w_{j,k} y_j}{\sum_{j=1}^{K} \mu_{i}}
$$

where $K$ is the number of behaviors; $w_{j,k}$ is a behavior weight of the $j$th behavior over the discrete time step $t$. By updating the behavior weights, the robot can take a multi-objective behavior according to the time series of perceptual information. The update amount of each behavior is calculated as follows,

$$
\begin{bmatrix}
\Delta w_{g_1}
\vdots
\Delta w_{g_L}
\end{bmatrix}
= \begin{bmatrix}
dw_{1,1} & dw_{1,2} & \cdots & dw_{1,L} \\
\vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
dw_{L,1} & dw_{L,2} & \cdots & dw_{L,L}
\end{bmatrix}
\begin{bmatrix}
\mu_{i}
\vdots
\mu_{i}
\vdots
\vdots
\vdots
\mu_{i}
\end{bmatrix}
$$

where $\mu_{i}$ is the parameter on the perceptual information; $L$ is the number of perceptual inputs. This method can be considered as a mixture of experts if the behavior coordinator is considered as a gating network.

B. Gesture Navigation

Remote control and tele-operation is one of the most important tasks for robots [11,27,28]. We explain the gesture navigation using touch interface. The trajectory drawn on the touch panel is updated by the linear smoothing. Figure 5 shows examples of the original trajectory and updated trajectory. This result shows that the trajectory is interpolated smoothly.

Figures 6 and 7 show examples of the gesture navigation on iPad where the robot is depicted at a point in the display. After a person draws a trajectory to be traced, the robot partner traces the trajectory step by step. The robot is depicted in the center of screen. In this gesture navigation, the robot
partner relatively traces the trajectory from the current position without moving to the starting point of the trajectory. Furthermore, the robot avoids the collision with other robots and obstacles based on the multi-objective behavior coordination.

**Fig.5.** Trajectories to be traced; the upper is the original trajectory drawn by the user, and the lower is the updated trajectory

**Fig.6.** An experimental result of the navigation of virtual robot partner based on the touch interface (Case 1)

**Fig.7.** An experimental result of the navigation of virtual robot partner based on the touch interface (Case 2)

**C. Trajectory Extraction Based on Spiking Neural Network**

Various types of artificial neural networks have been proposed to realize clustering, classification, nonlinear mapping, and control [19-23]. Basically, artificial neural networks are classified into pulse-coded neural networks and rate-coded neural networks from the viewpoint of abstraction level [21]. A pulse-coded neural network approximates the dynamics introduced the ignition phenomenon of a neuron, and the propagation mechanism of the pulse between neurons. Hodgkin-Huxley model is one of the classic neuronal spiking models with four differential equations. An integrate-and-fire model with a first-order linear differential equation is known as a neuron model of a higher abstraction level. A spike response model is slightly more general than the integrate-and-fire model, because the spike response model can choose kernels arbitrarily. On the other hand, rate-coded neural networks neglect the pulse structure, and therefore are considered as neuronal models of the higher level of abstraction. McCulloch-Pitts and Perceptron are well known as famous rate coding models [19]. One important feature of pulse-coded neural networks (SNNs) have been applied for memorizing spatial and temporal context. Therefore, we apply a SNN for memorizing several motion patterns of a human hand, because the human hand motion has a specific dynamics. In this paper, we use a modified simple spike response model to reduce the computational cost. First of all, the internal state of the neuron according to PSP with the weight connection. The PSP is calculated as follows;

$$h_i(t) = \tanh\left(\gamma \cdot \sum_{j=1}^{N} w_{ij} \cdot h_{j}^{\text{ref}}(t)\right) + h_i^{\text{syn}}(t) + h_i^{\text{ref}}(t)$$

(6)

Here hyperbolic tangent is used to avoid the bursting of neuronal fires, $h_i^{\text{ref}}(t)$ is the input to the $i$th neuron from the external environment, and $h_i^{\text{syn}}(t)$ including the output pulses from other neurons is calculated by,

$$h_i^{\text{syn}}(t) = \gamma^{\text{syn}} \cdot h_i(t-1) + \sum_{j=1}^{N} w_{ij} \cdot h_{j}^{\text{ref}}(t)$$

(7)

Furthermore, $h_i^{\text{ref}}(t)$ indicates the refractoriness factor of the neuron; $w_{ij}$ is the parameter of a weight coefficient from the $j$th to $i$th neuron; $h_i^{\text{ref}}(t)$ is the presynaptic action potential (PSP) approximately transmitted from the $j$th neuron at the discrete time $t$; $N$ is the number of neurons; $\gamma^{\text{syn}}$ is a temporal discount rate. When the internal state of the $i$th neuron is larger than the predefined threshold, a pulse is outputted as follows;

$$p_i(t) = \begin{cases} 1 & \text{if } h_i(t) \geq q_i \\ 0 & \text{otherwise} \end{cases}$$

(8)

where $q_i$ is a threshold for firing. Furthermore, $R$ is subtracted from the refractoriness value in the following,

$$h_i^{\text{ref}}(t) = \begin{cases} \gamma^{\text{ref}} \cdot h_i^{\text{ref}}(t-1) - R & \text{if } p_i(t-1) = 1 \\ \gamma^{\text{ref}} \cdot h_i^{\text{ref}}(t-1) & \text{otherwise} \end{cases}$$

(9)

where $\gamma^{\text{ref}}$ is a discount rate and $R>0$.

The presynaptic spike output is transmitted to the connected neuron according to PSP with the weight connection. The PSP is calculated as follows;

$$h_i^{\text{PSP}}(t) = \begin{cases} 1 & \text{if } p_i(t) = 1 \\ \gamma^{\text{PSP}} \cdot h_i^{\text{PSP}}(t-1) & \text{otherwise} \end{cases}$$

(10)

where $\gamma^{\text{PSP}}$ is the discount rate (0 < $\gamma^{\text{PSP}}$ < 1.0). Therefore, the postsynaptic action potential is excitatory if the weight parameter, $w_{ij}$ is positive. Here spiking neurons are arranged on the planar grid of an
image (Fig.8) and \( N=25 \) in this case. By the value of a position on the trajectory, the input to the \( i \)th neuron is calculated by the radial basis function as follows;

\[
h_{\text{out}}^{SOM}(t) = \exp \left( -\frac{\|G(t)-c_i\|^2}{2\sigma^2} \right)
\]

(11)

where \( c_{i}\in(c_{i,1}, c_{i,2}) \) is the position of the \( i \)th neuron; \( \sigma \) is a standard deviation. The trajectory of human gesture navigation is obtained by the sequence of \( G(t) = (G_1(t), G_2(t), \ldots, G_t(t)) \), \( t=1, 2, \ldots, T \). If the condition \( h_{\text{out}}^{SOM}(t-1) < h_{\text{out}}^{SOM}(t) \) is satisfied, the weight parameter is trained based on the temporal Hebbian learning rule as follows,

\[
w_{ij} \leftarrow \tanh \left( \gamma^{\text{temp}} \cdot w_{ij} + \xi^{\text{temp}} \cdot h_{\text{out}}^{SOM}(t-1) \cdot h_{\text{out}}^{SOM}(t) \right)
\]

(12)

where \( \gamma^{\text{temp}} \) is a discount rate and \( \xi^{\text{temp}} \) is a learning rate. Because the adjacent neurons along the trajectory of the human hand position become easily fired by the Hebbian learning, the SNN can memorize the temporally firing patterns corresponding to various hand motions.

### D. Clustering of Trajectories by Self-organizing Map

This subsection proposes a method for clustering trajectories obtained by the human gesture navigation. Cluster analysis is used for grouping or segmenting observations into subsets or clusters based on similarity [22,23]. Self-organizing map (SOM), \( K \)-means algorithm, and Gaussian mixture model are often applied as clustering algorithms. SOM can be used as an incremental learning method, while \( K \)-means algorithm and Gaussian mixture model use all observed data at the learning phase (batch learning). In this paper, we apply SOM for clustering spatiotemporal patterns of PSP obtained from the SNN.

SOM is often applied for extracting a relationship among observed data, since SOM can learn the hidden topological structure from the data. Figure 8 shows a conceptual figure of SOM. The inputs to SOM is given as the weighted sum of pulse outputs from neurons,

\[
v = (v_1, v_2, \ldots, v_N)
\]

(13)

where \( v_i = h_{\text{out}}^{SOM}(T) \); \( T \) is the final time step of input data. When the \( i \)th reference vector is represented by \( r_i \), the Euclidean distance between an input vector and a reference vector is defined as

\[
d_i = \|v - r_i\|
\]

(14)

where \( r_i = (r_{i,1}, r_{i,2}, \ldots, r_{i,N}) \) and the number of reference vectors (output units) is \( M \). Next, the \( k \)th output unit minimizing the distance \( d_i \) is selected by

\[
k = \arg \min_i d_i
\]

(15)

Furthermore, the reference vector of the \( i \)th output unit is trained by

\[
r_i \leftarrow r_i + \xi^{SOM} \cdot \zeta^{SOM} \cdot (v - r_i)
\]

(16)

where \( \xi^{SOM} \) is a learning rate \((0 < \xi^{SOM} < 1.0)\); \( \zeta^{SOM} \) is a neighborhood function \((0 < \zeta^{SOM} < 1.0)\). Accordingly, the selected output unit is the nearest pattern among the previously learned human navigation trajectory patterns.

![Fig.9 Extraction of topological structure in SOM](Image)

### IV. EXPERIMENTAL RESULTS

This section shows several experimental results of gesture navigation and its corresponding behavior learning. The number of spiking neurons is 25 \((N=10)\), and the number of reference vectors is 10 \((M=10)\). Therefore, the proposed method can learn 10 different types of trajectories at most. The time sequence used for the extraction of a trajectory is 10 \((T=10)\), and this is corresponding to the number of points interpolating a trajectory. Figure 10 shows the comparison results before and after learning behaviors. Each figure of Fig.10 shows experimental results of gesture navigation (upper), the PSP of SNN after the input of a trajectory (lower left), and the selected reference vector in SOM using the PSP (lower right) where the depth of red color indicates the value of PSP. Figures 10 (a) and (b) show an example of navigation before and after learning, respectively (Case 3). Before learning, the reference vector of SOM is not trained (Fig.10 (a) lower right), but SOM learns the target trajectory after several times (Fig.10 (b) lower right). Next, the person tries to show a different trajectory to a robot. Figures 10 (c) and (d) show an example of navigation before and after learning, respectively (Case 4). The PSP based on spike outputs does not cover the trajectory given to the robot partner at the first trial (Fig.10 (c) lower left), but after learning, the PSP successfully covers the trajectory based on the trained weight connections (Fig.10 (d) lower left). After several trials, the person tries to show the trajectory given to the robot partner at first again. Figures 10 (e) and (f) show an example of navigation before and after learning, respectively (Case 5). As a result, the trained SOM selects the nearest reference vector (Fig.10 (a) lower right), but SOM already forgets the first learned trajectory owing to many trials. However, SOM learns the target trajectory again after several times (Fig.10 (f) lower right). We intend to develop the autonomous update method of the parameters used for the learning of SNN and SOM as a future work.

We proposed an imitative learning method for human-like robot partners based on camera images [14,15], but the performance of image processing is strongly dependent on the lighting conditions of the surrounding environment. However, the tough interface can easily give a robot partner a target...
trajectory. In this way, we can use various teaching methods according to the quality of tasks.

![Fig.10. Experimental results of before and after learning](image)

**V. SUMMARY**

This paper discussed the human interface for interactive behavior learning for robot partners. The proposed method is composed of a spiking neural network for spatiotemporal hand motion extraction, and self-organizing map for unsupervised classification of a trajectory given by human hand motion using the tough interface. The experimental results show the effectiveness of the learning method for interactive behavior design, but we should use various types of sensory information in addition to tough interface.

Furthermore, we intend to discuss a total architecture of the behavior learning through multi-modal communication.

REFERENCES