

# Construction of Robotic Body Schema by Extracting Temporal Information from Sensory Inputs

Komei Sugiura<sup>1,2</sup>, Daisuke Matsubara<sup>1</sup>, and Osamu Katai<sup>1</sup>

<sup>1</sup>Graduate School of Informatics, Kyoto University, Kyoto, Japan

(Tel : +81-75-753-3592; E-mail: {sugiura, matsubara, katai}@sys.i.kyoto-u.ac.jp)

<sup>2</sup>ATR Spoken Language Communication Research Laboratories, Kyoto, Japan

**Abstract:** This paper proposes a method that incrementally develops the “body schema” of a robot. The method has three features: 1) estimation of light-sensor positions based on the Time Difference of Arrival (TDOA) of signals and multidimensional scaling (MDS); 2) incremental update of the estimation; and 3) no additional equipment. We carried out simulation experiments in which a mobile robot moves around environments or follows another robot. Each robot has several light sensors that collect data from which cross-correlation functions are derived and the TDOA is computed. Experimental results show that our method can estimate the positions of sensors deployed on the body and identify the sensors on the same part of the body.

**Keywords:** embodiment, robotic body schema, localization, incremental mapping, cross-correlation

## 1. INTRODUCTION

For human beings, body schema, which is the unconscious neural map of the spatial relations among body parts, is necessary for imitation, spatial perception, and goal-oriented behaviors. Recent research has revealed plasticity in the body schema that can be changed in minutes [7, 12]. This neural map indicates us where an external stimulus has been received.

How can we construct such a highly adaptive representation of the body (morphology) for robots?

Most previous studies in robotics have assumed that the internal model of the body is fixed and/or given by the designers. However, this assumption is not reasonable when the morphology is likely to change (e.g., modular robots). Also, a fixed model of the body might negatively affect the adaptive behavior of the robots [4].

In recent years, studies have begun to focus on the adaptive representation of robot bodies in the field of cognitive developmental robotics [1]. These studies implemented computational models that accomplished one of the functions performed by the human body schema. Some previous works have investigated the localization of sensors deployed on the body [6, 11]. Other researchers have focused on the extendable body schema of a robot [9, 14].

On the other hand, many other studies have been conducted on robotic mapping [13, 15] and localization in sensor networks [2, 3, 16]. By considering the construction of an adaptive body schema as a localization problem, it might be possible to introduce existing localization algorithms into the new problem domain. Such algorithms, however, do not share all assumptions with the localization of sensors on robots. For example, we cannot use a method that utilizes fixed landmarks [13] or wireless signals [3] since the sensors are not fixed to the environment or do they communicate by wireless signals.

This paper addresses a novel method for the construction of robotic body schema. Here, we define *robotic body schema* as the internal model of sensor location

in relative scale. Our method is based on Time Difference of Arrival (TDOA) and Multidimensional Scaling (MDS), so online it can estimate sensor positions accurately. Both this study and others are aimed at the localization of sensors by statistically processing sensory inputs. The main difference, however, is that our method estimates distances between sensors by extracting temporal information (TDOAs) from sensory inputs.

Human beings are thought to construct body schema based on multimodal sensory inputs [12]. However in this study, robotic body schema is developed based only on light sensors due to the simple implementation and industrial applicability such as localization for sensor networks where motion information is not available.

## 2. CONSTRUCTION OF ROBOTIC BODY SCHEMA

In this paper, we define *robotic body schema* as the internal model of sensor location in relative scale. The proposed method is discussed below in detail.

### 2.1 Outline of Proposed Method

Fig. 1 illustrates a schematic of the proposed method that consists of two modules: Similarity Extraction Module (SEM) and Incremental Mapping Module (IMM). First, SEM computes the cross-correlation between sensory inputs and estimates relative distances. IMM then incrementally updates the estimated distances and computes the positions of sensors. These modules are presented in the following subsections.

### 2.2 Similarity Extraction Module

SEM estimates relative distances and similarities between sensors by computing the cross-correlation functions for all pairs of sensors. Let  $X_i(t)$  and  $X_j(t)$  denote sensory inputs from the  $i$ th and  $j$ th sensors. The cross-correlation function is derived from the correlation coefficient between time series  $X_i(t)$  and time-shifted series  $X_j(t - \Delta t)$ . We define cross-correlation function as fol-

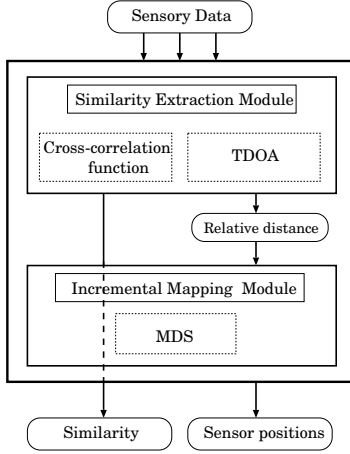


Fig. 1 Schematic of method

lows:

$$R_{ij}(t, \Delta t) = \frac{C_{ij}(t, \Delta t)}{\sqrt{C_{jj}(t, \Delta t)C_{ii}(t, 0)}}, \quad (1)$$

where  $C_{ij}(t, \Delta t)$  denotes cross-covariance between  $X_i(t)$  and  $X_j(t - \Delta t)$  at time  $t$ ,

$$C_{ij}(t, \Delta t) = \frac{1}{w} \sum_{t=0}^{w-1} (X_i(t) - \bar{X}_i)(X_j(t - \Delta t) - \bar{X}_j), \quad (2)$$

and  $w$  is the number of samples, or the width of the window.  $\bar{X}_i$  and  $\bar{X}_j$  are the averages of  $X_i(t)$  and  $X_j(t - \Delta t)$  in their own windows.

Now, consider the maximum value of  $R_{ij}(t, \Delta t)$  for  $\Delta t \in [-T, T]$ , where  $T$  is a positive constant. Then we get the following equations:

$$\Delta T_{ij} = \operatorname{argmax}_{\Delta t} R_{ij}(t, \Delta t), \quad (3)$$

$$\mathbf{f}_i = [\Delta T_{1i}, \Delta T_{2i}, \dots, \Delta T_{ni}]^T, \quad (4)$$

where  $[\cdot]^T$  stands for the transpose of a matrix and  $n$  denotes the number of sensors. Let us call  $\mathbf{f}_i$  a *relative distance vector*.

Assuming that  $\mathbf{f}_i$  is a set of the Time Difference of Arrival (TDOA) of signals traveling at constant speed, these TDOAs are proportional to the physical distances between the  $i$ th and other sensors. The TDOA method was applied to a sensor localization problem where the movements of a building's occupants were sensed [16]. In such localization problems, the physical positions of sensors are usually fixed. On the other hand, we will show that this simple TDOA method also works for the localization of sensors deployed on a robot (see Fig. 2). External objects cast shadows on the body, and even some parts of the body does so on other parts of the body. Our method measures the temporal distance of such changes generated by the shadows to estimate the relative distance.

For all sensors, we calculate cross-correlation function and TDOA and then obtain cross-correlation matrix  $\mathbf{R}(t)$

and relative distance matrix  $\mathbf{F}(t)$ :

$$\mathbf{R}(t) = [R_{ij}(t, \Delta T_{ij})] \quad (i, j = 1, 2, \dots, n) \quad (5)$$

$$\mathbf{F}(t) = [\mathbf{f}_1 \ \mathbf{f}_2 \ \dots \ \mathbf{f}_n]^T. \quad (6)$$

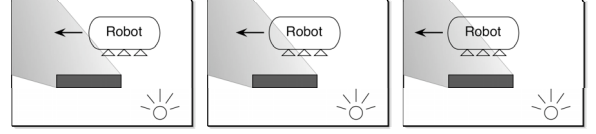


Fig. 2 Shadow casted on a robot moving left. Each triangle represents a light sensor.

### 2.3 Incremental Mapping Module

IMM first updates the above matrix using the following update rule:

$$\hat{\mathbf{F}}(t) = (1 - \lambda)\hat{\mathbf{F}}(t - 1) + \lambda\mathbf{F}(t), \quad (7)$$

where  $\hat{\mathbf{F}}(t)$  denotes the estimated relative distance matrix and  $\lambda$  denotes the learning rate.

Next, we calculate the location of sensors by Multidimensional Scaling (MDS) [5]. The MDS algorithm starts with a matrix of item-item similarities and then assigns a location of each item in a low-dimensional space. Applying the MDS algorithm to  $\hat{\mathbf{F}}(t)$ , we project the location of sensors on two or three dimensions.

### 2.4 Clustering Sensors

Computing TDOA every pairs of sensors might have some drawbacks. Suppose that both groups A and B consist of multiple sensors ( $a_1, a_2, \dots, b_1, b_2, \dots$ ) and that A is not close enough to B. It is then impossible to measure accurately the TDOA between  $a_i$  and  $b_j$  with constant window width  $w$ . In addition, TDOA computation related to a broken sensor is nothing but a waste of time. How can we select appropriate sensory data among all? In the discussion below we present a method that cluster sensors by similarity of sensory data.

Since robot movements are ruled by the laws of physics, sensors on an identical body part receive structured inputs. On the other hand, sensory inputs are less correlated if those sensors are deployed on different body parts. When visual and/or motion information is available, some statistical methods are applicable [8, 10, 17]. Here we introduce a method using cross-correlation.

First, we define similarity matrix  $[\mathcal{R}_{ij}(t)]$  as follows:

$$\mathcal{R}_{ij}(t) = \frac{1}{2T + 1} \sum_{\Delta t=-T}^T R_{ij}(t, \Delta t) \quad (8)$$

Obviously, a particular statistical estimation approximates the true similarity matrix by reducing noise. For simplicity, we assume that the following update rule estimates the true similarity matrix:

$$\hat{\mathcal{R}}_{ij}(t) \leftarrow (1 - \lambda)\hat{\mathcal{R}}_{ij}(t - 1) + \lambda\mathcal{R}_{ij}(t), \quad (9)$$

where  $\hat{\mathcal{R}}_{ij}(t)$  represents the estimation of  $\mathcal{R}_{ij}(t)$  at time  $t$ . With threshold  $\varepsilon$ , the following operation yields adjacency matrix  $\mathbf{A}$ :

$$A_{ij} = \begin{cases} 1 & \text{if } \hat{\mathcal{R}}_{ij} > \varepsilon \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

We then trace all the connected nodes to obtain a cluster of nodes.

### 3. EXPERIMENT I: ROBOTIC BODY MAPPING

In this section we present simulation experiments to evaluate the proposed method. Their objective was to qualitatively examine the localization error of our method. In Experiments I-a and I-b, localization in 2D and 3D spaces is examined.

#### 3.1 Experimental Setup

##### 3.1.1 Experiment I-a

In the experiments, we programmed a mobile robot to move in an arena under some conditions. The right-hand figure of Fig. 3 shows an example of the task environment. The arena is one-meter-square, and a light source is put at the center 500 mm above the ground.

To collect data, we used light sensors. While human beings are thought to construct body schema based on multimodal sensory inputs, we chose to develop robotic body schema based only on light sensors due to the simplicity of implementation and industrial applicability such as localization for sensor networks. Moreover, light sensors collect a sufficiently wide variety of data for statistical analysis; however touch/pressure sensors deployed on rigid bodies usually produce sparse data, which makes analysis unpractical. We assume constant luminous intensity regardless of the distance from the light source. Sensory inputs are computed as the inner product of both the directional vector of light and of light sensors. Random noise is added to the inputs.

The left-hand figure of Fig. 3 shows a schematic view of the robot used in the experiments. The robot is 55 mm in diameter, and its design is based on the Khepera robot of the K-Team Corporation. Twelve light sensors are deployed along the periphery of the robot. We implemented the task environment in Cyberbotics' robot simulator Webots.

We conducted 2D mapping experiments under three conditions: the robot (a) rotates, (b) moves randomly, or (c) avoids obstacles. The right-hand figure of Fig. 3 shows the task environment for condition (c). For obstacle avoidance, we implemented a simple controller based on conditional reflexes. The most remarkable results were obtained under the conditions shown in the middle column of Table 1.

##### 3.1.2 Experiment I-b

The left-hand figure of Fig. 4 illustrates the robot (Aibo of Sony) and the task environment used in the experiments. The arena is two-meters-square, and the light

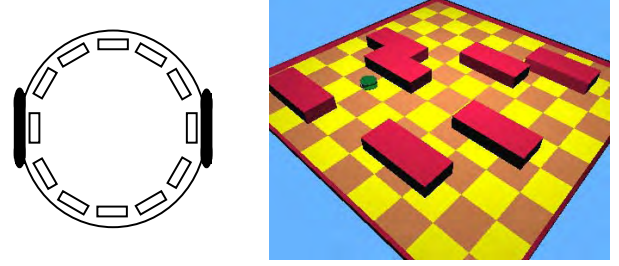


Fig. 3 Left: Schematic view of robot. Boxes represent light sensors. Right: Task environment for condition (c).

source is put 2 m above the ground. Light sensors are deployed on the head of the robot, as shown in the right-hand figure of Fig. 4. For simplicity, these sensors are positioned on symmetrical points on a sphere.

In experiment I-b, quadrupedal locomotion is performed by the robot (condition (d)). The right column of Table 1 shows the parameters used. Note that TDOA is computed every ten steps to reduce computational cost in both I-a and I-b.

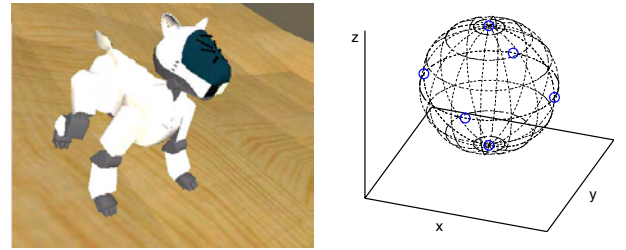


Fig. 4 Left: Aibo and environment. Right: Sensor layout. Each sensor, deployed on the head of Aibo, is represented by a circle.

Table 1 Parameters used in Experiment I

Parameter	Value I-a	Value I-b
Simulation step	64 ms	16 ms
Maximum action steps	10,000	40,000
Window width $w$	60	80
$T$	90	200
Learning rate $\lambda$	0.001	0.001

#### 3.2 Evaluation Method

In this subsection, we introduce an evaluation index "relative error". Relative distance (see 2.1) is not absolute distance but something proportional to it. Consequently, estimated position has arbitrariness in scale and is not absolute position. Therefore, we need an evaluation index for estimated positions.

Let  $\mathbf{x}_i$  and  $\hat{\mathbf{x}}_i$  denote the true and estimated position of the  $i$ th sensor, respectively.  $\mathbf{x}_1$  is used as a reference to normalize the absolute positions of other sensors. Each estimated position is normalized by using  $\hat{\mathbf{x}}_1$  as well. The true and estimated positions should also be rotated since MDS does not specify sensors directions. Therefore,  $\mathbf{x}_i$  and  $\hat{\mathbf{x}}_i$  are rotated so that  $\mathbf{x}_1$  and  $\hat{\mathbf{x}}_1$  are on x-axis, respec-

tively. Thus, we define relative error  $e_r$  as follows:

$$e_r = \frac{1}{n} \sum_{i=1}^n \left\| \frac{\mathbf{G}(-\arg \mathbf{x}_1) \mathbf{x}_i}{\|\mathbf{x}_1\|} - \frac{\mathbf{G}(-\arg \hat{\mathbf{x}}_1) \hat{\mathbf{x}}_i}{\|\hat{\mathbf{x}}_1\|} \right\|, \quad (11)$$

where  $n$  and  $\mathbf{G}(\phi)$  denote the number of sensors and a rotation matrix. Note that relative error is measured in terms of the Euclidean norm of vectors.

### 3.3 Results

#### 3.3.1 Experiment I-a

In Fig. 5, estimated positions under condition (c) are plotted. In the figure, ‘o’ and ‘x’ stand for the actual and estimated positions of the light sensors, respectively. The left-hand, middle, and right-hand figures show the results at 0, 2,000, and 10,000 steps, respectively. Fig. 5 shows that estimated positions converge into the actual positions as the number of steps increases.

Next, we quantitatively examined the results. Relative error is plotted against the number of steps in Fig. 6. In the figure, the solid, dotted, and broken lines represent the relative error of conditions (a), (b), and (c), respectively. Each line shows the average of 10 experiments. Fig. 6 shows that relative error under each condition decreases with an increase in the number of steps. Specifically, the relative error of condition (c) is approximately 0.3 at 1,000 steps and decreases to 0.1 at 8,000 steps.

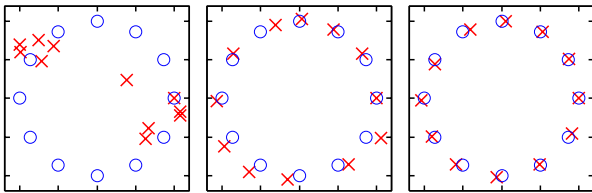


Fig. 5 Estimated sensor positions under condition (c). ‘o’: actual positions, ‘x’: estimated positions. Left: At 0 steps. Middle: 2,000 steps. Right: 10,000 steps.

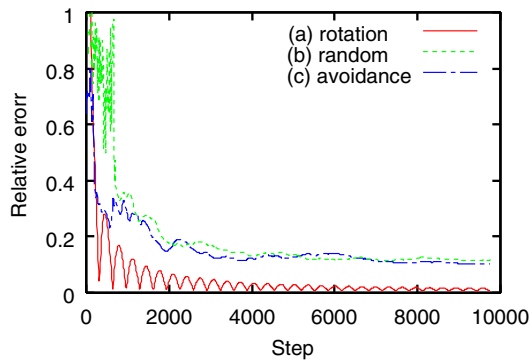


Fig. 6 Experimental result I-a. Relative error of estimated sensor positions.

#### 3.3.2 Experiment I-b

Fig. 7 also illustrates the variation of relative error in I-b. Relative error is large and unstable until 5,000 steps, and it converges to 0.2 at 10,000 steps.

Also, estimated positions under condition (d) are plotted in Fig. 8. The left-hand, middle, and right-hand figures show results at 0, 5,000, and 40,000 steps.

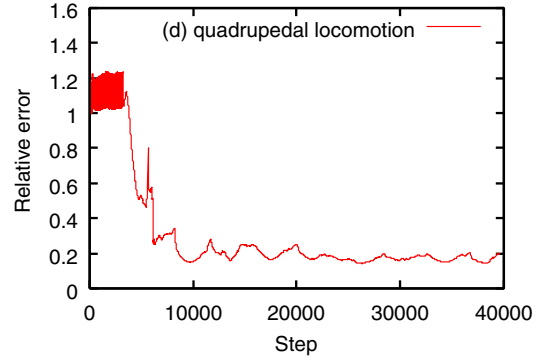


Fig. 7 Experimental result I-b. Relative error of estimated sensor positions.

### 3.4 Discussion

The experimental results are discussed below. Fig. 6 shows that condition (a) requires fewer steps to converge than conditions (b) and (c) and that the relative error of (c) is also smaller than the others. In condition (a), the robot rotated at a constant speed and obtained periodic data, which made the relative distance accurate. In condition (b) and (c), the sensory inputs diversely varied since the robot did not move periodically, which decreased the accuracy of the TDOA method. For example, let us consider the robot moving straight toward the light source. Since the sensors are symmetrically positioned, some pairs of sensors would obtain identical inputs in certain chunks, so the TDOA method would fail.

However, this problem can be solved by statistical learning. Fig. 6 verifies that after 8,000 steps (= 512 sec) relative error converges and that sensor positions were successfully estimated in this task environment in terms of relative error.

Comparing Fig. 6 and Fig. 7, relative error in condition (d) is larger than other conditions. This is because one of the light sensors was positioned below the jaw of Aibo, and it detected fewer changes in luminance. Since the TDOA method measures the time difference between changes in two time series, fewer changes produce larger estimation error. The right-hand figure of Fig. 8 also supports this conclusion. In the figure, the distance between the estimated and actual position of the bottom sensor is the largest.

## 4. EXPERIMENT II: IDENTIFYING SENSORS ON THE SAME BODY

The objective of the experiments is to investigate how our method clusters sensors on the same body part, since clustering related sensors can improve localization accuracy (see Section 2.4).

### 4.1 Experimental Setup

The left-hand figure of Fig. 9 shows a schematic of the robots used in Experiment II. The robots, named K1

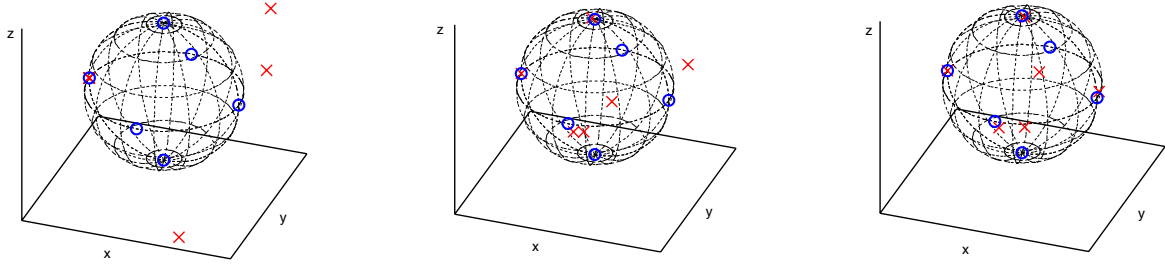


Fig. 8 Estimated positions of sensors under condition (d). ‘o’: actual positions, ‘x’: estimated positions. Left: At 0 steps. Middle: 5,000 steps. Right: 40,000 steps.

and K2, are connected with a virtual spring so that K2 follows K1. In the early stage of the experiments, we made the robots as if joined (see right-hand figure of Fig. 9). The virtual force of the spring was invalidated after 50,000 steps so that both robots can move freely. Then the robots were reconnected at 100,000 steps. Sensory data are recorded during the experiments, and a similarity matrix is computed offline. Table 2 shows the parameters used in the experiments.

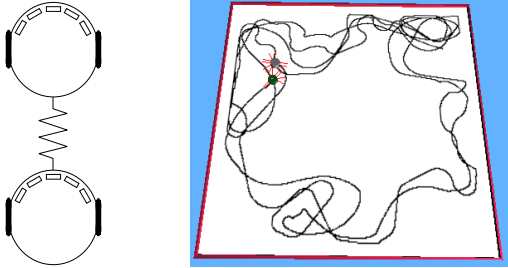


Fig. 9 Left: Robots connected with a virtual spring. White boxes represent light sensors. Right: K1 and K2 trajectories with a virtual spring.

Table 2 Parameters used in Experiment II

Parameter	Value
Simulation step	64 ms
Maximum action steps	150,000
Window width $w$	80
$T$	120
Learning rate $\lambda$	0.001
Threshold $\varepsilon$	0.1

## 4.2 Results

First, we quantitatively present the results of clustering sensors. Fig. 10 illustrates the number of clusters plotted against the number of steps. Solid and broken lines represent estimated and theoretical values, respectively. The average of ten experiments is shown in the figure. Note that the number of clusters obtained in each run is not continuous. The estimated number started from ten at the first step and decreased to one at approximately 30,000 steps. After the bodies were disconnected at 50,000 steps, it converged to two. Also, the number decreased again to one at approximately 130,000 steps after reconnection.

Next, we examine the qualitative results. Similarity matrices (see 2.4) is visualized in Fig. 11. Dotted lines show the boundary between K1’s and K2’s sensors.

In the figure, the left-hand, middle, and right-hand figures show the similarity matrix at 45,000, 95,000, and 145,000 steps, respectively. In each subfigure of Fig. 11, the top right and bottom left submatrices show similarity between K1’s and K2’s sensors. Some of their sensors have similarity, as K1’s sensors do in the left-hand figure. Specifically, some of the similarity values are more than 0.2 at 45,000 steps. On the other hand, the middle figure shows that all the values in the top right and bottom left submatrices are less than 0.1 at 95,000 steps. In the right-hand figure, some values recover to approximately 0.2 at 145,000 steps.

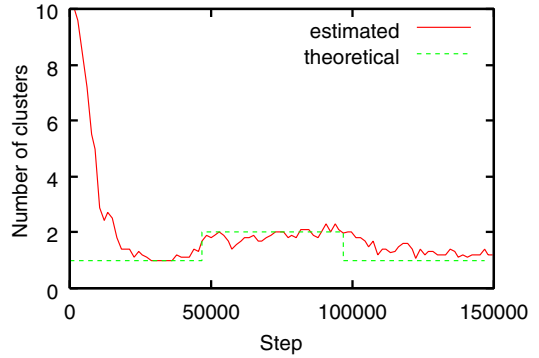


Fig. 10 Variation of number of clusters

## 4.3 Discussion

In this experiment, we assumed that highly correlated sensors were on the same body. Fig. 10 and Fig. 11 indicate that the proposed method categorized K1’s and K2’s sensors into two groups. Although some similarity values between particular sensors are less than 0.1, it is possible that they were categorized into the same group. Because the method traces every possible path in the adjacency matrix (see 2.4), some sensors have indirect relationships with others.

The experimental results indicate that the sensors are successfully clustered; however some breakthroughs are necessary to combine this method with the aforementioned localization technique. A threshold was determined based on the results of preliminary experiments that might not work for other types of experiments. Also, when the density of sensors is small and the relationships among sensors are not approximated by linear functions, a correlation-based approach might fail. To measure similarity in such cases, possible substitutes include

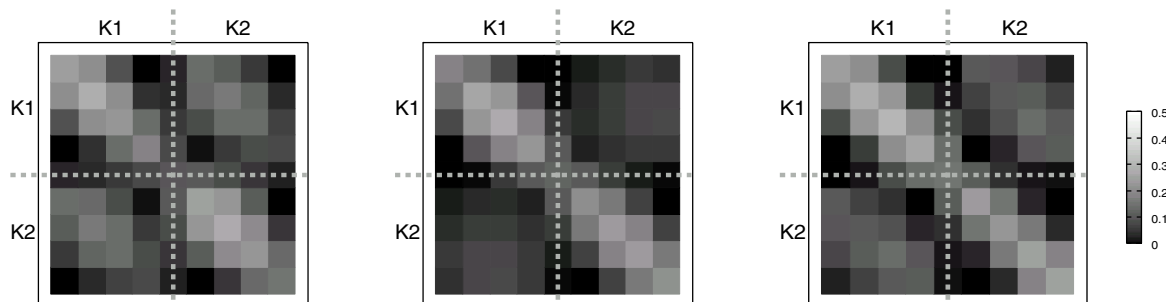


Fig. 11 Variation of similarity matrix. Left: Robots connected (45,000 steps). Middle: Robots disconnected (95,000 steps). Right: Robots reconnected (145,000 steps).

statistical measures such as mutual information and KL-divergence.

## 5. CONCLUSION

Constructing a geometrical model of a robot's body by low-cost sensors noticeably impacts robotics, sensor networks, and biological implications. Such adaptive representation of the body will provide more adaptability to robots when the body is likely to change. Our method recovers the geometry of the sensors of robots by simply computing TDOA within light-sensory data. This method does not need any additional devices in the environment, unlike existing methods such as motion capture equipment.

We carried out simulation experiments whose results indicated that our method successfully recovered the geometry of sensors. Then we evaluated our method by experiments in which the robot body changed and obtained the results where the sensors were clustered based on the changes in the body.

### Acknowledgments

The authors thank Dr. Naoto Iwahashi for useful comments and discussions. This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for JSPS Fellows, 18-2972, 2006.

## REFERENCES

- [1] M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi, "Cognitive developmental robotics as a new paradigm for the design of humanoid robots," *Robotics and Autonomous Systems*, vol. 37, pp. 185–193, 2001.
- [2] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, 2000.
- [3] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of International Conference on Mobile Computing and Networking (MOBICOM)*, 2003.
- [4] K. Hosoda, "What morphology brings to learning, what learning brings to morphology," *JRSJ*, vol. 22, no. 2, pp. 186–189, 2004, (in Japanese).
- [5] J. B. Kruskal and M. Wish, *Multidimensional Scaling*. Sage Publications, 1978.
- [6] Y. Kuniyoshi, Y. Yorozu, Y. Ohmura, T. Otani, A. Nagakubo, and T. Yamamoto, "From humanoid embodiment to theory of mind," in *Embodied Artificial Intelligence*, 2004, pp. 202–218.
- [7] A. Maravita and A. Iriki, "Tools for the body (schema)," *Trends in Cognitive Sciences*, vol. 8, no. 2, pp. 79–86, 2004.
- [8] G. Metta and P. Fitzpatrick, "Early integration of vision and manipulation," *Adaptive Behavior*, vol. 11, no. 2, pp. 109–128, 2003.
- [9] C. Nabeshima, M. Lungarella, and Y. Kuniyoshi, "Timing-based model of body schema adaptation and its role in perception and tool use: A robot case study," in *Proceedings of ICDL-05*, 2005, pp. 7–12.
- [10] L. Olsson, C. L. Nehaniv, and D. Polani, "Measuring informational distances between sensors and sensor integration," in *Proceedings of Artificial Life X*, 2006, pp. 316–322.
- [11] D. Pierce and B. Kuipers, "Map learning with uninterpreted sensors and effectors," *Artificial Intelligence*, vol. 92, pp. 169–229, 1997.
- [12] V. S. Ramachandran and S. Blakeslee, *Phantoms in the Brain: Probing the Mysteries of the Human Mind*. Harpercollins, 1998.
- [13] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicules*. Springer-Verlag New York, Inc., 1990, pp. 167–193.
- [14] A. Stoytchev, "Computational model for an extendable robot body schema," Georgia Institute of Technology, College of Computing, Technical Report GIT-CC-03-44, 2003.
- [15] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002, pp. 1–35.
- [16] C. R. Wren, "Large networks of ultra-low resolution sensors in buildings," in *Proceedings of International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, 2005, pp. 373–389.
- [17] Y. Yoshikawa, "Subjective robot imitation by finding invariance," Ph.D. dissertation, Osaka University, 2005.