Population aging has become a major problem in developed countries. As the labor force declines, robot arms are expected to replace human labor for simple tasks. A robotic arm attaches a tool specialized for a task and acquires the movement through teaching by an engineer with expert knowledge. However, the number of such engineers is limited; therefore, a teaching method that can be used by non-technical personnel is necessitated. As a teaching method, deep learning can be used to imitate human behavior and tool usage. However, deep learning requires a large amount of training data for learning. In this study, the target task of the robot is to sweep multiple pieces of dirt using a broom. The proposed learning system can estimate the initial parameters for deep learning based on experience, as well as the shape and physical properties of the tools. It can reduce the number of training data points when learning a new tool. A virtual reality system is used to move the robot arm easily and safely, as well as to create training data for imitation. In this study, cleaning experiments are conducted to evaluate the effectiveness of the proposed method. The experimental results confirm that the proposed method can accelerate the learning speed of deep learning and acquire cleaning ability using a small amount of training data.

Keywords: system integration, tool manipulation, imitation learning, deep learning, human support robot

1. Introduction

The purpose of this study is to enable the robot arm to infer the use of cleaning tools, such as brooms, and to learn how to use them directly from human behavior, thereby enabling the automation of tasks that have never been automated previously. Currently, robot arms are used only in large factories and are equipped with special end-effectors to perform tasks. In the future, the robot arm is expected to perform tasks not only in large factories, but also in nursing homes and other small facilities. It is not economical to fabricate specific tools for each task and an end-effector for each operation. Therefore, an intelligent system that can extend the functions of a robot using tools is required.

Whereas sweeping dirt is classified as a simple task for humans, it is not classified as such for robots. Different types of brooms with different tip shapes, tip elasticities, handle lengths, and other features exist. In many cases, they can be used variously. When the robot arm grasps a broom, its usage must be specified by a human in advance. Teaching an ideal movement to the robot arm is a difficult task that requires considerable expertise and time. Hence, it is difficult for the robot arm to acquire the ability to sweep using a broom. Even when the robot arm performs the same sweeping motion, if the shape of the broom tip is different, the dirt will move in different directions. A difference in elastic characteristics results in a difference traveled by the distance between dirt particles. Therefore, the use of a broom depends significantly on the shape and elastic characteristics of the broom. Whereas these factors can be sensibly understood by humans, they are difficult to implement explicitly on the robot arm.

One of the advantages of the proposed system is that it expands the area that the robot can clean using a broom. Although a household vacuum cleaner robot can sweep fine dust, it cannot collect dirt of a certain size. Examples include spilled food in a cafeteria or gauze in a hospital. Furthermore, it is not hygienic for the robot to directly touch and move garbage. In this study, we attempt to solve these problems.

Stoytchev et al. created a system of robotic arms that can move an object in the desired direction using a primitive shape tool [1] such as a stick. This system comprises a database that is used to create an environmental map that represents the relationship between the behavior of the robot arm and movement of the object; subsequently, the system uses inverse mapping to calculate the robot’s behavior based on the movement direction of the object. Sinapov et al. designed a system in which the robot moved objects with a rake and a hoe [2]. Their system allowed the robot to select an appropriate tool for a task. Tikhanoff et al. developed a system that can establish an action plan to pull the stuffed animal on the table that was outside the reach of a humanoid robot [3]. The robot was able to execute a motion plan to pull a stuffed animal using tools...
without explicitly programming it. Furthermore they allowed the robot to determine the desirable type of grasping tool for moving the object in the desired direction by calculating the tip shape of the tool as a feature of the image and point cloud [4, 5]. Wicaksono et al. attempted to create a learned model that represents tool usage acquired in a simulator to a real workspace [6].

In recent years, deep learning has been used to imitate human behavior [7–10]. Finn et al. proposed a method to directly learn the state space representation for reinforcement learning from a camera image using a deep space autoencoder [11]. Using the obtained state representations, they performed reinforcement learning using a local linear model and trained the robot to learn motor skills such as moving objects using a spatula and moving toy blocks. Levine et al. proposed an end-to-end method using deep learning that directly translates images into robot arm motor torques [12]. They demonstrated that perception and control can be trained simultaneously instead of separately. Rajeswaran et al. used deep reinforcement learning to train a 24 degree of freedom (DOF) robot hand to perform complex actions such as pen spinning and hitting a nail with a hammer on a simulator. In addition, by using human demonstrations, the learning time was reduced, thereby enabling the hand to acquire natural behaviors.

High-quality training data are important for imitation learning. In this study, training data were created by teleoperating a robot arm. This method allows an operator to intuitively control the robot arm and create high-quality training data. Research pertaining to operating robots using teleoperation has already been conducted, and it provides important insights for creating teaching data for robots to acquire their skills. Fujihira et al. created a haptic device for operating the manipulators used in surgery [13]. This device can present the reaction force from the manipulator and convey the softness of the organs in contact with the operator. Kamata et al. proposed a method for creating a haptic device with one degree of freedom to control a manipulator with six degrees of freedom [14]. The method enabled an easy assembly of products by operating the manipulator and creating a virtual plane in the workspace by the operator. Hoshino et al. created a non-contact remote control system for operating a lunar exploration robot from Earth [15]. In that study, methods for intuitively manipulating robots were investigated, and a non-contact operating system was used to create teaching data. Zhang et al. developed a teleoperation system using a virtual reality (VR) system to create high-quality training data [16]. In their study, a robot learned to imitate the grasping of a ball and the use of a nail hammer. In this study, we used a VR system to control the robot and create training data. However, the amount of training data required to make the robot behavior with deep learning is huge and takes time to create. Therefore, it costs a lot of time to train a robot to use a single tool. Previously we proposed a method to reduce the amount of training data required for learning how to use a cleaning tool by imitation learning [17]. However, this system was not practical because target dirt was only a single object. Moreover, the physical characteristics of the cleaning tools were disregarded. Hence, we improved the system in this study.

Langsfeld et al. investigated robot cleaning using tools that considered the deformation of an object in a twin-armed robot [18, 19]. Hess et al. developed a system to calculate the optimal path for a robot arm to polish a three-dimensional surface, such as a car or a chair [20]. Elliott et al. used machine learning to clean sand from a desk [21]. In those studies human skills were not limited; instead, actions were planned by predicting the next state of the experimental environment. In these cleaning studies, the robot used only one tool and had to perform recalculations using a different tool that was used by humans.

The remainder of this paper is organized as follows. In Section 2, the architecture of deep learning and the proposed method, including the calculation of the broom similarity algorithm, are described. In Section 3, the cleaning experiment and its results are presented. In Section 4, a discussion regarding the performance of the proposed method is presented. Section 5 presents the conclusions and challenges.

2. Broom Usage Learning System

In this section, the proposed learning system is described. In particular, the process by which the system imitates the use of a broom from a person, uses experience, and learns the use of a broom from a small amount of training data, is presented. In this study, broom usage represents the parameters of a deep neural network (DNN). In the proposed method, the robot learns to use a broom using a DNN. DNNs can represent complex broom usage; however, they require a large amount of training data. In this study, we attempted to reduce the amount of training data required to learn to use a broom by assuming the following. If the shape and physical properties of the two brooms are similar, then the usage of the two brooms is similar. In other words, the parameters of the DNN that represent the usage of the two brooms will have similar values. This is the memory process of retrieving the usage of a tool based on its shape and characteristics.

Additionally, the abovementioned process reduces the amount of training data. The shape and physical characteristics of the broom as well as the usage data are stored in the database. When a new broom is provided, the similarity between the shape of the broom provided and that of the broom in the database is calculated. The broom with the highest similarity is identified using the stored values in the DNN. Next, the broom is loaded into the DNN. Subsequently, the DNN attempts to learn the usage of the broom provided using additional training data. Because the DNN is provided with a usage similar to the usage of the new broom provided, it is assumed that the learning can be completed even with a small amount of additional training data provided by a human.
2.1. Experimental Environment

The experimental environment of this study is illustrated in Fig. 1. In the experiment, a seven-axis robot arm was placed, and the broom was fixed to its end-effector. The broom was fixed to a six-axis force sensor. Yellow dirt (10 mm × 10 mm × 10 mm cubes) and a dustpan were placed on the black table. An RGB camera was installed to observe AR markers and dirt, the position and angle of the broom and dustpan were observed using the AR markers, and the coordinates of the dirt were observed via image processing.

2.2. Broom Usage Acquisition

In this section, we describe the method to train a broom network. In addition, a system that learns the usage of a new broom with a small amount of training data is explained. An overview of the proposed system is presented in Fig. 2.

First, an experience database that stores multiple sets of data is set; subsequently, it is used to combine the broom point cloud, tip angle of broom \( \theta_{\text{database}} \), and tip elasticity \( \omega_{\text{database}} \) with broom usage \( \omega_{\text{database}} \). Next, the point cloud of a new broom that does not exist in the experience database is provided to the system. The system attempts to learn to use the broom; the similarity calculator calculates the similarities between the broom provided and the brooms in the database. The specific calculation method is described in Section 2.3. The similarity calculator identifies the broom with the highest similarity \( s_{\text{max}} \) among the brooms in the database and sends the parameter \( \omega_{\text{max}} \) to the broom network.

The broom network loads the \( \omega_{\text{max}} \) that belongs to the maximum similarity \( s_{\text{max}} \) broom. Next, a small additional amount of training data \( D_{\text{extra}} \) is loaded and it trains the robot to learn the usage of the provided broom. After learning, the system can extract observation data \( o_t \) from the experimental environment and use it as an input to output the robot’s tip velocity. The observed data at time \( t \) are substituted into Eq. (1).

\[
o_t = (x_{\text{arm}}^{-2t}, x_{\text{broom}}, x_{\text{dirt}}, x_{\text{dustpan}}, f_{\text{vertical}})
\]

In the equation above, \( o_t \) comprises components (a)–(e) listed below. Each observation data point was preprocessed so that it was normalized to the value \([-1, 1]\).

(a) \( x_{\text{arm}}^{-2t} \in \mathbb{R}^{18} \): End-effector position of the robot arm at time \( t = 2 \) to \( t \).
(b) \( x_{\text{broom}} \in \mathbb{R}^{6} \): Coordinates of AR markers attached to broom.
(c) \( x_{\text{dirt}} \in \mathbb{R}^{6} \): Coordinates of the centers of the dirt.
(d) \( x_{\text{dustpan}} \in \mathbb{R}^{6} \): Coordinates of the AR marker for dustpan.
(e) \( f_{\text{vertical}} \in \mathbb{R}^{1} \): Vertical component of the force applied to the broom and observed by force sensor.

The training data \( D_{\text{training}}, D_{\text{extra}} = \{o^{(i)}, u^{(i)}\} \) comprised the observed data and corresponding end-effector velocity commands. An overview of the training data creation system is presented in Fig. 3. The training data were created using the VR system, which was used as a motion capture system to observe the coordinates of the VR controllers. The end-effector trailed the displacement of the controller held by the operator. The operator performed cleaning by controlling the robot arm, and the training data were recorded simultaneously.

After learning using training data \( D_{\text{training}}, D_{\text{extra}} \), the broom network receives the observed data \( o_t \) in the experimental environment and predicts the target end-effector velocity \( u_t \).

The pose estimator receives the image from the RGB camera and estimates the coordinates of the two AR markers. Additionally, it extracts the center of mass of the dirt on the image. The extraction algorithm is shown in Fig. 4. First, the image is converted from the RGB color space to the HSV color space. Next, the image is binarized to extract the color of the dirt. Because the image after this process is noisy, noise is removed from it. The noise occupies only a few pixels in the image, i.e., less than the amount of garbage; this feature eases noise removal. The opening process removes small blobs of pixels that are not garbage. The closing process combines large and small pixels of dirt scattered around them. This enables the center of the dirt to be estimated, even if it is scattered when the dirt is moved by a broom. After labeling some blobs, a blob that occupies the largest pixel area is collected as a dirt. The blob center is calculated using the center of mass of the blob’s pixels. Using these processes, the coordinates of dirt were calculated. Because the dustpan and dirt appeared on the table, the \( z \)-coordinate of the dustpan was used as the \( z \)-coordinate of the dirt.

The robot arm controller receives the robot arm’s end-effector target velocity \( u_t \) and converts it to the target angular velocity \( \theta_t \) for the seven motors in the robot arm. The control cycle of the module was 40 Hz.
2.3. Similarity Calculation Algorithm

In this section, the algorithm for computing similarity, which can measure the similarity of two brooms, is explained. Currently, classification methods exist that use machine learning to determine whether two objects are similar or belong to the same category. However, these methods are not suitable for determining the type of broom. First, a significant amount of training data must be prepared to perform machine learning. Although many datasets are available, they require the use of large-scale machine learning architectures. In our proposed method, a dataset need not be prepared, and the broom type can be identified, which is difficult to achieve by the existing method. In addition, most existing methods use only images as inputs. Therefore, they cannot consider the broom size and physical characteristics. Therefore, in this study, a new classification method was developed. As this method is a numerical method, a dataset is not required, and the computational resources necessitated are lower than those of machine learning methods.

The similarity calculation algorithm is described in this section. The calculation procedure is shown in Fig. 5. The algorithm uses three types of inputs: broom point clouds, tip angle of broom, and elasticity. Two point clouds are required: the first is the point cloud of the broom provided, \( p_s \). The second is the point cloud, \( p_d \), which is stored in the experience database. Two types of calculations were performed for these two point clouds.
In the first calculation, an object oriented bounding box (OBB) is created for each of the two point clouds; subsequently, the volume $v_g$ of the OBB of $p_g$ and the volume $v_d$ of the OBB of $p_d$ are calculated. Using these two values, the subtraction of volume $\Delta v$ is calculated. This calculation compares the size of each broom. The volume $v_g$ of the bounding box can be obtained from the point cloud $p_g$, and $v_g$ can be obtained from $p_d$.

The second calculation uses the moment of inertia around the center of mass for each point cloud. By performing principal component analysis (PCA) on $p_g, p_d$, the center of mass and basis of the point cloud can be obtained. Using PCA, a unique basis for each point cloud is created. Subsequently, the moments of inertia, $I_{xg}, I_{yg},$ and $I_{zg}$ from $p_g$ are obtained. Similarly, the moments of inertia $I_{xd}, I_{yd},$ and $I_{zd}$ from $p_d$ are obtained. Next, $\Delta I_x, \Delta I_y,$ and $\Delta I_z$ are derived by calculating each subtraction.

In addition, physical characteristics are included in the calculations. In this method, the elasticity of the broom was used for similar calculations. The measurement method is as follows. The tip of the broom fixed to the end-effector was set to 0.045 m from the floor, and the robot moved up and down with a cosine wave of amplitude 0.06 m and period 2 Hz (Fig. 6). The vertical upward reaction force generated by the broom touching the floor was measured using a force sensor.

The amount of pushing of the broom tip was set to 0.015 m. Although it was confirmed that the broom pushing amounts were 0.005 and 0.025 m, the vertical reaction force can be measured appropriately, and the pushing amount did not affect the similarity calculation.

One cycle was extracted from the reaction force time-series data, and the data were approximated via quadratic equations; the quadratic coefficients used were the elasticity elements $\alpha_x$ and $\alpha_y$. The components of the similarity between the two brooms are represented by Eqs. (2)–(7).

\[
\Delta I_x = I_{xg} - I_{xd} \quad \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots (2)
\]

\[
\Delta I_y = I_{yg} - I_{yd} \quad \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots (3)
\]

\[
\Delta I_z = I_{zg} - I_{zd} \quad \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots (4)
\]

\[
\Delta v = v_g - v_d \quad \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots (5)
\]

\[
\Delta \theta = \theta_g - \theta_d \quad \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots (6)
\]

\[
\Delta \alpha = \alpha_g - \alpha_d \quad \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots (7)
\]

To calculate similarity $s$, the values obtained using Eqs. (2)–(7) are substituted into Eq. (8).

\[
s = \frac{1}{\sqrt{w_1 \left( \Delta I_x^2 + \Delta I_y^2 + \Delta I_z^2 \right) + \Delta v^2 + w_2 \Delta \theta^2 + w_3 \Delta \alpha^2}} \quad \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots (8)
\]

The term $w$ in the Eq. (8) is a weight, which is used to match the digits of $\Delta v$ and other elements. In this case, $w_1$, $w_2$, and $w_3$ are 0.1, 0.1, and 0.0001 respectively.

2.4. Broom Network

The architecture of the broom network used in the proposed method to imitate the use of brooms is described in this section. The architecture is illustrated in Fig. 7. The broom network comprised eight layers, with 37 dimensions in the input layer, 30 dimensions in the hidden layer, and six dimensions in the output layer. Each layer was connected by a full connection, the activation function was a rectifier, and the loss function was the least-squares error. The mini-batch size of the training data was 8.

This paper describes a method for learning new broom usage based on a small amount of training data. The following was assumed: if the similarity $s$ between two brooms is great, then the parameter values of the broom network are similar. Therefore, the parameters with the greatest similarity $s_{\text{max}}$ in the experimental database were loaded and used as the initial values. This enables new broom usage to be learned using a small amount of training data.

3. Acquisition of Broom Usage and Cleaning Experiment

In an experiment, the proposed method was investigated to determine whether new broom usage can be learned using a small amount of training data. Three brooms were prepared, as shown in Fig. 8, and the robot was trained in advance to use two of them, while storing the results in a database. Subsequently, the usage of the remaining broom was learned using a small amount of training data.

3.1. Experimental Conditions

The position of the dirt in the experimental environment changed randomly during each trial. Similarly, the initial position of the end-effector changed randomly based on the position of the AR marker on the broom and can be observed using a camera. The positions of the dustpan and camera remained fixed at all times.

3.2. Preparation for Learning Broom Usage

An experiment was conducted using normal brooms to determine the extent to which the proposed method can reduce the amount of training data. To prepare for the experiment, the parameters, point clouds, and elasticity
were stored in the database, except for those of the normal broom. The calculated similarities for all brooms are presented in Table 1. The normal broom showed a maximum similarity $s_{\text{max}} = 0.3080$ with respect to the sponge broom. Therefore, the learned parameters of the sponge broom were loaded into the broom usage network as initial values at the starting point of training. Subsequently, the network learned to use the broom using a small amount of training data.

### Table 1. Similarity of brooms: bold numbers are the maximum value in each broom.

<table>
<thead>
<tr>
<th></th>
<th>Normal broom</th>
<th>Brush broom</th>
<th>Sponge broom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal broom</td>
<td>$\infty$</td>
<td>0.2591</td>
<td>0.3080</td>
</tr>
<tr>
<td>Brush broom</td>
<td>0.2591</td>
<td>$\infty$</td>
<td>0.5811</td>
</tr>
<tr>
<td>Sponge broom</td>
<td>0.3080</td>
<td>0.5811</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

### 3.3. Cleaning Experiment Results

Figure 9 shows the usage of a normal broom as a result of learning. As shown by the series of images, the robot successfully used a broom to move to a dustpan. Therefore, it can be inferred that the robot successfully acquired the usage of the broom using a small amount of training data.

In Fig. 10, each line in the figure represents the five-point average of the validation loss, and an epoch less...
which prevented all dirt to be collected into the dustpan. The first issue can be addressed by increasing the sweeping time from single to multiple in a trial. Meanwhile, the second issue may be solved using the relative coordinates between the current positions of the broom tip and dustpan in the DNN.

Another experiment was conducted using not only yellow cubes, but also dirt that was more similar to those in the actual cleaning environment; subsequently, the cleaning ability of the system was verified. Stones were prepared for gardening and sand was used for the sand mold. The success rate of the stones for gardening was 61%, and that of sand mold was 15%, indicating the ability of the proposed system to clean actual dirt. The reason for the low success rate when using industrial sand is that the sand particles are extremely fine, which prevented them from adhering to the broom tip. Additionally, insufficient force to press the broom against the ground is a contributing factor. The proposed system is effective for dirt with a size of 0.01 m; however, it requires multiple trials to clean fine particles.

The effect of the shape of the broom on the calculation time required for learning a DNN was analyzed. DNNs that have learned the usage of a normal broom with a smaller handle (0.26 m smaller than the original normal broom) and the usage of a sponge broom were prepared. Subsequently, two DNNs were prepared to learn the usage of a normal broom. The calculation time required for learning was verified. The result indicated that the validation loss was 0.002, which was attained 85% earlier in a DNN that loaded the sponge broom usage as an initial value. The experiment demonstrated that a larger similarity afforded a shorter learning time.

5. Conclusions

In this study, a fast imitation learning system for a robot arm to use housework tools was proposed and demonstrated. The proposed system is based on the assumption that if the shape and physical features of a tool are similar, then the parameters of deep learning will be similar as well. To use experience and reduce the amount of training data, the system searches for similar brooms among the tools it has learned and uses the deep learning parameters of that broom as its initial values. In the experiment, the training data required for imitation learning by the proposed system and the relationship between the number of training data and cleaning ability were verified.

In this study, a method to calculate the similarity between two brooms was developed. This method does not involve large-scale machine learning, as in conventional classification systems, but uses three-dimensional features and the elasticity of the broom tip. Deep learning was used to learn the usage of the broom using a VR system that can yield high-quality training data by remotely controlling a robot arm. The initial parameters of the most similar broom were loaded to the deep network from the experience database, and the additional learning of the
broom provided allowed the robot to learn the usage of a new broom. The experiment revealed that new broom usage can be learned using a small number of demonstrations.

References:


<table>
<thead>
<tr>
<th>Name</th>
<th>Takahito Yamashita</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affiliation</td>
<td>Assistant Professor, Department of Mechanical Engineering, College of Science and Engineering, Aoyama Gakuin University</td>
</tr>
<tr>
<td>Address</td>
<td>5-10-1 Fuchinobe, Chuo-ku, Sagamihara, Kanagawa 252-5258, Japan</td>
</tr>
<tr>
<td>Brief Biographical History</td>
<td>2015-2020 Nissei Corporation</td>
</tr>
<tr>
<td></td>
<td>2020- Assistant Professor, Aoyama Gakuin University</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Naoki Uchiyama</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affiliation</td>
<td>Professor, Department of Mechanical Engineering, Toyohashi University of Technology</td>
</tr>
<tr>
<td>Address</td>
<td>1-1 Tempaku-cho, Toyohashi, Aichi 441-8580, Japan</td>
</tr>
<tr>
<td>Brief Biographical History</td>
<td>1995- Research Associate, Toyohashi University of Technology</td>
</tr>
<tr>
<td></td>
<td>2001-2002 Visiting Scholar, University of California, Davis</td>
</tr>
<tr>
<td></td>
<td>2015- Professor, Toyohashi University of Technology</td>
</tr>
<tr>
<td>Membership in Academic Societies</td>
<td>Institute of Electrical and Electronics Engineers (IEEE)</td>
</tr>
<tr>
<td></td>
<td>Society of Automotive Engineers of Japan (JSAE)</td>
</tr>
<tr>
<td></td>
<td>Robotics Society of Japan (RSJ)</td>
</tr>
<tr>
<td></td>
<td>Japan Society for Precision Engineering (JSPE)</td>
</tr>
<tr>
<td></td>
<td>Society of Instrument and Control Engineers (SICE)</td>
</tr>
<tr>
<td></td>
<td>Institute of Systems, Control and Information Engineers (ISCIIE)</td>
</tr>
<tr>
<td></td>
<td>Japan Society of Mechanical Engineers (JSME)</td>
</tr>
</tbody>
</table>