芸術科学会論文誌 Vol. 13, No. 4, pp. 198 – 217

A System for Recognizing User Actions on an Interactive Surface using Accelerometers

Naoya Isoyama[1)] **Tsutomu Terada**[1,2)]

Masahiko Tsukamoto[1)]

1) Graduate School of Engineering, Kobe University
2) PRESTO, JST

isoyama (at) stu.kobe-u.ac.jp
tsutomu (at) eedept.kobe-u.ac.jp
tuka (at) kobe-u.ac.jp

Abstract
It is important for systems to recognize user actions using sensors and cameras when constructing interactive systems and artworks. Conventional systems have tackled to make systems recognize wide varieties of user actions and to install sensing devices with various environmental restrictions. However, since the recognition methods in conventional systems are specialized to their own work, they cannot be applied to other systems and a specialist for activity recognition is required to construct the systems. In addition, conventional systems took a long time to select recognition algorithms and to set the recognition parameters. This means that they cannot have enough flexibility to change the actions to be recognized or to adapt to changing environments. This paper proposes a method of adding interactivity to various surfaces and recognizing the positions and intensities of performed action by using multiple accelerometers. Our method has functions that enable easy settings and maintenance even by beginners in activity recognition. Participants in an experiment on constructing interactive surfaces constructed a system that could recognize two actions at two points in 51 minutes on average. Moreover, we confirmed the effectiveness of our approach with two actual artworks in long-term media-arts exhibitions.

Keywords
interactive system, media art, accelerometers, recognition method, surface

1 Introduction

Advances in computer technologies, especially human computer interaction techniques, have expanded the means of art expression. By making art or venues for exhibitions interactive, artists can depict what they want to explain more richly, and audiences can feel like they have entered into the world of the art piece.

There are currently many systems that make arts interactive. These systems are typically small and recognize user actions (e.g., user gestures and the position at which the user inputs them) very accurately. For example, HoloWall [1] uses a camera and infrared illumination to locate objects near a glass wall. It can not only recognize when something is touching the wall but also objects approaching the wall without making any contact. However, HoloWall needs enough space because it needs to have its camera placed behind the surface, and it also has limitations in that the area where it can recognize actions needs to be within the angle of view of the camera. PingPongPlus [2] can detect the location where a ball has hit a table by using multiple microphones. Although it may be easy to apply the system hardware to another table, users need to reconstruct the algorithm for other tables because its recognition algorithm for location detection is specialized for a ping-pong table. Since the recognition methods in conventional systems are specialized for their own work, they cannot be applied to other systems and specialists in activity recognition are required to construct the systems. In addition, conventional systems took a long time to select recognition algorithms and to set the recognition parameters. This means that they cannot have enough flexibility to change the actions to be recognized or to adapt to changing environments.

We created a framework that could uniformly recognize user actions in a variety of situations, and implement software that had a function to utilize recognition to set an interactive system even without experts in activity recognition. The proposed method added interactivity to various surfaces and recognized the positions and intensities of performed actions by using multiple accelerometers. Our method had functions that enable easy setting and maintenance even by beginners in activity recognition. Fabricators who want to set the interactive system only have to fix the accelerometers to the material and perform actions to be recognized several times for the settings and maintenance. The accelerometers are small enough to be installed invisibly in existing environments, and the proposed system can be used in conjunction with other systems. We confirmed the effectiveness of our approach through two actual artworks in long-term media-arts exhibitions [3].

The remainder of this paper is organized as follows. Section 2 explains related work. Section 3 describes the design, system structure, and recognition method in our system. Sections 4 and 5 explain how we implemented the calibration software and evaluated its performance. We then present the two actual applications in Sections 6. Finally, we conclude this paper and discuss our future work in Section 7.

2 Related Work

Several studies have proposed methods of recognizing user actions for interactive systems. For example, Holowall [1] enables users to interact with a computerized wall using their fingers, hands, and own bodies. It recognizes user actions by using infrared lights and a video camera with an IR filter, which is installed behind the wall. The camera captures the images on the back surface of the wall, which are illuminated by the IR lights. HINOCO [4] is an installation system that detects the motion of humans and drapes by using a camera and detects the positions of user actions by using a laser range finder. Fukasawa has proposed a system that recognizes user gestures in front of a wall [5]. It tracks multiple people and recognizes hand gestures by using a camera. ZeroTouch [6] is a flat-panel optical multi-touch technology using a linear array of modulated light receivers that surrounds the periphery of a display to detect touches. It allows precise sensing of hands, fingers, and other objects within a 2-dimensional plane frame. It tracks touch up to 30 concurrent points. These systems recognize user actions that are performed on surfaces with hands the same as our system does. However, Holowall needs to have its camera placed behind the surface, and it needs enough space. HINOCO and Fukasawa's systems are difficult to install in public spaces because they need to have their cameras placed at spots in the public eye. Since methods using cameras cannot recognize user actions without body movements (e.g. breathing) and the intensity of actions, they need to be integrated with other methods if the system is to recognize such actions. Although ZeroTouch can detect the positions of actions without a camera, it cannot flexibly change areas where the system can recognize user actions.

There is a system apart from actions with hands that can detect the location where a ball has hit a table. PingPongPlus [2] is a digitally enhanced ping-pong game. When a ball hits the table, the sound travels through the table at roughly twice its speed in air, and eight microphones mounted on the underside of the table pick up the sound. When a microphone detects a hit, a time value is assigned to that microphone. The time values are evaluated by an algorithm that determines the location of the hit. This system is similar to our system in that multiple sensors are installed on the surface. However, since it can only be used for hard surfaces where the high frequency waves travel easily, it cannot be used with the soft surfaces such as fabrics.

There have also been studies on recognizing various user actions other than touch. BYU-BYU-View [7] enables users to interact with virtual environments on a screen through their own breath and wind emissions from the device. It uses a screen made from a wind-permeable material. Wind passes through this screen, behind which is an array of wind-input detection sensors. Jellyfish Party [8] is a Mixed Reality installation using a head-mounted display to create virtual soap bubbles and jellyfish in real space in response to breath input. It uses a device equipped with a spirometer to create bubbles in response to user breathing. livePic [9] and Thermo-Tablet [10] detect both touch and breath by using a thermo-infrared camera located behind the surface. However, fabricators who set the interactive system by using BYU-BYU-View cannot project high resolution pictures because its screen has to be made of coarse material that can let wind pass through it. Users need to hold the input device for the system in their hand in Jellyfish Party, which is not suitable for public spaces. Although livePic and Thermo-Tablet can not only recognize breathing but also touch the same as our system can, the thermo-infrared camera is too expensive to be used casually.

It is important for this type of system to not only recognize the occurrence of touches but also the detailed properties of touch such as intensity. WrinkleSurface [11] is a touch panel that recognizes multi-touches. It consists of a silicone rubber panel, an acrylic panel as its base, and LED modules. The system uses elastic material to detect wrinkles that are made by touching the panel. Touchë [12] recognizes gestures on the surface and utilizes Swept Frequency Capacitive Sensing to recognize human hands and body configurations. It can detect the number and shapes of fingers that are used to touch the surface. As previously stated, if the number of recognizable input types increases, the expression of interactive arts can be enhanced. However, it is difficult to change areas where the system can recognize user actions because WrinkleSurface needs to have a the special device (which takes a lot of time to generate) placed onto the surface. Touchë requires conductivity for objects where the method can detect actions. If fabricators want to apply it to non-conductive objects, they have to coat the objects with conductive ink or tape.

As we previously mentioned, these systems achieved novel recognition functions. However, since these recognition mechanisms were specialized for their own systems, they could not be applied to other artworks. In other words, professionals in activity recognition are required to construct such interactive systems in related works since appropriate settings for recognition and algorithms vary according to different situations. We created a generalized framework of activity recognition in this study for constructing interactive surfaces in a variety of situations, and implemented software thar enables artists or general people to manage recognition settings without expert knowledge. In addition, one of the most important problems for such interactive systems is in the maintenance of artworks. Usually such systems need to be maintained correctly by adjusting sensors or fixing problems in changing situations. However, since experts are not always ready to fix problems, these systems should have functions for maintenance. Our system has maintenance functions such as automatic calibration to determine the threshold, error notification when problems in sensors occur, and the addition/modification of gestures to be recognized by the system. These functions allow general users to construct and maintain interactive surfaces without experts.

3 Proposed System

Our aim in this study was to implement an interactive system that could easily be installed in various environments by a fabricator who was not familiar with activity recognition. The proposed system recognizes user actions that sway the surface of a target object.

3.1 System design

Principles

We designed our proposed system with seven principles in mind:

- **No wearable devices:** When using the system in public, people should not have to wear any devices to interact with the system.
- **No cameras:** It is not practical to place cameras in most public spaces.
- **Invisible to audience:** Since exposed devices/ cables might spoil the image of a showpiece, the

devices in the system should be invisible to visitors.

- **Scalability:** The system should target surfaces of various sizes.
- **Various materials:** The system must be able to be installed in various types of materials such as fabric, paper, and thin board.
- **Easy maintenance:** Even an amateur who is not familiar with sensors should be able to maintain the system.
- **Integration with other methods:** There are various conventional methods (e.g., a position detection methods using cameras or gesture recognition methods using Swept Frequency Capacitive Sensing) that are specialized in detecting specific activities with high level of accuracy. It should be possible to integrate our method with such methods to improve recognition accuracy.

Targets for recognition

Our system recognizes three activities that are important for interactive systems.

1. **Kinds:** The system classifies types of actions such as patting, pushings and breathing on the surface.
2. **Intensity:** It recognizes intensity at several granularities.
3. **Positions:** It detects positions where users perform actions.

Our system uses multiple accelerometers on the surface of an object that acquires actions and recognizes them by time-series data from the accelerometers. If a user touches the surface, it sways. By attaching an accelerometer to the surface, the system can detect the swing. By using the difference of each acceleration data, the system can recognize activities stated above. Figure 1 shows an example of acceleration data when a user touches a fabric whose top and bottom edges are fixed to the ceiling and floor of a room. Four accelerometers are attached to the four corners on the fabric in the form of a rectangle (1000 × 950 mm) in these examples. These graphs provide one-axis acceleration data for each accelerometer whose direction is front-to-back. The graph at left indicates that we touched the upper left of the rectangle, and that on the right indicates that we touched the upper right of the rectangle. We touched at the timings indicates by the vertical line, which

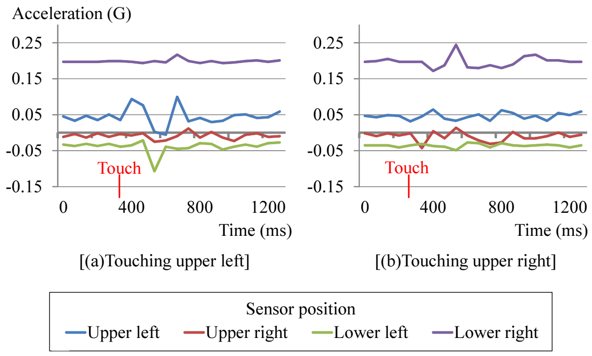


Figure 1: Examples of acceleration data when touching fabric.

crosses with each horizontal line in the graph. The waveforms indicates that the sensors placed near the positions of the touches detect stronger waves. This means that the positions and intensities of touches can be detected from the data of multiple accelerometers.

In addition, our system can be installed in existing environments since the layout of the accelerometers is not restricted. This means that artists do not need to consider any design restrictions caused by the system in recognizing user actions. This characteristic creates an advantage in that artists and the system engineers can create/design showpieces individually.

Functions for easy setting

Since the recognition methods in conventional systems are specialized for their own work, specialists in activity recognition are required to construct the systems. In addition, conventional systems take a long time in selecting recognition algorithms and in setting the recognition parameters. Therefore, the proposed system gathers acceleration data when fabricators perform actions that they want to recognize multiple times, and the system creates a classifier by using machine learning. Fabricators can easily construct an interactive system with our system. In addition, the system allows them to easily add new kinds of actions and install it in other environments. We employed a decision tree (DT) and a support vector machine (SVM) as context recognition algorithms.

3.2 System structure

Figure 2 outlines the structure of our system. It consists of three-axis accelerometers, a PC, microcomputer(s), and a surface that perceives user actions. The surface can be made from various types of materials such as fabric, paper, and film. The

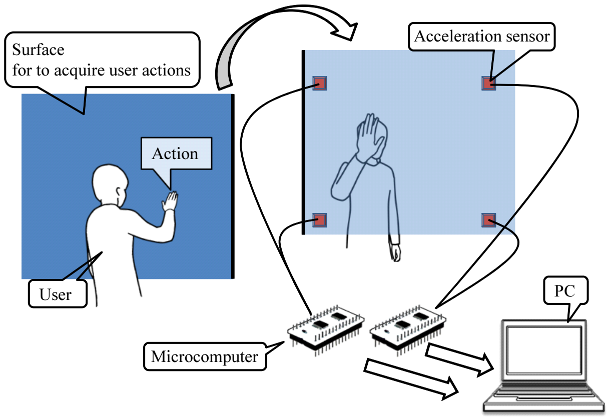


Figure 2: System structure.

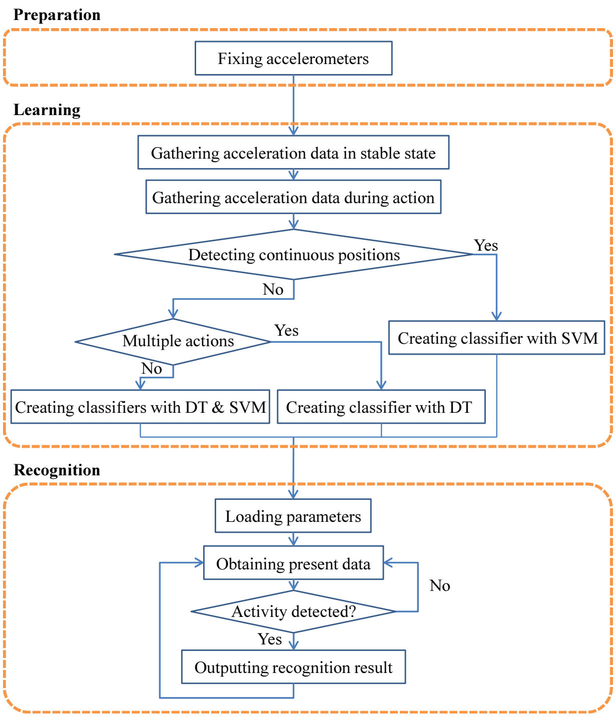


Figure 3: Flowchart for proposed method.

accelerometers are installed on its back. The number and the positions of the accelerometers can be determined according to the situation (e.g., location restrictions of the showpieces or characteristics of the materials). The sensed data are sent to the PC through the microcomputers.

3.3 Recognition method

Figure 3 shows the processing flow for the proposed method which is divided into three steps: Preparation, Learning, and Recognition.

Preparation

The system prepares to gather the learning data in this step. First, a fabricator installs multiple accelerometers on the target surface. Next, he/she determines the type of recognition; i.e., area or linear. The former means that the surface is divided into several areas and the system outputs the area where the user performed an action. The latter plots the point (x, y) where the user performed an action.

Learning

The system gathers the learning data in this step, determines the recognition algorithm, and selects the feature values to be used in activity recognition. First, the system gathers the data in a stable state. Although the input surface sways when a human gets close, the system needs to determine if he/she is performing an action.

Next, the system gathers acceleration data as learning data when a user actually performs actions on the surface. He/she performs an action in the area type in each area several times (ten or more times are recommended). He/she performs an action in the linear type at nine points to calculate the coordination shown in Figure 4.

After the data are gathered, the system calculates the feature values and creates the classifier. The system calculates the threshold of acceleration data, which it recognizes at the start of user action using the maximum and minimum values of the data in the stable state. The system calculates feature values with one-second acceleration data from the start of user action. Three are to types of feature values, including mean, variance, and crossingcounts (complete list of feature values is given in Appendix).

The system constructs a classifier by using these feature values. If the fabricator wants to recognize multiple kinds of action in the area type, the system creates a DT. The system in this research created a DT with J48graft [14]. If he/she wants to recognize one kind of action in the area type, the system create a DT and SVM and chooses the better of the two in the results from cross validation of the learning data. The system in this research used SMOreg [15] as the SVM. Since our method using SMOreg as SVM could only detect positions, the system could not recognize multiple kinds of actions in the area type by using SVM. If the fabricator wants to recognize one kind of action in the linear type, the system creates an SVM.

When the system uses DT, it calculates the feature value of the node and pursues the edges of the DT in turn. We will next describe how to treat SMOreg. The system calculates coefficients for every sensor with SMOreg, and calculates distances that are from a sensor to a point where a user would perform an action by using their coefficients. The point is expected to be near a position that is on a circumference whose radius is the calculated distance and whose center is every sensor. Therefore, the system determines that a recognition point is a point where the variance in values that is divided by the distance between an optional point and every sensor by using the calculated distance that is the shortest. In this way, the system can detect a certain point in a continuous area. The system does not use the feature values that are compared among sensors because it calculates the distance from each sensor. The proposed system cannot presently handle multiple kinds of action in the linear type because coefficients calculated with SMOreg cannot classify multiple kinds. When the system uses SMOreg in the area type, it determines a point that is the nearest of all calculated points as the result.

Recognition

After a recognition model is constructed , the system is used to recognize actual user actions. First, the system loads the parameters for recognition created in the previous step, and starts to obtain the acceleration data. If the acquired data exceed the threshold, the system starts to process for recognition by calculating the feature values.

After the recognition process is finished, the system outputs the recognition result so that other software can also use the result through a communication method such as OSC and UDP.

4 Implementation

We implemented a prototype of the calibration software explained in Section 3.3. The software has three modes, i.e., Preparation, Learning, and Recognition modes.

Preparation mode

The system prepares to gather the learning data in this mode. Figure 4 has a snapshot of this mode being used. After multiple accelerometers are installed on the target surface and the microcomputers are connected to a PC, the user activates the software. First, he/she inputs the size of the surface in the console window. If he/she is using it in the area type, he/she also inputs the number of point that he/she wants to recognize. The system sets nine points by default in the linear type. The system draws circles in the windows of Figure 4 that are positions where he/she wants to perform actions, which can be moved by dragging.

Figure 4: Snapshot of software's Preparation mode.

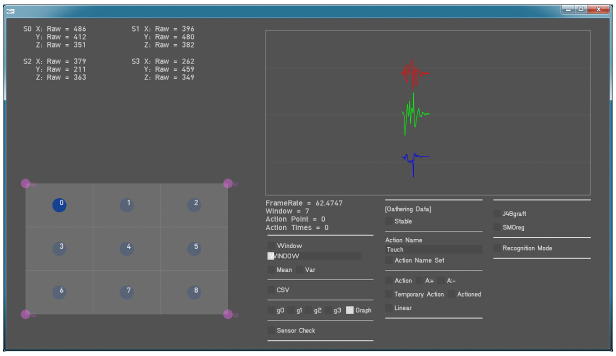


Figure 5: Snapshot of software's Learning mode.

Learning mode

The system gathers the learning data in this mode, and creates a classifier. Figure 5 has a snapshot of this mode being used.

The system displays a graphical user interface (GUI) to control the settings for learning in the bottom half of the figure. The top left part draws the current data on accelerometers. There are several buttons in the bottom right of the figure labelled Mean, Var, Window, CSV, g0, g1, g2, g3, Graph, Sensor Check, Stable, Action Name Set, Action, A+, A-, Temporary Action, Actioned, Linear, J48graft, SMOreg and Recognition Mode, a slider labelled WINDOW, and a text box above the "Action Name Set" button. If a user push as the "Mean" or "Var" button in the left column of the bottom right GUI, the system draws the mean values or variance values of data under the current data. The window width for calculating mean and variance values is changed by using the "WINDOW" slider in the state of the pushed "Window" button. If the user pushes "CSV", the system saves the current data and the current time in the CSV format. If the system is in a state where the “Graph” button is pushed, it draws a graph of the current data at the top right. The drawn data are from the three dimensional data of one sensor. By pushing the “g0”–“g3” radio buttons, the system converts the sensors for drawing the graph. Additionally, the system has a “Sensor Check” button, which checks whether sensors are malfunctioning. If the system receives data from malfunctioning sensors, it raises an alert and does not receive data from the malfunctioning sensor until it is repaired. The functions mentioned above are not needed to create the classifier, but the user can use them for reference.

The user creates the classifier by using the following flow. First, the system gathers data in a stable state. The system records data for the stable state when the “Stable” toggle button is pushed.

The system next gathers data when the user is performing an action. He/she selects one of the action points by clicking one of the points graphically displayed at the bottom left of Figure 5. The action name is set by typing a name into the text box in the center column of the bottom right GUI (It is labelled as “Touch” in Figure 5). The system records data as the action that was named when the “Action” button is pushing. If the user cannot identify the action point by the time the action ends (e.g., the action of throwing a ball), he/she can set the point and name the action after performing it by using the “Temporary Action” button and “Actioned” button. If the user has an action that he/she does not want to recognize, he/she should just type “None” in the text box. The system ignores recorded actions labelled “None”. The system can record data any number of times.

After the action is recorded, the system calculates the feature values and the user creates the classifier by pushing the “J48graft” or “SMOreg” button. All the feature values that we implemented in this research are provided in the Appendix. The system uses Weka [13], which is a popular suite of machine learning software to create the classifier.

The system createsa DT by using the J48graft of Weka when the user wants to recognize areas, the kinds of actions, or their intensities. Also, the system creates an SVM by using the SMOreg of Weka as an SVM when the user wants to recognize the positions.

The system moves into Recognition mode when the “Recognition Mode” button is pushed.

Recognition mode

The system in this mode recognizes the action by using the classifier created in the Learning mode. Figure 6 has a snapshot of the Recognition mode (UDP Mode) being used.

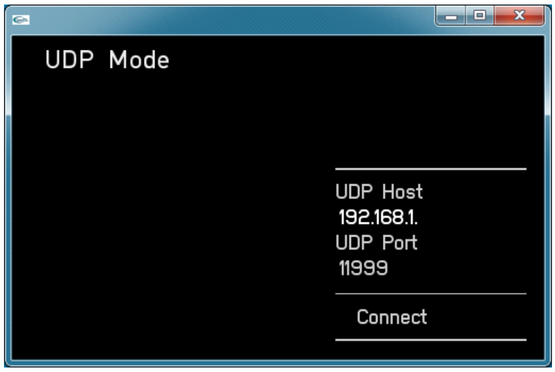


Figure 6: Snapshot of software’s Recognition mode.

The recognition results (the position and the name of the detected action) can be desplayed on our software or transmitted via UDP or OSC. After they are selected, the system loads the classifier and the threshold values from the data in the stable state. Then, the system starts to obtain the acceleration data, and starts to recognize action if the data is over the threshold value.

The system in this mode checks whether sensors are malfunctioning at fixed intervals. If it finds any errors, it displays an alert. It this happens, the user enters Gathering & Learning modes again or repairs the sensor.

We implemented two applications that interacted with receiving output.

Figure 7 has a screenshot of the sound output application. The circles indicate action points, and the user moves sound files via drag-and-drop operations into the circles. If the application receives the recognition data through UDP or OSC, it starts the file that has been assigned.

Figure 8 has a snapshot of the function output application. The user moves function names via drag-and-drop operations into the circles. If the application receives the recognition data through UDP or OSC, it activates the assigned software or executes the assigned command key.

5 Evaluation

We carried out two experiments by using the implemented software where the systems of both consisted of a note PC (CPU: Core i7 2.80 GHz, and RAM of 8 GB), four accelerometers (#KXM52-1050 XYZ ± 2 G), and two microcomputers (Arduino Nano). The software received the data at 60 Hz.

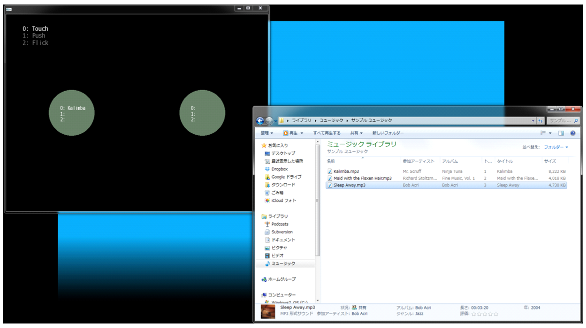


Figure 7: Screenshot of sound output application.

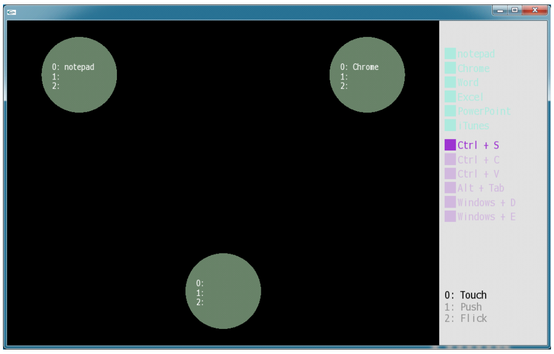


Figure 8: Snapshot of function output application.

Table 1: Sizes of each material and rectangle

Material	Size	
	Material (mm)	Rectangle (mm)
Nylon (top)	1500 x 1170	1000 x 1150
Nylon (bottom)		1000 x 1150
Cotton	1500 x 900	940 x 870
Quadrilateral table	600 x 900	530 x 820
Corkboard	450 x 600	420 x 570
Paper box	250 x 190 x 165	-

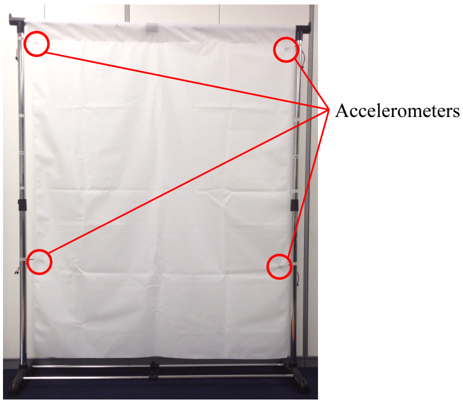


Figure 9: Fabric made of 100% nylon.

5.1 Use of several materials

5.1.1 Experimental purposes

We evaluated whether the proposed system worked well in the first experiment for several materials that were soft fabric, hard board and so on because our aim was to develop a system that could be set interactively onto several materials. We investigated the recognition rates in cases where the proposed method was used with several materials while the feature values were calculated automatically. In addition, we explored what kinds of characteristics the system had for each material. We considered how we could upgrade the proposed system from the results.

5.1.2 Experimental setup

We have prepared a fabric (made of 100% nylon), a fabric (made of 100% cotton), a quadrilateral table, a corkboard, and a paper box. The cotton was harder fabric than that made of nylon. We installed four accelerometers on them in the form of a rectangle. All input areas were inside the rectangles containing the four accelerometers. Table 1 lists the sizes of the materials and rectangles. We hanged the nylon and the cotton on a hanger rack. Figures 9 - 14 are photographs that show how the accelerometers were installed onto them. The accelerometers were sewn onto the nylon and the cotton fabric. The accelerometers were attached to the quadrilateral table, the corkboard, and the paper box with vinyl tape. The accelerometers were attached to the center of each of the paper box's sides (Figure 14) only for the paper box, and we performed actions on four surfaces. After the accelerometers were installed, we performed various kinds of actions and points. Data where we performed actions ten times per action and per point were regarded as one data set. A data set was used to create a classifier. We calculated the recognition results from other data sets based on this classifier, and we computed the accuracy rate and average error. We repeated this calculation for all data sets.

5.1.3 Experimental results

Nylon, cotton, table, and corkboard
Nine points and one kind (patting)

We performed actions on nine points where each

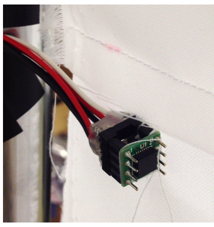


Figure 10: Sewing accelerometers onto nylon fabric.

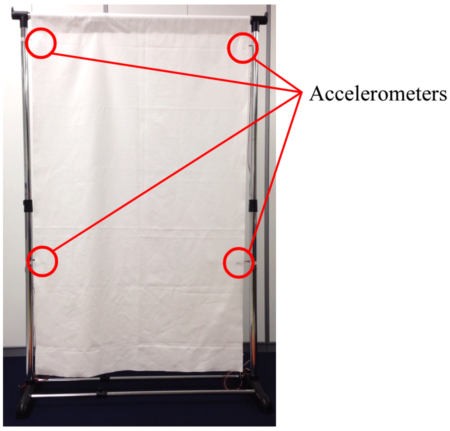


Figure 11: Fabric made of 100% cotton.

Figure 12: Accelerometers attached to table.

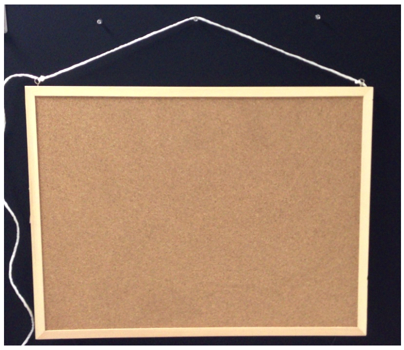


Figure 13: Corkboard.

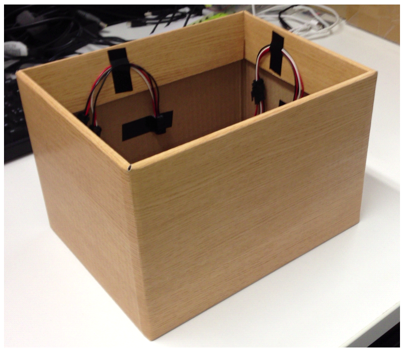


Figure 14: Paper box.

center part divided the input area into nine zones. The materials were nylon (both at the top and bottom part), cotton, the quadrilateral table, and the corkboard. The first kind of action was a patting. We gathered six data sets (total of 540 actions). We gathered five data sets (total of 450 actions) only for the table.

Table 2 summarizes the results for the accuracy rate, and Table 3 summarizes them for average error.

For example, the top-left value (65.6) in Table 2 lists the accuracy rate when the classifier was used (created from the first data set by using J48graft) for data in the other five data sets (total of 450 action data). The value that is under it (48.2 in the table) is the accuracy rate when SMOreg was used. The system regarded a point that was the closest to each action point (nine points in this case) as a recognized point. The top-left value (259.5) in Table 3 is the average error (i.e., average distance from a correct point to a calculated point) where the classifier was used (i.e., created from the first data set by using SMOreg) for the data in the other five data sets.

The accuracy rates, in all cases, for J48graft are better than those for SMOreg. The average for all accuracy rates is 66.3% for nylon (top). Most of the misrecognized points adjoined a correct point. The system misrecognized vertical points, and recognized correctly horizontal points. The accuracy rate calculated for the sixth data set based on the classifier created by the sixth data set had a low value. It appeared to be the cause of the poor accuracy rate. Its average was 58.0% for nylon (bottom). The results ware higher when the accelerometers were sewn onto the top. Most of the misrecognized points adjoined correct points, and were vertical in similar to the points at of the top. The average for cot-

Table 2: Result for accuracy rate (9 points and 1 kind of action (patting))

Material	Classifier	Accuracy rate (%)						
		1	2	3	4	5	6	Ave.
Nylon (top)	J48graft	65.6	65.8	72.2	69.8	70.0	54.7	66.3
	SMOreg	48.2	46.0	51.6	47.3	50.9	45.3	48.2
Nylon (bottom)	J48graft	58.4	60.4	63.8	55.8	50.4	59.1	58.0
	SMOreg	38.0	37.3	30.0	22.9	34.2	38.4	33.5
Cotton	J48graft	53.6	64.7	66.9	54.2	42.0	56.0	56.4
	SMOreg	32.2	34.2	36.4	31.8	30.0	22.2	31.1
Table	J48graft	19.7	27.8	31.9	33.1	27.5		28.0
	SMOreg	22.5	21.4	29.7	24.4	22.2		24.0
Corkboard	J48graft	25.8	191.1	36.0	33.3	41.6	35.6	31.9
	SMOreg	23.8	20.9	19.1	22.0	22.0	22.9	21.8

Table 3: Results for average error (9 points and 1 kind of action (patting))

Material	Average error (mm)						
	1	2	3	4	5	6	Ave.
Nylon (top)	259.5	276.7	246.2	246.2	235.9	247.4	252.0
Nylon (bottom)	307.6	319.6	377.0	324.4	312.0	315.0	328.9
Cotton	284.6	312.5	300.0	282.3	268.0	333.1	296.7
Table	295.2	259.8	231.0	246.0	288.1		264.0
Corkboard	193.7	178.4	189.4	186.2	200.8	196.1	191.0

ton was 56.4%. The results for when accelerometers were sewn onto the top of the fabric made of nylon ware better than those for cotton. The accuracy rate that was calculated for the fifth data set based on the classifier created with the fifth data set was low, and the fifth data set results were poor. The average for the table was 28.0%, which was relatively low. The system did not recognize the top-left point in many cases. The accuracy rate calculated for the first data set based on the classifier created with the first data set was low, and the first data set results were poor. The average for the corkboard was 31.9%. The accuracy rate calculated for the first data set based on the classifier created with the first data set was low, and the first data set results were poor. The second data set was also poor. It was difficult to achieve accurate recognition because the input area was too small. Moreover, the corkboard was the hardest material used at this evaluation. The system correctly recognized points vertically in many cases.

The average of all average errors was 252.0 mm for nylon (top) when considering the results for average errors. The average error needed to be less than 100 mm because the length of a hand that performs actions is approximately 200 mm. The dispersion of values was low, but a value of approximately 800 mm was calculated on rare occasions. The average for nylon (bottom) was 328.9 mm. The average errors for the bottom were greater those for the top. The average for cotton was 296.7 mm. The results for nylon were better than those for cotton and similar to J48graft. The average for the table was 264.0 mm. The dispersion of values was low. The average for the corkboard was 191.0 mm, which was better than that for nylon because it was smaller.

Nylon and cotton

Four points and two kinds (patting and pushing)

We performed actions at four points where each center part divided the input area into four zones. The materials were nylon (both at the top and bottom) and cotton. The kinds of actions were patting and pushing. We have gathered six data sets (total of 480 actions).

Table 4 summarizes the results. The average of all accuracy rates for nylon (top) is 77.3%. The system misrecognized kinds of actions, but correctly recognized positions. The accuracy rate calculated for the third data set based on the classifier created with the third data set was low, and the third data set results were poor. The average for nylon (bottom) was 60.5%. The system misrecognized positions, and

Table 4: Results for accuracy rate (4 points and 2 kinds of action (patting and pushing))

Material	Accuracy rate (%)						
	1	2	3	4	5	6	Ave.
Nylon (top)	71.8	89.5	64.8	86.5	71.5	79.5	77.3
Nylon (bottom)	60.3	58.3	57.0	59.5	55.5	72.5	60.5
Cotton	51.5	64.5	68.3	73.2	77.8	51.0	64.4

correctly recognized kinds of actions unlike the nylon at the top. The average for cotton was 64.4%. The results for where the accelerometers were sewn onto the top of the fabric made of nylon was better than those for cotton. The system did not recognize the top-left point in many cases.

Table
Four points and two kinds (weak and strong patting)

The material was the table. We performed actions at four points where each center part divided the input area into four zones. The kinds of actions were weak patting and strong patting. We gathered six data sets (total of 480 actions).

Table 5 lists the results. The average of all accuracy rates was 46.1%. The misrecognitions were not particularly regular, but some of them correctly recognized kinds of actions.

Corkboard
One point and two kinds (patting and pushing)

The material is the corkboard. We performed actions at a point that was at the center of the corkboard. The kinds of actions were patting and pushing. We gathered six data sets (total of 120 actions).

Table 6 summurizes the results. The average of all accuracy rates is 83.1%. The accuracy rate calculated for the fourth data set based on the classifier created with the fourth data set was low, and the fourth data set results were poor.

Corkboard
Two points and two kinds (patting and pushing)

The material was corkboard. We performed actions at two points that was at the each center of part that is divided the input area into two zones from side to side. The kinds of actions were patting and pushing. We gathered six data sets (total of 240 actions).

Table 7 lists the results. The average of all accuracy rates was 71.7%. The system misrecognized positions, and correctly recognized kinds of actions.

Table 5: Results for accuracy rate in recognizing table (4 points and 2 kinds of actions (weak and strong patting))

Accuracy rate (%)						
1	2	3	4	5	6	Ave.
54.0	43.5	45.8	44.0	44.5	45.0	46.1

Table 6: Results for accuracy rate in recognizing corkboard (1 point and 2 kinds of actions (patting and pushing))

Accuracy rate (%)						
1	2	3	4	5	6	Ave.
87.0	85.0	87.0	61.0	89.0	90.0	83.1

Table 7: Results for accuracy rate in recognizing corkboard (2 points and 2 kinds of actions (patting and pushing))

Accuracy rate (%)						
1	2	3	4	5	6	Ave.
67.0	59.5	88.0	66.0	74.5	75.0	71.7

Paperbox
Eight points and one kind (patting)

The material is the paper box. We performed actions at eight points where the center part divided each side into two zones from side to side. The kind of action was patting. We gathered six data sets (total of 480 actions).

Table 8 lists the results. The average of all accuracy rates was 22.1%. The accuracy rate calculated for the third data set based on the classifier created with the third data set was low, and the third data set results were poor. The misrecognitions were scattered.

Table 8: Results for accuracy rate in recognizing paper box (8 points and 1 kind of action (patting))

Accuracy rate (%)						
1	2	3	4	5	6	Ave.
20.8	25.5	15.8	27.5	30.8	12.5	22.1

Table 9: Results for accuracy rate in paper box (4 points and 3 kinds of actions (patting, pushing, and lifting))

Accuracy rate (%)						
1	2	3	4	5	6	Ave.
41.7	23.5	47.7	23.7	25.2	44.7	34.4

Paper box
Four points and three kinds (patting, pushing, and lifting)

The material was a paper box. We performed actions at four points that were the centers of each side of the paper box's. The kinds of actions were patting, pushing, and lifting. We gathered six data sets (total of 720 actions).

Table 9 lists the results. The average of all accuracy rates was 34.4%. The accuracy rate calculated for the second data set based on the classifier created with the second data set was low, and the second data set results were poor. The fourth and fifth data data set results were also poor.

5.1.4 Consideration

We found from the results of evaluating the five materials that the proposed method was more effective for soft material. The system could not recognize many points in the hard material. Therefore, we need to improve our proposed method for hard materials like table tops.

There were some cases that the accuracy rate (i.e., calculated for one data set based on the classifier created with the own data set) for their data become lower. For that reason, the system needs to calculate the accuracy rate that uses the classifier created with its own data set, and raise alerts to gather learning data again if the rate is too low.

There are also situations where the system does not recognize certain points at all. If the system cannot recognize certain points at all in the recognition step, it needs to raise alerts to return to the learning step.

5.2 Setting interactive system

5.2.1 Experimental purpose

We carried out an experiment on system construction in the second experiment because it was important for the fabricator to construct an interactive system easily in this study. We investigated how long a fabricator took to establish an interactive system and whether he/she could do this effectively by using the proposed system. We also examined what problems occurred during the setting process. We found areas we could improve with the proposed system from the results.

5.2.2 Experimental setup

Participants set up an interactive system that was designated by the author after the implemented software was explained to them. The target system should have recognized two kinds of actions (patting and pushing) that were performed at two points on the nylon fabric used in Section 5.1. The fabric had four accelerometers sewn onto it with thread. The participants were four males in their 20s, and they were labelled "P1 - P4". P1 was the author. P3 was familiar with the characteristics of accelerometers, while P2 and P4 were not. We have measured the time from when the participants started to install the accelerometers when they activated the software, which was recorded as "T1". In addition, we measured the time from when the participants started to gather the learning data to when they activated the recognition mode, which was recorded as "T2". After they had finished the settings, they performed each kind of action at each point ten times. We counted the number of correct answers out of a total of 40.

5.2.3 Experimental results

Table 10 summarizes the results. The units of time in the results were minutes and seconds (i.e., "minutes : seconds"). The interactive systems that were set up by P2 and P4 always output the results notwithstanding no action. Therefore, they set the system again. P2 set it from scratch again. Because P4 had only fixed the lead cables with vinyl tape in 2 minutes and 43 seconds, his T1 was recorded as "+2:43". The "T1+T2" of P4 (2nd) was the time that was the T1 of P4 (1st) plus the T2 of P4 (2nd) plus 2:43.

The average of all the participants times was 47 minutes and 43 seconds. The average of all the correct answers was 28 (70%). The average time except for P1 (author) was 51 minutes and 22 seconds. The

Table 10: Results for setting interactive system

Participant	Time			Number of correct (/40)
	T1	T2	T1+T2	
P1	31:10	5:35	36:45	34 (85%)
P2 (1st)	22:20	6:17	28:37	-
P2 (2nd)	50:02	5:49	55:51	30 (75%)
P3	45:30	9:37	55:07	23 (57.5%)
P4 (1st)	34:10	7:20	41:30	-
P4 (2nd)	+2:43	6:16	43:09	25 (62.5%)

average number of correct answers except for P1 (author) was 26 (65%). Almost all the results for misrecognition by P1, P2, and P3 were related to kinds of actions (i.e., almost all recognized positions were correct answers).

The participants made some comments in questionnaires that they answered freely. P3 said that the system should sound whenever a fabricator performed actions as feedback, and the sounds should differ according to each kind of action. P2 and P3 said that it was hard for them to install the accelerometers without tilting them. Note that none of the participants said they could not construct such an interactive system without the proposed system.

5.2.4 Consideration

P2 and P4 had not been able to set the interactive system at first. It seems that why they failed was because they roughly installed the accelerometers. They sewed them too loosely. Therefore, the state of the accelerometers changed, and the acceleration data were always over the threshold value. Moreover, all the participants took too long to install the accelerometers. To resolve these problems, we need to make an attachment that will help to install accelerometers onto surfaces. In addition, the system needs to raise alerts when the states of accelerometers change, not just when accelerometers malfunction.

The number of correct answers P3 gave was low. It seems that there were for differences between patting and the pushing. Therefore, the system needs to raise alerts if the cross validation of learning data is poor.

6 Applications

We had exhibited two different showpieces using our system in two long-term media-arts exhibitions. The system consisted of a desktop PC (CPU Intel Core 2 Quad 2.83 GHz with 3.25 GB of RAM), four accelerometers (#KXM52-1050 XYZ ±2 G), and two microcomputers (Arduino Nano).

Figure 15: Snapshot of 34°_41.38’N 135°_30.7’E.

6.1 34°_41.38’N 135°_30.7’E

“34°_41.38’N 135°_30.7’E” was the title of a work of art. This installation art filled an entire room. We created the showpiece together with Kazunari Sako, who is a conceptual artist. The exhibition was held from August 10 to September 2, 2012 in the city of Kobe in Japan. Figure 15 has a snapshot of the installation art. The two screens in the figure depict of a newspaper page projected from two rear projectors. When a visitor breathes onto the screen at left, (Figure 16), characters, figures, and the lines around the point of breathing move to the screen at right, whose coordinates are the same (Figure 17).

The system for this installation was required to recognize both breathing actions and their intensities and positions. In addition, since the system projects approximately 12,000 objects, such as characters in the 1920 × 1080 pixels, and visual effects were applied to individual characters, the characters needed to be recognized one by one. For that reason, the system needed to precisely detect the positions where visitors breathed. However, the proposed method only detected approximate positions that visitors breathe onto. Therefore, the system precisely detected the positions with a method using a depth sensor (ASUSTek Xtion PRO LIVE [16]). Although the system could not recognize when visitors breathed onto the screen, it could detect the positions and the timing of action by integrating both methods. We tested and confirmed that the proposed method could extend conventional methods, and integration could be easily achieved by constructing this application.

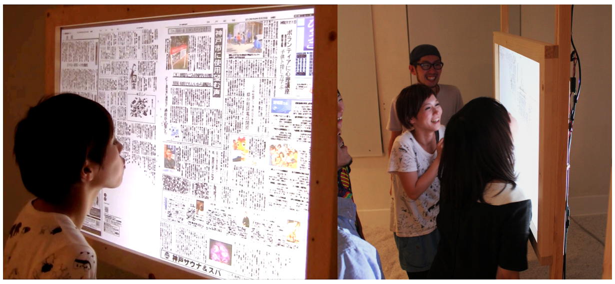


Figure 16: Breathing onto the screen.

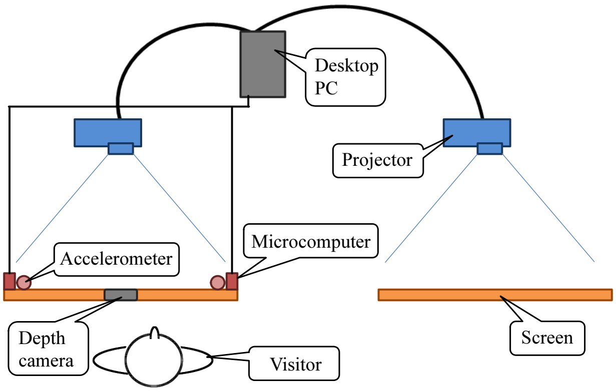


Figure 18: Structure of 34°_41.38'N 135°_30.7'E.

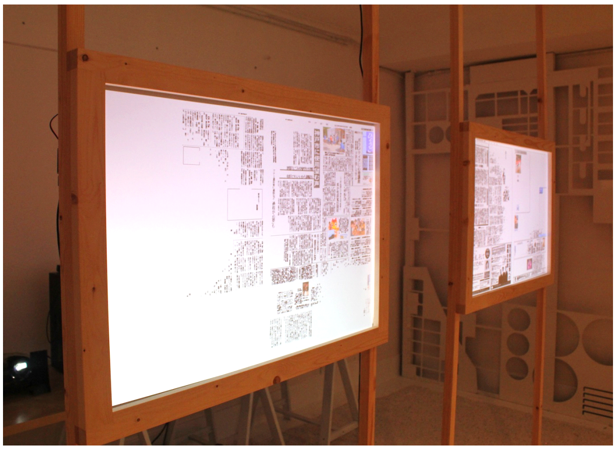


Figure 17: Characters moving from left screen to right screen.

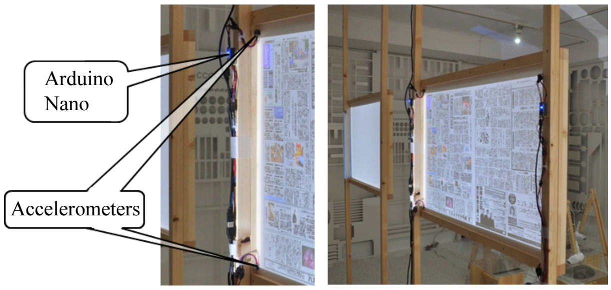


Figure 19: Back of screen.

6.1.1 System structure

The system structure is outlined in Figure 18. We used Theaterhouse #TPW1200TK+60 for the two screens, which is a material for both front and rear projections, and placed the screens in frames that were the size of a Japanese newspaper page (540 × 810 mm). An image was projected onto the screen from the rear, and the accelerometers were installed on the four corners of the back of the screen with a bonding agent, as shown in Figure 19. The system could detect breathing actions and their intensity with the accelerometers. Additionally, it can detect the approximate positions of actions. However, since this showpiece required accurate positions of breathing, we used an additional approach with a depth camera to accurately detect the positions of visitor heads. Our system acquired the height of the top of the head from the depth camera, and it then output the position of the mouth 200 mm below the top of the head. The value of 200 mm was determined on the basis of a preliminary experiment and was the average of differences between the top of the head and the position of the mouth. In addition, the system output the horizontal position of the mouth as the center of gravity of the head acquired from the depth camera. We used Xtion PRO LIVE as the depth camera (see Figure 20).

6.1.2 Action recognition

When visitors moved their heads close to the screen, their actions created fine waves that the accelerometers detected. We established data where visitors got close to the screen as a stable state. Furthermore, the system recognized the intensity of breathing from the amplitude of the detected waves.

When the system recognized an action, the characters, figures, and lines around the mouth's position dispersed. The radius of dispersion was determined by the intensity of breathing.

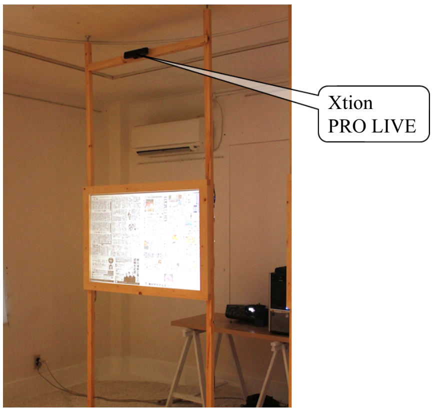


Figure 20: Setting of depth camera.

6.1.3 Discussion

Many visitors had fun with the breathing aspect: they enjoyed the phenomenon of viewing what they could not see in daily life that the interactivity provided.

The action of breathing can be recognized by BYU-BYU-View [7] or by installing a large number of sensors on the screen. However, if the showpiece used these methods, the image on the screen was often coarse, or the sensors were visible as part of the screen. The resolution of the image remained fine grained by using our method, and visitors could read the characters of the newspaper on the screen. Moreover, our system could be installed more inexpensively than livePic [9] or ThermoTablet [10].

Although the proposed method only detected approximate positions, it can be combined with other systems that detect positions more precisely as was previously stated. We tested and confirmed that our method can be integrated with other methods that detected user positions by using a depth camera.

6.2 White Parallel Small Space

"White Parallel Small Space (WPSS)" is an installation art piece filling an entire room at the Designer Show House, which is an event at which artists repair and arrange the rooms of an old building. We created a showpiece together with Makiko Issha, who is an interior coordinator. The exhibition was held from October 13 to November 4, 2012 in the city of Osaka in Japan. Figure 21 has two snapshots of our showpiece. Two projectors projected countless white circles and three images onto drapes that had been hung from wall to wall, as seen in Figure 22. The countless white circles slowly dropped, like snowflakes. When a visitor touched any of the three images (Figure 23), the image he/she touched moved right, changed to a video clip, and was played. Here, the image is a thumbnail of the video clip.

Figure 21: Snapshots of WPSS.

Figure 22: Three images on drapes.

Figure 23: Touching image.

The system for the installation needed to recognize which of the three images had been touched. We could not install any cameras due to space and concept restrictions. This is because there was no space behind the drapes and all devices used had to be invisible to visitors. Therefore, we used our method individually without any other approaches to identify the position at which visitors had touched the drapes.

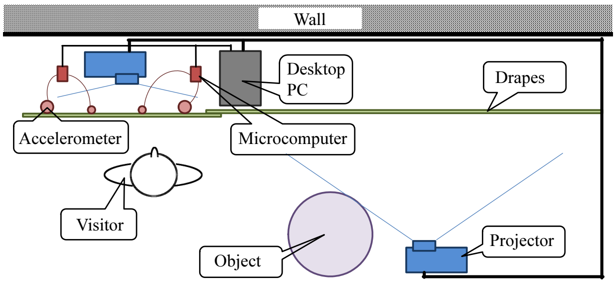


Figure 24: System structure of WPSS.

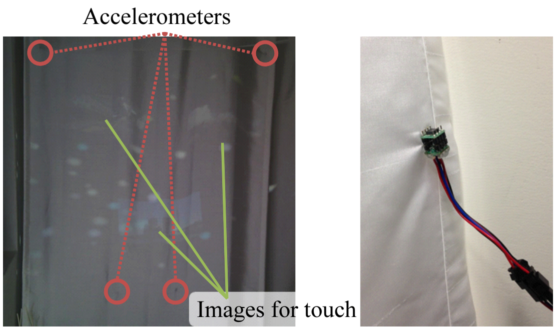


Figure 25: Sewing on accelerometers.

6.2.1 System structure

Figure 24 outlines the system structure. The projector on the left has an ultra-short focus (RICOH IPSiO PJ WX4130) and projects three thumbnails and white circles. The other projector on the right presents the movie after someone has touched one of the thumbnails. The drapes are hung from the roof and reach the floor. Four accelerometers are sewn into the drapes in the form of a trapezoid (height: 950 mm, top: 1000 mm, and bottom: 250 mm), as seen in Figure 25.

6.2.2 Action recognition

The system detects a touch action by the amplitude of the waves that travel on the surface of the drapes. Opening/closing the door of the room and wind flows create fine waves on the drapes that the accelerometers can detect. We established that the data here were in a stable state.

The system detected the touched position using the time it took for the waves to reach the accelerometers. The drapes have characteristics where the waves travel slower horizontally than vertically, which means that the differences in when they arrive horizontally at multiple accelerometers can be used to calculate the touched positions, while there is little difference when they arrive vertically. Therefore, it is easy to recognize the position by using feature values that are related to arrival timing.

All the devices are hidden from the view of visitors. Although one projector is placed in a visible area, it is obscured by an object, as shown in Figure 24.

6.2.3 Discussion

Visitors were intrigued that the drapes, which are everyday objects, became interactive. The unexpected contrivance surprised them, and delighted them as in a magic show. The system was able to detect the touched position by only using the proposed method. There were few misrecognitions, although they did occur, e.g., when visitors passed their hand across the surface and the system moved an image that they had not touched.

The room was small (1500 × 3700 mm), and the wall and the drapes were close (700 mm). Therefore, if we wanted to use a camera behind the drapes, the area where the system could recognize touching would be too narrow because the angle of the camera view was limited. The proposed method could applied to a wide area that the system can recognize. In addition, our method can obscure devices from audiences.

7 Conclusion

We proposed a system to add interactivity to art showpieces that recognized actions, their intensity, and the positions at which they had been performed by attaching accelerometers to the surface of input materials. The new system only used small accelerometers and did not require specialized devices or cameras. Moreover, the installation environment was not restricted by the size of surfaces, and our method could be integrated with other systems. We found from the results of evaluating five materials that the proposed approach was more effective with soft materials. Additionally, we tested and confirmed that the proposed system worked well through two actual cases of use and that it could be adapted to various situations and restrictions.

In future, we intend to develop a system that integrates other kinds of sensors that depends on the situation. If the system combines distance sensors where a corkboard is the input surface, it should be able to recognize that the pushed side of the board approaches the posterior wall and the other side with-draws. The recognition rate should increase if microphones are adopted in the system.

Moreover, it is likely that the proposed method will be able to recognize the degree of intensity more precisely if the system also uses SMOreg to recognize intensity. We intend to add additional functions to the calibration software.

References

[1] N. Matsushita and J. Rekimoto: HoloWall: Designing a Finger, Hand, Body, and Object Sensitive Wall, *Proc. of the 10th annual ACM Symposium on User Interface Software and Technology (UIST 1997)*, pp. 209–210 (Oct. 1997).

[2] H. Ishii, C. Wisneski, J. Orbanes, B. Chun, and J. Paradiso: PingPongPlus: Design of an Athletic-tangible Interface for Computer-supported Cooperative Play, *Proc. of the 17th SIGCHI conference on Human Factors in Computing Systems (CHI 1999)*, pp. 394–401 (May 1999).

[3] N. Isoyama, T. Terada, and M. Tsukamoto: An Interactive Surface that Recognizes User Actions using Accelerometers, *Proc. of the 12th NICOGRAPH International 2013*, pp. 72–80 (June 2013).

[4] Y. Arita, T. Miyake, N. Yamamoto, P. Ravindra S De Silva, and M. Okada: HINOCO - An Interface to Colligate the Virtual-World and Real-World, *Proc. of the 13th Virtual Reality International Conference (VRIC 2011)*, pp. 18–19 (Apr. 2011).

[5] T. Fukasawa, K. Fukuchi, and H. Koike: A Vision-Based Non-contact Interactive Advertisement with a Display Wall, *Proc. of the 5th International Conference on Entertainment Computing (ICEC 2006)*, pp. 394–397 (poster) (Sep. 2006).

[6] J. Moeller and A. Kerne: ZeroTouch: an Optical Multi-Touch and Free-Air Interaction Architecture, *Proc. of the 30th International Conference on Human Factors in Computing Systems (CHI 2012)*, pp. 2165–2174 (May 2012).

[7] E. Sawada, S. Ida, T. Awaji, K. Morishita, T. Aruga, R. Takeichi, T. Fujii, H. Kimura, T. Nakamura, M. Furukawa, N. Shimizu, T. Tokiwa, H. Nii, M. Sugimoto, M. Inami: BYU-BYU-View: a Wind Communication Interface, *Proc. of the 34th International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH 2007)* (Aug. 2007).

[8] Y. Okuno, H. Kakuta, and T. Takayama: Jellyfish Party: Blowing Soap Bubbles in Mixed Reality Space, *Proc. of the 2nd International Symposium on Mixed and Augmented Reality (ISMAR 2003)*, pp. 358–359 (Oct. 2003).

[9] M. Katsura and M. Inakage: livePic, *Proc. of the 33th International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH 2006)* (July 2006).

[10] D. Iwai and K. Sato: Heat Sensation in Image Creation with Thermal Vision, *Proc. of the 2nd International Conference on Advances in Computer Entertainment Technology (ACE 2005)*, pp. 213–216 (June 2005).

[11] Y. Sakamoto, T. Yoshikawa, T. Oe, B. Shizuki, and J. Tanaka: Constructing an Elastic Touch Panel with Embedded IR-LEDs using Silicone Rubber, *Proc. of IADIS International Conference Interfaces and Human Computer Interaction (IHCI 2012)*, pp. 21–23 (July 2012).

[12] M. Sato, I. Poupyrev, and C. Harrison: Touchë: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects, *Proc. of the 30th International Conference on Human Factors in Computing Systems (CHI 2012)*, pp. 483–492 (May 2012).

[13] Weka 3: Data Mining Software in Java, `http://www.cs.waikato.ac.nz/ml/weka/`.

[14] G. Webb: Decision Tree Grafting From the All-Tests-But-One Partition, *Proc. of the 16th international joint conference on Artificial intelligence (IJCAI 1999)*, pp. 702–707 (July 1999).

[15] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya and K. R. K. Murthy: Improvements to the SMO Algorithm for SVM Regression, *Proc. of Neural Networks, IEEE Transactions on*, Vol. 11, No. 5, pp. 1188–1193 (Sep. 2000).

[16] ASUS: Xtion PRO LIVE, `http://www.asus.co.jp/Multimedia/Motion_Sensor/Xtion_PRO_LIVE/`.

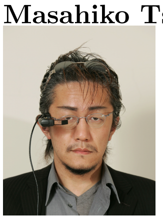
Naoya Isoyama

Naoya Isoyama is in the doctoral course at Graduate School of Engineering, Kobe University. He received B.Eng. and M.Eng. degrees from Kobe University in 2010, 2012, respectively. His research interest is wearable computing, ubiquitous computing, and entertainment computing.

Tsutomu Terada

Tsutomu Terada is an Associate Professor at Graduate School of Engineering, Kobe University. He received B.Eng., M.Eng., and Ph.D. degrees from Osaka University in 1997, 1999, and 2003, respectively. He has been an Assistant Professor at Cybermedia Center of Osaka University and a Lecture in 2000 and 2005, respectively. He is currently investigating the wearable computing, ubiquitous computing, and entertainment computing. He is a member of IEEE and IEICE.

Masahiko Tsukamoto



Masahiko Tsukamoto is a Professor at Graduate School of Engineering, Kobe University. He received B.Eng., M.Eng., and Ph.D. degrees from Kyoto University in 1987, 1989, and 1994, respectively. From 1989 to 1995, he was a research engineer of Sharp Corporation. From 1995 to 1996, he has been an Assistant Professor at the Department of Information Systems Engineering, Osaka University and since 1996, he has been an Associate Professor at the same department. He is currently investigating the wearable computing and ubiquitous computing. He is a member of eight learned societies, including ACM and IEEE.

A List of All Feature Values

Type of feature value	Axis	Window Size
mean	x, y, z, norm	1sec, 0.5sec, 0.25sec
variance	x, y, z, norm	1sec, 0.5sec, 0.25sec
maximum	x, y, z, norm	1sec, 0.5sec, 0.25sec
minimum	x, y, z, norm	1sec, 0.5sec, 0.25sec
peak-to-peak	x, y, z, norm	1sec, 0.5sec, 0.25sec
crossingcounts	x, y, z, norm	1sec, 0.5sec, 0.25sec
difference from mean of every sensor's peak-to-peak	x, y, z, norm	1sec, 0.5sec, 0.25sec
order of variance	x, y, z, norm	1sec, 0.5sec, 0.25sec
order of peak-to-peak	x, y, z, norm	1sec, 0.5sec, 0.25sec
order of crossingcounts	x, y, z, norm	1sec, 0.5sec, 0.25sec
order of difference from mean of every sensor's peak-to-peak	x, y, z, norm	1sec, 0.5sec, 0.25sec
variance of variance per five data	x, y, z, norm	1sec, 0.5sec, 0.25sec
variance of variance per ten data	x, y, z, norm	1sec, 0.5sec, 0.25sec
mean of variance per five data	x, y, z, norm	1sec, 0.5sec, 0.25sec
mean of variance per ten data	x, y, z, norm	1sec, 0.5sec, 0.25sec
order of variance of variance per five data	x, y, z, norm	1sec, 0.5sec, 0.25sec
order of variance of variance per ten data	x, y, z, norm	1sec, 0.5sec, 0.25sec
order of mean of variance per five data	x, y, z, norm	1sec, 0.5sec, 0.25sec
order of mean of variance per ten data	x, y, z, norm	1sec, 0.5sec, 0.25sec
difference of 1st and 2nd half (mean)	x, y, z, norm	1sec
difference of 1st and 2nd half (variance)	x, y, z, norm	1sec
difference of 1st and 2nd half (maximum)	x, y, z, norm	1sec
difference of 1st and 2nd half (minimum)	x, y, z, norm	1sec
difference of 1st and 2nd half (peak-to-peak)	x, y, z, norm	1sec
difference of 1st and 2nd half (crossingcounts)	x, y, z, norm	1sec
difference of 1st and 2nd half (difference from mean of every sensor's peak-to-peak)	x, y, z, norm	1sec
division of 1st and 2nd half (mean)	x, y, z, norm	1sec
division of 1st and 2nd half (variance)	x, y, z, norm	1sec
division of 1st and 2nd half (maximum)	x, y, z, norm	1sec
division of 1st and 2nd half (minimum)	x, y, z, norm	1sec
division of 1st and 2nd half (peak-to-peak)	x, y, z, norm	1sec
division of 1st and 2nd half (crossingcounts)	x, y, z, norm	1sec
division of 1st and 2nd half (difference from mean of every sensor's peak-to-peak)	x, y, z, norm	1sec
each difference of 1st, 2nd, 3rd, and 4th quarter (mean)	x, y, z, norm	1sec
each difference of 1st, 2nd, 3rd, and 4th quarter (variance)	x, y, z, norm	1sec
each difference of 1st, 2nd, 3rd, and 4th quarter (maximum)	x, y, z, norm	1sec
each difference of 1st, 2nd, 3rd, and 4th quarter (minimum)	x, y, z, norm	1sec
each difference of 1st, 2nd, 3rd, and 4th quarter (peak-to-peak)	x, y, z, norm	1sec
each difference of 1st, 2nd, 3rd, and 4th quarter (crossingcounts)	x, y, z, norm	1sec
each difference of 1st, 2nd, 3rd, and 4th quarter (difference from mean of every sensor's peak-to-peak)	x, y, z, norm	1sec
each division of 1st, 2nd, 3rd, and 4th quarter (mean)	x, y, z, norm	1sec
each division of 1st, 2nd, 3rd, and 4th quarter (variance)	x, y, z, norm	1sec
each division of 1st, 2nd, 3rd, and 4th quarter (maximum)	x, y, z, norm	1sec
each division of 1st, 2nd, 3rd, and 4th quarter (minimum)	x, y, z, norm	1sec
each division of 1st, 2nd, 3rd, and 4th quarter (peak-to-peak)	x, y, z, norm	1sec
each division of 1st, 2nd, 3rd, and 4th quarter (crossingcounts)	x, y, z, norm	1sec
each division of 1st, 2nd, 3rd, and 4th quarter (difference from mean of every sensor's peak-to-peak)	x, y, z, norm	1sec

Type of feature value	Axis	Window Size
difference among each sensor (variance)	x, y, z, norm	1sec, 0.5sec
difference among each sensor (maximum)	x, y, z, norm	1sec, 0.5sec
difference among each sensor (minimum)	x, y, z, norm	1sec, 0.5sec
difference among each sensor (peak-to-peak)	x, y, z, norm	1sec, 0.5sec
difference among each sensor (crossingcounts)	x, y, z, norm	1sec, 0.5sec
difference among each sensor (mean of variance per five data)	x, y, z, norm	1sec, 0.5sec
difference among each sensor (mean of variance per ten data)	x, y, z, norm	1sec, 0.5sec
difference among each sensor (variance of variance per five data)	x, y, z, norm	1sec, 0.5sec
difference among each sensor (variance of variance per ten data)	x, y, z, norm	1sec, 0.5sec
division among each sensor (variance)	x, y, z, norm	1sec, 0.5sec
division among each sensor (maximum)	x, y, z, norm	1sec, 0.5sec
division among each sensor (minimum)	x, y, z, norm	1sec, 0.5sec
division among each sensor (peak-to-peak)	x, y, z, norm	1sec, 0.5sec
division among each sensor (crossingcounts)	x, y, z, norm	1sec, 0.5sec
division among each sensor (mean of variance per five data)	x, y, z, norm	1sec, 0.5sec
division among each sensor (mean of variance per ten data)	x, y, z, norm	1sec, 0.5sec
division among each sensor (variance of variance per five data)	x, y, z, norm	1sec, 0.5sec
division among each sensor (variance of variance per ten data)	x, y, z, norm	1sec, 0.5sec
variance of each sensor's arrival timing	x, y, z, norm	
difference of each sensor's arrival timing	x, y, z, norm	
time from arrival timing to maximum	x, y, z, norm	
time from arrival timing to minimum	x, y, z, norm	
which is bigger with respect to difference between mean and maximum or minimum	x, y, z, norm	