Electronic Preprint for Journal of Information Processing Vol.25

Regular Paper

Activity Recognition and User Identification based on Tabletop Activities with Load Cells

Kazuya Murao[1,a)] Junna Imai[2,†1] Tsutomu Terada[2,3,b)] Masahiko Tsukamoto[2,c)]

Received: April 5, 2016, Accepted: October 4, 2016

Abstract: There have been several studies on object detection and activity recognition on a table conducted thus far. Most of these studies use image processing with cameras or a specially configured table with electrodes and an RFID reader. In private homes, methods using cameras are not preferable since cameras might invade the privacy of inhabitants and give them the impression of being monitored. In addition, it is difficult to apply the specially configured system to off-the-shelf tables. In this work, we propose a system that recognizes activities conducted on a table and identifies which user conducted the activities with load cells only. The proposed system uses four load cells installed on the four corners of the table or under the four legs of the table. User privacy is protected because only the data on actions through the load cells is obtained. Load cells are easily installed on off-the-shelf tables with four legs and installing our system does not change the appearance of the table. The results of experiments using a table we manufactured revealed that the weight error was 38 g, the position error was 6.8 cm, the average recall of recognition for four activities was 0.96, and the average recalls of user identification were 0.65 for ten users and 0.89 for four users.

Keywords: activity recognition, user identification, table, load cells

1. Introduction

There have been several systems for user activity recognition with sensors proposed thus far. In the open air and in public spaces such as offices, user activities can be obtained with sensors in a smartphone that the users carry or with sensors that they wear. However, it is difficult to collect data in private homes, as users do not carry the devices all the time. The basic approach for activity recognition in homes is installing sensors directly in the environment, such as ceiling [12], [16], wall [15], floor [1], furniture, and home appliance [11]. In particular, various daily activities occur on tables, and recognizing tabletop activities will enable a rich life-log at home and situated control of appliances.

Systems using image processing [2], [4], [5], [10] have been proposed for recognizing tabletop activities, but these systems create the feeling of being kept under surveillance for inhabitants, which is not appropriate for home use from the viewpoint of privacy. Moreover, placing pressure sensors and/or electrodes over a tabletop requires many sensors and a remodelling of the table, which is difficult when using off-the-shelf tables. Methods using extra devices such as trays [3], [8], [9] have been proposed, but the usage of the table is then limited by the type of extra device.

In this work, we propose a system that can recognize tabletop activities and identify users from the data of load cells attached to the four corners/legs of a table. User privacy is protected since the obtained data is load information only and activities other than tabletop activities cannot be inferred. Our system consists of a communication circuit board and four load cells that can easily be attached to off-the-shelf four-leg tables. Our system provides three functions: object detection, activity recognition, and user identification. The object detection function detects an object placed on the table and estimates its weight and position, and detects object removed from the table. The activity recognition function recognizes four kinds of activities that can be performed on a table: typing, moving a mouse, writing, and wiping. The user identification function identifies users from the data of the four activities. For activity recognition and user identification, the system learns user activities in advance, and recognizes activities and identifies users with machine learning.

This paper is organized as follows. Section 2 introduces related work and Section 3 explains the proposed system. We introduce an implementation of a table with the proposed system in Section 4 and present the evaluation experiments in Section 5. We conclude in Section 6 with a brief summary and mention of future work.

2. Related Work

This section introduces studies on activity recognition with sensors installed into a building and installed around a table.

2.1 Indoor Activity Recognition

There has been done many researches on user activity recognition using sensors installed on environments such as wall and ceiling. Murao et al. [12] proposed a method that estimates tra-

1 College of Information Science and Engineering, Ritsumeikan University Kusatsu, Shiga 525–8577, Japan
2 Graduate School of Engineering, Kobe University, Kobe, Hyogo 567–0013, Japan
3 PRESTO, Japan Science Technology Agency, Kawaguchi, Saitama 332–0012, Japan
†1 Presently with SONY
a) murao@cs.ritsumei.ac.jp
b) tsutomu@eedept.kobe-u.ac.jp
c) tuka@kobe-u.ac.jp

jectories of inhabitants with the small number of infra-red sensors installed on the ceiling. This work has difficulty of tracking inhabitants when multiple people get together in a room. Wren et al. [16] proposed a method that detects human movement by placing a lot of infra-red sensors on the ceiling. Sensors are placed in lattice at intervals of few meters, and events such as passing through, turning around, and crossing over are detected by analyzing the order of outputs from adjacent sensors. The number of sensors is hundreds, which leads to high cost of installation and maintenance. Health management systems for elderly people proposed by Wilson et al. [15] uses infra-red sensors and pressure-sensitive mattress. Addlesee et al. [1] proposed gait pattern classification using a specially-configured floor with multiple load cells. These works focus on movements of users among rooms or in a room, and did not recognize user activities.

2.2 Tabletop Activity Recognition

As researches on user activity recognition with sensors around a table, systems using camera and image processing technologies realize accurate recognition of objects and user activities on the table. Koike et al. [6] proposed interface prototypes on the augmented interface system that they had constructed, EnhancedDesk. One of the prototypes is Interactive Textbook, which is a learning support system using image recognition with a camera and image projection to a table by a overhead projector. Recognizing visual markers printed on books and hand movement, the system automatically retrieves digital contents from its database and projects them onto the desk. Inoue et al. [4] proposed Another Dish Recommender: a system that recommends additional dishes at an appropriate timing considering the progress of having a meal by recognizing activities during the meal. This system recognizes dishes and user's hands on a transparent tabletop board from camera image in real-time. A marker is attached to the dish. Recommended dishes are displayed on the table. This system was assumed to be used in a restaurant. Joutou et al. [5] proposed a system that makes food logs by using image recognition with Multiple Kernel Learning (MKL), which is one of the machine learning algorithms. In the experiment, 61.43% recognition accuracy on average was achieved for 50 kinds of foods. This system uses a camera, therefore it creates a feeling of being kept under surveillance for inhabitants, which is not appropriate for home use from the viewpoint of privacy.

Schmidt et al. [13] proposed a system of context acquisition based on load sensing. Their system uses four load cells placed at four corners of a table, floor, shelf, or similar item, and estimates the position of an object placed on its surface. The results of their experiment showed errors of approximately 2% of the surface length in each direction. In addition, these events on the table—setting an object down, removing an object, and knocking an object over—are detected. The system can also be installed on a floor to track users and estimate activities. However, the activities are domain knowledge-based, i.e., reliant upon the whereabouts of furniture, walls, and devices. The published results showed that 94% of the events were classified correctly, 6% were missed, and no events were misclassified. In another work by the same authors [14], touching, clicking, and tracking events can be detected. The novel points of our research are the recognition of tabletop activities based on machine learning and the identification of users performing activities.

3. Proposed system

In this section, we describe the system flow and detailed algorithms of object detection, activity recognition, and user identification.

3.1 System overview

We assume that four load cells are embedded in the four corners of a tabletop board (tabletop type) or attached under the legs of the table (leg-attachment type), as shown in **Fig. 1**. We implemented these two prototypes and will explain in detail in later section. The tabletop type is applicable to tables with a detachable tabletop board. The effect of noise from the floor is small because the sensors are far from the floor. The leg-attachment type is applicable to off-the-shelf tables that have four legs. Each load cell outputs weight values given to each point. The sampling frequency is 15 Hz. Our system targets the actions of placing objects on the table, removing objects from the table, and four specific activities on the table: typing, moving a computer mouse, writing, and wiping. We assume that only one object is added or removed at a time, that is, that multiple objects are neither added nor removed at the same time. The user does not slide objects on the tabletop surface and objects do not move or change positions by themselves like a ball or a toy car. A life-log application records individual activities often performed on the table, therefore, we adopted use of a laptop (typing and moving a computer mouse), study (writing), and cleaning (wiping) as activities performed in a daily life.

Figure 2 shows the flow of the proposed system. Our method first calibrates raw data and then the current state on the table is classified into stable state and active state. When the state changes from active to stable and the weight has changed from the last stable state, our system detects the addition or removal of an object. If an object is added, the weight and position of the object is calculated, and if an object is removed, it is identified from among the objects on the table. While in the active state, our system recognizes the activity and identifies the user. Our system has collected data of load cells with ground truth captured from

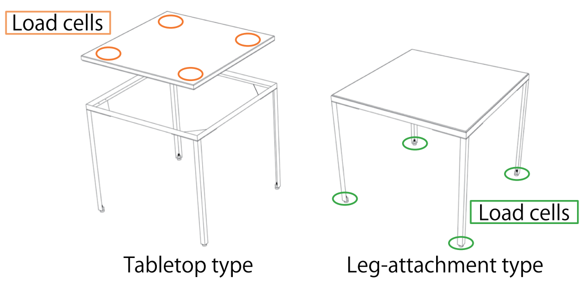

Fig. 1 Placement of load cells.

Fig. 2 Flow of the proposed method.

the users and constructed models for activity recognition and user identification in advance.

3.2 Calibration

Weight values obtained through load cells include the load given to the tabletop surface and the weight of the table itself. The weight of the table differs depending on the table. Since the load given to the tabletop surface is required, our system calibrates the offset of the weight of the table in advance. Let the raw sensor values of the load cells at time t be $s_i(t)$ $(1 \leq i \leq 4)$. Calibrated sensor values are obtained by

$$m_i(t) = s_i(t) - \frac{1}{N_c} \sum_{t'=T_c-N_c+1}^{T_c} s_i(t') \ (1 \leq i \leq 4), \tag{1}$$

where N_c is window size and $t = T_c$ is the time at which the calibration takes place. N_c is set to 15 samples (1 sec) in this paper. T_c is the time at which nothing is on the table and nobody is touching the table. $m_i(t)$ will ideally be zero value when there is nothing on the table and nobody is touching the table.

3.3 State Classification

There are two states on the table: stable state and active state. Stable state is the state when the load given to the table is constant. While in a stable state, the user is not touching the table. Objects such as a cup or laptop computer have been placed on the table, or there is nothing on the table. Active state is the state when the load given to the table is fluctuating. While in an active state, the user is touching the table by typing, cleaning, and so on, or objects are being placed on or removed from the table. Tabletop activities occur while in the active state, so in this paper our system classifies the tabletop state into stable or active states.

The detailed algorithm of state classification is as follows. Total load $m(t)$ is defined by $m(t) = \sum_{i=1}^{4} m_i(t)$. Supposing time $t = T$ now, the running average of total load $\mu(T)$ and running variance of total load $\sigma^2(T)$ are calculated over an N-sample window by the following equations. N is set to 15 samples (1 second) in this paper.

$$\mu(T) = \frac{1}{N} \sum_{t=T-N+1}^{T} m(t) \tag{2}$$

$$\sigma^2(T) = \frac{1}{N} \sum_{t=T-N+1}^{T} (m(t) - \mu(T))^2 \tag{3}$$

Our method sets the initial state to stable sate. Then, when $\sigma^2(T) > \alpha$ is satisfied, the current state changes to active state, where α is a threshold and set to 0.005 kg^2. While in active state, when $\sigma^2(T) \leq \alpha$ is satisfied for one second, the current state changes to stable state. These values were decided from the preliminary experiment. In our system, object detection is executed when the state has changed from active state to stable state. Activity recognition and user identification are periodically conducted while in active state.

3.4 Object Detection

When the state changes from active to stable, the system detects the addition or removal of an object by calculating the difference of total loads before and after the previous active state, as shown in **Fig. 3**. If the addition of an object is detected, the system calculates the position and weight of the added object. If the removal of an object is detected, the system identifies the removed object from among the objects on the table.

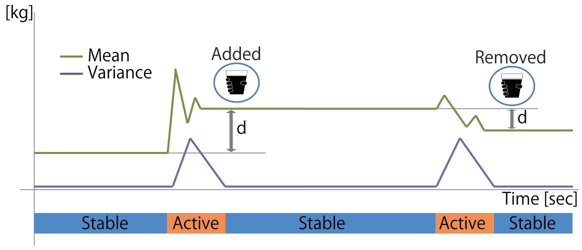

Fig. 3 Object detection.

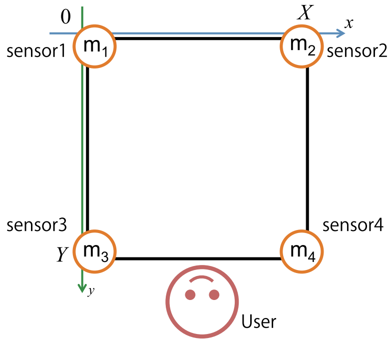

Fig. 4 Coordinates of sensors, and relationship of user and table.

The detailed algorithm is as follows. Assuming time $t = T$ is now and the time when the previous stable state started is T_{PS}, the change in the load during the previous active state is calculated by $d = m(T) - m(T_{PS})$. Finally, the time at which the previous stable state started is renewed by $T_{PS} = T$.

3.4.1 Addition of an Object

If $d \geq \beta$ is satisfied, the system judges that an object has been added onto the table, where β is a threshold and set to 0.2 kg. In this case, the system stores the difference of load values of each sensor $M_{i,j} = m_i(T) - m_i(T_{PS})$ $(1 \leq i \leq 4)$ for the future retrieval of removed objects, where i is the number of sensors and j is the sequential number of the object. The estimated weight of the object is the sum of the difference of load values of each sensor $M_j = \sum_{i=1}^{4} M_{i,j}$.

Figure 4 shows coordinates and positions of sensors. If we assume that the points of sensor1, sensor2, sensor3, and sensor4 are $(0, 0)$, $(X, 0)$, $(0, Y)$, and (X, Y), respectively, the estimated point of the object is obtained by

$$(x, y) = \left(\frac{M_{2,j} + M_{4,j}}{\sum_{i=1}^{4} M_{i,j}} X, \frac{M_{3,j} + M_{4,j}}{\sum_{i=1}^{4} M_{i,j}} Y \right). \tag{4}$$

3.4.2 Removal of an Object

If $d \leq -\beta$ is satisfied, the system judges that an object has been removed from the table. In this case, the system identifies which object has been removed. The estimated weight of the removed object is obtained by $M_{i,j_{rem}} = m_i(T_{PS}) - m_i(T)$, calculating the difference of the load values of each sensor as well as the addition of the object. If we suppose that there are J objects on the table, the system has stored load values of each sensor for each object. For the sake of simplicity, assume that all the objects 1 to J are on the table and no objects have been removed thus far. The system calculates the difference between $M_{i,j_{rem}}$ and $M_{i,j}$ $(1 \leq j \leq J)$ and finds the object whose distance is the smallest as the removed object by $\arg\min_j \sum_{i=1}^{4} |M_{i,j_{rem}} - M_{i,j}|$.

3.5 Activity Recognition and User Identification

The proposed system recognizes activities performed during active state. The system calculates twelve kinds of feature value over a window and recognizes activities with random forests that have learned data with ground truth in advance. In addition to the activity recognition, our system identifies users with random forests that have learned each activity labelled with a user ID. Both the activity recognition and the user identification use the same set of feature values.

The twelve kinds of feature values $f_1(t), \cdots, f_{12}(t)$ are defined as follows. $f_1(t)$ and $f_2(t)$ are running average and running variance of fluctuation of the load from the previous stable state

$$f_1(t) = \frac{1}{N} \sum_{i=t-N+1}^{t} m'(i) \tag{5}$$

$$f_2(t) = \frac{1}{N} \sum_{i=t-N+1}^{t} \left(m'(i) - \overline{m'}(t)\right)^2 \tag{6}$$

respectively, where $m'(t) = \sum_{i=1}^{4} (m_i(t) - m_i(t_{PS}))$, $\overline{m'}(t) = f_t(t)$, and N is a window size set to 15 samples (1 second) in this paper. These features focus on the load given to the table. $f_3(t)$ and $f_4(t)$ are running variance of centroid of the load $x'(t) = \left(\frac{m'_2(t)+m'_4(t)}{\sum_{i=1}^{4} m'_i(t)}\right) X$ and $y'(t) = \left(\frac{m'_3(t)+m'_4(t)}{\sum_{i=1}^{4} m'_i(t)}\right) Y$ calculated with the following equations,

$$f_3(t) = \frac{1}{N} \sum_{i=t-N+1}^{t} \left(x'(i) - \overline{x'}(t)\right)^2 \tag{7}$$

$$f_4(t) = \frac{1}{N} \sum_{i=t-N+1}^{t} \left(y'(i) - \overline{y'}(t)\right)^2, \tag{8}$$

where $\overline{x'}(t) = \sum_{i=t-N+1}^{t} x'(i)$ and $\overline{y'}(t) = \sum_{i=t-N+1}^{t} y'(i)$. $f_5(t)$ is the Euclidean distance between starting point and end point on the table over an N-sample window obtained by

$$f_5(t) = \sqrt{(x'(t) - x'(t-N+1))^2 + (y'(t) - y'(t-N+1))^2}. \tag{9}$$

$f_6(t)$ is the inclination of single regression over an N-sample window obtained by

$$f_6(t) = \frac{\sum_{i=t-N+1}^{t} \left(x'(i) - \overline{x'}(t)\right)\left(y'(i) - \overline{y'}(t)\right)}{\sum_{i=t-N+1}^{t} \left(x'(i) - \overline{x'}(t)\right)^2}. \tag{10}$$

$f_1(t)$ to $f_6(t)$ focus on the movements on the table.

$f_7(t)$ to $f_{12}(t)$ focus on the angle of movement of load centroid. $\theta(t)$ $(0° < \theta(t) \leq 360°)$ is the inclination angle of a line that connects centroids at time $t-1$ and t, as shown in **Fig. 5**. $f_7(t)$ is the median of a set $A = \{\theta(t-n), \theta(t-(n-1)), \cdots, \theta(t)\}$. $f_8(t)$ is the variance of $\theta(t)$ over an N-sample window obtained by

$$f_8(t) = \frac{1}{N} \sum_{i=t-N+1}^{t} \left(\theta(i) - \bar{\theta}(t)\right)^2 \tag{11}$$

$f_9(t)$, $f_{10}(t)$, $f_{11}(t)$, and $f_{12}(t)$ are the number of apexes whose $\theta(t)$ meets the conditions: $65° \leq \theta(t) \leq 115°$, $155° \leq \theta(t) \leq 205°$, $245° \leq \theta(t) \leq 295°$, and $335° \leq 360°||0° < \theta(t) \leq 25°$, respectively. These feature values were used in the reading activity recognition based on gaze tracking [7].

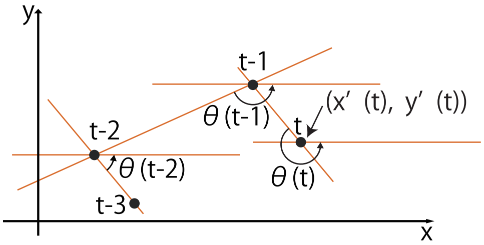

Fig. 5 How to obtain $\theta(t)$.

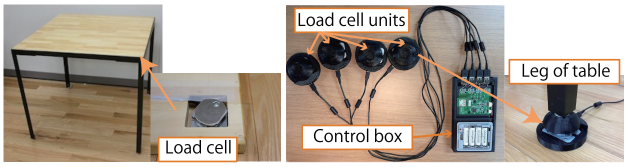

Fig. 6 Tabletop type table (left) and leg-attachment type (right).

4. Hardware Implementation

We implemented two prototypes: a tabletop type and a leg-attachment type. The tabletop type is a specially configured table consisting of a frame with four legs and a detachable square tabletop board. Four load cells are embedded in the tabletop board as shown in **Fig. 6**. The height of each leg is adjustable with a spacer, which enables the tabletop board to be kept flat. Each load cell is wired to the central circuit board that transmits the data to a PC via Bluetooth. The central circuit board uses four AA dry batteries or a 5 V AC power supply. The size and weight of the tabletop is 30(H)×790(W)×790(D) mm and 10.0 kg and that of the frame is 690(H)×800(W)×800(D) mm and 9.1 kg.

Leg-attachment type is an external device consisting of four load cell units and a control box, as shown in Fig. 6. The load cell unit consists of a bottom cover part, a load cell, and an upper part. The load cell is covered with metal and the bottom cover part prevents the load cell from scratching the floor. The upper part connects the load cell and a leg of the table. The upper parts were made using a 3D printer, enabling easy adjustment to any shape of table leg. Acrylonitrile butadiene styrene (ABS) resin was used as the filament of the 3D printer. Each load cell unit is wired to the control box that transmits the data to a PC via Bluetooth. The control box uses four AA dry batteries. The size and weight of the load cell unit is 42(H)×85(W)×85(D) mm and 156 g and the size and weight of control box is 24(H)×198(W)×105(D) mm and 205 g.

The load cells used for both prototypes are sensors installed in a Wii balance board [*1] by Nintendo. Specifications of the load cell are listed in **Table 1**. We developed an application that receives data with Visual C# and WiimoteLib [*2].

[*1] Wii Balance Board http://wiifit.com/what-is-wii-fit-plus/
[*2] WiimoteLib - .NET managed library for using a Nintendo Wii Remote http://www.wiimotelib.org/

Table 1 Specifications of load cell.

Max load with four units	130 [kg]
Error	±0.2 [kg]
Sampling rate	15 [Hz]
Size	20(H)×55(W)×45(D) [mm]
Weight	120 [g]

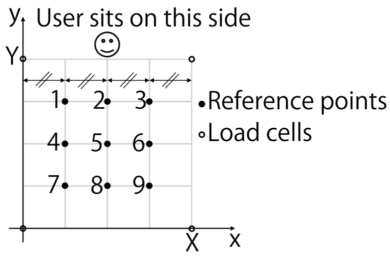

Fig. 7 Reference points on the table.

5. Evaluation

We performed experiments to evaluate the accuracies of object detection, activity recognition, and user identification. In the experiments, data is collected with the tabletop prototype implemented in the previous section to know the baseline of the system since the tabletop type produces less noisy data than leg-attachment type.

5.1 Object Detection

5.1.1 Setup

We conducted two types of experiment for object detection. In the first one, one of the authors placed and removed three types of bottle that weigh 300 g, 600 g, and 900 g on nine reference points on the table, as shown in **Fig. 7**. Addition and removal of the bottles was conducted in the following manner: 300-g bottle on point 1 five times → 600-g bottle on point 1 five times→ 900-g bottle on point 1 five times→ $\cdots$ → 900 [g] bottle on point 9 five times. Only one bottle was placed on the table at a time. In total, bottles were placed $3 \times 9 \times 5 = 135$ times.

5.1.2 Results and Consideration

Figure 8 shows plots of the detected points. **Figure 9** shows the distribution of position error for each reference point. Position errors and weight errors are listed in **Table 2**. For the results of position error, the average error over all trials was 6.8 cm and the maximum error was 50.7 cm. The average and standard deviation of position error of heavy objects were smaller, as heavy objects were less affected by potential error of the sensor. At the same time, the position error at points close to the load cells became large, as a point close to one corner is far from the other three corners and weight was not distributed ideally to the sensors due to the bend of the tabletop board.

For the results of weight error, the average error over all trials was 38 g and the maximum was 109 g. **Figure 10** shows the distribution of weight error for each reference point, indicating that the detected weight for the 300-g bottles was smaller than 300 g, that for the 600-g bottles was close to 600 g, and that for the 900-g bottles was larger than 900 g. This is also confirmed by the result that the standard deviation of weight error for 600-g bottle is the smallest in Table 2. This means that light weight was absorbed

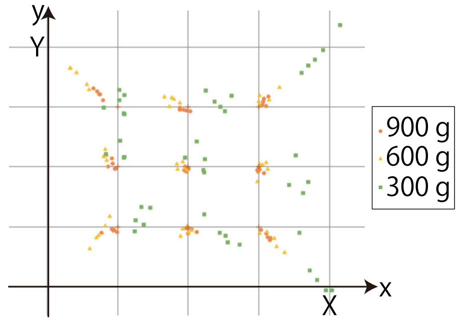

Fig. 8 Plots of detected points of bottle placements.

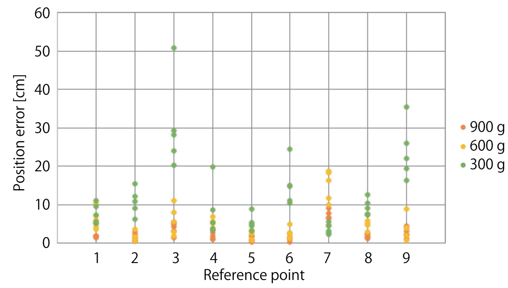

Fig. 9 Distribution of position error.

Table 2 Position error and weight error.

Bottle	Position		Weight	
	Avg±Std [cm]	Max [cm]	Avg±Std [g]	Max [g]
300 g	12.7±10.0	50.7	24±23	50
600 g	5.3±4.4	16.3	14±13	38
900 g	2.5±2.0	9.0	77±17	109

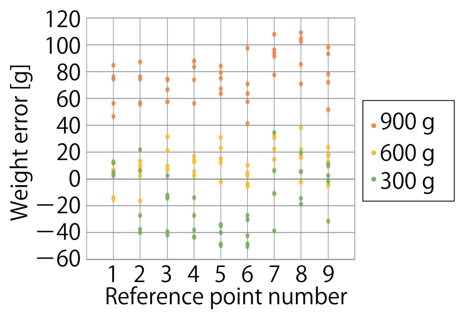

Fig. 10 Distribution of weight error.

and heavy weight was overestimated due to the characteristics of the tabletop board e.g., the bend. Further investigation is needed to clarify the cause.

We assume that the result of object detection is used for projection of object's information over the object, recording a progress of having a meal, and improving activity recognition accuracy. Therefore, 38-g error in the object weight estimation can be used for identifying objects in our daily life as a weight of objects ranges widely from smarphone (100 g) to laptop (2,000 g). Identifying dishes such as rice, soup, and main dish, and detecting decrease of these dishes would also be possible. In addition, considering the size of objects such as cup and plate, the objects are not mistaken for each other with 6.8-cm position error.

In this paper, we assume that objects are added and removed one by one and object position does not change. With respect to object removal, the removed objects can be identified by seeing the object alignment after the removal since all objects before the removal are known. In the case of addition, the system would find an addition of multiple objects by detecting the irregular change in the waveform unless the objects are placed at exactly same time. Addition and removal occurring at the same time can also be found since irregular waveforms would appear as well. When the object is slid on the table, the current algorithm cannot detect it since total load value does not change, however it can be detected by seeing the change in load values of each sensor. Implementation of these functions are our future work.

In the second experiment, one of the authors placed three 500-g bottles on arbitrary reference points on the table one by one and then removed them one by one. This set was iterated ten times. In total, bottles were placed and removed $3 \times 10 = 30$ times. Addition and removal of objects were detected with 100% accuracy. However, we have found that when placing ten objects on the table, our system sometimes made mistakes in the object detection, so further investigation is needed to determine the limits of our system.

5.2 Activity Recognition and User Identification

5.2.1 Setup

In this experiment, ten subjects denoted A to J (nine males and one female aged 21 to 25 years old) performed four kinds of activities: Typing, Moving a mouse, Writing, and Wiping. Typing is playing a typing game with the keyboard of a laptop PC. Moving a mouse is playing a shooting game with an external mouse connected to a laptop PC. Writing is transcribing texts. Wiping is cleaning the table with a towel. The subjects performed these activities for a few minutes, 10 seconds of which was used for training and 20 seconds of which was used for testing. We used the random forest of Weka [*3]. The number of trees was 100. Accuracies of activity recognition and user identification were calculated for a group of all ten subjects (group A) and for a group of four subjects A–D (group B) representing a family. The way of evaluation in both groups is same. Activity recognition was conducted user-dependently, i.e., data for all the subject in a group was used for training and then activities are recognized with the model for each subject. User identification was conducted for each activity, i.e., data on one activity for all the subjects in a group was used for training and then users are identified with the model for each subject.

5.2.2 Results and Consideration

Confusion matrices of activity recognition for groups A and B are shown in **Tables 3** and **4**, respectively. Average recalls over the four activities for groups A and B were 0.94 and 0.96, respectively. This confirms that the activities were recognized with a high degree of accuracy regardless of the number of subjects.

Confusion matrices of user identification that identifies the user out of group A for typing, moving a mouse, writing, and wiping are shown in **Tables 5**, **6**, **7**, and **8**, respectively.

Average recall over four activities was 0.65. Focusing on the activities, average recalls were 0.74 for typing, 0.80 for moving a mouse, 0.59 for writing, and 0.48 for wiping. The result of wiping was lower than the other activities, which would be caused by

Table 3 Confusion matrix of activity recognition (10 people).

Output→ Input↓	Typing	Moving a mouse	Writing	Wiping	Recall
Typing	2876	90	78	0	0.94
Mouse	19	3165	5	1	0.99
Writing	331	23	1968	34	0.84
Wiping	0	15	64	3117	0.98
Precison	0.89	0.96	0.93	0.99	—

Table 4 Confusion matrix of activity recognition (4 people).

Output→ Input↓	Typing	Moving a mouse	Writing	Wiping	Recall
Typing	1151	46	28	0	0.94
Mouse	17	1222	14	0	0.98
Writing	53	4	954	14	0.93
Wiping	3	0	26	1198	0.98
Precision	0.94	0.96	0.93	0.99	—

Table 5 Confusion matrix of user identification for typing (10 people).

Output→ Input↓	A	B	C	D	E	F	G	H	I	J	Recall
A	203	0	0	0	109	5	0	0	0	2	0.64
B	77	191	17	0	0	0	0	0	0	0	0.67
C	0	0	237	7	0	0	1	74	0	0	0.74
D	0	16	0	304	0	0	0	0	0	0	0.95
E	2	0	5	0	242	0	1	0	38	0	0.84
F	5	0	0	23	0	245	0	0	0	0	0.90
G	35	0	0	1	13	0	238	0	0	32	0.75
H	0	0	58	0	4	0	19	185	0	48	0.59
I	8	0	0	0	2	5	5	0	270	0	0.93
J	43	0	13	0	78	0	29	24	1	129	0.41
Precision	0.54	0.92	0.72	0.91	0.54	0.96	0.81	0.65	0.87	0.61	—

Table 6 Confusion matrix of user identification for moving a mouse (10 people).

Output→ Input↓	A	B	C	D	E	F	G	H	I	J	Recall
A	276	19	4	0	0	0	0	0	0	12	0.89
B	0	258	0	0	0	62	0	0	0	0	0.81
C	0	0	311	0	3	0	6	0	0	0	0.97
D	0	0	0	285	11	24	0	0	0	0	0.89
E	0	0	43	9	181	0	66	0	20	1	0.57
F	8	176	0	0	0	134	0	2	0	0	0.42
G	0	0	0	0	0	0	318	0	2	0	0.99
H	0	62	0	0	0	0	0	257	0	0	0.80
I	0	0	0	112	0	0	0	6	202	0	0.63
J	0	0	0	0	0	0	0	0	0	319	1.00
Precision	0.97	0.50	0.87	0.70	0.93	0.61	0.82	0.97	0.90	0.96	—

Table 7 Confusion matrix of user identification for writing (10 people).

Output→ Input↓	A	B	C	D	E	F	G	H	I	J	Recall
A	92	2	0	0	6	0	0	22	13	12	0.63
B	6	187	0	33	11	33	0	15	0	1	0.66
C	7	1	218	2	1	12	15	2	9	7	0.79
D	24	8	1	108	14	17	0	1	0	6	0.60
E	4	11	0	3	174	44	0	35	0	3	0.63
F	0	1	0	42	1	100	22	45	0	1	0.47
G	0	0	51	17	0	62	96	67	0	13	0.31
H	17	6	2	29	13	2	14	168	4	11	0.63
I	9	2	9	6	51	0	4	10	161	4	0.63
J	36	4	6	0	12	2	0	7	0	88	0.57
Precision	0.47	0.85	0.76	0.45	0.62	0.37	0.63	0.45	0.86	0.60	—

common wiping activity over the subjects.

Confusion matrices of user identification for four people (group B) for typing, moving a mouse, writing, and wiping are shown in **Tables 9**, **10**, **11**, and **12**, respectively.

[*3] Weka 3 - Data Mining with Open Source Machine Learning Software in Java http://www.cs.waikato.ac.nz/ml/weka/.

Table 8 Confusion matrix of user identification for wiping (10 people).

Output→ Input↓	A	B	C	D	E	F	G	H	I	J	Recall
A	201	0	1	5	36	0	0	38	17	19	0.63
B	1	112	0	0	0	135	11	0	0	61	0.35
C	8	3	147	77	30	0	0	47	7	0	0.46
D	12	0	90	178	1	1	4	0	5	0	0.61
E	29	30	6	43	119	0	0	26	59	0	0.38
F	0	7	0	0	11	181	119	0	0	2	0.57
G	0	0	7	0	0	103	210	0	0	0	0.66
H	27	9	76	28	24	3	0	89	19	40	0.28
I	17	22	0	1	77	0	0	55	121	27	0.38
J	0	7	0	0	0	156	2	0	0	155	0.48
Precision	0.68	0.59	0.45	0.54	0.40	0.31	0.61	0.35	0.53	0.51	—

Table 9 Confusion matrix of user identification for typing (4 people).

Output→ Input↓	A	B	C	D	Recall
A	312	0	2	0	0.99
B	45	201	23	1	0.74
C	0	0	314	5	0.98
D	0	2	0	318	0.99
Precision	0.87	0.99	0.92	0.98	—

Table 10 Confusion matrix of user identification for moving a mouse (4 people).

Output→ Input↓	A	B	C	D	Recall
A	263	16	13	1	0.90
B	0	312	0	8	0.98
C	0	0	320	0	1.00
D	0	14	0	306	0.96
Precision	1.00	0.91	0.96	0.97	—

Table 11 Confusion matrix of user identification for writing (4 people).

Output→ Input↓	A	B	C	D	Recall
A	194	7	3	12	0.90
B	20	211	2	54	0.73
C	14	2	268	5	0.93
D	15	10	1	207	0.89
Precision	0.80	0.92	0.98	0.74	—

Table 12 Confusion matrix of user identification for wiping (4 people).

Output→ Input↓	A	B	C	D	Recall
A	289	11	5	5	0.93
B	2	315	1	0	0.99
C	46	19	153	94	0.49
D	17	0	77	193	0.67
Precision	0.82	0.91	0.65	0.66	—

Average recalls were 0.93 for typing, 0.96 for moving a mouse, 0.86 for writing, and 0.77 for wiping, with an average of 0.89 for all four activities. The result for group A was low since the number of candidate users was high, which made it difficult to identify them on the basis of single tabletop activity. In contrast, the result for group B was high. Tables in a living room or dining room are generally shared at home with small number of people. Overall, the results demonstrate that the proposed system is able to identify the user from the activities on the table. Moreover, these performances will be improved by seeing the consequence of activities and by filtering.

6. Conclusion

In this paper, we proposed a system that recognizes tabletop activities and identifies users by using load cells attached to the four corners of a table. The proposed system classifies the state of the table as stable and active and then calculates the position and weight of objects on the table in the stable state. The system also recognizes activities performed on the table and identifies which the user performed the activities in the active state. Experimental results showed that our system could detect objects with a 38-g mean absolute error of weight and a 6.8-cm mean absolute error of position and could detect the addition and removal of objects without mistakes. In addition, our system recognized four types of activity with 0.94 recall and identified users with 0.65 recall for ten subjects and with 0.89 recall for four subjects. We have implemented two prototypes: tabletop type and leg-attachment type. Though the tabletop prototype was used in the experiment, we prospect that similar trend would appear with the leg-attachment type, however, accuracy would drop since sensors are further from the tabletop and sensor value is more noisy. Four simple activities were targeted in this paper. We plan to expand our system to recognize more complicated and combined activities such as dining as a sequence of simple activities.

Acknowledgments This research was supported in part by a Grant-in-Aid for Scientific Research (B) (15H02698) of the Japanese Ministry of Education, Culture, Sports, Science and Technology, and by the Japan Science and Technology Agency, PRESTO.

References

[1] Addlesee, M.D., Jones, A., Livesey, F. and Samaria, F.: The ORL Active Floor, *IEEE Personal Communications*, Vol.4, No.5, pp.35–41 (1997).

[2] Fukuchi, K. and Rekimoto, J.: Interaction Techniques for SmartSkin, *Proc. 15th ACM Symposium on User Interface Software and Technology* (*UIST 2002*), pp.49–50 (2002).

[3] Chang, K.-H., Liu, S.-Y., Chu, H.-H., Hsu, J.Y.-J., Chen, C., Lin, T.-Y., Chen, C.-Y. and Huang, P.: The Diet-Aware Dining Table: Observing Dietary Behaviors over a Tabletop Surface, *Proc. International Conference on Pervasive Coputing* (*Pervasive 2006*), pp.366–382 (2006).

[4] Inoue, T. and Matsusaka, Y.: A System to Recommend Dishes by the Real Time Recognition of Dining Activity, *Proc. IEEE International Conference on Systems Man and Cybernetics* (*SMC 2009*), pp.2448–2452 (2009).

[5] Joutou, T. and Yanai, K.: A Food Image Recognition System with Multiple Kernel Learning, *Proc. 16th IEEE International Conference on Image Processing* (*ICIP 2010*), pp.285–288 (2010).

[6] Koike, H., Sato, Y., Kobayashi, Y., Tobita, H. and Kobayashi, M.: Interactive Textbook and Interactive Venn Diagram : Natural and Intuitive Interface on Augmented Desk System, *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems* (*CHI 2000*), pp.121–128 (2000).

[7] Kunze, K., Utsumi, Y., Shiga, Y., Kise, K. and Bulling, A.: I know what you are reading: Recognition of document types using mobile eye tracking, *Proc. 17th Annual International Symposium on International Symposium on Wearable Computers*, pp.113–116 (2013).

[8] Lin, T.-Y., Chang, K.-H., Liu, S.-Y. and Chu, H.-H.: A Persuasive Game to Encourage Healthy Dietary Behaviors of Young Children, *Adjunct Proc. 8th International Conference on Ubiquitous Computing* (*UbiComp 2006*) (2006).

[9] Lo, J.-L., Lin, T.-Y., Chu, H.-H., Chou, H.-C., Chen, J.-H., Hsu, J.Y.-J. and Huang, P.: Playful Tray: Adopting Ubicomp and Persuasive Techniques into Play-based Occupational Therapy for Reducing Poor Eating Behavior in Young Children, *Proc. 9th International Conference on Ubiquitous Computing* (*Ubicomp 2007*), pp.38–55 (2007).

[10] Mori, M., Kurihara, K., Tsukada, K. and Siio, I.: Dining Presenter: Augmented Reality System for a Dining Tabletop, *Adjunct Proc. In-*

ternational Conference on Ubiquitous Computing (*Ubicomp 2009*), pp.168–169 (2009).
[11] Mori, T., Asaki, K., Noguchi, H. and Sato, T.: Accumulation and Summarization of Human Daily Action Data in One-room-type Sensing System, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS 2001*), pp.2349–2354 (2000).
[12] Murao, K., Terada, T., Yano, A. and Matsukura, R.: Detecting Room-to-Room Movement by Passive Infrared Sensors in Home Environments, *Proc. 1st Workshop on Recent Advances in Behavior Prediction and Pro-active Pervasive Computing* (*AwareCast 2012*), pp.3:1–12 (2012).
[13] Schmidt, A., Strohbach, M., van Laerhoven, K., Friday, A. and Gellersen, H.-W.: Context Acquisition Based on Load Sensing, *Proc. 4th International Conference on Ubiquitous Computing* (*UbiComp 2002*), pp.333–350 (2002).
[14] Schmidt, A., Strohbach, M., van Laerhoven, K. and Gellersen, H.-W.: Ubiquitous Interaction - Using Surfaces in Everyday Environments as Pointing Devices, *Proc. 7th ERCIM Workshop* (2002).
[15] Wilson, D. and Atkeson, C.: Simultaneous Tracking and Activity Recognition (STAR) Using Many Anonymous, Binary Sensors, *Proc. 3rd International Conference on Pervasive Computing* (*Pervasive 2005*), pp.62–79 (2005).
[16] Wren, C.R. and Tapia, E.M.: Toward Scalable Activity Recognition for Sensor Networks, *Proc. 2nd International Workshop on Location- and Context-Awareness* (*LoCA 2006*), pp.168–185 (2006).

Kazuya Murao is an Assistant Professor at College of Information Science and Engineering, Ritsumeikan University, Japan. He received B.Eng., M.Info.Sci., and Ph.D. degrees from Osaka University in 2006, 2008, and 2010, respectively. Prof. Murao is working on wearable computing, ubiquitous computing, and context aware systems. He is a member of IPSJ, IEEE and ACM.

Junna Imai received B.Eng. and M.Eng. degrees from Kobe University in 2013 and 2015, respectively. Mr. Imai is currently working at SONY.

Tsutomu Terada is an Associate Professor at the Graduate School of Engineering, Kobe University, Japan. He is also a PRESTO researcher at the Japan Science and Technology Agency, Japan. He received B.Eng., M.Eng., and Ph.D. degrees from Osaka University in 1997, 1999, and 2003, respectively. Prof. Terada is working on wearable computing, ubiquitous computing, and entertainment computing. He is a member of IPSJ, IEEE, ACM, and IEICE.

Masahiko Tsukamoto is a Professor at Graduate School of Engineering, Kobe University, Japan. He received B.Eng., M.Eng. and Ph.D. from Kyoto University in 1987, 1989 and 1994, respectively. Prof. Tsukamoto is working on wearable computing and ubiquitous computing. He is a member of IPSJ, IEEE, ACM, and IEEE.