



Evaluating Sensor Placement and Gesture Selection for Mobile Devices

Murao, Kazuya
Terada, Tsutomu
Yano, Ai
Matsukura, Ryuichi

(Citation)

ヒューマンインタフェース学会論文誌, 15(3):281-292

(Issue Date)

2013-08-25

(Resource Type)

journal article

(Version)

Version of Record

(Rights)

© 2013 ヒューマンインタフェース学会

本著作物の著作権は特定非営利活動法人ヒューマンインタフェース学会に帰属します。本著作物は著作権者であるヒューマンインタフェース学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」に従うことをお願い致します。

(URL)

<https://hdl.handle.net/20.500.14094/0100492900>



Evaluating Sensor Placement and Gesture Selection for Mobile Devices

Kazuya Murao^{*1}, Tsutomu Terada^{*1 *2}, Ai Yano^{*3} and Ryuichi Matsukura^{*4}

Abstract – Mobile phones and video game controllers using gesture recognition technologies enable easy and intuitive operations, such as scrolling browser and drawing objects. However, usually only one of each kind of sensor is installed in a device, and the effect of multiple homogeneous sensors on recognition accuracy has not been investigated. Moreover, the effect of the differences in the motion of a gesture has not been examined. We have investigated the use of a test mobile device with nine accelerometers and nine gyroscopes. We captured data for 27 kinds of gestures for a mobile tablet. We experimentally investigated the effects on recognition accuracy of changing the number and positions of sensors and the number and kinds of gestures.

Keywords : Gesture recognition, Accelerometer, Mobile device, Sensor placement, Gesture selection

1. Introduction

Along with the downsizing of computers, many kinds of devices containing small sensors have been released, and their applications have attracted a great deal of attention. In particular, an accelerometer is installed in most current mobile devices, such as the iPhone and Android-powered devices, and video game controllers for Wii and PS3, which enables easy and intuitive operations. Usually only one accelerometer is installed in commercial devices for energy saving and small footprint purposes. A high degree of accuracy is required since failure in gesture recognition deteriorates interface usability. The effect on accuracy of changing the number of sensors has not been investigated. Using multiple sensors increases the processing time for recognition, which would lead to less interface usability. In addition, whether recognition accuracy is affected by the number and kinds of gestures has not been studied.

We conducted an experiment capturing 27 kinds of gestures with 9 accelerometers and 9 gyroscopes and investigated the effect on recognition accuracy of changing the number of sensors and their positions and changing the number and kinds of gestures.

This paper is organized as follows. Section 2 describes the factors that affect gesture recognition

accuracy. Section 3 describes the experiment conducted to evaluate the effect of the number and position of sensors and of the number and kinds of gestures. The results are presented and discussed in Section 4. An application using gesture recognition technology is described in Section 5. The key points are summarized and future work is mentioned in Section 6.

2. Factors in gesture recognition accuracy

The important factors affecting gesture recognition accuracy include the number of sensors, position of sensors, number of gestures, and kinds of gestures. While the form of device is an important factor in terms of affordance, it is ignored here since it is not realistic to change the form of the device for recognition accuracy without taking its design into consideration.

2.1 Number and positions of sensors

Table 1 summarizes the number and kinds of sensors commonly installed on commercial devices. Apple's iPod and iPhone and Nintendo's Wii remote and Nunchuk use a 3-axis accelerometer. Nintendo's Wii motion plus, a device for enhancing the Wii remote, contains a 2-axis gyro and a 1-axis gyro, which together detect 3-axis angular velocity. Wii motion plus can detect twist of wrist to give a ball spin in a table tennis game. HTC Dream (also marketed as T-Mobile G1 in the US and parts of Europe), Android-powered phone, has a 3-axis accelerometer and a 3-axis geomagnetic sensor. One device usually has only

^{*1}: Graduate School of Engineering, Kobe University, Japan

^{*2}: PRESTO, Japan Science and Technology Agency, Japan

^{*3}: Fujitsu Laboratories Ltd., Japan

^{*4}: Fujitsu Ltd., Japan

Table 1 Sensors commonly installed in commercial devices.

Device	Number and kind	Manufacturer
Apple Inc. iPod nano (5th generation)	One 3-axis accelerometer	STMicroelectronics
Apple Inc. iPhone 3GS	One 3-axis accelerometer	STMicroelectronics
Apple Inc. iPhone 4S	One 3-axis accelerometer	STMicroelectronics
HTC Dream (T-Mobile G1)	One 3-axis accelerometer One 3-axis geomagnetic sensor	Asahi Kasei Microsystem
Nintendo Wii remote	One 3-axis accelerometer	STMicroelectronics
Nintendo Nunchuk	One 3-axis accelerometer	Analog Devices, Inc.
Nintendo Wii motion plus	One 2-axis gyroscope One 1-axis gyroscope	INVENSENSE EPSON TOYOCOM

one sensor of each kind, and, as far as we know, no device with multiple homogeneous sensors has been released. The positions of sensors installed in the iPod, iPhone 3GS, iPhone 4S, Wii remote, and HTC Dream are shown in Figure 1 with ■ marks. The accelerometer and a geomagnetic sensor are implemented in the same position in the HTC Dream. As shown in the figure, there is no consistent positioning for the sensors. Sensor positions are not coordinated, especially in the iPod, iPhone 3GS, and iPhone 4S. This indicates that the positions probably depend on hardware limitations.

Murao et al.^[6] measured the recognition accuracies of daily movements for all sensor combinations of five 3-axis accelerometers attached to a subject's wrists, ankles, and hip. The results show that recognition accuracies depend on the number of sensors and their combination. Therefore, the effect of the number and position of sensors should be taken into consideration in addition to energy consumption, implementation space, and wiring when developing new devices.

In the field of activity recognition with sensors, many studies have tackled on improving recognition accuracy from the viewpoints of filtering, feature selection, and parameter tuning of classifiers. This paper clarifies the effect of using multiple sensors and of changing positions for gesture recognition performance, which has not been reported thus far and is important information for hardware developers and application designers.

2.2 Number and kinds of gestures

The number and kinds of gestures also affects the recognition accuracy. Many systems using gesture recognition technology with accelerometers have

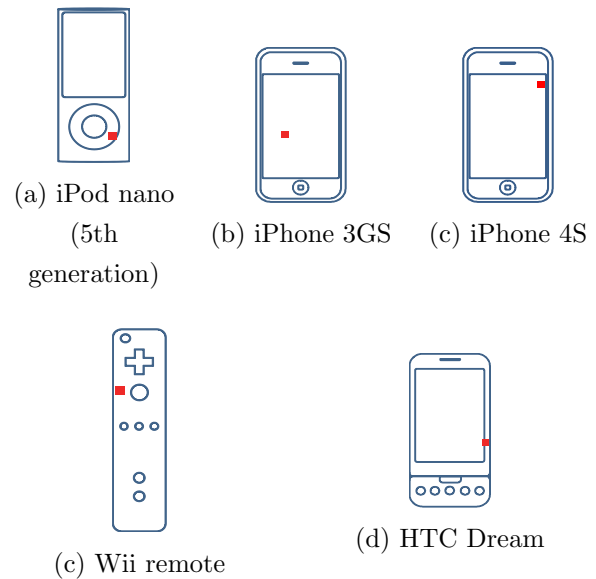


Fig.1 Positions of sensors installed in commercial devices.

been proposed. The method proposed by Ravi et al.^[10] recognizes eight activities including vacuuming and brushing teeth with an accelerometer attached to the pelvic region. Naya et al.^[8] proposed workers' routine activity recognition system to support their daily work. This system recognizes nurse's activities such as drip injection and vital check. The method proposed by Graeme et al.^[1] annotates video-recorded activities by gesture recognition using one accelerometer mounted on the wrist since annotating video is difficult only by analyzing video. It uses hidden Markov models (HMMs)^[9] for the recognition, resulting in one mistake in 30 trials for three kung-fu martial art movements; cut, elbow, and punch. The Georgia Tech Gesture Toolkit^[12] has been proposed as a tool to support gesture recognition. The toolkit enables ordinary users to use HMM toolkit^[2] with ease, which facilitates using

HMM but requires specialized knowledge of speech recognition. One of its applications, gesture recognition with a 3-axis accelerometer on the wrist and on the elbow, achieved 93.3% accuracy for ten kinds of gestures such as grinding and sawing. The system proposed by Holger et al. [5] recognizes ten daily short actions, such as pushing a button and drinking, and achieves approximately 80% precision and recall. Innovative point of this study is that it partitions the stream of sensor into several segments that represent atomic human movement by using the sliding-window and bottom-up (SWAB) algorithm [3].

The system proposed by Liu et al. [4] recognizes eight gestures including drawing a line or a circle (recommended by Nokia laboratories), with one 3-axis accelerometer. They captured more than 4,000 samples for 8 test subjects over a long period, using dynamic time-warping (DTW) [7] as a recognition algorithm. Accuracy of 98.6% was achieved by successively renewing the training data. Moreover, several studies proposed gesture recognition system using one 3-axis accelerometer installed in a mobile phone [14]~[16]. These systems recognize seventeen, nine, and eleven kinds of gestures, respectively, including drawing a number or a alphabet, tilting a phone, and shaking a phone. They achieved accuracy of 90-97.5% using SVM, HMM, and Bayesian Network as a classifier, respectively. These systems use mobile phones which are commercially available, therefore the number of sensor is one and its position is fixed.

In these studies, the number of gestures to be recognized was not many, and the effect of the number and kinds of gestures was not reported. Though increasing the number of gestures to be recognized would enhance the interface, recognition accuracy would drop. Our investigation clarified the effect of the number and kinds of gestures, enabling application engineers to better select gestures used in applications.

3. Experiment

This section describes the gesture recognition experiment we conducted to investigate the effect on recognition accuracy of changing the number of sensors and their positions and the number and kinds of gestures.

3.1 Setup

Data on 27 kinds of gestures (Table 2) that could be performed using a mobile tablet such as the iPad for controlling applications such as browser and game we assume were captured 10 times for each gesture for 8 subjects (7 male and 1 female) 21-26 years. A total of 2,160 samples were collected using 9 sensors placed on a test board, as shown in Figure 2. The board was W117×H155×D16 [mm] and weighed 200 [g]. The sensors used in the experiment were WAA-006 sensor, made by Wireless-T Inc. ¹ Each sensor comprised a wireless 3-axis accelerometer and a 3-axis gyroscope. All the subjects were right-handed and performed the gestures while holding the lower right edge of the board with their right hand. A laptop PC placed nearby the subjects instructed gestures to perform in a random order and received the data from the sensors via Bluetooth. The sampling frequency was 50 [Hz]. To reduce errors due to individual interpretation, the instructions were not given verbally. Instead, one of the authors demonstrated the actual movement for each gesture.

3.2 Preprocessing

Since our objective was to investigate the effect of sensors and gestures, the subjects were instructed to stand still before and after each gesture to mark the starting and ending points of each gesture. The waveform for each gesture was then extracted by checking for displacements in the captured data. If $|x(t) - \bar{x}(t)|$, where $x(t)$ is a captured data point and $\bar{x}(t)$ is the moving average, exceeded a threshold ϵ , the system detected a starting point. Otherwise, the system judged that the subject was not gesturing. We set ϵ to 200 [mG] since the fluctuation in the data while the subjects were stationary was up to 100 [mG]. Since the current value $x(t)$ might have temporarily entered the region of $x(t) \pm \epsilon$ even while the subject was gesturing, the gesture was judged to have ended only after $x(t)$ had been within the region for more than 0.25 [s]. This interval was based on the result of pilot studies. The data for one gesture consisted of 54 sequences (9 sensors× 2 types × 3 axes). The collected data were manually labeled.

3.3 Recognition

We employed two types of method to investigate the recognition accuracy: support vector machine

1: Wireless Technologies, Inc. <http://www.wireless-t.jp>

Table 2 List of gestures.

ID	Description	Illustration
1	Tilt to the near side	
2	Tilt to the far side	
3	Tilt to the left side	
4	Tilt to the right side	
5	Tap upper side twice	
6	Tap left side twice	
7	Swing twice to the left side quickly	
8	Swing twice to the right side quickly	
9	Shuffle cards	
10	Tap lateral edge as though sifting	
11	Scoop	
12	Lay cards	
13	Gather cards	
14	Rap table with the longer lateral edge	
15	Rap table with the surface of the board	
16	Knock the board twice	
17	Turn the board over	
18	Rotate clockwise on the table	
19	Shift up	
20	Shift down	
21	Shift left	
22	Shift right	
23	Shift diagonally up	
24	Shift diagonally down	
25	Draw a circle	
26	Draw a triangle	
27	Draw a square	

(SVM) and dynamic time warping (DTW). In SVM, the feature values are extracted from the sequence and fed to the support vector machine. In DTW, the raw data of the sequence is used. The characteristics of both methods are that SVM recognizes quickly but does not suit gesture recognition since the information on the gesture trajectory is missed



Fig. 2 Experimental board with nine sensors.

by feature extraction, and that DTW can conduct precise recognition and suits gesture recognition but takes long time for the recognition since DTW compares all the data points in the sequence.

In addition, in our evaluation, training data and testing data are selected in two ways: intra-subject and inter-subject. Intra-subject means the averaged accuracy over the eight subjects after four-fold cross-validation was conducted for each subject. Inter-subject means the use of one subject's data for testing and the other subjects' data for training, with the average accuracy over all subjects.

3.3.1 Dynamic time-warping

Time-series data has been used in various fields such as science, medicine, economics, and engineering. Calculation of similarity between time-series data is required in order to perform data-mining. A simple approach to measure similarity is to use the Euclidean distance, but the results are susceptible to temporal distortion, and the number of samples in two data sequences must be equal. A better approach is to use dynamic time-warping (DTW). The DTW algorithm can be used to calculate the temporal non-linear elastic distance between two sequences, which may vary in time, and the number of both samples need not be equal. For example, it can be used to find the similarity for *draw a circle in the air* when the rotating speeds differed.

The algorithm works as follows. For the sake of simplicity, we assume there is data for one axis. Testing data $X = (x_1, \dots, x_m)$ and training data $Y = (y_1, \dots, y_n)$, with length m and n , respectively, are compared, an $m \times n$ matrix $\mathbf{d}$ is defined: $\mathbf{d}(x_i, y_j) = |x_i - y_j|$. Next, warping path $W = (w_1, \dots, w_k)$, which is a path of pairs of indices of X and Y , is found. W meets three conditions.

- Boundary
 $w_1 = (1, 1), w_k = (m, n)$
- Seriality
 $w_k = (a, b), w_{k-1} = (a', b')$
 $\Rightarrow a - a' \leq 1 \wedge b - b' \leq 1$
- Monotony
 $w_k = (a, b), w_{k-1} = (a', b')$
 $\Rightarrow a - a' \geq 0 \wedge b - b' \geq 0$

To find the path with the lowest cost that meets these conditions, the following steps are applied.

1. Initialization
 $DTW(0, 0) = 0$
 $DTW(i, 0) = \infty$ for $i = 1, \dots, m$
 $DTW(0, j) = \infty$ for $j = 1, \dots, n$
2. Do for $i = 1$ to m
 Do for $j = 1$ to n
 $DTW(i, j) = d(x_i, y_j) +$

$$\min \begin{cases} DTW(i-1, j-1) \\ DTW(i-1, j) \\ DTW(i, j-1) \end{cases}$$

3. Output

Return $DTW(m, n)/n$

The obtained cost $DTW(m, n)$ is the distance between X and Y . The returned $DTW(m, n)$ is divided by n since DTW distance increases with the length of the training data. The system recognizes gestures on the basis of the training data. The distances for all the training data are calculated, and the training data with the shortest distance is identified. A gesture labeled with the training data is then output. The DTW algorithm can be used for multiple axes of accelerometers and gyroscopes. The DTW calculation is carried out for each axis, and the sum of the distances for all axes is used as the distance of the gesture.

3.3.2 Support vector machine

Our evaluation in SVM uses the mean and variance over first one second of the sequence as feature values. One 3-axis gesture sequence produces a 6-dimensional feature vector. There are two reasons for using the data of first one second: Firstly, using whole sequence causes delay since the processing is conducted after the gesture finishes. Secondly, recognition accuracy of one second was better than that of whole sequence since movements around the end points added noise.

Moreover, since the feature values extracted from

these sensors cannot be treated equally due to the different scales of acceleration and angular velocity, the D -dimension feature vectors were normalized by $\mathbf{Z}(t) = (\mathbf{X}(t) - \mathbf{M})/\mathbf{S}$, where $\mathbf{M} = (m_1, m_2, \dots, m_D)$ and $\mathbf{S} = (s_1, s_2, \dots, s_D)$ are mean $m_i = \frac{1}{N} \sum_{t=1}^N x_i(t)$ and standard deviation $s_i = \sqrt{\frac{1}{N} \sum_{t=1}^N \{x_i(t) - m_i\}^2}$ of the feature vectors before normalization $\mathbf{X}(t) = (x_1(t), x_2(t), \dots, x_D(t))$, and N is the number of samples. The mean and variance of $\mathbf{Z}$ become 0 and 1 after this conversion, respectively. For a real-time system, $\mathbf{M}$ and $\mathbf{S}$ are obtained from training data since data to be recognized have not been obtained before using the system.

SVM is a classification algorithm that often provides competitive or superior accuracy for a large variety of real-world classification tasks^[11]. Other classifiers such as C4.5, Self-organizing maps, and K-nearest neighbor are often used for activity recognition using accelerometers. Tendency of their performance is similar to that of SVM, however, SVM performed the best^[8]. Consider the problem of separating a set of training data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_J, y_J)$ into two classes, where $\mathbf{x}_i \in R^N$ is a feature vector and $y_i \in \{-1, +1\}$ is its class label. Assuming that the classes can be separated by the hyperplane, $\mathbf{w} * \mathbf{x}_i + b$, and no knowledge about the data distribution is given beforehand, the optimal hyperplane is the one with the maximum distance to the closest points in the training dataset. We can find the optimal values for w and b by solving

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i(\mathbf{w} * \mathbf{x}_i + b) \geq 1, \quad \forall i = 1, \dots, n. \end{aligned} \quad (1)$$

DTW^[7] is often used as a gesture recognition algorithm since DTW measures the similarity of two time series in detail. Since the mean and variance have information on the orientation and intensity of a body movement, but have no information on its trajectory, such as rotating arm clockwise or counterclockwise, gestures are preferably recognized with DTW. DTW, however, scans all the data in a time series, which causes delay and disables immediate recognition. Therefore, SVM^[11] can be an option for mobile devices that do not have enough processing resource.

In addition, unconscious and undefined gestures may occur since a device moves due to pressing a

button and walking with holding the device. A general way to cope with them in an actual use is to employ NULL class for a classifier. When the distance to the closest training data is not smaller than the threshold for DTW, or when the distance to the hyperplane is not larger than the threshold for SVM, the classifier judges the input gesture is not any of gestures to be recognized and does not output recognition result.

4. Results and consideration

4.1 Number and positions of sensors

4.1.1 Number

The recognition accuracy on average over the gestures, the maximum accuracy, and the minimum accuracy for each number of sensors are shown in Table 3. The rows show the results of accelerometer only, gyroscope only, and both. Tables 3(a) and 3(b) are for intra-subject analysis and inter-subject analysis for DTW, respectively. Tables 3(c) and 3(d) are for intra-subject analysis and inter-subject analysis for SVM, respectively. For example, for two sensors, the number of combination was ${}_9C_2 = 36$. "Ave." shows the average accuracy over the 36 combinations, and "Max." and "Min." show the maximum and minimum accuracies among the 36 combinations, respectively.

The results for DTW show that the accuracies remained almost the same when the number of sensors was increased. However, the results for SVM show that the accuracies slightly increase but this growth is throttled as the number of sensors was increased. Their tendency differ and they would be caused by complicated factors including loss of information due to feature extraction, curse of dimension, or parameter setting for SVM, therefore it is hard to investigate the effect of sensors and gestures from the results for SVM. Comparison of the intra-subject and inter-subject results for both DTW and SVM shows that the intra-subject ones were much better. This is because the training data included only the data for the subject making the gestures. Comparison of the results for the two kinds of sensors for both DTW and SVM shows that the gyro was slightly inferior to the accelerometer and that using both an accelerometer and gyro improved accuracy. This is related to the kinds of gestures, which is discussed in Section 4.2.

The calculation cost increases with the number of

sensors. The devices used were an Android-powered tablet (Nexus 7; Android 4.2 OS; NVIDIA Tegra 3 (T30L) Quad-core 1.3 GHz CPU) and an Android-powered phone (HTC One X; Android 4.0 OS; NVIDIA Tegra 3 (T30) Quad-core 1.5GHz CPU). Table 4 shows the operation time for extracting feature values and recognizing one gesture with SVM and for recognizing one gesture with DTW, with one sensor and with nine accelerometers and nine gyroscopes, respectively. This operation time is an average over 10,000 iterations and linearly increases in response to the number of sensors. The number of sequences for training data for DTW was one for each gesture (total of 27 sequences). Length of a gesture was 100 samples (2 seconds) on average. Even though the operation time could be longer if other applications are running in the background or the CPU is poor, the delay with SVM would not be significant, but the delay with DTW with multiple sensors would likely irritate the user and lead to less usability.

4.1.2 Position

To evaluate the position of sensors, the accuracy of one sensor for each position was measured. The results are shown in Figure 3. The horizontal axis shows the sensor positions corresponding to the number in Figure 2, and the vertical axis shows recognition accuracy. The results for the intra-subject analysis for DTW show 0.98-0.99 accuracy with negligible difference, between the number and kind of sensor. The results for the inter-subject analysis for DTW show that the accuracy was lower and that the differences between the positions were negligible. These data are supported by the data in Table 3: the range of accuracies among the same number of sensors was quite small.

The gyroscope results show uniform accuracy against sensor position. While some may think that shifting gestures do not involve gyration components, shifting gestures are actually not parallel shifting. They are sectoral shifting centered about the wrist or elbow, which produces angular values, resulting in accurate recognition with a gyroscope. Using an accelerometer and gyroscope together improves accuracy because they covers their drawbacks each other.

4.1.3 Summary

Recognition results for SVM are much affected by complicated factors such as feature extraction, curse

Evaluating Sensor Placement and Gesture Selection for Mobile Devices

Table 3 Recognition accuracy vs. number of sensors.

(a) Intra-subject analysis for DTW

		Number of sensors								
		1	2	3	4	5	6	7	8	9
Accelerometer	Ave.	0.989	0.992	0.992	0.993	0.993	0.993	0.994	0.994	0.994
	Max.	0.994	0.995	0.995	0.995	0.995	0.995	0.995	0.994	0.994
	Min.	0.982	0.985	0.985	0.988	0.989	0.991	0.992	0.993	0.994
Gyroscope	Ave.	0.991	0.992	0.992	0.992	0.992	0.992	0.992	0.992	0.992
	Max.	0.993	0.994	0.993	0.994	0.993	0.993	0.992	0.992	0.992
	Min.	0.988	0.990	0.990	0.990	0.990	0.991	0.991	0.991	0.992
Both sensors	Ave.	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998
	Max.	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998
	Min.	0.996	0.997	0.997	0.997	0.997	0.998	0.998	0.998	0.998

(b) Inter-subject analysis for DTW

		Number of sensors								
		1	2	3	4	5	6	7	8	9
Accelerometer	Ave.	0.682	0.692	0.694	0.696	0.697	0.697	0.698	0.698	0.697
	Max.	0.702	0.702	0.704	0.705	0.706	0.704	0.703	0.701	0.697
	Min.	0.661	0.674	0.682	0.687	0.688	0.691	0.693	0.695	0.697
Gyroscope	Ave.	0.638	0.639	0.639	0.639	0.639	0.639	0.639	0.639	0.640
	Max.	0.642	0.646	0.644	0.644	0.643	0.644	0.642	0.642	0.640
	Min.	0.631	0.629	0.632	0.634	0.633	0.635	0.635	0.637	0.640
Both sensors	Ave.	0.751	0.752	0.753	0.753	0.753	0.753	0.752	0.752	0.753
	Max.	0.763	0.762	0.759	0.760	0.757	0.756	0.756	0.755	0.753
	Min.	0.744	0.740	0.744	0.746	0.747	0.748	0.749	0.750	0.753

(c) Intra-subject analysis for SVM

		Number of sensors								
		1	2	3	4	5	6	7	8	9
Accelerometer	Ave.	0.849	0.891	0.902	0.907	0.909	0.910	0.910	0.910	0.910
	Max.	0.864	0.903	0.910	0.914	0.914	0.915	0.913	0.912	0.910
	Min.	0.827	0.871	0.894	0.899	0.902	0.905	0.906	0.907	0.910
Gyroscope	Ave.	0.702	0.737	0.753	0.761	0.770	0.778	0.785	0.792	0.797
	Max.	0.728	0.777	0.783	0.788	0.793	0.795	0.797	0.797	0.797
	Min.	0.696	0.719	0.728	0.731	0.743	0.747	0.755	0.763	0.797
Both sensors	Ave.	0.901	0.908	0.912	0.913	0.912	0.912	0.911	0.911	0.912
	Max.	0.910	0.915	0.920	0.917	0.917	0.916	0.914	0.912	0.912
	Min.	0.889	0.902	0.907	0.909	0.908	0.907	0.906	0.908	0.912

(d) Inter-subject analysis for SVM

		Number of sensors								
		1	2	3	4	5	6	7	8	9
Accelerometer	Ave.	0.584	0.701	0.747	0.775	0.794	0.806	0.815	0.824	0.829
	Max.	0.627	0.738	0.771	0.796	0.811	0.818	0.825	0.830	0.829
	Min.	0.530	0.643	0.692	0.743	0.767	0.787	0.807	0.820	0.829
Gyroscope	Ave.	0.469	0.521	0.550	0.574	0.592	0.609	0.625	0.642	0.650
	Max.	0.512	0.599	0.619	0.632	0.639	0.646	0.651	0.654	0.650
	Min.	0.455	0.480	0.501	0.517	0.535	0.550	0.566	0.580	0.650
Both sensors	Ave.	0.799	0.847	0.874	0.893	0.906	0.915	0.924	0.930	0.936
	Max.	0.817	0.871	0.893	0.908	0.920	0.924	0.931	0.934	0.936
	Min.	0.777	0.824	0.849	0.868	0.886	0.898	0.912	0.925	0.936

of dimension, and parameter setting, therefore gestures should be recognized through analysis in raw data. Using multiple homogeneous sensors is not meaningful much for accuracy. The sensor position does not affect accuracy much. Using an accelerometer and gyroscope together slightly improves accuracy.

4.2 Number and kinds of gestures

4.2.1 Number

We calculated the overall accuracy as we reduced the number of gestures one by one, starting from one with the lowest accuracy. The results are plotted in Figure 4; the gestures excluded at each step are shown beneath the graph by sensor type for one

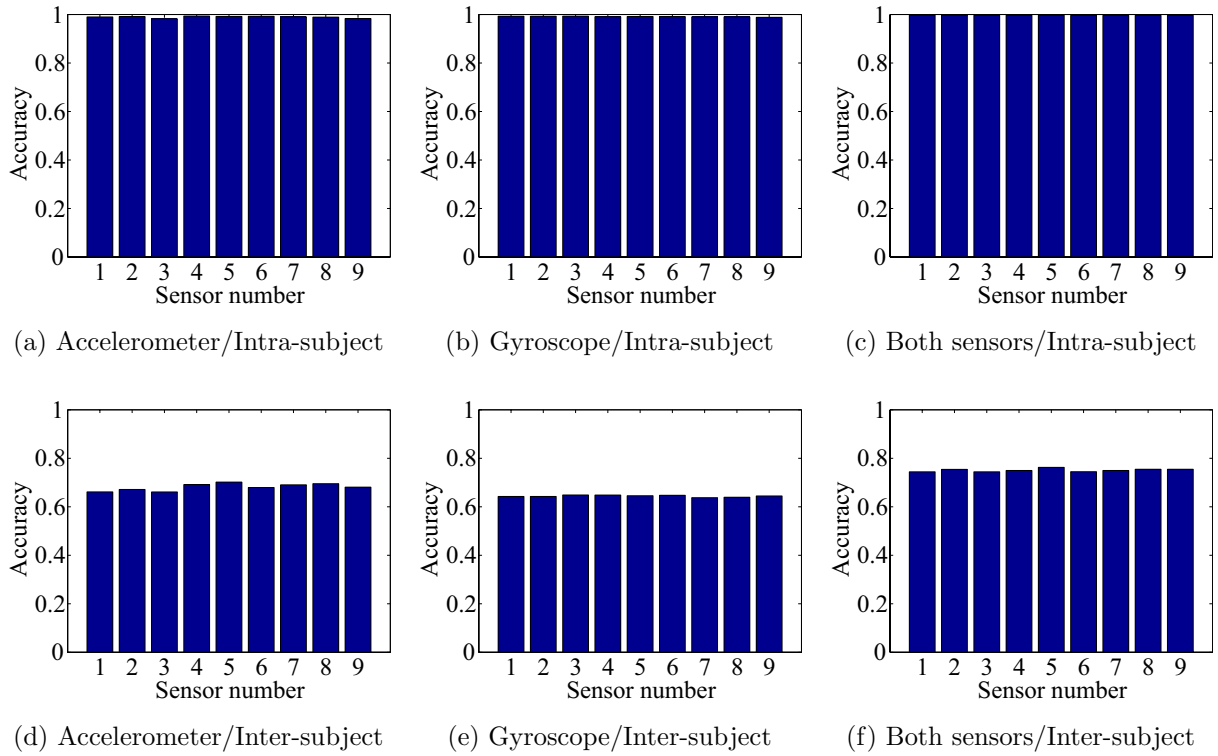


Fig. 3 Recognition accuracy vs. sensor position.

Table 4 Operation time for recognizing one gesture with SVM and DTW [ms].

Method	Sensor	1 sensor	
		1 sensor	9 accelerometers and 9 gyroscopes
DTW	Nexus 7	52	940
	One X	59	1100
SVM	Nexus 7	0.36	6.4
	One X	0.48	8.7

test subject. Specifically, 26 gestures were subject to recognition after excluding gesture 5 since gesture 5 had the lowest accuracy of the 27 gestures. (In this evaluation, sensor 5 was used for the accelerometer, gyroscope, and both sensors.) The accuracy for the accelerometer, gyroscope, and both sensors reached 1.0 when the number of gestures was respectively 13, 8, and 16. The results for the accelerometer show that, though the average accuracy for 27 gestures was 0.5+, an average accuracy of 0.9+ could be obtained for as many as 16 gestures by selecting ones that are easily recognized, and for as many as 22 by also using a gyroscope. The results excluded for the accelerometer, gyroscope, and both sensors were the tapping and shifting gestures. These results are for one test subject, but the same tendency was observed for the other subjects.

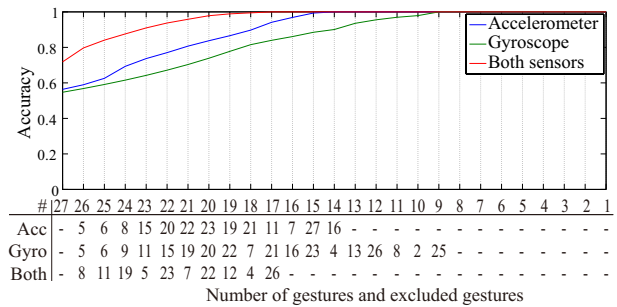


Fig. 4 Accuracy vs. number of gestures.

4.2.2 Kinds

The recognition accuracies for each gesture with sensors are shown in Table 5. From the intra-subject analysis, the accuracies of all gestures were virtually 1.0 regardless of sensor kind. This is because the training data included only data for the subject making the gestures. Moreover, the results between trials were consistent. From the intra-subject analysis for accelerometer, the accuracies for the “Lay cards” (12) and “Gather cards” (13) gestures were lower than those of the other gestures. This is because noise appeared in the data for these gestures when scraping the table surface.

The inter-subject accuracy was much less since the training data did not include data for the subject making the gestures. It was almost zero for gestures

Table 5 Accuracy for each gesture with sensor 5.

Gesture	Intra analysis			Inter analysis		
	Acc	Gyro	Both	Acc	Gyro	Both
1 	0.974	0.987	0.987	0.804	0.744	0.830
2 	1.000	1.000	1.000	0.692	0.736	0.665
3 	1.000	1.000	1.000	0.857	0.857	0.857
4 	1.000	0.984	1.000	0.697	0.810	0.841
5 	1.000	0.984	1.000	0.000	0.029	0.075
6 	0.984	1.000	1.000	0.043	0.016	0.189
7 	1.000	1.000	1.000	0.757	0.786	0.914
8 	1.000	1.000	1.000	0.425	0.813	0.756
9 	1.000	1.000	1.000	0.714	0.311	0.857
10 	1.000	0.986	1.000	0.686	0.600	0.800
11 	0.986	0.941	0.986	0.817	0.514	0.608
12 	0.943	0.986	0.986	0.586	0.668	0.683
13 	0.956	1.000	1.000	0.557	0.610	0.714
14 	1.000	1.000	1.000	1.000	0.971	1.000
15 	0.971	1.000	1.000	0.339	0.686	0.914
16 	1.000	1.000	1.000	0.557	0.200	0.386
17 	1.000	1.000	1.000	1.000	0.811	1.000
18 	1.000	1.000	1.000	1.000	1.000	1.000
19 	1.000	0.986	1.000	0.857	0.500	0.800
20 	0.987	0.973	0.987	0.711	0.145	0.415
21 	1.000	1.000	1.000	0.868	0.900	1.000
22 	1.000	1.000	1.000	0.681	0.667	0.922
23 	0.986	1.000	1.000	0.854	0.700	0.900
24 	1.000	0.986	1.000	0.760	0.832	0.864
25 	1.000	1.000	1.000	0.970	0.984	1.000
26 	1.000	1.000	1.000	0.787	0.629	0.614
27 	1.000	0.968	1.000	0.929	0.913	0.986
Average	0.992	0.992	0.998	0.702	0.646	0.763

5 and 6. These gestures were recognized as various gestures, such as “Swing twice to the left side quickly” and “Knock the board twice.” This is attributed to the high degree of freedom of tapping. In short, tapping involves the tapping points, strength, and timing and the state of the hand holding the board (not moving, absorbing shock, or hitting the left hand with the board).

In addition, the accuracy for gesture 2 was less than that for gesture 1. This is because tilting to the far side is physically more difficult than tilting to the near side. The accuracy for gesture 22 was less than that for gesture 21. This is because all the subjects held the board with their right hand, and shifting right is harder than shifting left. Gestures involving rotation and shifting are simple and easy to use as interface, but they are affected by the orientation of the device and the state of holding it.

The gyroscope accuracies for gestures 19 and 20 were less than those of gestures 21 to 24. This is because shifting up and down is done parallel to the sur-

face of the board while shifting right, left, diagonally up, and diagonally down involve sectoral movement, which produces a rotation component that appears in the gyration values. Accelerometer mainly obtains acceleration produced by parallel shift, shock, and tilting, while gyroscope obtains gyration produced by rotation movements. Therefore, combining an accelerometer with a gyroscope improves accuracy by approximately 6% since each sensor obtained different components.

Lastly, Table 6 and 7 show the sensors with the best accuracy of all sensor kinds and positions for each gesture with intra-subject analysis and inter-subject analysis, respectively. The intra-subject results show that most of all gestures marked the best accuracy with any accelerometer. Gyroscope worked well for gesture 12 and 13 since these gestures involve sectoral movement. The inter-subject results show that gestures are not recognized well only with one accelerometer. Gesture 5 and 6 are quite hardly recognized even with both accelerometer and gyroscope. Comparing to the result for fixed sensor as shown in Table Table 5, accuracies improved by selectively using appropriate kind and position of sensor. For these result, it would be meaningful to select gestures according to the sensor kind and position installed in a commercially available phone, or to decide sensor kind and position for the gestures that developers assume when they make a new device.

4.2.3 Summary

Some gestures had low reproducibility, which degraded recognition accuracy. The training for these gestures should thus be done using data captured from many users or the user who is to use the device. An adequately high degree of accuracy can be achieved without excluding many gestures by selecting gestures that do not conflict with each other and that have high reproducibility.

5. Application

As examples of gesture-based video game, several video games for Wii allow gesture input with accelerometers, which employ tiling and shifting gestures to attack and block while battle, and shifting and drawing a circle gesture with B button to move

Table 6 The best accuracy of all sensor kinds and positions for each gesture with intra-subject analysis.

Gesture	Accuracy	Sensor	Position
1 	0.987	1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
2 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
3 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
4 	1.000	1 acc 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
5 	1.000	1 acc 1 both	[1][2][5][6][8][9] [1][2][3][4][5][6][7][8][9]
6 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][6][8][9] [2][3][4][5][6][8] [1][2][4][5][6][7][8][9]
7 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8] [1][2][3][4][5][6][7][8][9]
8 	1.000	1 acc 1 gyro 1 both	[1][2][3][5][6][7][8][9] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
9 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
10 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][6] [1][2][3][4][5][6][7][8][9]
11 	0.986	1 acc	[1][4][5][6][8][9]
12 	0.986	1 acc 1 gyro 1 both	[4] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][7][8][9]
13 	1.000	1 acc 1 gyro 1 both	[4] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
14 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][5][6][8][9] [1][2][3][4][5][6][7][8][9]
15 	1.000	1 gyro 1 both	[1][2][3][4][5][7][8] [1][2][3][4][5][7][8][9]
16 	1.000	1 acc 1 gyro 1 both	[2][3][5][6][7][8][9] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
17 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
18 	1.000	1 acc 1 gyro 1 both	[2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
19 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [7] [1][2][3][4][5][6][7][8][9]
20 	1.000	1 acc	[1][3][6][7][9]
21 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
22 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
23 	1.000	1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
24 	1.000	1 acc 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
25 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
26 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
27 	1.000	1 acc 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]

Table 7 The best accuracy of all sensor kinds and positions for each gesture with inter-subject analysis.

Gesture	Accuracy	Sensor	Position
1 	0.844	1 acc	[2][3]
2 	0.781	4 acc	[1,2,7,9]
3 	0.871	1 acc	[3][6]
4 	0.857	1 acc 1 both	[7] [7]
5 	0.122	1 both	[2]
6 	0.189	1 both	[5]
7 	0.914	1 both	[5][7][8]
8 	0.827	1 gyro	[8][9]
9 	0.857	1 both	[1][2][3][4][5][6][7]
10 	0.843	2 both	[6,9]
11 	0.833	2 acc	[1,4][1,5]
12 	0.714	1 acc 1 both	[7] [1][7][8][9]
13 	0.729	1 both	[9]
14 	1.000	1 acc 1 gyro 1 both	[1][2][3][4][5][6][7][8][9] [7] [1][2][3][4][5][6][7][8][9]
15 	0.914	1 both	[5]
16 	0.557	1 acc	[5]
17 	1.000	1 acc 1 both	[1][2][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
18 	1.000	1 acc 1 gyro 1 both	[2][4][5][6][7][8][9] [1][3][4][5][6][7][8][9] [1][2][3][4][5][6][7][8][9]
19 	0.886	2 acc	[2,8]
20 	0.898	3 acc	[1,3,4]
21 	1.000	1 both	[5][7][8]
22 	0.938	1 both	[9]
23 	0.929	1 both	[1][2][3]
24 	0.896	1 both	[8]
25 	1.000	1 acc 1 both	[6][7][8][9] [1][2][3][4][5][6][7][8][9]
26 	0.816	1 both	[8]
27 	1.000	1 both	[4]

or use a skill ² ³.

An example of an application that was enhanced with gesture recognition technology is the gesture-based browser shown in Figure 5. This browser uses several of the gestures used in our evaluation. For example, tilt invokes *page-up/down*, swing invokes *go back/forward*, rap invokes *stop loading/reload*, shift invokes *scroll/zoom*, and circle invokes *add bookmark*.

We do not think all the functions should be assigned to gestures. However, assigning many functions to the limited number of buttons forces users to select functions from a pop-up menu after pressing a “menu” button. A first level has six to eight choices at most, forcing users to select “more” button to go to the next pop-up to select other functions. Such functions can be invoked with a single action

2: <http://www.gamefly.com/Rent-Kung-Fu-Panda-Legendary-Warriors/134180/GameplayControls/>

3: <http://www.gamefly.com/Rent-Mortal-Kombat-Armageddon/124700/GameplayControls/>



Fig. 5 A gesture-based browser.

by using a gesture.

Recent smartphones have a 5-inch display; for example, the GALAXY S III by SAMSUNG⁴ has a 4.8-inch display and the ONE X by HTC⁵ has a 4.7-inch display. Users sometimes use both hands to touch the display since the thumb usually does not reach the far side of the display. Even in this case, gesture-based interaction can be done with one hand (except for tapping and knocking) and does not interrupt other activities, such as eating and writing.

For the results obtained in the previous section, it has been confirmed that some gestures are hardly recognized and there is compatibility between sensor and gesture. These knowledge would be meaningful: sensor positions can be taken into consideration to achieve higher performance with the gestures that the developers assume when developing a new device, moreover, when developing an application for a commercially available device, the functions can be assigned to gestures that are relatively easy to be recognized considering the sensor position and kind installed in the device.

6. Conclusion

We investigated the effect of gesture recognition accuracy of the number and positions of sensors and of the number and kinds of gestures for mobile devices. We experimentally evaluated the accuracies

for 27 kinds of gestures measure using a board on which nine accelerometers and nine gyroscopes had been placed.

Using multiple homogeneous sensors did not improve accuracy by much. On the other hand, using both an accelerometer and gyroscope improved accuracy. Recognition accuracy did not depend on sensor position so much.

Gestures involving shifting the side on which the device was held and tilting the far side of the device, which produced a small amount of movement due to physical constraints, and tapping, which had less consistency due to the high degree of freedom, were barely recognized with an accelerometer and gyroscope together. However, 0.9+ accuracy was achieved for 16 kinds of gestures with an accelerometer and 22 kinds of gestures with both an accelerometer and gyroscope by selecting gestures that do not conflict with each other and do have high reproducibility.

Acknowledgment

This research was supported in part by a Grant-in-aid for PRESTO from Japan Science and Technology Agency and Microsoft Research Asia Windows Phone Academic Program.

Reference

- [1] Graeme, S. C., Svetha V., Geoff A. W. W., and Hung H. B.: Hierarchical Recognition of Intentional Human Gestures for Sports Video Annotation; ICPR 2002, Vol.2, pp.1082-1085 (2002).
- [2] HTK Hidden Markov Model Toolkit home page: <http://htk.eng.kcam.ac.uk/>.
- [3] Keogh, E., Chu, S., Hart, D., and Pazzani, M.: An online algorithm for segmenting time series; ICDM 2001, pp.289-296 (2001).
- [4] Liu, J., Wang, Z., Zhong, L., Wiekramasuriya, J., and Vasudevan, V.: uWave: Accelerometerbased Personalized Gesture Recognition and its Applications; PerCom 2009, pp.1-9 (2009).
- [5] Junker, H., Amft, O., Lukowicz, P., and Tröster G.: Gesture spotting with body-worn inertial sensors to detect user activities, Journal of Pattern Recognition, pp.2010-2024 (2008).
- [6] Murao K., Terada T., Yoshinari T., and Shojiro N.: A Context-Aware System that Changes Sensor Combinations Considering Energy Consumption; Pervasive 2008, pp.197-212 (2008).
- [7] Myers, C., and Rabiner, L. R.: A Comparative Study of Several Dynamic Time-warping Algorithms for Connected Word Recognition; The Bell System Technical Journal, Vol.60, pp.1389-1409 (1981).
- [8] Naya, F., Ohmura, R., Takayanagi, F., Noma, H., and Kogure, K.: Workers' Routine Activity Recog-

4: <http://www.samsung.com/uk/consumer/mobile-devices/smartphones/android/GT-I9300MBDBTU>
 5: <http://www.htc.com/uk/smartphones/htc-one-x/>

- inition using Body Movement and Location Information; ISWC 2006, pp.105-108 (2006).
- [9] Rabiner, L.R. and Juang, B.H.: A tutorial on Hidden Markov Models and selected applications in speech recognition; Proceedings of IEEE, pp.257-286 (1989).
 - [10] Ravi, N., Dandekar, N., Mysore, P., and Littman, M.L.: Activity Recognition from Accelerometer Data; AAAI 2005, pp.1541-1546 (2005).
 - [11] Vapnik, V.: The Nature of Statistical Learning Theory; Springer (1995).
 - [12] Westeyn, T., Brashear, H., Atrash, A., and Starner, T.: Georgia Tech Gesture Toolkit: Supporting Experiments in Gesture Recognition; ICMI 2003, pp.85-92 (2003).
 - [13] Wireless Technologies, Inc.: <http://www.wireless-t.jp>.
 - [14] He, Z.: Accelerometer Based Gesture Recognition Using Fusion Features and SVM; Journal of Software, Academy publisher, Vol. 6, No. 6, pp. 1042-1049 (2011).
 - [15] Joselli, M. and Clua, E.: gRmobile: A Framework for Touch and Accelerometer Gesture Recognition for Mobile Games; SBGAMES 2009, pp. 141-150 (2009).
 - [16] Choi, E.S., Bang, W.C., Cho, S.J., Yang, J., Kim, D.Y., and Kim, S.R.: Beatbox Music Phone: Gesture-based Interactive Mobile Phone using a Tri-axis Accelerometer; ICIT 2005, pp. 97-102 (2005).

(received Dec. 12, 2012, revised Apr. 30, 2013)

Biography

Kazuya Murao



Kazuya Murao is an Assistant Professor at Graduate School of Engineering, Kobe 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 investigating the wearable computing, ubiquitous computing, and context aware systems. He is a member of IEEE and IPSJ.

Tsutomu Terada (Member)



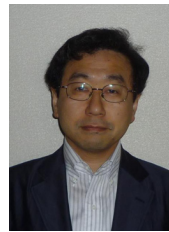
Tsutomu Terada is an Associate Professor at Graduate School of Engineering, Kobe University, Japan. He received B.Eng., M.Eng., and Ph.D. degrees from Osaka University in 1997, 1999, and 2003, respectively. Prof. Terada is working on wearable computing, ubiquitous computing, and entertainment computing.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

Ai Yano



Ai Yano is a researcher at Human Centric Computing Laboratories, Fujitsu Laboratories Ltd, Japan. She received B.Edu. degree from Shizuoka University in 1992. Ms. Yano has previously worked on human activity detection and analysis, and she is currently investigating context aware services.

Ryuichi Matsukura



Ryuichi Matsukura is a Senior Manager at Network Solution Business Unit, Fujitsu Ltd, Japan. He received B.Eng and M.Eng from Tohoku University in 1986 and 1988, respectively. Mr. Matsukura is currently investigating the human interaction support technology and context computing. He is a member of ACM.