Two-stage Recognition of Raw Acceleration Signals for 3-D Gesture-Understanding Cell Phones

Sung-Jung Cho, Eunseok Choi, Won-Chul Bang, Jing Yang, Junil Sohn, Dong Yoon Kim, Young-Bum Lee, Sangryong Kim

Two-stage Recognition of Raw Acceleration Signals for 3-D Gesture-Understanding Cell Phones

Sung-Jung Cho, Eunseok Choi, Won-Chul Bang, Jing Yang,
Junil Sohn, Dong Yoon Kim, Young-Bum Lee and Sangryong Kim
Interaction Lab.
Samsung Advanced Institute of Technology
{sung-jung.cho, eunseok.choi, wc.bang, jing.yang,
ji.sohn, kdy 2891, srkim}@samsung.com

Abstract

As many functionalities like cameras and MP3 players are converged to cell phones, more intuitive interaction methods are essential beyond tiny keypads. In this paper, we present gesture-based interactions and their two-stage recognition algorithm. Acceleration signals are generated from accelerometer. At the first stage, they are hierarchically modelled and matched as basic component and their relationships by Bayesian networks. At the second stage, they are further classified by SVMs for resolving confusing pairs. Our system showed enough recognition performance for commercialization; with 100 novice users, the average recognition rate was 96.9% on 11 gestures (digits 1-9, O, X*). The algorithms have been adopted in the world-first gesture-recognizing Samsung cell phones since 2005.*

Keywords: 3-D gesture recognition, acceleration, Bayesian networks, support vector machines, sensor-based interaction

1. INTRODUCTION

As the role of cell phones has evolved from mere voice communication devices to our daily life assistants, they have employed more functionalities like cameras, MP3, and web browsing. Even though the evolution enables users to enjoy various functions at any time, it incurs the usage problem of controlling many functions with tiny screen and keypads. Therefore, intuitive and interesting interaction methods are essential in mobile devices.

These days, a new kind of interaction technology that understands users' movement has emerged due to the rapid development of sensor technology. An accelerometer measures the amount of acceleration of a device in motion. Analysis of acceleration signals enables three kinds of gesture interaction methods: tilt detection, shake detection and gesture recognition [1,2].

The tilt detection algorithm interprets the posture of a device. When a user holds it in a static posture, its tilt angle is calculated by measuring the ratio of gravity components in tri-axis. It is used for moving cursors in a menu tree or virtual objects [3,4]. By combining tilt-based input and RFID-based object identification, information on physical objects can be browsed [5].

The shake detection algorithm interprets occurrences of users' shake movement. When a user shakes a phone, acceleration signals in a time interval are analyzed about whether they exceed threshold values. It is used for counting the number of walking steps in Fujitsu's cell phone F672i and Pantech's PH-S6500 [3]. Also, shaking patterns are used for identifying users and devices [6, 7].

The gesture recognition algorithm interprets dynamic movement patterns in the 3-D space. When a user draws a trajectory in the air for inputting commands or characters, the relationship between acceleration signals over the whole input is analyzed. We proposed a remote controller prototype, *Magic Wand*, for controlling TVs by gestures in the air [8-9] with accelerometers and gyroscopes. The 3-D trajectories are estimated by employing inertial navigation theory and then classified. Mäntyjärvi et al. also published a gesture-interactive remote controller for controlling DVD players with an accelerometer for recognizing eight gestures [10].

The goal of our research is to commercialize gesture-understanding technologies in cell phones for supporting interesting and intuitive interaction experiences to users. The previous researches have following limitations for the goal. First, tilt-based input is analogous to four-direction keys so that users' interests and curiosity are not so large. Second, the previous shaking detection algorithms do not handle applications with real-time response requirements. Third, previous gesture recognition algorithms are not of commercial quality because recognition approach after trajectory estimation requires gyroscopes which are not still fitted well to cell phones in terms of size and cost [8]. Also Mäntyjärvi et al.'s work lacks in the user-independent recognition capability [10].

In this paper, we present two-stage gesture recognition system and their applications in the world-first commercialized gesture interactive cell phone (Samsung SCH-S310 and E760 released in 2005) [11,12]. The gesture recognition algorithm enables the inputting of characters and symbols by drawing them in the air as an intuitive interaction. Acceleration signals generated according to users' motion are hierarchically modelled and matched as basic component and their relationships by Bayesian networks. Then they are further classified by SVMs for resolving confusing pairs.

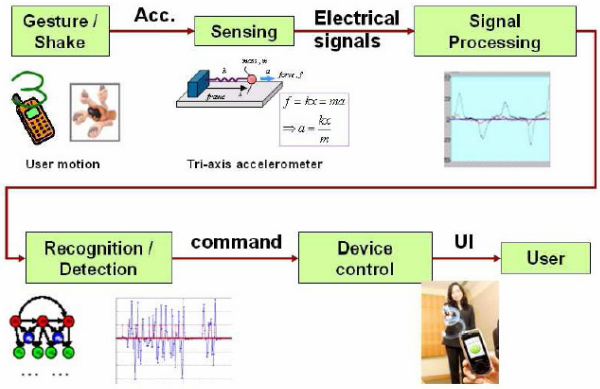


Fig. 1. Overview of Gesture Interactive Cell Phone

The rest of this paper is organized as follows. Section 2 describes the overview of the gesture interactive cell phone. Section 3 presents the gesture recognition algorithm. Section 4 shows experimental results and Section 5 concludes the paper.

2. GESTURE INTERACTIVE CELL PHONES

Figure 1 shows the overview of gesture interactive cell phones. When a user draws gestures, the movement is sensed by an accelerometer. Then the acceleration signal is processed and normalized. It is then classified into a gesture by the gesture recognition algorithm. Finally, the corresponding function is executed and shown to users.

2.1. Gesture-based Interaction Applications

Among the three interaction categories, we choose real-time shake-based interaction and gesture-based interaction for supporting intuitive interface and entertainment applications. They are targeted for young generations from 10's to 30's who are very sensitive to new trends. Samsung cell phone design groups utilize their expertise in developing application ideas with us.

The real-time shake-based interaction supports mainly entertainment applications. Figure 2 shows screen shots of *Dices* and *Random balls*. In *Dices*, the dices start rolling when the cell phone is shaken and stop when not shaken. In *Random balls*, the balls are rotating when shaken and randomly selected. The games look very realistic because they resemble our activity of shaking dices and balls in the real world. The shaking movement is also used for generating music; musical instrument sound like drum and portion of songs are played to the shaken time.

The gesture-based interaction supports command input for *speed dialing, gesture-to-sound generation, song navigation* and *message deletion* (Figure 3). By using

Fig. 2. Game applications: *Dices* and *Random balls*

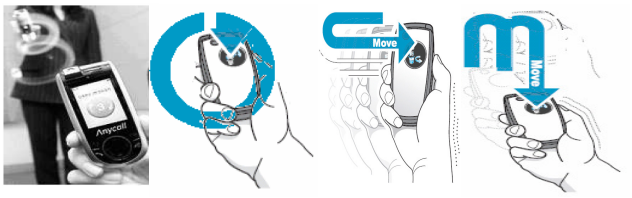


Fig. 3. Gesture applications: speed dialing, sound generation, MP3 control, and message

gestures, user does not need to pay attention to keypads, which may be useful for the blind. Also it is convenient for slide-up style cell phones whose keypads are hided by the upper screen.

In *speed dialing*, a user makes a phone call to the registered phone number by drawing digit shapes in the air. In *gesture-to-sound generation*, the user draws **O** or **X** on the air. Then, the sound corresponding to the gesture is played such as ' I love you' or 'Oh, No~'. It is used for expressing the user's emotion in interesting way. In *song navigation*, the user moves to the next or the previous song in the song list by shaking it rightward or leftward while listening to music. In *message deletion*, the user deletes messages like advertisement by shaking the phone vertically twice

Table 1 shows the phone functions and their gesture shapes [2]. In *idle* phone context where the phone waits for phone calls, a user can dial phone number or generate gesture sounds by drawing shapes of {1-9, O, X}. In *MP3 played* context , he can select the previous or the next song by gestures. In *message received* state, he can delete newly arrived message by gestures.

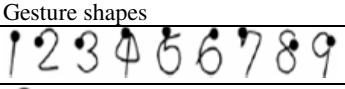
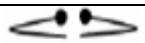
Context	Function	Gesture shapes
Idle	Speed dialing	
	Gesture-to-sound	
MP3 played	Song navigation	
Message received	Message deletion	

Table 1. Gesture phone functions and their corresponding shapes

2.2. Hardware Components

The gesture recognition capability requires additional hardware components: a gesture button, analog-to-digital converter (ADC), and tri-axis accelerometers. The gesture button is used for indicating the start and the ending time of gesture input. The ADC digitizes acceleration signals for digital processing. The accelerometer generates acceleration signals while the button is pressed. We use the accelerometer KXM 52 from Kionix [13] because it senses the range of +- 2G (gravity force) acceleration at 100 Hz sampling frequency which is enough for sensing accelerations due to hand movements

3. Gesture recognition algorithm

The recognition framework consists of five steps: calibration, preprocessing, feature extraction, recognition and confusing pair discrimination (Figure 4).

When a user presses the gesture activation button, the calibration module converts the digital sensor output values in the range of 0 to 255 into physical acceleration values m/s^2 by rescaling and translating them. The preprocessing step normalizes acceleration signal variations like slow or fast writing. The feature extraction step finds feature points and divides acceleration signals into primitives at those feature points. The recognition step matches the gesture input with Bayesian network-based gesture models and finds the most probable gesture model given the input. Finally, the confusing pair discrimination step is invoked for further discriminating frequently confusing gestures.

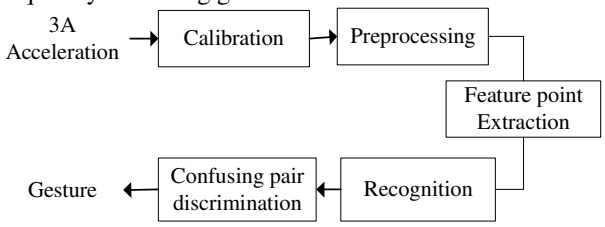


Fig. 4. Recognition framework

3.1. Preprocessing

The preprocessing step consists of four sub steps: motion-area detection, normalization, Gaussian smoothing and resampling.

- Motion-area detection

It detects the interval of a gesture motion in time. By utilizing acceleration signals within the motion interval, a gesture is recognized regardless of whether users make pause during gesture input.

The signals in motion interval are different from those in pause interval in that their variances are significantly larger. A sample point $\mathbf{A}(t) = [a_x(t), a_y(t), a_z(t)]$ is classified into motion area point by comparing it with K-previous samples:

$$\exists_{k\in\{1,\ldots,K\}} \left\| \mathbf{A}(t) - A(t-k) \right\| \geq threshold \qquad (1)$$

The motion area points in neighbor are merged consecutively. If several motion area groups are found, then the biggest motion area is selected.

- Norm normalization

The sensed acceleration signal contains not only the gesture movement acceleration but also earth gravity. The gravity amounts are different according to the posture of the sensor in the 3D space. Also, the gesture acceleration signal size changes according to the writing force. Those variations are normalized as follows.

The gravity components are approximately removed by subtracting the mean of accelerations at each time.

$$A_1(t) = A(t) - \overline{A} \qquad (2)$$

Writing force is approximately represented by the norm of tri-axis acceleration signals. Therefore, it is normalized by making the mean of norm as one:

$$A_2(t) = A_1(t) / \overline{\left\| A_1 \right\|} \qquad (3)$$

- Gaussian Smoothing

The obtained tri-axis accelerations contain measure noises and a user's unintended hand trembling. Hence it is necessary to get rid of such noises for extracting reliable features.

1-D Gaussian smoothing is adopted for each axis with Gaussian kernel of zero-mean. The convolution weight for smoothing is given as

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}} \qquad (4)$$

where σ is the standard deviation.

The data after Gaussian smoothing is given as follows:

$$\mathbf{A}_3(t) = \sum_{i=t-k}^{t+k} G(i-t) A_2(t) \qquad (5)$$

- Resampling

Because acceleration signals are sampled in equal-time interval, the number of sampled points is variable according to the gesture input speed; the fast moving interval has small number of points and vice versa. Therefore, the writing speed should be normalized.

Our approach is to resample the signals to have same lengths in the acceleration-length space. Then any neighboring two resampled points have same distance in the space. The length of acceleration signals from time 1 to T in the original space is represented as follows:

$$R_1^T(A) = \sum_{t=2}^{T} \left\| A_3(t) - A_3(t-1) \right\| \qquad (6)$$

The new points are resampled with following constraints:

$$\begin{aligned} A_4(t) = \{ & P(i) \text{ on traject ory } \{A_3(1),\ldots,A_3(T)\}, \\ & \text{s.t. } R_h^i(A_3) = U, \text{and } P(h) = A_4(t-1)\} \end{aligned} \qquad (7)$$

where U denotes predetermined unit length which is determined by experiment.

3.2. Feature Point Extraction

In the proposed gesture recognition system, primitives need to be extracted as basic modeling units. Here, a primitive means a portion of acceleration signals whose values increase or decrease monotonously within the interval. The boundary between successive primitives has the property that the signal value has local maximal or minimal value. A point is classified as local minima or maxima if it satisfies one of following conditions:

$$\begin{cases} \text{maxima if } A^w(t) \geq \forall_{k\leq K} A^w(t\pm k) \text{ for } \exists w \in \{x, y, z\} \\ \text{minima if } A^w(t) \leq \forall_{k\leq K} A^w(t\pm k) \text{ for } \exists w \in \{x, y, z\} \end{cases}, \qquad (8)$$

For the measured acceleration signals, the number of extracted local maximal or minimal points is generally excessive than the number of proper primitive boundaries. Therefore, they are filtered with the conditions that primitives should have significantly large

lengths and the local minima and maxima point should have sufficiently small and large value respectively.

3.3. Recognition Algorithms

3.3.1. Bayesian network modeling of relationships

A Bayesian network is a directed acyclic graph (DAG) whose nodes represent random variables and whose arcs relationships between them [14].

In this paper, the relationship between random variables are represented by the conditional Gaussian distribution. When a multivariate random variable X depends on $X_1,\cdots,X_n$, the conditional probability distribution is given as follows:

$$P(X=x|X_1=x_1,\cdots,X_n=x_n) = (2\pi)^{-d/2}|\Sigma|^{-1/2}\exp[-\frac{1}{2}[x-u]^T\Sigma^{-1}[x-u]] \quad (9)$$

The mean μ is determined from the linear weight sum of dependant variables $Z=[x_1^T,\cdots,x_n^T,1]$ as follows:

$$u = WZ^T. \quad (10)$$

where W is a $d\times k$ linear regression matrix, d and k are the dimension of X and Z^T respectively.

3.3.2. Gesture model

A gesture is represented hierarchically by modeling its primitives and relationships among the primitives. In the first level, a gesture model is composed of primitive models and their relationships. A primitive represents a portion of acceleration signals whose values increase or decrease monotonously. In the second level, a primitive model is composed of point models and their relationships [15,16]. Finally, a point model is represent as 3-D Gaussian distribution for the values of X, Y, and Z-axis acceleration. A point model corresponds to a single node in Bayesian networks.

A primitive model is composed of point models and their relationships, called as WPRs (within-primitive relationships). It is constructed by recursively adding mid point models and specifying their WPRs. A mid point is the point at which the lengths of the left and the right partial primitives are equal. Because all the points are resampled in equal distance in acceleration space, the time index of a mid point becomes the mean of those of two partial primitive end points.

A WPR is represented as the dependency of a mid point from two end points of a primitive. Figure 5 shows the recursive construction example of a primitive model. Figure 5 (a) shows an example of primitive instances. At the first recursion ($d=1$), IP_1 is added for modeling ip_1's with the WPR from EP_0 and EP_1 (Figure 5 (b)). At $d=2$, IP_2 and IP_3 are added for the left and the right partial primitives (Figure 5 (c)). This recursion stops when the covariances of newly added point models become smaller than a predetermined threshold.

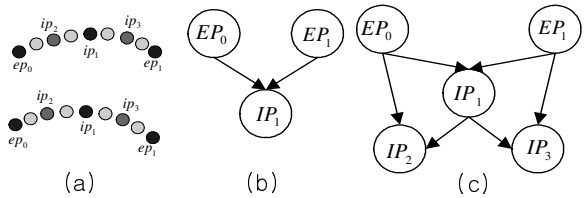


Fig. 5. Example of recursive construction of a primitive model

A gesture model is constructed by concatenating primitive models according to their input order and specifying inter-primitive relationships (IPR). IPRs are represented by dependencies among primitive boundary points. Figure 6 shows a Bayesian network based gesture model with N primitives and the recursion depth $d=2$. EP_i's are the primitive boundary point models and $IP_{i,j}$'s are the internal point models of the i-th primitive. The right end point of the previous primitive is shared with the left one of the following primitive. IPRs are represented by the arcs between EP_i's, and WPRs are represented by the incoming arcs to $IP_{i,j}$'s.

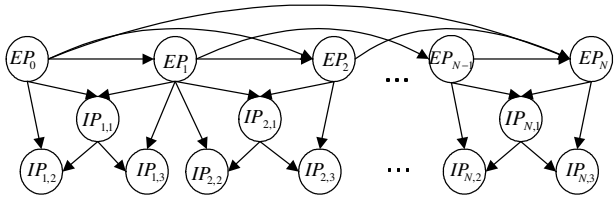


Fig. 6. Gesture model with N primitives and the primitive recursion depth of 2

3.3.3. Matching algorithm

Each gesture class m has a corresponding gesture model λ_m. A gesture input, a tri-axis acceleration sequence of $A(1),\cdots,A(T)$, is recognized by finding the gesture model λ^* which produces the highest model likelihood as follows:

$$\lambda^* = \arg\max_m P(\lambda_m|A(1),\cdots,A(T)) = \arg\max_m P(\lambda_m)P(A(1),\cdots,A(T)|\lambda_m) \quad (11)$$

The model likelihood is calculated by matching primitive internal point models (IP's) and primitive end point models (EP's) of gesture models (Figure 6) with the input point sequence. Because primitive boundaries are not explicitly specified in the input acceleration sequence, all the possible primitive segmentations should be searched. After a gesture input is segmented into primitives, primitive end points are matched to EP's. Then each primitive instance is recursively matched to mid points IP's. When a gesture model G with N basic primitive models matches the input

points $A(1), \cdots, A(T)$, and one stroke segmentation instance is denoted as $\gamma_i = (t_0, t_1, \cdots, t_N)$, $t_0 = 1 \le t_1, \cdots, \le t_N = T$ and the whole set as Γ, the likelihood is calculated as follows:

$$P(A(1), \cdots, A(T) \mid \lambda_m) = \sum_{\gamma_i \in \Gamma} \prod_{j=0}^{N} P(EP_j = A(t_j) \mid A(t_0), \cdots, A(t_{j-1})) \prod_{j=1}^{N} \prod_{k=1}^{2^d - 1} P(IP_{j,k} = ip_{j,k}(A(t_{j-1}, t_j)) \mid pa(IP_{j,k}))) \quad (12)$$

In Eq.(14), $A(t_{j-1}, t_j) = (A(t_{j-1}), A(t_{j-1}+1), \cdots, A(t_j))$ and $ip_{j,k}$ represents the k-th recursively sampled point of the j-th primitive input. The matching probabilities of EP's can be interpreted as the probabilities of global primitive value similarities and those of IP's as the probabilities of local primitive shape similarities.

The training algorithm of the proposed gesture models is found in [15,16].

3.4. Confusion Pair Discrimination

Because writers cannot get visual feedback of written trajectories during their writing, gestures with similar movement history such as **O** and **6** are frequently confusing as shown in Figure 7. The most effective feature to discriminate the two is positions of their last parts; the last part of **O** is located in the upper position and that of **6** is in the middle. However, position information is not explicitly available in the acceleration signals. The approach to integrate signals twice for estimating positions does not work in practice because of double integration error accumulation.

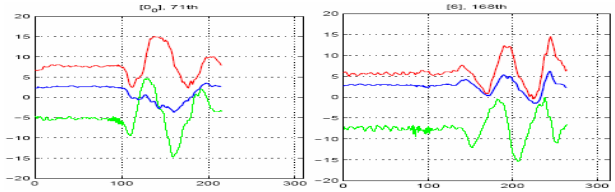


Fig. 7. Confusing pair: data '0' and '6'

Therefore, we adopt confusion pair discrimination algorithm based on support vector machines (SVM) [17]. SVM is well known for its high recognition performance in binary classes. It classifies binary classes by finding the hyper plane which separates the two classes with the maximum margin. One confusion pair is modeled with one SVM. For the case of **O** and **6**, all the data of class **O** and **6** are fed for training the SVM classifier of **O** and **6**.

For training and testing of SVM, gesture data is preprocessed with the steps of motion-area detection, norm-normalization and Gaussian smoothing as explained in Section 3. Then, the preprocessed data is resampled into the fixed number of points, *M*, because SVM requires fixed dimension of data. The length of acceleration signals from time *1* to *T* in the original space is represented as follows:

$$L = R_1^T(A) = \sum_{t=2}^{T} \left\| A_3(t) - A_3(t-1) \right\| \quad (13)$$

The new points are resampled with following constraints:

$$A^4(t) = \{P(i) \text{ on traject ory } A_3(1), \ldots, A_3(T), \text{ s.t. } R_h^i(A_3) = L/M, \text{ and } P(h) = A_4(t-1)\} \quad (14)$$

For SVM features, combination of X, Y, and Z-axis values and their norm values are used.

4. EXPERIMENTAL RESULTS

4.1. Gesture Shapes

Table 1 shows gesture shapes for all the applications. The shapes should be easy to learn and remember. In *gesture-to-sound*, **O** denotes positive sound because it usually means good or OK. **X** denotes negative sound because it usually means bad or not OK. In *song navigation*, '>' shape is mapped to the next song title and '<' is mapped to the previous one because the right and left direction is usually mapped to 'next' and 'previous'.

The design of digit shapes is complicated because writing styles are different according to nationalities, ages and genders of users. Therefore, we surveyed 120 people in our company on their writing styles. Table 1 presents the most popular digit shapes except 6 whose ending part becomes elongated for discrimination from 0.

4.2. Data Collection

In order to evaluate the proposed system, we collected data from 100 writers. Because the cell phone is targeted for young generation, all the writers were of 20's and 30's and did not have any experience of using it. The phone was attached to a PC by serial port during data collection. Acceleration signals were generated from the phone and then transmitted and saved in the PC.

Figure 8 shows the proper hand posture and the activity in data collection. A user draws gestures of 14 classes (1-9, O, X, <, >, M) by three times while looking at their labels and representative shapes shown on PC screen. They were asked to hold the phone with its screen facing up about 60 degree from the earth plane and write characters on imaginary vertical plane in front of them. Because the gesture activation button is located in the right side of the phone, the hand stays the most comfortably at the posture.

4.3. Evaluation Result

The recognition performance is measured with 11 gestures (1-9 and O, X) that are recognized at the same time in the *idle* state. The other states such as *MP3*

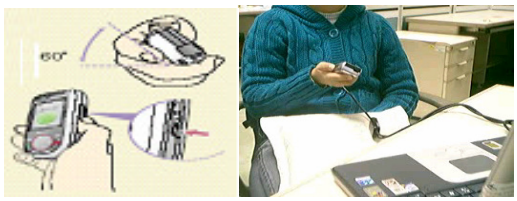


Fig. 8. Data collection activity

played or *message received* have only 2 classes and 1 class to recognize respectively. Therefore, the *idle* state has the most difficult recognition task.

In order to measure the user independent recognition performance, 100 users' database were divided into four folds without any common data. Then, four-folds generalization test was performed and the four recognition rates were averaged.

Table 3 shows the confusion table of BN based recognizer. The row and the column denotes data classes and the recognized class. The average recognition rate is 96.3%. The table shows that almost 30% of errors come from the pair of **O** and **6** (35 among 114 errors).

	Recognized as										
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**X**
0	258 (266)	2	0	0	1	0	16 (8)	2	0	0	0
1	1	289	1	0	0	0	0	3	1	0	1
2	0	0	282	5	0	5	0	0	0	0	2
3	0	1	2	288	1	1	0	0	0	0	1
4	0	0	1	0	289	0	0	3	1	0	0
5	0	0	0	3	1	289	0	1	1	0	0
6	19 (9)	0	0	0	1	0	246 (256)	1	0	4	0
7	0	1	0	0	3	0	0	255	0	1	1
8	0	0	0	0	0	2	0	2	255	1	0
9	3	0	0	0	2	0	9	3	0	249	0
X	0	1	1	1	0	0	0	1	0	0	291

Table 3. Confusion table for the gesture recognizer: ( ) in the cells denote results by SVM

The SVM discriminator for 0 and 6 was trained with the feature of 10 resampled points as explained in Eq. (14). Its four-fold generalization result was 97.6%.

By combining the BN classifier and the SVM pairwise discriminator, the recognition performance is further enhanced (Table 3, digits in ()). The combination scheme is as follows; First, a data is classified by the BN recognizer. If the recognition candidate with top score belongs to either **O** or **6**, SVM pairwise discriminator is called. Otherwise, the recognition candidate becomes the final recognition result. Table 3 shows that about half of errors in **O** and **6** are resolved so that the overall recognition rate is 96.9%. Even though SVM can classify **O** and **6** with the accuracy of 97.6%, the errors were reduced only by the half in the final result because the errors by SVM and BN are largely overlapped.

5. Conclusions

In the paper, we presented a new kind of interaction method based on a hand motion as intuitive and convenient input methods. People can make speed dialing, navigate songs, delete messages, and generate gesture sounds by drawing their gesture shapes in the air. The interaction method has the advantage that users do not need to pay attention to tiny keypads. Our algorithms were employed in the world-first motion-understanding cell phone, Samsung SCH-S310 and E760.

The gesture recognition algorithm models basic components and their relationships of acceleration signals. Acceleration signals are normalized by removing gravity components and writing speed variations. Then, the signals are divided into primitives at feature points. The primitives and their relationships are modeled with conditional Gaussian distributions. The robustness of its recognition capability is further enhanced by discriminating confusing pairs based on SVMs.

We evaluated the performance of the algorithms with 100 users who did not have any experience of using the phone. For 11 gestures (digits *1-9, O, X*), the average recognition rate was 96.3% with Bayesian networks. About 30% of the recognition errors came from the pair of (*O, 6*). With pairwise discriminator, the recognition rate was further enhanced into 96.9%.

Our next research is targeted for making motion-based interaction as one of basic interaction means in mobile devices. For the purpose, continuous gestures need to be recognized for inputting consecutive digits, English words or Korean characters, which will enable people to input short messages by drawing them in the air.

6. REFERENCES

1. E.-S. Choi, et. al, "Beatbox Music Phone: Gesture Interactive Cell phone using Tri-axis Accelerometer," IEEE Int. Conference on Industrial Technology, 2005
2. W.C. Bang, et. al, "SCH-S310: Gesture Understanding Cell phone," Proc. 7th Int Conf on Human Computer Interaction with Mobile Devices and Services, 2005
3. Article, "Digital Appliance, Controlled by Movement," Nikkei Electronics April 11, 2005
4. Rekimoto. Et. al., "Tilt Operations for Small Screen Interfaces (Tech Note)". *UIST*, 1996, pp. 167-168
5. A. Feldman, E.M. Tapia, S. Sadi, P. Maes, C. Schmandt, "ReachMedia: On-the-move interaction with everyday objects," *ISWC*, Osaka, Japan, 2005
6. S.N. Patel, J.S. Pierce, G.D. Abowd, "A Gesture-based Authentication Scheme for Untrusted Public Terminals," *UIST*, Santa Fe, USA, 2004
7. L.E. Holmquist, et. al, "Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts," *Ubicomp*, Atlanta, USA, 2001, pp 116-122
8. S.-J. Cho, et. al, "MagicWand: A hand-drawn gesture input device in 3-D space with inertial sensors," *IWFHR*, Tokyo, 2004
9. W.-C. Bang, et. al., "Self-contained spatial input device for wearable computers," *ISWC,* USA, 2003, pp. 26–34.
10. J. Mäntyjärvi, et. al, "Enabling fast and effortless customisation in accelerometer based gesture interaction," *MUM, College* Park, Maryland, 2004 pp. 25-31.
11. Samsung SCH-S310 product web site (in Korean), http://www.anycall.com/i_world/i_view/view_detail_2005.jsp?PFID=SCH-S310&page=1&real=feature&sar=
12. Samsung GSM E750 product manual, http://www.samsung.com/uk/products/mobilephones/mobilephones/sgh_e760msaxeu.asp (in English)
13. Kionix Inc., http://www.kionix.com/.
14. F. Jensen, An Introduction to Bayesian Networks, Springer, New York, 1996
15. S.-J. Cho et. al., "Bayesian network modeling of strokes and their relationships for on-line handwriting recognition," *Pattern Recognition*, vol. 37, no. 2, pp. 253–264, 2004.
16. S.-J. Cho, et. al., "Bayesian Network Modeling of Hangul Characters for On-line Handwritten Recognition," proc. 7th ICDAR, 2003, pp 207-211
17. N. Cristianini et. al., *Introduction to Support vector machines* , Cambridge University Press, 2000