Motion-Understanding Cell Phones for Intelligent User Interaction and Entertainment

조성정, 최은석, 방원철, 양징, 조준기, 기은광, 손준일, 김동윤, 김상룡
삼성종합기술원
{sung-jung.cho, eunseok.choi, wc.bang, jing.yang, handle.cho,
ek.ki, ji.sohn, kdy 2891, srkim}@samsung.com

지능형 UI 와 Entertainment 를 위한 동작 이해 휴대기기

Sung-Jung Cho, Eunseok Choi, Won-Chul Bang, Jing Yang, Joonkee Cho,
Eunkwang Ki, Junil Sohn, Dong Yoon Kim and Sangryong Kim
Samsung Advanced Institute of Technology

Abstract

As many functionalities such as cameras and MP3 players are converged to mobile phones, more intuitive and interesting interaction methods are essential. In this paper, we present applications and their enabling technologies for gesture interactive cell phones. They employ gesture recognition and real-time shake detection algorithm for supporting motion-based user interface and entertainment applications respectively. The gesture recognition algorithm classifies users' movement into one of predefined gestures by modeling basic components of acceleration signals and their relationships. The recognition performance is further enhanced by discriminating frequently confusing classes with support vector machines. The shake detection algorithm detects in real time the exact motion moment when the phone is shaken significantly by utilizing variance and mean of acceleration signals. The gesture interaction algorithms show reliable performance for commercialization; with 100 novice users, the average recognition rate was 96.9% on 11 gestures (digits *1-9, O, X*) and users' movements were detected in real time. We have applied the motion understanding technologies to Samsung cell phones in Korean, American, Chinese and European markets since May 2005.

Keyword : Motion recognition, sensor-based interaction, gesture recognition, accelerometer, gyroscopes, Bayesian networks, support vector machines, input devices

1. Introduction.

As the role of mobile phones has evolved from mere voice communication devices to our daily life assistants, they have employed more functionalities such as cameras, MP3, games and web browsing. Even though the evolution enables users to enjoy multimedia and games at any time, it gives users difficulties in using complicated 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 detects users' movement has emerged due to the rapid development of accelerometer sensor technology. The accelerometer measures the amount of acceleration of a device in motion. The 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 may be browsed [5].

The shake detection algorithm interprets occurrences

of users' shake movement. When a user shakes a phone, acceleration signals in time interval are analyzed about whether they exceed some 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 with accelerometers and gyroscopes [8-9]. VTT also published a gesture-interactive DVD remote controller with an accelerometer for recognizing eight gestures [10].

The goal of our research is to commercialize motion-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 it requires gyroscopes [8] or lacks in the user-independent recognition capability [10].

In this paper, we present two motion-understanding technologies with their applications in the world-first commercialized gesture interactive cell phone (Samsung SCH-S310, released on May 2005 in Korea) [11,12]. One is gesture recognition algorithm which classifies users' movement into one of predefined classes. It enables the inputting of characters and symbols by drawing them in the air as an intuitive interaction. The other is real-time shaking detection algorithm which detects in real time the exact moment when the phone is shaken significantly. It supports entertainment applications such as music generation.

The rest of this paper is organized as follows. Section 2 describes the overview of the gesture interactive mobile phone SCH-S310. Section 3 presents the gesture recognition algorithm. Then the real-time shake detection algorithm is explained in Section 4. Section 5 shows experimental results and Section 6 concludes the paper.

2. GESTURE INTERACTIVE MOBILE PHONES

Figure 1 shows the overview of gesture interactive mobile phones. When a user draws gestures or shakes the mobile phone, the movement is sensed by a tri-axis accelerometer. Then signal processing algorithm is employed to normalize the sensed accelerations. The normalized signals are classified into a gesture by the gesture recognition algorithm or converted into a shake time sequence by the real-time shake-detection algorithm. Then, the corresponding function is executed and its result is presented to users by user interface (UI).

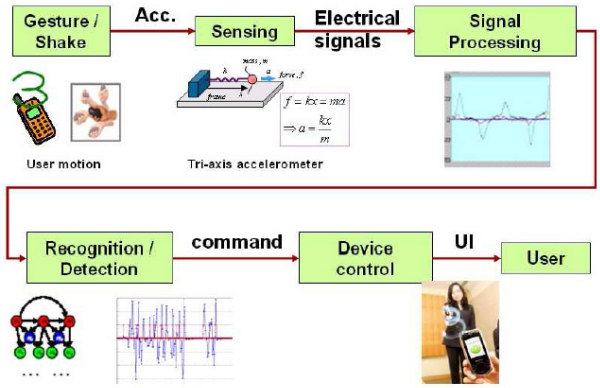


Figure 1. Overview of Gesture Interactive Mobile Phone

2.1 Gesture-based Interaction Applications

Among the three gesture interaction categories, we chose the gesture recognition and the real-time shake detection algorithm for supporting intuitive interface and entertainment applications. They are targeted for young generations of from 10's to 30's, who are very acceptable to new technologies, sensitive to new trends and have great interest in music. Samsung mobile phone design groups utilize their expertise in developing those commercial application ideas with us.

The shake detection algorithm supports mainly entertainment applications. Figure 2 shows screen shots

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

Figure 3. Music applications: *Orgols* and *Beat box*

of *Dices* and *Random balls*. In *Dices*, the dices start rolling when the mobile phone is shaken and keep rolling while shaken continuously. In *Random balls*, the balls are rotating when shaken and randomly selected when not shaken any more. The games look very realistic because they resemble our activity of shaking dices and balls in the real world

The algorithm also supports music applications: *Beat box* and *Orgols* (Figure 3). In *Beat box*, a musical instrument sound is played to the shaken time. Currently, about 30 kinds of recorded sounds are played such as drums, tambourines and human voices. In *Orgols*, one musical note is played to one shake time. A song composed of several musical notes is played to the rhythm of the user's shake movement.

The gesture recognition supports a gesture-based command input for *speed dialing, gesture-to-sound generation, song navigation* and *message deletion* (Figure 4). Because a user does not need to pay attention to keypads, the applications may be useful for the blind. Also it is convenient for slide-up style mobile phones whose keypads are hidden by the upper screen.

In *speed dialing*, a user makes a phone call to the registered phone number by drawing the digit shape in

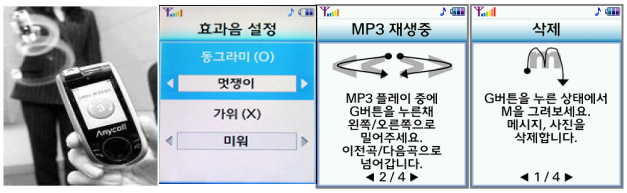


Figure 4. Gesture applications: speed dialing, sound generation, MP3 control, and message deletion

the air. In *gesture-to-sound generation*, the user draws **O** or **X** on the air. Then, enrolled sound to the gesture is played such as ' I love you' or 'Oh, No~'. It is used for expressing the user's emotion in a funny way. In *song navigation*, the user moves to the next or the previous song in the song list by shaking it rightward or leftward. In *message deletion*, the user deletes the newly arrived unwanted message such as advertisement or spam by shaking the phone vertically twice

Table 1 shows the gesture-enabled phone functions and the gesture shapes [2]. Gestures are used in three different phone contexts. First, the *idle* context indicates the state when the phone does not execute any function and waits for phone calls or a user's input. In this state, a user can dial a phone number or generate gesture sounds by drawing shapes of {1-9, O, X}. The *MP3 played* context denotes the state when the MP3 function is selected. He can select the previous or the next song by gesture. The *message received* state is entered when he receives a new message. He can delete it by shaking vertically twice. Because three contexts accept different gesture shapes, the number of recognition classes can be reduced by utilizing the context information.

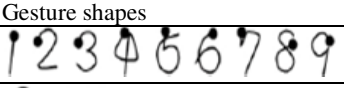
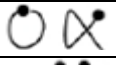
Context	Function	Gesture shapes
Idle	Speed dialing	
	Gesture-to-sound	
MP3 played	Song navigation	
Message received	Message deletion	

Table 1. Gesture-enabled phone functions and their shapes

2.2 Hardware Components

The gesture interactive mobile phone requires additional hardware components: tri-axis accelerometers, gesture buttons and analog-to-digital converter (ADC). The gesture button is used for indicating the start and the ending time of gesture input. The accelerometer generates acceleration signals when the button is pressed. The ADC digitizes acceleration signals for digital processing.

It is worth of noting the specification of the

accelerometer KXM 52 from Kionix [13]. By analyzing gesture signals from 100 users, we found that acceleration values of typical users' gestures range from –2G (gravity force) to +2G. The KXM 52 satisfies the requirement and is adequate for mobile phones; its size is small, supports 3.3V and less than 1.5 mA current.

2.3 Software Components

The mobile phone requires following SW components: calibration, axis-transformation, gesture recognition and shake detection module (Figure 5).

When a user presses the gesture activation button or selects gesture-enabled games, the accelerometer measures accelerations with the sampling rate of 100 Hz. 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 axis transformation step is employed for normalizing sensor layout variations according to mobile phone models. All the axis directions are aligned to those of a reference phone posture.

The gesture recognition module is executed for classifying the sensor signal into one of predefined gesture classes (presented in Section 3). When an application requires shaking time, the shaking detection algorithm is employed (presented in Section 4).

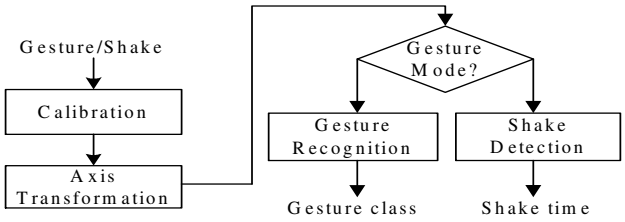


Fig 5. Data flow between SW components

3. Gesture recognition algorithm

The recognition framework consists of four steps for robust performance: preprocessing, feature extraction, recognition and confusing pair discrimination (Figure 6).

3.1 Preprocessing

The preprocessing step normalizes acceleration signal

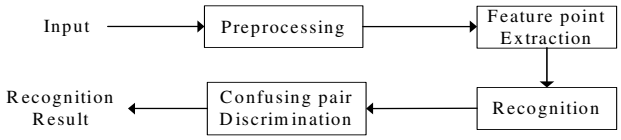


Figure 6. Recognition framework

variations according to the gesture input condition. For example, a user may hold a phone in various postures; its screen side may turn to the sky or turn to him. Also, he may write gestures very fast or very slowly.

The preprocessing step consists of four sub steps: motion-area detection, normalization, Gaussian smoothing and resampling. The motion-area detection step extracts the interval during which the real gesture motion is contained by comparing signal variances in sliding time windows. The normalization step removes gravity components from the input signals. The Gaussian smoothing step removes noises by users' hand tremblin g. The resampling step normalizes the writing speed variations according to users.

3.2 Feature Point Extraction

The feature point extraction step finds feature points where the property of acceleration signal changes drastically. Acceleration signals are divided into primitives at those feature points. Here, a primitive denotes a portion of acceleration signals whose values increase or decrease monotonously within the interval. Local minimal or maximal points in signal values are extracted as feature points.

3.3 Recognition

The recognition step matches the gesture input with gesture models and finds the most probable gesture model given the input. It employs Bayesian network for modeling acceleration primitives and their relationships.

A gesture is represented hierarchically by modeling its primitives and relationships among them. In the first level, a gesture model is composed of primitive models and their relationships. In the second level, a primitive model is composed of point models and their relationships. Finally, a point is modeled with 3-D Gaussian distribution for its X, Y, and Z-axis values. The

detailed description of models and training and test algorithms are found in found in [14,15].

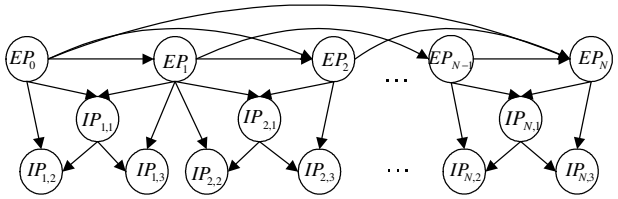


Figure 7. Gesture model with N primitives and the primitive recursion depth of 2

3.4 Confusion Pair Discrimination

The confusing pair discrimination step is invoked for further discriminating frequently confusing gestures. For instance, gestures with similar movement history such as **O** and **6** have similar acceleration signals and are frequently confusing. The only way to discriminate the two is to examine positions of their last parts; the last part of **O** is located high and that of **6** is in the middle. However, position information is not explicitly included in the acceleration signals.

As the discrimination method, we employed support vector machines (SVM) [16]. They are well known for the high recognition performance in binary classes by finding the hyperplane with the maximum margin between classes. 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 of the SVM.

4. REAL-TIME SHAKE DETECTION ALGORITHM [1]

To detect the motion for *Dice* and *Random balls* applications, we compared the predefined threshold value ρ and maximal standard deviation $\sigma_{\max}$ of the acceleration signals A_{bx}, A_{by}, and A_{bz} for the moving window size *n*. This can be expressed as follows:

$$\lambda(k) = \begin{cases} 1, \text{if } \sigma_{\max} \geq \rho \\ 0, \text{otherwise,} \end{cases} \quad (1)$$

where *k* is the current sample, and $\sigma_{\max} = \max\{\sigma(A_{bx}), \sigma(A_{by}), \sigma(A_{bz})\}$ for the moving window size *n*. Using $\lambda(k)$ in Eq. (1), the start sample k_s and end sample k_e of motion is determined when $\lambda(k)$ change from 0 to 1 and from 0 to 1, respectively. Figure 2 shows the relation between the state of the shaking motion and the graphic user interface of *Dice* application.

To decide the start time of sound for the *Beat box* and *Orgol* applications, we compared the predefined threshold values $\{\eta_1, \eta_2\}$ with the difference between the acceleration signal of a current sample *k* and that of previous sample *k*-1 for each axis. The detail expression is as follows:

$$\xi_m(k) = \begin{cases} 1, \text{if } |\delta \mathrm{A}_{\mathrm{bm}}(k)| \geq \eta_1 \text{ and } \mathrm{A}_{\mathrm{bm}}(k) > \eta_2 \\ 0, \text{otherwise,} \end{cases} \quad (2)$$

where *m* can be *x*, *y*, and *z* for the acceleration signal of *x*,*y*, and *z* axis, and $\delta A_{bm}(k) = A_{bm}(k) - A_{bm}(k-1)$. The start time t_s of motion is determined by *decision-making process* using $\xi_x(k), \xi_y(k), \xi_z(k)$ in Eq. (2).

5. EXPERIMENTAL RESULTS

5.1 Gesture shapes

Gesture shapes were designed with focusing on the easiness of remembrance. Table 1 shows gesture shapes for all the applications. 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 and '<' is mapped to the previous one because the right direction is usually mapped to 'next' and the left usually mapped to '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. For each digit, several writing shape candidates were presented to and selected by them.

Table 2 shows the survey result. For each digit, different writing styles are denoted by the subscripts of a,

b, c and so on. The population ratio of each style is denoted beside the digit label. Because a mobile phone has a limited memory size and computing power, we select the most popular shape from each digit and build the gesture model for it. However, in case of **6**, 6_b is chosen even though 6_a is more popular because the shape of 6_a is so similar to that of 0_a that both are frequently confused. The selected shapes are denoted with thick fonts in Table 2.

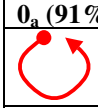
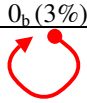
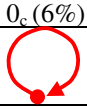
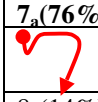
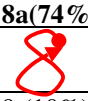
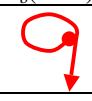
0_a (91%)	0_b (3%)	0_c (6%)	**1_a (92%)**	1_b (7%)
				
2_a (80%)	2_b(20%)	**3(100%)**	**4_a (94%)**	4_b(6%)
				
5_a(80%)	5_b(17%)	5_c (4%)	**6_a(7%)**	6_b(93%)
				
7_a(76%)	7_b(2%)	7_c(16%)	7_d(6%)	**8a(74%)**
				
8_b(14%)	8_c(8%)	**9_a(76%)**	9_b(13%)	9_c(10%)
				

Table 2. Writing shapes of digits and their popularity (Dots denote writing-starting points)

5.2 Data Collection

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

A user draws gestures while looking at their labels and representative shapes shown on PC screen. They were asked to hold the mobile phone with its screen facing up about 60 degree from the earth plane and write characters on imaginary vertical plane. Because the gesture activation button is located in the right side of the phone, the hand is comfortable at the posture. Figure 11 shows the hand posture and one picture of a user in data collection. All the data collection activities were video-recorded for the purpose of further analysis.

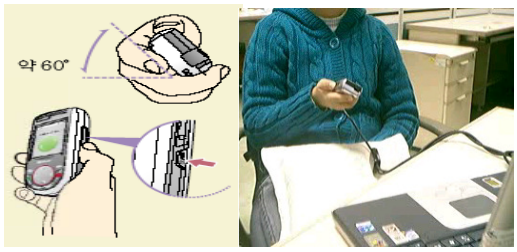


Figure 8. Data collection activity

For experimentation, each writer wrote data from 14 classes (1-9, O, X, <, >, M) by three times. For shake detection, he shook the phone rhythmically three times in horizontal and vertical direction.

5.3 Gesture Recognition Performance

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

Among the collected data, noisy data were removed. Because users were not used to the phone, they sometimes released the activation button too early without inputting meaningful gestures. Therefore, the acceleration data with less than 0.4 seconds length or without any movements were discarded.

In order to measure the user independent recognition performance, 100 users' database was divided into four folds. In each turn, three folds were used for training and the other for testing. Then the four recognition rates were averaged.

Table 3 shows the confusion table of the Bayesian-network (BN) based recognizer. The column denotes the data classes and the row denotes 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). SVM is employed to discriminate the confusing pair. By combining the BN classifier and the SVM pairwise discriminator, the recognition performance is further enhanced (Table 3, digits in ()). About half of errors in **O** and **6** are resolved so that the overall recognition rate is 96.9%, which was determined to be sufficient for commercialization by the Samsung mobile phone manufacturing division.

	Recognized as										
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**X**
0	258	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	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 where 6 and 0 meets denote errors by SVM

5.4 Evaluation of Shake Detection Algorithm

To test the shake detection algorithm for *Dice* and *Random balls* applications, we shake the phone once and several times. Figure 9 shows the results of the shake detection method using Eq. (1). The start and end time of the shaking motion are determined when $\lambda(k)$ changes from 0 to 1 and from 1 to 0, respectively. By adjusting the predefined threshold value, it is possible to control the sensitivity of shake detection algorithm.

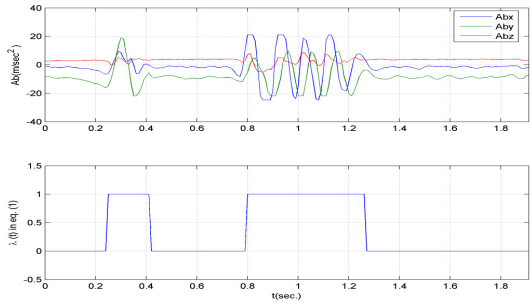


Figure 9. Results of the shaking detection for *Dice* and *Random ball* applications.

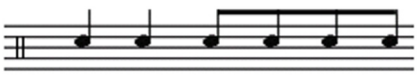


Figure 10. A Part of musical note for percussion instruments

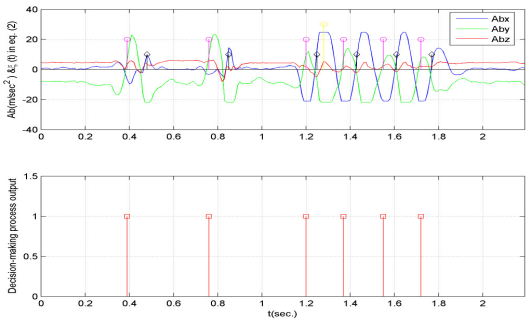


Figure 11. Results of the shaking detection for Beatbox and Electric orgel applications.

The phone has been shaken to the musical note for percussion instruments (Figure 10) to test the shake detection algorithm for *Beat box* and *Orgol* applications. Its results are depicted in Figure 11. The symbol ◇ and ◦ in Figure 11 are $10\xi_x(k)$ and in Eq. (2) for A_{bx}, and A_{by} respectively, where the constants have been multiplied by $\xi_x(k)$ and $\xi_y(k)$ to represent the results more effectively. The symbol □ in Figure 11 is the start time t_s of shaking motion gotten from the *decision-making process* using $\xi_x(k)$ and $\xi_y(k)$. These results demonstrate that it can be used as percussion instruments for entertainment.

6. Summary

As cell phones have evolved into our daily life assistants, the need for a convenient input method beyond their tiny screens and keypads has become much stronger. In the paper, we presented a new kind of interaction method based on a hand motion which was employed in the world-first motion-understanding cell phone, Samsung SCH-S310. It is convenient because users do not have to pay attention to tiny screens and keypads while inputting commands. Also, it is natural and intuitive for entertainment applications with shaking activities such as musical instruments and dice games.

We have developed two kinds of algorithms for supporting motion-understanding applications: gesture recognition algorithm and real-time shaking detection algorithm. The former supports a gesture-based input method for speed dialing, music song navigation, message deletion and gesture-to-sound generation. It classifies users' movement into one of predefined gestures by modeling acceleration primitives and their relationships with Bayesian networks. The recognition performance is further enhanced by discriminating confusing pairs based on support vector machines. The latter algorithm supports entertainment applications such as music instruments and dice games. It detects the exact motion time when the phone is shaken significantly by calculating the variance and the mean of signals.

We evaluated the performance of the algorithms with 100 users who do 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*). Therefore, the confusing pair was separately modeled and classified by support vector machines. Then, the recognition rate was further enhanced into 96.9%. For the shake detection algorithm, it detected users' movement moment in real time with negligible delay.

We have found that users have positive attitudes to the proposed technologies by monitoring mobile phone users' communities: *Daum* SCH-S310 Café (more than 2,000 members) [17] and *Cetizen* SCH-S310 community [18]. Articles on the sites show that the motion-understanding applications are interesting and intuitive even though they require some learning time for getting used to them. Also, the proposed technologies have been employed in Samsung phones targeted to USA, Chinese, Russian, and European markets.

Our next research is targeted for making motion-based interaction as one of basic interaction means in mobile devices by extending the gesture recognition capability to consecutive digits, English words or Korean characters drawn in the air and identifying movement directions in the shaking applications. We expect that it will give users more pleasances and convenience.

REFERENCES

1. E.-S. Choi, et. al, "Beatbox Music Phone: Gesture Interactive Mobile Phone using Tri-axis Accelerometer," IEEE Int. Conference on Industrial Technology, 2005
2. W.C. Bang, et. al, "SCH-S310: Gesture Understanding Mobile 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 SCH-S310 product manual, http://downloadcenter.samsung.com/content/UM/200505/20050516094238218_SCH-S310_UG_SKT_Rev1.0_050516.pdf (in Korean)
13. Kionix Inc., http://www.kionix.com/.
14. 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.
15. S.-J. Cho, et. al., "Bayesian Network Modeling of Hangul Characters for On-line Handwritten Recognition," proc. 7th ICDAR, 2003, pp 207-211
16. N. Cristianini et. al., *Introduction to Support vector machines* , Cambridge University Press, 2000
17. Daum Café SCH-S310 users' community, http://cafe.daum.net/S310?nil_profile=p&nil_Cafetxt=1 (in Korean, membership is required)
18. Cetizen SCH-S310 users' community http://moim.cetizen.com/sch-s310 (In Korean, membership is required)