Recognizing and Analyzing of User’s Continuous Action in Mobile Systems

Jonghun BAEK, Ik-Jin JANG, Nonmembers, and Byoung-Ju YUN, Member

SUMMARY As a result of the growth of sensor-enabled mobile devices, in recent years, users can utilize diverse digital contents everywhere and anytime. However, the interfaces of mobile applications are often unnatural due to limited computational capability, miniaturized input/output controls, and so on. To complement the poor user interface (UI) and fully utilize mobility as feature of mobile devices, we explore possibilities for a new UI of mobile devices. This paper describes the method for recognizing and analyzing a user’s continuous action including the user’s various gestures and postures. The application example we created is mobile game called AM-Fishing game on mobile devices that employ the accelerometer as the main interaction modality. The demonstration shows the evaluation for the system usability.

key words: human computer interaction, continuous action, mobile game, accelerometer, gesture recognition, posture estimation

1. Introduction

Advances in mobile computing and micro electro mechanical systems (MEMS) enable the development of games which lead about user’s interest on the mobile phone, PDAs, and other portable devices. Presently, mobile devices have been made popular to everyone, and users always carry with them. In this sense, mobile games are becoming more popular, in particular with the recent release of the portable devices such as Sony’s PlayStation Portable or Samsung’s SCH-G100/SPH-1000. It shows that this area has huge potential for growth.

Traditional desktop-based UI has been developed on the basis that user’s activities are static states. UI design for desktop devices can use all of its visual resources. The representative desktop-based interaction mechanisms are keyboard, mouse and joystick. In general, these are very graphical and still more detailed for desktop-based applications. In contrast, interaction mechanisms of mobile devices can not utilize all or any of their visual resources by reasons not only that activities of users are dynamic states [1] but also mobile devices have the limited resource and the small LCD display. Games on mobile devices rely on key-pad, stylus-pen and input-panel. When using these mechanisms to manipulate mobile devices, it may lead to the time delay and the difficult operation because the small button may be manipulated repeatedly. So, it may make the users to lose their interests in the games [2].

Several studies have proposed some techniques about input/output interaction methods to solve these problems; voice recognition and gesture recognition are representative cases [1]–[4]. Voice recognition comes hard to apply to it because it is unsuitable for UI of mobile games. The other way to design UI is to recognize the gesture of the user. The user’s gesture-based input by interaction between human and computer enable user to approach easily more than desktop PC. The gesture recognition needs additional devices that can detect user’s gesture and must be able to embed in mobile device. The trend of this field is much using the embedded camera [5]. The camera-based UI may be uncomfortable for use because user’s gesture must be transmitted to mobile device under condition that user carries it in hand. Additional, the camera is comparative high cost. Therefore, it needs new method that can detect user’s gesture easier and faster in mobile environment. In other words, proper sensors are necessary to detect action or to recognize gesture of the user in the case of mobile and portable devices, and a processing capacity and mobile convenience should be considered before applying sensor. MEMS accelerometer is suitable sensor that coincides in this purpose.

To use the gesture as input/output interface of mobile devices, each gesture should be definitely discriminated each other. Otherwise, useless situation or inappropriate gestures can cause malfunction. Therefore, we sort context information in context-aware computing.

In this paper, to supplement lacking the UI of mobile devices, we propose the novel UI that recognizes user’s gestures, such as throwing, tilting, pushing & pulling, and snatching, and estimates the user’s posture. The application example we created is a fishing game. Although 2-axis accelerometer expresses the user’s actions in detail less than 3-axis accelerometer, 2-axis accelerometer enable to detect the gestures and postures used in this research. For this reason we used 2-axis accelerometer, it can later easily combine them to form 3-axis accelerometer.

To improve efficiency that is expected by applying the accelerometer (ADXL202EB), with considering many restriction elements such as processing capacity, we propose the structure of the system that embedded in the accelerometer and the algorithm of the signal processing of the accelerometer for mobile devices, which is the motion-based interaction technology and context-aware computing technology.
2. Related Works

There have been various studies for accelerometer based gesture interaction [6]–[9]. These studies are classified into two categories: using the static acceleration and using the dynamic acceleration. The former case is that user’s gesture is relatively slow or small movement such as tilting, and rotating, and its complexity is low. The latter case is that user’s gesture is relatively fast or large movement such as throwing, swing, snatching, and pushing & pulling more than the former case, and its complexity is high. Since a control event is generated by detecting the only one gesture, each of these research areas is discrete gesture recognition.

In this paper, we use not only static acceleration (right/left tilting) but also dynamic acceleration (throwing, snatching, pushing, and pulling). Additionally, we use the user’s posture as interaction information. The key point of our study is to recognize the user’s various gestures and postures generating in order of precedence.

3. Context Information

Unlike designing UI for traditional desktop applications, mobile devices can sort complex context information. Therefore many kinds of context information in context-aware computing environment are sorted [3], [10].

Traditional desktop applications wait until user inputs data through the keyboard or mouse before execute special action. Context-aware applications with sensors, however, can offer or require information even through non-conscious input of user by estimating motion of user at specific time and in particular place. Context-aware computing analyzes changing contexts in ubiquitous computing environment, and discriminates whether the information is valid or not, and then it generates the control event that requests or delivers information to execute the application.

In this paper, we detect context information called the “gesture” and “behavior”, and calculate displacement amount of computer from the acceleration in order to gesture recognition and posture estimation. These user’s gestures and behaviors are very important factors in ubiquitous computing environment. For examples, pulling & pushing can be used to control the TV (e.g. zoom in and out the display, etc.), a tilting can be used to control the TV (e.g. volume and channel control, etc.) and scrolling the display of mobile devices, a snatching can be used to control the bell of the cellular phone and to converter the windows of the PDAs, etc., and a posture can be used to automatically turn on mobile devices or receive any information from the object that attached RFID tag when user puts mobile devices toward their chest in ubiquitous computing environment. In particular, the user’s posture estimation technology can correct the various postures of the user such as walking, running and sitting in daily life as well as be applied to the sport science by the fusion of the sensors.

4. System Description

UI for mobile devices is designed by a general-structure that is available to the sensor’s output at the same time in diverse control interface to increase the expected efficiency by applying the sensor, because mobile devices have many restriction elements such as processing capability, limited resources (e.g. memory, battery). Figure 1 is our system that is designed to use the accelerometer by a general-purpose in mobile devices.

Figure 2 shows that the system generates the control event by processing the signal detected from the accelerometer in software structure of mobile devices with the RTOS (real time operation system) environment. The accelerometer’s output is passed through a dispatcher that asks the GPIO for the signal of the accelerometer and passes through two parallel filters; a lowpass filter for a static acceleration such as tilt and gravity and a highpass filter for a dynamic acceleration such as vibration, shock, and impulse. The lowpass and highpass filtered output then goes to a signal processing for each application and then it generates a gesture-based control event and controls or executes the applications. To decrease overheads that are generated by at-

![Fig. 1](image1.png)

**Fig. 1** Architecture of mobile devices that embedded the accelerometer.

![Fig. 2](image2.png)

**Fig. 2** Block diagram of mobile devices with the RTOS environment.
taching the accelerometer to mobile devices, we designed the structure that control events do not pass to the control module but to a task module.

5. Accelerometer Signal Processing for Continuous Action Recognition

In the game of the existing mobile devices, control of the direction and action of a character is done by using the keypad. In our hand gesture- and posture-based mobile interaction technique, we show the usability for the use of the accelerometer that is embedded in mobile devices and the effect of the input.

In the process of the user’s motion recognition, actually, there are several cases that the acceleration signal for the user’s action often misrecognized. For examples, there are malfunction by the user’s hand tremble, the non-necessary gesture, and misrecognition. In addition, in the case that several gestures are occurred almost at the same time in one action, how we can distinguish each gesture. To resolve these limitations and problems, we proposed that the signal processing method for the gesture recognition and posture estimation is to define each user’s posture and gesture stage by stage as considering a typical accelerometer signal for the action. Namely, a representation scheme for user’s action is necessary for the description of each posture and gesture. Therefore we use a finite state machine that has a finite number of possible stages.

5.1 Scenario for the Fishing Game

We assume that the action for fishing game consists of the nine stages as followings:

- In the initial stage, the position of the mobile phone is assumed that user is gripping the mobile phone in an attention state (see Fig. 3 (a) and section A in the Fig. 4 (b)).
- First, in the beginning stage, the user takes the mobile phone toward their chest (see Fig. 3 (b) and Sect. B in the Fig. 4 (b)). It informs the beginning of the game by estimating of the user’s posture (see Fig. 3 (c) and Sect. C in the Fig. 4 (b)).
- Third, in the throwing stage, we determine the movement distance, velocity, power, and path-direction of the fishing rod for the user’s action (see Fig. 3 (d) and Sect. D in the Fig. 4 (b)).
- Fourth, in the waiting stage, at the time, the user takes the mobile phone toward their chest and then waits a bite of the fishes (see Fig. 3 (e) and section E in the Fig. 4 (b)).
- Fifth, in the snatching stage, snatching motion that user tried to land it immediately when the fish are on the feed (see Fig. 3 (f) and Sect. F in the Fig. 4 (b)).
- Sixth, in the pull & push stage, pulling & pushing gesture indicate that user pulls the fishing rod toward user’s chest like user reels in fish in real fishing to consume the power of the fish, and then user watches the mobile phone to check the state of the fish (ex. power of the fish) (see Fig. 3 (g) and Sect. G in the Fig. 4 (b)).
- Seventh, in the right and left tilting stage, tilting gesture indicate that user tilts the fishing rod to take the fish by their chest direction and to consume the power of the fish (see Fig. 3 (h) and Sect. H in the Fig. 4 (b)).
- Finally, in the finishing stage, the user takes the mobile phone toward their chest again and then user verifies the experimental result (see Fig. 3 (i) and Sect. I in the Fig. 4 (b)).

5.2 Gesture Recognition

An experiment was conducted to assess the usability of the accelerometer as interaction technique for mobile games. Thirty subjects (27 males and 3 females, ages 23 to 33, all were colleagues who volunteered for the study) participated in this study. First, we observed the output patterns of the accelerometer for the fishing action as the above scenario (see Fig. 5).

Figure 5 was the result of the experiment for the representative eight subjects among of the thirty subjects. Though the amplitudes are different each other for their action, the all output patterns are almost similar in all the experiments. To analysis the acceleration signal of the gestures for the fishing action, we considered the typical signal type
for the fishing action (see Fig. 4(b)).

In the Fig. 4, the direction of the mobile phone is parallel to the Earth’s surface, and X- and Y-axis are pointing the right and forward direction, respectively (see Fig. 4(a)). Therefore, the output values of the X- and Y-axis are both 0 g. The section A in the Fig. 4(b) indicates that the user’s posture is “attention”. The direction of the mobile phone of this case is that X- and Y-axis are pointing the right and the Earth’s gravitational field, respectively, and the output values are 0 g (0°) and −1 g (−90°), respectively. The section B in the Fig. 4(b) indicates that user’s posture changed from “attention” to beginning stage. In this case, the output values of Y-axis gradually increase because the Y-axis is changed from direction of the Earth’s gravitational field to its reverse. And X-axis nearly remains the same. The section C in the Fig. 4(b) indicates that user’s posture changed from beginning stage to ready stage. In this case, the output values of Y-axis are decreasing up to about 0 g because the direction of the mobile phone is parallel to the Earth’s surface (see Fig. 3(c)). The section D in the Fig. 4(b) is the throwing stage, and the accelerometer signal is a dynamic acceleration. The section E in Fig. 4(b) is the waiting stage and indicates that user watches the mobile phone so that the user checks the bite of the fishes. Therefore, the output values of Y-axis gradually increase the same with beginning stage. The section F in the Fig. 4(b) is the snatching stage, and the accelerometer signal is the dynamic acceleration the same throwing stage. The section G in the Fig. 4(b) is the pull & push stage, and the output values of the accelerometer gradually decrease and then increase. The section H in the Fig. 4(b) is the tilting stage, and the output values of the accelerometer decrease to −1 g when user tilt to the right and increase to +1 g when user tilt to the left. Finally, the section I in the Fig. 4(b) is the finishing stage that user ascertains the practice result, and the user’s posture is the same with beginning stage about the output of the accelerometer.

Generally, the output acceleration of this accelerometer has been converted to the acceleration that varies between ±1 g. However, in the fishing action, the acceleration having over ±1 g is detected because the dynamic acceleration such as throwing and snatching exists. Namely, in the fishing action, all of the gesture and posture are a static acceleration except for the throwing and snatching gestures (see Fig. 4(b)).

5.3 Posture Estimation

To estimate user’s posture carrying the mobile phone, we calculate θ for each state as shown in Fig. 3. The accelerometer uses the force of gravity as an input vector to determine orientation of an object in space. The X- and Y-axis of the accelerometer are pointing the right and forward direction, respectively. If the accelerometer is ideal state, the accelerometer’s digital outputs (X- and Y-axis) are 0 g.
Table 1
Output of X- and Y-axis respond to changes in tilt.

<table>
<thead>
<tr>
<th>Orientation to horizon</th>
<th>X output</th>
<th>Y output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PWM (%)</td>
<td>Acceleration (g)</td>
</tr>
<tr>
<td>90</td>
<td>53.40</td>
<td>0.000</td>
</tr>
<tr>
<td>75</td>
<td>50.43</td>
<td>-0.232</td>
</tr>
<tr>
<td>60</td>
<td>47.41</td>
<td>-0.468</td>
</tr>
<tr>
<td>45</td>
<td>44.80</td>
<td>-0.672</td>
</tr>
<tr>
<td>30</td>
<td>42.77</td>
<td>-0.831</td>
</tr>
<tr>
<td>15</td>
<td>41.51</td>
<td>-0.929</td>
</tr>
<tr>
<td>0</td>
<td>40.60</td>
<td>-1.000</td>
</tr>
<tr>
<td>-15</td>
<td>41.36</td>
<td>-0.941</td>
</tr>
<tr>
<td>-30</td>
<td>42.55</td>
<td>-0.848</td>
</tr>
<tr>
<td>-45</td>
<td>44.72</td>
<td>-0.678</td>
</tr>
<tr>
<td>-60</td>
<td>47.25</td>
<td>-0.481</td>
</tr>
<tr>
<td>-75</td>
<td>50.19</td>
<td>-0.251</td>
</tr>
<tr>
<td>-90</td>
<td>53.40</td>
<td>-0.000</td>
</tr>
</tbody>
</table>

Table 2
Threshold value for estimating user posture; min, max, mean and standard deviation values of Y-axis for each posture.

<table>
<thead>
<tr>
<th>Posture</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>T(Threshold value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attention</td>
<td>-1.0000</td>
<td>-0.9730</td>
<td>-0.9852</td>
<td>0.0040</td>
<td>-0.9918 &lt; T &lt; -0.9786</td>
</tr>
<tr>
<td>Watching the mobile device</td>
<td>0.6260</td>
<td>0.7580</td>
<td>0.6853</td>
<td>0.0282</td>
<td>0.6388 &lt; T &lt; 0.7318</td>
</tr>
<tr>
<td>Ready</td>
<td>-0.3540</td>
<td>0.1620</td>
<td>-0.0703</td>
<td>0.1168</td>
<td>-0.2631 &lt; T &lt; 0.1225</td>
</tr>
<tr>
<td>Waiting the bite</td>
<td>0.5840</td>
<td>0.8520</td>
<td>0.6724</td>
<td>0.0784</td>
<td>0.5840 &lt; T &lt; 0.8017</td>
</tr>
<tr>
<td>Watching after tilting</td>
<td>0.5960</td>
<td>0.6500</td>
<td>0.6290</td>
<td>0.0098</td>
<td>0.6127 &lt; T &lt; 0.6452</td>
</tr>
</tbody>
</table>

when it is parallel to the earth’s surface and a scale factor is 12.5% duty cycle change per g. However, the accelerometer may not always be level when the applications using the accelerometer are executed on the mobile phone, and calibration is recommended by the manufacturer. The method to do a more accurate calibration is to slowly rotate the accelerometer 360° and to make measurements at ±90°. The following Table 1 shows the calibration results with the changes in the X- and Y-axis as the accelerometer that embedded in the mobile phone is tilted ±90° through gravity. These experimental results were recorded at each Y-axis orientation to horizon. If the accelerometer is ideal state, the duty cycle is changed 62.5% into 37.5% both X- and Y-axis when angle is changed ±90°. However, in this experiment, the duty cycles of the X- and Y-axis are (66.2−40.6)/2=12.8%/g and (60.1−36.4)/2=11.9%/g, respectively [11].

To determine a range of the θ for the user’s posture, with the various postures of the thirty participants, we collected data a two-dimensional time series for about three seconds at the sampling rate of the 140 samples/s from the outputs of the accelerometer. We examined the optimal threshold value to determine the convergence of the user’s posture. Here, the convergence of the user posture was determined using the threshold valued and following form,

\[\mu - z \times \frac{\sigma}{\sqrt{n}} < \text{Threshold value} < \mu + z \times \frac{\sigma}{\sqrt{n}}\]  \(1\)

where, \(\mu\) is mean of the population which is output of the accelerometer, \(z\) is normal distribution, \(\sigma\) is variance for the population, and \(n\) is number of samples. The 90% (\(z=1.65\)) of the total samples is used for the confidence interval. The following Table 2 shows the result of the confidence interval.

Through the experiment mentioned above (Table 1), the angle of the mobile phone for each user’s posture is estimated: it is about −90° for attention, about 39.5°~46.8° for watching the mobile phone, about −16°~7° for ready, about 35.5°~54° for waiting the bite, and about 37.6°~40° for watching the mobile phone after tilting gesture.

5.4 Implementation

The gestures used in the fishing game are the throwing, snatching, and tilt; the throwing gesture is expressed as the movement distance and the velocity of the fishing rod by measuring the acceleration for the user’s throwing gesture, the path-direction of the fishing rod is expressed by X-axis of the accelerometer when user throw the fishing rod. We consider the range of the power and the tilt in the throwing stage in order to determine the movement distance, the velocity, and the path-direction of the fishing rod. Because it is difficult to distinguish between the dynamic acceleration like the throwing and the static acceleration like the posture, and the noises as the unnecessary user’s action and hand tremble have to be removed, the state machine is needed. The state machine for the mobile fishing game recognizes and processes the user’s action by the designed scenario by treating only valid gesture or posture in each state. The pros and cons of applying finite state machine to the AM-Fishing game are follows in Table 3. The input of the state machine is the accelerometer signal by lowpass filter and the output is gesture-based control event. If this system is misrecognized in a stage, do all over again. Figure 6 is the structure of the state machine.

To demonstrate our application in games we have de-
Table 3  Pros and cons of applying finite state machine.

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supports and encourages parallelism</td>
<td>More closely oriented toward the action than object</td>
</tr>
<tr>
<td>Visually clear</td>
<td>Link between object and action not very clear</td>
</tr>
<tr>
<td>Simple and robust</td>
<td>Not predictive for scientific computation</td>
</tr>
<tr>
<td>Can recognize user’s actions in order of precedence</td>
<td>When misrecognized, do all over again</td>
</tr>
<tr>
<td>Can directly set up the appropriate threshold value according to subject</td>
<td></td>
</tr>
<tr>
<td>Can remove the noises caused by user’s unnecessary actions and hand trembling</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 6  Finite state machine for the AM-Fishing game; E_A for attention event, E_W for watching event, E_R for ready event, E_th for throwing event, E_b for a bit event, E_S for snatching event, E_P for pull & push event, E_T for tilting event, E_Wt for watching event after tilting, and E_Tp and E_Tw for timer event.

Table 4  Recognition accuracy of the gestures and postures in the AM-Fishing game.

<table>
<thead>
<tr>
<th>Subjects</th>
<th>Number of plays</th>
<th>Number of functions</th>
<th>Number of malfunctions</th>
<th>Recognition accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapted subjects</td>
<td>90</td>
<td>85</td>
<td>5</td>
<td>94.4%</td>
</tr>
<tr>
<td>First-time subjects</td>
<td>90</td>
<td>82</td>
<td>8</td>
<td>91.1%</td>
</tr>
</tbody>
</table>

Fig. 7  AM-Fishing game; (a) throwing mode, (b) snatching mode, and (c) tilting mode on the mobile device.

developed mobile game called AM-Fishing game (see Fig. 7). In the game the user has to throwing the mobile phone with his real arm into a virtual fishing site.

6. Experimental Results

We measured the recognition accuracy rate to evaluate the performance of the AM-Fishing game. The sampling rate was about 20 samples per second in mobile phone. We define adapted subjects and first-time subjects as the participants, and the malfunction includes non-function, misrecognition, and overtime. Table 4 shows the recognition accuracy of the AM-Fishing game that the state machine is applied. Thirty subjects (fifteen adapted subjects and fifteen first-time subjects in the AM-Fishing game) participated in this experiment and they played the games three times, each. An average success rate for the games was about 92.8%. This result is useful in terms of the gesture- and posture-based control interface and technique for inducing the interest and realism for mobile games as well as providing the convenience of the control. For example, in the AM-Fishing game, we can improve the realism by using the gesture and posture in the control interface. Also, this enables to induce extra-interest through the dynamic motion for the control of the game. This is because that the interface for controlling the motion of game characters enables user to feel higher realism as we prefer to use joystick instead of keyboard in the personnel computer-based game.
7. Conclusion

We implemented the new interaction by estimating and detecting the context information as user's various gestures and postures with 2-axis accelerometer. We considered the typical fishing action by the output type of the accelerometer for 30 participants. Because it is difficult to distinguish between the dynamic acceleration and the static acceleration, and the noises as the unnecessary user's action and hand tremble have to be removed, the state machine is needed. The state machine for the mobile fishing game recognizes and processes the user’s action by the designed scenario by treating only valid gesture or posture in each state. As a result, this will enhance not only the convenience to use but also the interest and realism for mobile games.

References