A Gesture-Based Control for Handheld Devices Using Accelerometer

Ikjin Jang and Wonbae Park

Dept. of Information & Communication, Kyungpook National University,
Daegu 702-701, Republic of Korea
frog30@korea.com, wbpark@ee.knu.ac.kr

Abstract. The current paper presents how the signals from an accelerometer can be processed to accurately recognize user gestures after applying a small accelerometer to a handheld device. For gesture-based control to be effective in handheld devices, the overheads involved in recognizing gestures should be minimal and the gestures accurately recognized in real operational environments. Therefore, the signals detected from accelerometers were classified into acceleration and dynamic acceleration, then the signal patterns of the accelerometers in relation to simple gestures were analyzed. A device control module was created and its operating process compared to that of a normal control device to evaluate the usability of gesture recognition. As a result, gesture-based control was found to be easy to use, reduced the preparation process, a produced a rapid system reaction. Accordingly, gesture-based control would seem to be an effective user interface for handheld devices primarily used in mobile environments.

1 Introduction

For mobile convenience, handheld devices do not provide the user interfaces included in notebooks, such as a keyboard and mouse. Instead, they use a keypad, stylus pen, and/or input-panel as the input device, and a small LCD device to output several lines of text or simple images. However, these input devices are time-consuming and difficult to use, as the small buttons on the display need to be manipulated several times. Extensive research has already been conducted on new input/output methods to solve this problem, particularly in relation to voice recognition and gesture recognition [1,3,5,6,9,10]. Yet, proper sensors are necessary to detect user actions or recognize gestures in the case of handheld devices, plus the processing capacity and mobile convenience should also be considered when applying sensors. Small accelerometers, the most appropriate sensors for handheld devices, measure mobility and actions through acceleration, thus the size and moving directions detected by the sensors can be used to classify simple user gestures. Accordingly, the current paper investigates the application of small accelerometers to handheld devices, and describes the gesture-based interactions required to replace a keypad or stylus pen operation.

A. Sanfeliu et al. (Eds.): CIARP 2004, LNCS 3287, pp. 259–266, 2004.

2 Related Works

Accelerometers are used in various fields and have been extensively studied in the context of gesture recognition In particular, Ken Hinckley studied how to recognize various user gestures using several sensors altogether (Tilt Sensor, Touch Sensor and Proximity Sensor) [4]. In addition, accelerometers have also been used in pointing devices, where finger movements are detected based on attaching accelerometers to the finger parts and hand of a glove[8], and in handwriting-recognition, where pen movements are detected based on attaching a pair of accelerometers to a pen [7]. When applying these case studies to the user interfaces of handheld devices, the operational environment and capacity of handheld devices must be considered, plus the overheads due to the inclusion of accelerometers must be minimized. Therefore, it is necessary to study how to recognize simple gestures efficiently, while also minimizing the number of accelerometers. In this paper, signal patterns make for gestures easily expressed in a real operational environment, and its usability be evaluated by simple examples.

3 Signal Processing of the Accelerometer for Gesture Awareness

The signal which is detected from the Accelerometers is divided with static acceleration and dynamic acceleration. Static acceleration is like tilt or gravity, and dynamic acceleration is like shock and vibration.

3.1 Static Acceleration (e.g., Gravity)

An accelerometer can detect gesture changes based on its horizontal status[2]. When an accelerometer is in a perfect horizontal status and there is no movement at all, the detected values for the dual axis are all zeros. If the position is then changed in a specific direction, the detected value will differ from that of the horizontal status even if there is no movement at all, and this value is sustained until the horizontal status is recovered. Fig. 1 shows the signal type detected when an accelerometer changes its position and produces signals for one-axis.

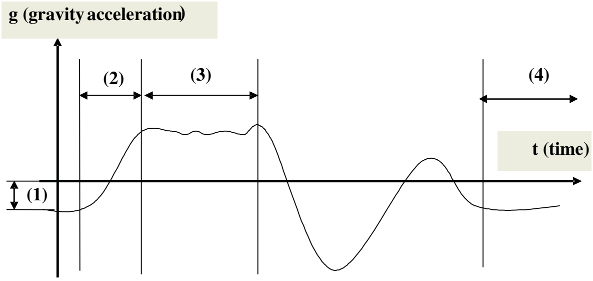


Fig. 1. Signal type of static acceleration

In Fig.1, the differences in section (1) mean that the signals detected from the accelerometer differed from those in the horizontal status, because the display of the handheld device used was tilted in the user direction. Meanwhile, the signals in section (2) are the signals detected when the position of the handheld device was change and those in section (3) were detected when a specific position was sustained for a while. Finally, the signals in section (4) were those detected when the handheld device returned to its original position. As such, static acceleration can detect a position change from the original status and determine which direction the position is changed in relation to the original position if the changed status is sustained for a while. The steps shown in Fig.2 are necessary to recognize gestures from static acceleration signals detected by an accelerometer.

The signals detected from an accelerometer are processed for each dual-axis, as shown in Fig.2, to grasp a change in the position of the accelerometer in one or two directions. The processes performed in each step of Fig.2 are outlined below.

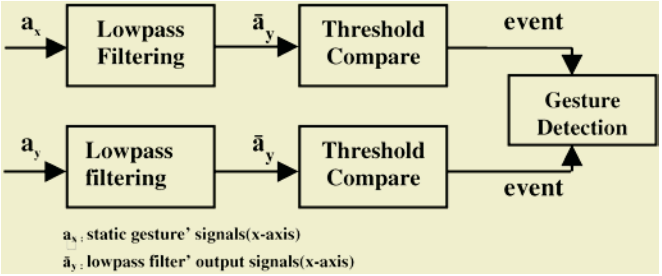


Fig. 2. The steps to recognize gestures from static acceleration

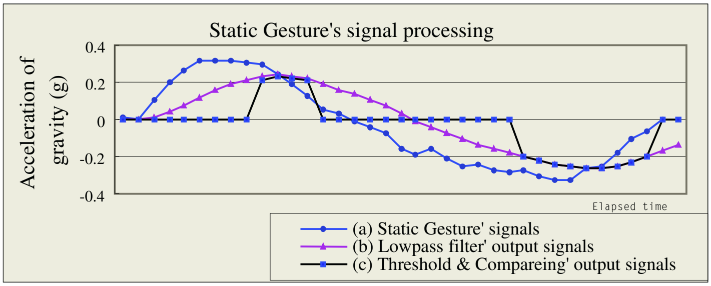


Fig. 3. Signal processing about Static gesture (1-axis' signal)

Low Pass Filtering

This makes signal changes consecutive by processing a trivial movement. The current study used a first-order Butterworth filter, as a simple filter is preferable considering the capacity of handheld devices. Fig. 3 shows the output signals at each step in the signal processing, where (a) shows the signals detected from the accelerometer and (b) shows the output signals of the lowpass filter.

Thresholding and Comparing

If the detected signals are lower than a threshold value, they are ignored. As such, Fig. 3 (c) shows the status of trivial movements that are eliminated at the Threshold

& Comparing step (which ignores signals where the gravity acceleration is smaller than 0.2g)

3.2 Dynamic Acceleration (e.g., Vibration)

Dynamic acceleration occurs when a sudden movement or short shock is transmitted to the accelerometer, resulting in short-term peaks in the detected signals. These kinds of signals happen when user gestures are not consecutive or from slight knocking or sudden shocks. Fig.4 shows an example of dynamic acceleration, based on signals from knocking gestures.

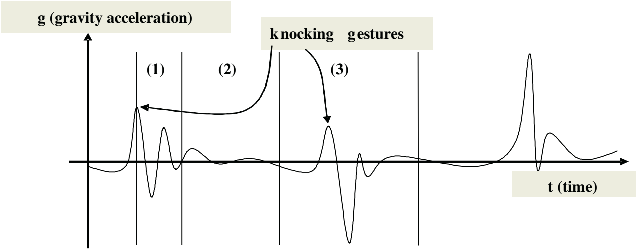


Fig. 4. Knocking Gesture's signal form

In Fig.4, section (1) shows several peaks generated by one knocking gesture. Although knocking gestures can occur for various reasons (bag, pocket, or hand etc.), the peak duration is normally within 100 ms(milliseconds). Therefore, the problem of 'multiple peaks' can be solved by debouncing, i.e. the identification of section (1) as a 'dead period', because users cannot generate knocking gestures within intervals of 100 ms. If knocking gestures occur while a user is moving, multiple peaks and consecutive vibrations can happen simultaneously. Therefore, a silent period, as marked in section (2), needs to be set to identify regular knocking signals, as distinct from vibration due to user movements. In addition, a subsequent signal needs to be detected some time after the former signal is detected. That is, another knocking gesture needs to exist in section (3) to be effective. Fig.5 shows the processes necessary to recognize user gestures using dynamic acceleration signals, and the output signals from each step ((a)~(d)) are shown in Fig.6.

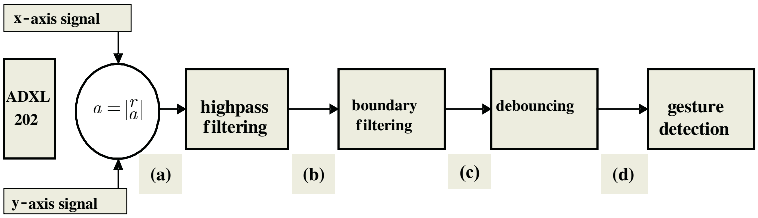


Fig. 5. Dynamic Acceleration's signal processing step

In Fig.5, the first step marked by a circle is the process used to obtain the amplitude from the dual-axis signals, and the content of each step is outlined below.

Highpass Filtering

When handheld devices are not used horizontally, the accelerometer is under the influence of gravity acceleration. Therefore, changing the position of the accelerometer also changes the detected signals. Fig.6 (a) shows the signal detected from an accelerometer in the case of a knocking gesture, i.e. seizing the handheld device. Because this signal is detected as the accelerometer is tilted in a random direction, the x-axis signal and y-axis signal sustain random values. Therefore, when considering various operational environments, the DC components should be eliminated first in order to process the detected signals in the same way, regardless of the tilt direction of the accelerometer. Fig.6 (b) shows the signals after eliminating the DC components using a highpass filter.

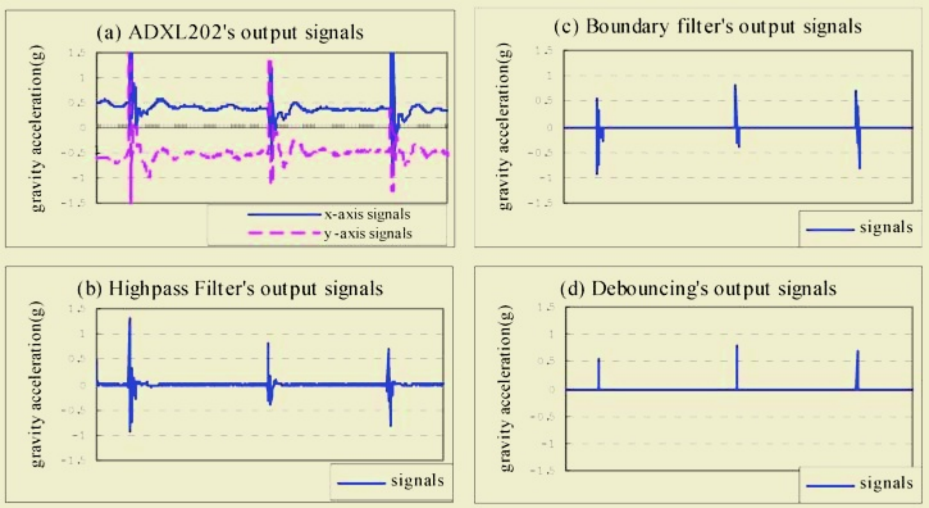


Fig. 6. Output signals about Fig.5 (Dynamic Acceleration's processing step)

Boundary Filtering

This process eliminates trivial signals to prevent the transmission of unwanted gestures arising from slight vibrations or movements. It removed signals that below 0.2g by the experimentation. Fig.6 (c) shows the output signal of the boundary-filtering step.

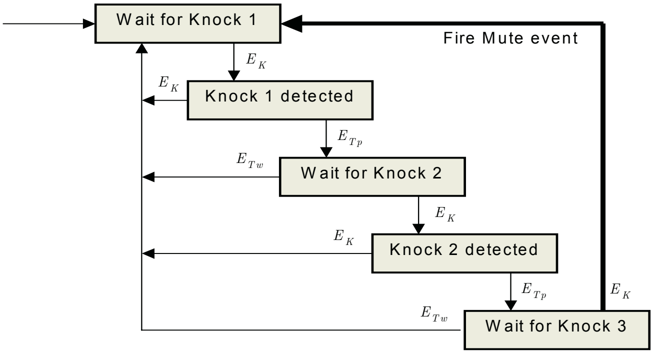
Debouncing

In the case of several peaks generated by one gesture, debouncing avoids recognizing such peaks as different gestures. As such, the multiple peaks shown in Fig.4 (1) are ignored. Fig.6 (d) shows the elimination of multiple peaks through debouncing.

4 Examples of Gesture-Based Control

4.1 Example 1: Bell and Sound Control

This is the function of stopping the bell using a knocking gesture without withdrawing the handheld device when it rings. Bell control is difficult in places such as convention halls or public meetings, thus rapid control through gestures is desirable. To control a bell using a knocking gesture, it is important that other gestures are not be wrongfully recognized. Therefore, regular knocking gestures more than 3 times are necessary, plus a state machine is needed to detect regular knocking gestures more than 3 times, as shown in Fig.7.



Ek : Knock Event, Etp : Timer Event, Etw : Timer Event
Tp : Minimum interval between two knocking events
Tw : Maximum waiting interval for a knocking event

Fig. 7. State machine for knocking gesture awareness

For effective knocking gesture detection, the Tp and Tw need to be properly selected. In the current experiment, Tp was 200 milliseconds, while Tw was 500 milliseconds. This should be adjustable by users. Fig 8 shows an example of bell control using knocking gestures, which demonstrates the step-by-step process of recognizing knocking gestures.

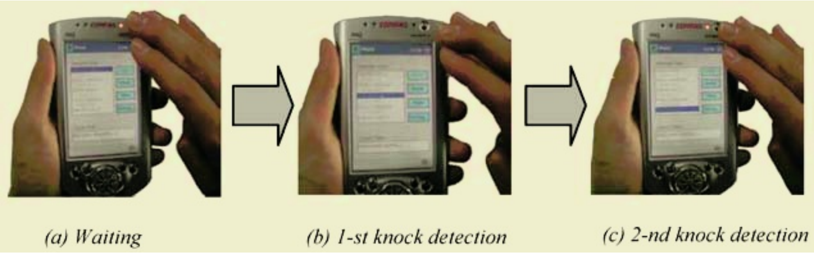

(a) Waiting *(b) 1-st knock detection* *(c) 2-nd knock detection*

Fig. 8. Bell Control Application

4.2 Example 2: Scrolling

To scroll a handheld device, a small button must be clicked using a stylus pen. However, it is not easy to click the scroll button on a small display panel. To scroll using gestures, information about the direction of the user gestures should be included. An example of gestures including information about the user direction is the tilt method. In the case of an accelerometer, if a one-axis signal is changed, this indicates a tilt in the related direction, while if both dual axes change, this indicates a complex tilt in two directions. When using tilting gestures to scroll, there can be a slight movement or vibration, because handheld devices are not fixed. Therefore, a dead zone should be set, and signals from the dead zone should not impact the scrolling. In the current experiments, a 0.2g dead zone was applied. Furthermore, since there is no signal larger than 2g from a tilting gesture, any signal larger than 2g was ignored. Fig.9 shows an example of Tilt & Scroll.

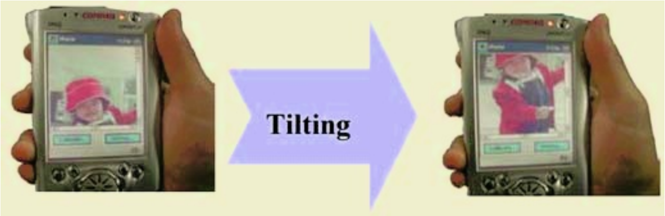


Fig. 9. Scrolling Application

4.3 Example 3: Display Mode Alteration (Pivot)

Handheld devices use rectangular display panels, and the display modes change according to the content and user preference. Sensor detection of axis-rotation is necessary to change display modes, yet the signals detected from dual axes are not changed if there is an axis-rotation when the accelerometer is in a horizontal status. However, if the display status tilted in the user direction is set as the original status, then a display rotation can be easily determined, because the signals detected from the dual axes will differ from the original signals if axis-rotation occurs. Fig.10 shows an example of realizing ‘display mode alteration’ based on detecting axis-rotation in a handheld device.

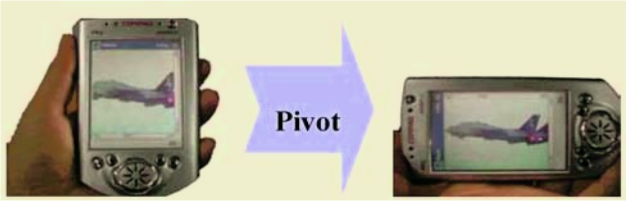


Fig. 10. Display Mode Alteration

5 Conclusions

The use of a keyboard or mouse as a user interface for handheld devices is awkward and impractical, therefore, the current study designed and realized a system that uses gesture-based interactions as user interfaces after applying accelerometers to handheld devices. As such, the signal features of accelerometers towards gestures that can easily be used in the real operational environments of handheld devices were surveyed, plus the signals detected from accelerometers were processed after classifying them according to the kind of gesture. To recognize the signals detected from accelerometers as user gestures, the signal pattern of each gesture was analyzed and recognized as a valid gesture only when a signal pattern matched a gesture pattern. In the current research, only one accelerometer was used based on the capacity of handheld devices and their mobile convenience. In addition, simple filters were used for the signal-processing step. Further research on using multiple sensors, designing precise filters, and algorithms that can exactly distinguish various gestures is necessary to identify various user gestures.

References

1. Bartlett, J.F., “Rock ‘n’ Scroll Is Here to Stay”, IEEE Computer Graphics and Applications, 2000.
2. Analog Devices, “ADXL202E: Low-Cost ?2g Dual-Axis Accelerometer With Duty Cycle Output Data Sheet (Rev. A, 10/00)”, http://www.analog.com/UploadedFiles/Datasheets/567227477ADXL202E_a.pdf
3. Ehreumann, M.; Lutticke, T.; Dillmann, R., “Dynamic gestures as an input device for directing a mobile platform”, Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on , vol.3 , 2001, pp: 2596-2601.
4. Ken Hinckley.; Jeff Pierce.; Mike Sinclair.; Eric Horvitz, “Sensing techniques for mobile interaction”, Symposium on User Interface Software and Technology, CHI Letters 2, 2000, pp: 91-100.
5. Mantyla, V-M,; Mantyjarvi, J.; Seppanen, T.; Tuulari, E., “Hand gesture recognition of a mobile device user”, ICME 2000. 2000 IEEE International Conference on Multimedia and Expo, 2000. vol. 1, 2000, pp: 281 -284
6. Matthieu, B.; Hedvig, S.;Jan-Olof, E., “Recognition of gestures in the context of speech”, Pattern Recognition, 2002. Proceedings. 16th International Conference on, vol. 1, 2002, pp: 356 –359.
7. Milner, B., “Handwriting recognition using acceleration-based motion detection”, IEE Colloquium on Document Image Processing and Multimedia, 1999.
8. Perng, J. K.; Fisher, B.; Hollar, S.; Pister, K. S. J., “Acceleration sensing glove(ASG)”, The Third International Symposium on Wearable Computers, 1999, pp: 178 – 180.
9. Randell, C.; Muller, H., “Context awareness by analyzing accelerometer data”, The Fourth International Symposium on Wearable Computers(ISWC00), 2000, pp: 175-176.
10. Sawada, H.; Hata, S.; Hashimoto, S., “Gesture recognition for human-friendly interface in designer-consumer cooperate design system”, Robot and Human Interaction, 1999. RO-MAN '99. 8th IEEE International Workshop on, 1999, pp: 400 –405.