Self-Defined Gesture Recognition on Keyless Handheld Devices using MEMS 3D Accelerometer

Shiqi Zhang [1, 2]
zhangshiqihit@yahoo.com.cn

Chun Yuan [1]
yuanc@sz.tsinghua.edu.cn

Yan Zhang [2]
ianzhang@hit.edu.cn

[1] *Information Sci. and Tec. Division*
Graduate School at Shenzhen
Tsinghua University
Nanshan, Shenzhen, Guangdong, China

[2] *Dept. of Electronic and Info. Eng.*
Shenzhen Graduate School
Harbin Institute of Technology
Nanshan, Shenzhen, Guangdong, China

Abstract

A novel gesture recognition method is introduced in this paper, and it is designed for keyless handheld devices specially. This terminal contains a three dimensional MEMS accelerometer and a Single Chip Microcontroller, which enable kinds of gesture recognitions, such as inclination, swaying and handwritten Arabic numerals. In this way, we could operate a handheld device without any key just by gesture recognition. It is noted that the same track of a numeral may come from different strokes, for example, Arabic numeral "0" could be written clockwise or anti-clockwise, so self-defined gesture recognition is adopted in Arabic numeral recognition, by which every user could "write" in their own way. Using keyless handheld devices, young guys could get more fun and old people or blind persons could get more convenience. A large number of experiments have been done to verify its reliability and the Correct Rates (C.R.) prove this terminal could satisfy users' basic requirements.

1. Introduction

In recent years, handheld devices have been greatly used as one of the most important daily tools in our lives. At the same time, an explosive growth of information on Internet has made multi-media resources almost ubiquitous. Many related technologies have come out until now; however, the operation manner is still the old way with many keys. The keys take up large space for such a small handheld device and it is still not an easy task to make proper use of these keys for many people, such as the old ones and the blind. To make this human-machine interface more human-friendly, many new devices and technologies have been developed these years [1]-[6]. All of them could be roughly divided into three parts. And the pros and cons will be analyzed briefly now.

The first part is based on vision. As [11], [13] states, most vision-based recognition methods collect visual information through a camera, and then the data would be processed on a computer. The whole process must be set up on a hypothesis that the camera is fixed firmly, if not, the visual information would be unreliable. However, it is difficult to do that for handheld devices like mobile phones and PDAs. So, vision-based recognition isn't suitable for handheld devices, although it has high precision and mature theories.

Secondly, several hearing-based recognition methods have been developed [10]. These systems make human-machine interaction simpler, but a quiet ambience is necessary, otherwise all kinds of noises could be regard as the voice by mistake. Besides, in some public places like libraries, maybe it is impolite to operate a handheld device by voice command.

At last, ultrasonic waves, infrared or optical sensing are the technologies currently used for human-machine interaction in companies like Interactive Whiteboard Technology. Their costs are usually very high and they are not very convenient to take all the time. Until 2005, the price of MEMS (Micro-electromechanical Systems) accelerometer has been reduced greatly and some gesture recognition methods based on MEMS 3D accelerometer have been developed [2], [12]. In these designs, Arabic numerals together with other basic gestures could be recognized. However, everybody writes numerals in different ways, for example, "0" could be written clockwise or anti-clockwise and "5" may have two strokes or only one. So, robust Arabic numeral recognition systems designed for users should be defined by themselves.

Thus, a new handheld device without any key is set up, in this system a 3-axes MEMS accelerometer is used to sample acceleration signals in three different directions respectively and a demo-program on J2ME platform is designed as an interface between users and handheld devices. There are three kinds of recognitions in this system as Table 1 shows.

Table 1. Gesture Recognition Taxonomy

Types	Usage	Gesture Model
Swaying	Browse Contents	Pre-defined
Inclination	Rotate Images	Pre-defined
Arabic Numerals	Dial a Number	Self-defined

For Arabic numeral recognition, the goal is to distinguish ten Arabic numerals from '0' to '9' according

978-0-7695-3304-9/08 $25.00 © 2008 IEEE
DOI 10.1109/ICNC.2008.387

to acceleration models when one writes a number in the air. Self-defined gesture recognition method is used in this part to make users be able to "write" in their own ways. According to [12], specific strokes of Arabic numerals are adopted in numeral writing to simplify gesture shapes. In this way, it is easy to keep "dialing" away from keys. We could also do Arabic numeral recognition using trajectory analyses, and a brief comparison between trajectory analyses and acceleration is presented.

2. System Skeleton

In aspect of hardware, this terminal is made up with three parts. Figure 1 shows an overall structure. Accelerometer is fixed on a mini-board and powered by the board too; it detects acceleration in three directions and transforms these values into voltages. These current signals would be transmitted to a Single Chip Machine (SCM) by wires on Printed Circuit Board (PCB). Then single chip microcomputer, CT-298 completes most of the other tasks, such as A/D conversion, gesture and motion detection and communication with terminal. The way to transmit digital data with demo terminal is through a serial port, which connects a mini serial port on mini board and a personal computer.

The 3-dimensional accelerometer MMA7260T is made by Freescale Semiconductor, which has selectable measure span (1.5g/2g/4g/6g), a high sensitivity (800mV/g), low current consumption (500μA) with sleep mode (3μA) and a very small package (6mm x 6mm x 1.45mm QFN). All of these features make MMA7260T very suitable for handheld device applications.

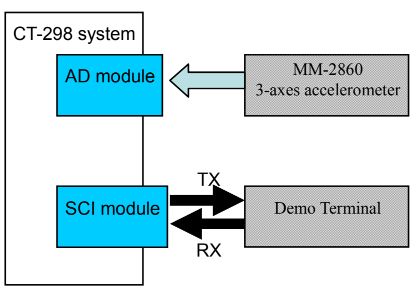

Figure 1. System skeleton

SCM MC9S08QG8 is a microcontroller also made by Freescale Semiconductor. It is an 8-bit HCS08 central processor unit or CPU and has many excellent features, such as 20-MHz working frequency, 8 Kbytes FLASH, 512 bytes RAM, 8-channel 10-bit analog-to-digital converter, serial communications interface module and 16-pin plastic dual in-line package. In actual fact, both the MEMS accelerometer and the SCM have been widely used in embedded system like toys and handheld phones.

Digital data are sent out from SCM, and then a demonstrative program is set up to receive the data and display a UI to intercommunicate with users.

3. Gesture Recognitions

There are three kinds of recognitions in this system and they are used in different items separately. The first part, swaying recognition completes most operation commands; the second part, swaying recognition is used in image browser module; the final part, Arabic numeral recognition, is only used to input telephone numbers.

3.1 Swaying Recognition

To operate a handheld device, basic commands must be able to be executed easily, like functions of a joystick. Four simple gestures are defined as "go to", "back", "up" and "down", using these four commands one could easily control the terminal browser. For example, one wants to dial a number, what he/she needs to do is just to shake the handheld phone back and forward to make the arrow point to right item, and shake it rightward. Comparing with the operation manner before, it is much simpler for everybody, especially the persons having eye-diseases.

Figure 2 presents some snapshots during operations. From top-left to bottom-right, they are main menu, file browser, video content browser, video player, audio content browser and audio player. By swaying it in four different directions, users could browse the device's file system freely. The gesture commands, when viewing "Video List" are the same as "File Browser". If a user wants to stop a video and go back to "Video List", a leftward swaying would be OK. The method to control audio player is the same as Video Player. Besides, when an audio file is being played, swaying the device backward/forward could change the volume.

Figure 2. Snapshots of Browser

3.2 Inclination Recognition

When an Image Brower is opened up, the Inclination Mode would be active automatically. The goal of this module is to make image always "upward" no matter in which direction the device is placed. When this mode is active, the SCM will send signals to terminal all the time telling it how the handheld device is placed. Based on the data from hardware, demo program could rotate the image as soon as the device is rotated. However, the rotation of image is in reverse direction as Figure 3 shows.

Figure 3. Image Browser in Inclination Mode

3.3 Arabic Numeral Recognition

For mobile phones, ten "number" keys take up large space comparing with the small body. To make handheld devices smaller, next generation of operation method should get rid of these keys. For this system, Arabic numeral recognition would be active when a user chooses "Dial a number" item, and a self-defined method would be used here.

In the part of swaying recognition, commands could be figured out by SCM. However, when we need to distinguish ten different kinds of motions, the performance of SCM couldn't satisfy our requirement. Thus, the processing is designed to be made on a stronger terminal which is a platform based on Java2 Micro Edition (J2ME). The reason of choosing J2ME platform is that most of the mobile phones and PDAs on market support J2ME, and Java is cross-platform and one of the most widely used language in handheld devices.

With a handheld device that only has a 3-dimensional accelerometer, but doesn't have a gyroscope, a user must obey some rules to record the correct acceleration values when writing a numeral in the air. Before introducing the rules, some definitions would be presented.

$$V_s = \begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} \quad (1)$$

where, V_s is the acceleration vector when the sensor is held still, and x_s, y_s, z_s are accelerations in three directions

Considering that the trajectory is in a 2-D plane, we could ignore one dimension of acceleration to simplify our calculations. Thus, we get a simplified instantaneous acceleration vector V_s^1, which is a 2×1 one.

$$V_s^1 = \begin{pmatrix} x_s \\ y_s \end{pmatrix} \quad (2)$$

We define a 3D coordinate space C according to the accelerometer's physical structure. At the same time, there is a natural coordinate space C^1. If the device is not placed horizontally, there will be $x_s \neq 0$, $y_s \neq 0$, which means the acceleration vector would not be a zero one although it doesn't move at all. In this condition, we must make sure of two points to keep the output value meaningful. The aim is to make C and C^1 keep still relatively.

1) Directions of x, y axes in C must be fixed.
2) Accelerometer's center of gravity must be moved in the same plane which is vertical by z-axis in C.

According to the results got from experiments, we found that it is too hard for most people to satisfy the two points above. Therefore, we lay the sensor on a slippery horizontal plane, say a desktop, to reduce the affection from resistance and keep $x_s = 0$ and $y_s = 0$. And then all the followed motions and data are made out on this desk.

Figure 4 shows a main flow chart of Arabic numeral Recognition. The first step of the whole process is to load vectors, and then every vector is sent to terminal directly to avoid complex operations in SCM. Program in demo terminal would do a beginning detection according to the vector serial it gets. In practical operation, a threshold B is defined to detect whether the acceleration sensor is moved or not. To avoid inaccurate judgments, only when a number of sequential vectors are all bigger than the threshold we could make sure it is a beginning of a motion. Expressions below show the process of beginning detection.

$$\left|x_{s_i}\right| + \left|y_{s_i}\right| < B \qquad i = 1 \cdots 5 \quad (3)$$

where, x_{si} and y_{si} are accelerations in x axis and y axis of five sequential input vectors

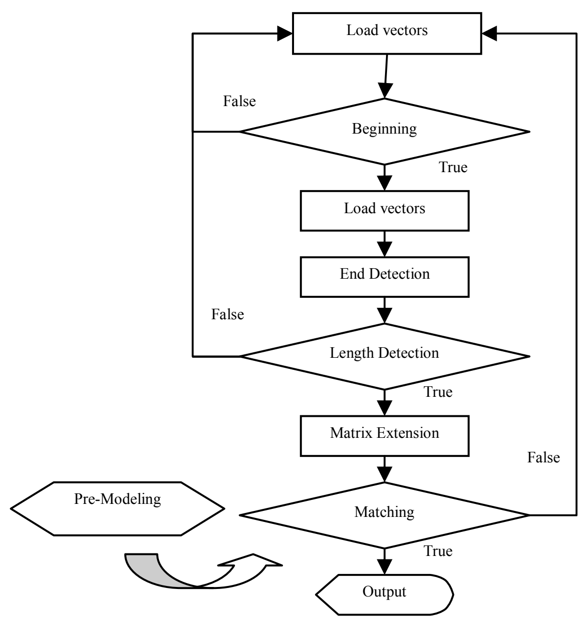

Figure 4. Flow Chart of Arabic numeral Recognition

After the beginning is detected, vectors that express the details of motion would be loaded into a buffer, and then some vectors signing an End would be detected later. At that time, a serial of vectors which contain all the information of acceleration are recorded. But the matrixes made up with vectors may be not of the same size. Therefore, length detection would be carried out. If this vector serial or matrix A_t's length L_t is longer than the length L we have set already, it will be returned to the first step and announce a failure at once, and if not, this big matrix would be extended into a $2\times L$ matrix as the expression below. For $L_t < i \le L$, there is $x_{Lt} = y_{Lt} = 0$.

Before the processing of matrix A, standard models should be constructed in advance. For numerals from "0" to "9", ten models, A_0, A_1, A_2 … A_9, are set up. A minus A_M leaves a $2\times L$ matrix W_M, which can't express the degree of similarity between A and A_M. Therefore every unit's absolute value in matrix W_M is added together to a number δ_M that could imply this level.

$$A_t = \left(V_1^1 V_2^1 \cdots\cdots V_{L_t}^{\ 1}\right) = \begin{pmatrix} x_1\, x_2 \cdots\cdots x_{L_t} \\ y_1\, y_2 \qquad y_{L_t} \end{pmatrix} \tag{4}$$

$$A = \begin{pmatrix} x_1\, x_2 \cdots\cdots x_{L_t} \cdots\cdots x_L \\ y_1\, y_2 \qquad y_{L_t} \qquad y_L \end{pmatrix} \tag{5}$$

where, A_t is the original acceleration matrix, and A is the matrix after extension

$$A_M = \begin{pmatrix} x_{M1}\, x_{M2} \cdots\cdots x_{ML} \\ y_{M1}\, y_{M2} \qquad y_{ML} \end{pmatrix} \qquad M = 0\cdots\cdots 9 \tag{6}$$

$$W_M = \begin{pmatrix} x_1 - x_{M1}\;\; x_2 - x_{M2} \cdots\cdots x_L - x_{ML} \\ y_1 - y_{M2}\; y_2 - y_{M2} \qquad y_L - y_{ML} \end{pmatrix} \qquad M = 0\cdots\cdots 9 \tag{7}$$

$$\delta_M = \sum_{i=1}^{L}\left(\left|x_i - x_{Mi}\right| + \left|y_i - y_{Mi}\right|\right) \qquad M = 0\cdots\cdots 9 \tag{8}$$

In this way, for every single A, ten evaluative numbers δ_M could be got. What we need to do now is just to find out the smallest one, and the sequence number would be the digit that a motion means. At the same time, δ_M should be verified to be smaller than a threshold value δ_T to make sure that the sampling is not meaningless. For M = 0, 1, 2 … 9, if we could get $\delta_D < \delta_M$, $\delta_D < \delta_T$ then the digit "D" would just be the output digit.

In this part, an entire process of Arabic numeral recognition has been listed and the performance of this arithmetic would be exhibited in following parts.

4 Experimental Results

All the experiments should be divided into two parts, basic operations with swaying recognition and inclinations recognition included and Arabic numeral recognition.

The first part contains nearly all the basic commands, such as browsing the file system, playing audio and video contents and entering or exiting items. It is very interesting for most people to operate a mobile phone without keys. This system has been operated by more than twenty classmates already and before they lose the curiosity all of them have got the capability to use this device freely. After recording 500 operations from twenty persons, a statistics is calculated out as Table 2. C.R. stands for correct recognition rate.

From table 2, we could find that all the swaying operations have low C.R. without filter. After analyzing the acceleration curve, we could know the reason is the second peak in the acceleration curve when a user stops a motion. System may mistake this peak as an inverse command, which would output an unwanted result. So, we update the program with a filter added. This filter has a simple function to disregard the second peak when it is quite near the first one. After that, C.R. has got a significant raise. Another interesting point in this experiment is everyone's C.R. gets higher as they control this "mobile phone" for a longer time and will nearly achieve 100% at last.

Table 2. C.R. of Basic Motions

Motions Name	without filter	with filter
Swaying Leftward	82%	90%
Swaying Rightward	80%	95%
Swaying Forward	85%	94%
Swaying Backward	87%	95%
Inclination Leftward	85%	98%
Inclination Rightward	85%	98%

To recognize Arabic numerals, users should write the numerals in a special way, which is demonstrated in Figure 5. In this manner, the models are set up. Figure 6 shows a user's models, in which from numeral "9" to "0" are in the order of top-left to bottom-right.

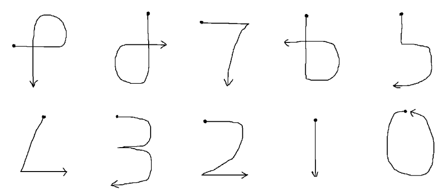

Figure 5. Special Strokes

Trajectory analyses have also been used here to recognize Arabic numerals; however, it is almost impossible for a system without Gyroscope to avoid unexpectable drift. Just for a comparison, we record two pictures as Figure 7, which are both achieved by integrating accelerations two times. Unfortunately, many times of manual regulations are unavoidable any way for every single time of recognition. Hence, this method is discarded.

To exam the performance of Arabic numeral Recognition, we made more than 200 motions on different models. And a curve comes out according to experimental results. As Table 3 shows, C.R.s of "1", "3", "7" are relatively higher than the others because their "tracks" of acceleration are more easy-distinguished from other numerals in shape. However, for "0" and "6", their "tracks" of acceleration are very alike, so more mistakes may be made there.

Totally speaking, the outcome shows this terminal could satisfy the civilian requirement and this keyless method has made the process of operating a mobile phone much more interesting than before.

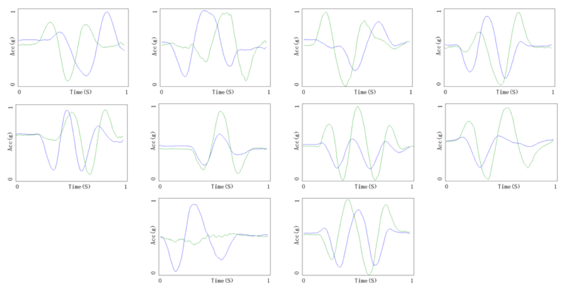

Figure 6. Models of Arabic numerals

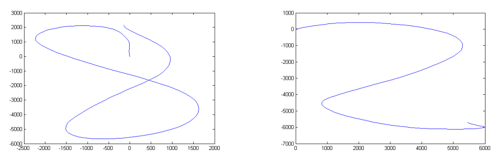

Figure 7. Recognition on Trajectory for number “8”, “2”

Table 3. C.R. of Arabic numerals

Numerals	0	1	2	3	4
C.R. (%)	85	95	93	95	93
Numerals	5	6	7	8	9
C.R. (%)	90	87	94	90	93

5 Conclusions and Further Researches

This paper presents a novel handheld terminal, which consists of a small-size, high-precision, low-cost MEMS accelerometer, an 8-bit high-performance SCM and a demo program on J2ME platform. According to the digital data from accelerometer, this terminal could distinguish users’ gestures. Thus, all the commands could be recognized by gestures. Then to set up “dialing a number” module, self-defined Arabic numeral recognition arithmetic is put forward, which could distinguish ten gestures at one time. At last, some experimental results are presented to verify its feasibility.

Current development effort now focuses on finding out a more effective method to construct prototype models of the ten numerals. Furthermore, the whole software system of Arabic numeral Recognition would be integrated together with other modules and the outlook would be made more human-friendly.

6 Preferences

[1] Youichi Horry, Issyu Nakajima “A Passive-Style Keyless Mobile Terminal” IEEE Tran. on Consumer Electronics, Vol. 49, No. 3, AUGUST 2003

[2] Guanglie Zhang, Guangyi Shi, “Towards an Ubiquitous Wireless Digital Writing Instrument Using MEMS Motion Sensing Technology”, IEEE Proc. on Advanced Intelligent Mechatronics, California, USA, pp.795-800, 24-28 July, 2005

[3] Paul Keir, Jocelyn Elgoyhen and Martin Naef “Gesture-recognition with Non-referenced Tracking” Proceedings of the 2006 IEEE Symposium on 3D User Interfaces, Glasgow, June 2006

[4] Joel F. Bartlett “Rock ’n’ Scroll Is Here to Stay” Information Appliances, Computer Graphics and Applications, IEEE, May/June 2000, 40-45

[5] I.J. Jang , W.B. Park, “Signal Processing of the Accelerometer for Gesture Awareness on Handheld Devices”, IEEE international Workshop on Robot and Human Interactive Communication Millbrae. California. USA, Oct. 31 -Nov. 2. 2003

[6] Yen-Ting Chen, Kuo-Tsung Tseng “Developing a Multiple-angle Hand Gesture Recognition System for Human Machine Interactions”, IEEE Industrial Electronics Society, Nov. 5-8, 2007, Taipei, Taiwan, pp.489-492

[7] Hee-Deok Yang, A-Yeon Park, “Gesture Spotting and Recognition for Human–Robot Interaction” IEEE Tran. on Robotis, VOL. 23, NO. 2, April 2007 pp.256-270

[8] Keith J. Rebello, “Applications of MEMS in Surgery” Proceedings of the IEEE, VOL. 92, NO. 1, JANUARY 2004, pp.43-55

[9] Jinhui Lan, Zhaohui Zhang, “Research on Application of MEMS Accelerometer in Target Classification by Advanced Information Processing”, IEEE Proc. on Nano/Micro Engineered and Molecular Systems, January 18-21, 2006, Zhuhai, China, pp.363-367

[10] De Mori, Renato, “Spoken language understanding: a survey”, Automatic Speech Recognition & Understanding, Avignon, France, 9-13 Dec. 2007 pp.365 – 376

[11] REN Haibing, ZHU Yuanxin, XU Guangyou , IN Xueyin , ZHANG Xiaoping, “Vision Based recognition of Hand Gesture s :A Survey”, ACTA ELECTRONICA SINICA, Vol. 28 No. 2 Feb. 2000, pp.118-121

[12] Sung-Jung Cho, Jong Koo Oh, “Magic Wand: A Hand-Drawn Gesture Input Device in 3-D Space with Inertial Sensors” Proceedings of the 9th Intel Workshop on Frontiers in Handwriting Recognition

[13] Sushmita Mitra, Tinku Acharya, “Gesture Recognition: A Survey”, IEEE Transactions on Systems, Man, And Cybernetics-part C: Applications and Reviews, Vol. 37, NO. 3, May 2007, pp.311-324