FreeDigiter: A Contact–free Device for Gesture Control

Christian Metzger
ETH - Swiss Federal Institute of Technology
Electronics Laboratory
Zurich, Switzerland
cmetzger@ee.ethz.ch

Matt Anderson and Thad Starner
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0760 USA
{fiend, thad}@cc.gatech.edu

Abstract

We present FreeDigiter, an interface for mobile devices which enables rapid entry of digits using finger gestures. FreeDigiter senses via an infrared proximity sensor and a dual axis accelerometer and requires little signal processing. Initial experiments attain accuracy rates of 99.0%, and the system is tolerant to highly varying lighting conditions. The FreeDigiter system requires little power and could be implemented in a very small form factor appropriate for in–ear hearing aids, small MP3 players, and hands–free mobile phone headsets.

1 Introduction

Electronic devices are becoming smaller; however, the same trend can not continue for the devices' user interfaces. For instance, due to the physical size of the human hand, buttons cannot be made any smaller and still be reliably pressed or seen. Age and illness exacerbate this effect.

Figure 1. MP3 players and phone headsets could shrink to the scale of a hearing aid but would require an alternative to button interfaces due to their size.

Traditional button interfaces may also be inappropriate in situations where visual attention is limited or where user contact may be unhygienic or messy. For example, manipulating a car radio or mobile phone often requires visual attention to search for the buttons, which may be dangerous while driving. A surgeon may not want to touch medical equipment while operating but still wish to control devices such as microscopes or medical monitors. Similarly, car mechanics may not wish to touch their computerized diagnostic equipment while their hands are covered in grease. Non-contact gesture recognition can overcome many of these limitations but has had limited application in mobile computing to date.

FreeDigiter uses hand gesture recognition to control mobile devices. In contrast to many existing gesture recognition systems, data analysis is simple, requiring the filtering of the binary output of a proximity sensor. This simplicity allows for small and inexpensive hardware with low power consumption and high wearability. FreeDigiter is reliable under varying lighting conditions and offers a very simple interaction: a user passes fingers in front of the device, the system counts the total number of fingers moved past the sensor, and the system executes the command associated with that number. In this paper, we describe the FreeDigiter system and present initial experiments on the system's usability and reliability.

2 Related Work

As seen in the annual proceedings of the International Symposium on Wearable Computing and the International Conference of Ubiquitous Computing, many mobile gesture recognition systems employ accelerometers [1, 2]. Accelerometers are an ideal sensor for mobile systems due to their low power, low cost, and reliability. Takahashi [5] described an early system that retrieved detailed information about hand and finger motion using a network of accelerometers mounted on a glove. More recently, Herenandez-Rebollar demonstrated his Acceleglove, which recognizes isolated American Sign Language signs [3]. However, gloves reduce the user's tactile sense, are often uncomfortable to wear, and are not suitable for daily wear for most users.

With the Gesture Pendant [4], our previous work on mobile gesture recognition, we use a camera ringed by infrared (IR) light emitting diodes (LEDs) and an IR pass filter over

Figure 2. The user moves two fingers past FreeDigiter's proximity sensor to indicate option two.

the lens. The LEDs illuminate the hand as it passes in front of the camera allowing easier recognition of hand shape and motion. Unfortunately, this system requires considerable processing and power and can be confused by strong illumination sources such as the sun or halogen lamps.

3 FreeDigiter

FreeDigiter is designed to allow simple gestural control of mobile devices such as MP3 players and mobile phones. Gestures are made by moving the fingers forward or backward past the sensor as shown in Figure 2. FreeDigiter employs a proximity sensor which emits infrared light that is reflected by the fingers. In order to differentiate the fingers from one another, the user must spread his fingers so that there are gaps between them. However, the user does not need to spread his fingers excessively as the proximity sensor has a rapid measurement time.

Numbers exceeding five have to be divided into smaller numbers. It does not matter how a number is split as long as the sum results in the required number for a command. Once triggered, FreeDigiter has a relatively long active detection period for counting fingers, allowing the user to go back and forth multiple times. A user generates a zero by holding his palm in front of the sensor for a short period of time, triggering the proximity sensor for a relatively long period of time compared to the moving fingers.

Example 1: Seven could be entered by moving the hand with four fingers held up past the sensor and then holding up three fingers on a return pass. Note that the hand's direction and order of numbers of fingers does not matter. In fact, the user could wave one finger in front of the sensor seven times for the same effect (though the physical process would require more time).

Example 2: Twenty could be entered by moving the hand past the sensor five times with four fingers held up.

This system offers high accuracy in detecting the fingers and works reliably under nearly any lighting conditions. Data analysis consists of counting the transitions in the proximity sensor's digital output. This simple analysis allows us to create a small and inexpensive device that consumes only a fraction of the power required for computer vision approaches. The result is an easily wearable device that does not impair wearing comfort.

4 Hardware

FreeDigiter uses an off-the-shelf Sharp proximity sensor GP2Y0D340K for gesture control. This sensor emits infrared light and detects the reflections from the fingers within the range of 10cm to 60cm. The greatest advantage of this sensor is its reliability in direct sunlight as well as in complete darkness. The proximity sensor measures 15x9.6x8.7mm, but could be made smaller. Its 6.4ms measurement cycle time allows a high temporal resolution and enables the detection of individual fingers moving past the sensor. As a finger passes the sensor, a peak with slightly variable length is produced. The shape of the pulses is shown in Figure 5.

We use a PIC16LF873 microcontroller to read the output of the proximity sensor. The microcontroller prepares data packets to be sent wirelessly using a Taiyo Yuden EYMF2CAMM Bluetooth module. Signal processing is performed by the receiving device, in this case a Linux computer. A picture of the implementation is shown in Figure 3 and a block diagram of the hardware components is shown in Figure 4. The entire FreeDigiter module draws a typical current of 58mA at 3.6V resulting in a power consumption of 210mW during transmission mode. The proximity sensor alone draws 16.2mA at 3.6V.

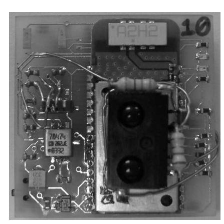

Figure 3. Proximity sensor mounted on Bluetooth module

5 Recognition

The proximity sensor emits infrared light that is reflected when a finger passes it. This reflection induces a current in the sensor that results in one pulse per finger. The total number of fingers, that is the sum of the number of pulses, is mapped to a command.

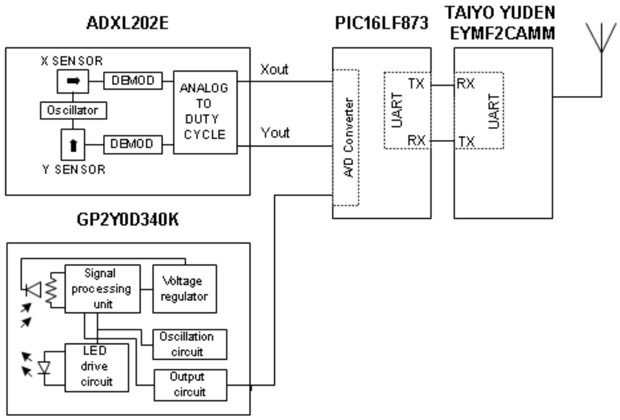

Figure 4. Block diagram of accelerometer, proximity sensor, microcontroller, and Bluetooth module

FreeDigiter's client software contains a simple state machine to recognize these pulses. The software analyzes the absolute value of the signal received from the PIC microcontroller and detects changes. Whenever a rising edge of a pulse which exceeds a certain threshold is detected, the state machine counts it as a detected finger. The state machine waits for further raising edges until, after 170 sequential non–detections, the programm executes the command corresponding to the total number detected and resets the state machine.

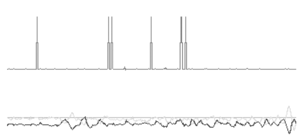

Figure 5. Above: Proximity sensor signal indicating an alternating progression of one and two fingers being sensed. Below: Accelerometer output (x and y axes)

6 Applications

FreeDigiter is designed to be suitable for various scenarios where a contact–free control system with simple commands might be needed. To inform our design, we are experimenting with two applications that can be controlled by entering numbers.

The first application, an MP3 player, plays the track whose number is entered by moving the appropriate number of fingers past the proximity sensor. With the trends in consumer electronics miniaturization, MP3 players of the near future could be reduced in size so that they consist of nothing more than an earplug. However, before such a device becomes practical, a new way to control the player is needed because they would no longer allow space to place buttons.

The second application simulates the functions of a mobile phone and offers a more sophisticated scheme where commands are linked to numbers. Our prototype application allows the user to accept a call, speed–dial, or prompt complete phone numbers without touching the phone or headset. The phone is simulated on a desktop PC. Since it is easier to enter small numbers, we chose to split commands into layered menus. The system starts in its default mode and offers the following options:

1. answer call (`one finger`)
2. speed-dial (`two fingers`)
3. dial phone number (`three fingers`)

After selection, the simulation proceeds to an appropriate submenu. For instance, after having chosen *answer call* the user either takes the call (`one finger`) or has the computer play a pre-recorded message (`two fingers`) that says that the user is currently driving in heavy traffic but intends to return the call in five minutes.

6.1 Initial Evaluation

We designed two experiments to collect data about the accuracy with which an experienced user can enter specific numbers. For the purposes of these initial studies, FreeDigiter was held in front of the user as if it was incorporated into a current portable consumer electronics device. Note that this test allows the user to see the position of his hand in relation to the proximity sensor. This feedback would not be available for an in-ear system.

Motivated by the MP3 player application, we created a task where the user would gesture numbers ranging from 1 to 20 (4+4+4+4+4 fingers) corresponding to possible tracks to select. The first two authors ran a total of 100 trials where they entered "tracks" as specified by a random number generator. 95% of the numbers were recognized correctly.

In the mobile phone application domain, only digits from 0 to 9 are required for dialing and for accessing the types of menu systems as described above. In 150 test gestures, the correct digit was recognized 99.0% of the time. This result is probably due to the fact that smaller numbers do not require the hand to go back and forth as many times. Thus, a entry system that specifies numbers larger than nine in a digit-by-digit fashion instead of explicit counting of fingers may prove more accurate.

Anecdotally, we found that often when the thumb is included in entering numbers, the palm covers the sensor. If the user tries to enter a number beginning with the thumb, he usually brings the hand further up with respect to the proximity sensor to compensate for the shortness of the thumb. However, this movement causes the hand to be too high relative to the sensor for distinguishing the fingers.

We suggest a strategy that restricts the use of the fingers to index, middle, ring, and little finger. While this strategy requires that the hand go back and forth more often, it prevents the palm from improperly covering the sensor.

7 Reliability

After proving that the system works under laboratory conditions, we evaluated the error rate for the use in "real world" mobile situations. Our goal for this experiment was to determine the number of false positives — errors where the sensor is active but no gestures are being made. One researcher wore the device as incorporated into a pair of headphones for four hours during an ordinary day in our laboratory. During that period, 136 false positives were detected. In order to discover what causes these errors, one researcher monitored the false detections, while the first researcher walked around. Note that since the system is wireless, this monitoring did not adversely effect the subject's movements.

Using this procedure, we found that the false positives were due to doorways, walls, corners, and people that were passed. We ascribe this problem to the long detection range of the proximity sensor (60cm). Therefore, the error rate depends considerably on what the user is doing and on his environment. Perhaps the proximity sensor can be tuned to have a smaller detection distance while still providing invariance to lighting conditions.

7.1 Accelerometer Switch

In order to prevent the system from detecting invalid inputs while turned on but not in use, we decided to integrated a modal switch into the existing system. Since no real switch is applicable for a contact–free device, a "soft" switch is needed.

FreeDigiter was extended with a low power, high resolution 2–axis Analog Devices ADXL202JE accelerometer. The PIC's A/D converter reads the analog signal from the accelerometer with 10 bit resolution. To begin gesture recognition, the user briskly nods his head to the right while facing forward. Another nod turns off gesture recognition.

To determine the effectiveness of this strategy, one researcher wore the headset system for 20 minutes to determine if normal activity would incorrectly trigger the accelerometer. No such false detections occurred, providing preliminary evidence that the head nod system may be an effective way to avoid unintentional digit entry.

8 Future Work

While the current system counts fingers by detecting rising edges of pulses from the proximity sensor, a more sophisticated detection system would exploit the varying pulse lengths caused by fingers versus doorways, walls, and corners. An analog proximity sensor which senses distance could also help distinguish between fingers and objects in the background, though experimentation is necessary to determine such a system's tolerance to variable lighting. More sophisticated algorithms together with fine-tuning of the proximity sensor may remove the need for the accelerometer trigger. However, the accelerometer and proximity sensor combination may prove useful in detecting a larger variation of interesting gestures. For example, "yes" and "no" head nods may be incorporated into the system for confirmation of finger gesture commands.

In order to determine FreeDigiters applicability in portable consumer electronics, a controlled HCI study on novice users should be performed. This study would determine gesture recognition accuracy as well as intuitiveness and user satisfaction with respect to example tasks.

9 Conclusion

We have described FreeDigiter, a gesture recognition system that allows entry of numbers by counting fingers swept past its proximity sensor. The system is high accuracy, lighting tolerant, requires little power, and is inexpensive. While our focus has been on creating mobile interfaces where buttons are inappropriate, FreeDigiter could be incorporated into stationary devices as well, allowing contact–free interactions in dirty or hazardous areas.

10 Acknowledgements

This work funded in part by NSF Career Grant #0093291 and the NIDRR Wireless RERC. Special thanks to Helene Brashear and Kent Lyons for their help.

References

[1] *Proceedings of the International Symposium on Wearable Computers*. IEEE Computer Society, Los Alamitos, CA, 1997-2003.

[2] *Proceedings of the International Conference on Ubiquitous Computing*. Springer-Verlag, Berlin, 1998-2003.

[3] J. Hernandez-Rebollar. *A New Instrumented Approach for Translating the American Sign Language into Sound and Text*. PhD thesis, George Washington University, Washington, DC, August 2003.

[4] T. Starner, J. Auxier, D. Ashbrook, and M. Gandy. The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In *Proceedings of IEEE International Symposium on Wearable Computing (ISWC 2000)*, 2000.

[5] T. Takahashi and F. Kishino. Hand gesture coding based on experiments using a hand gesture interface device. *SIGCHI Bulletin*, 23(2):67–73, 1991.