2011

Study of accelerometer assisted single key positioning user input systems

Chunming Gao
Michigan Technological University

Follow this and additional works at: https://digitalcommons.mtu.edu/etds

STUDY OF ACCELEROMETER ASSISTED SINGLE KEY POSITIONING USER INPUT SYSTEMS

By

CHUNMING GAO

A DISSERTATION

Submitted in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

(Computer Science)

MICHIGAN TECHNOLOGICAL UNIVERSITY

2011

This dissertation, "Study of Accelerometer Assisted Single Key Positioning User Input Systems", is hereby approved in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY in the field of Computer Science.

DEPARTMENT:
Computer Science

Signatures:

Dissertation Advisor: ________________________________
Dr. Robert Pastel

Co-Advisor: ________________________________
Dr. Jindong Tan

Department Chair: ________________________________
Dr. Steven Carr

Date: ________________________________

To my wife and our son

Table of Contents

List of Figures ………………………………………………………………………… 8

List of Tables ………………………………………………………………………… 12

Acknowledgments …………..………………………………………………………… 14

Abstract ……………………………………………………………………………… 16

1 Introduction ………………………………………………………………… 17

2 Background and Related Work……………………………………………… 20

2.1 Demands on Alternative Text Input Mechanisms…………............ 20

2.2 Research on Text Input with Various Media………….…………….... 21

2.3 Development of One-Handed Eyes-Free Text Input ……………...……….. 23

3 Single Key Positioning Input Systems……..…………………………………… 25

3.1 Input-Output Mappings ……………………………………………... 26

3.2 Clockwise Composition …………………………………………….. 27

3.3 Single Key Positioning Input System Models ………………………… 29

3.3.1 The 4+1 Positioning Input System……………………………. 29

3.3.2 The 8+1 Positioning Input System 30
3.4 The Modified Finite State Automaton Model 37
3.5 Evaluation of the Finite State Automaton Application to the Positioning Input System 39
4 Accelerometer Assisted Single Key Positioning User Input 41
4.1 Single Key Triggered Positioning Signal Selection 42
4.2 Determining Tilt Positions 43
4.3 Accelerometers' Effects during Clicking 47
5 User Test and Usability Study of the 4+1 Positioning Input System 50
5.1 Introduction 50
5.2 Methods 51
5.2.1 Participants 51
5.2.2 Apparatus 51
5.2.3 Procedures 51
5.3 Task 1: Tilt-Click Operations without Feedback 53
5.3.1 Task Goals 53
5.3.2 Task Procedures 53
5.3.3 Results 54
5.3.4 Discussion 59
5.4 Task 2: Number Input Operations over the 4+1 System 60
5.4.1 Task Goals 60
5.4.2 Task Procedures 61

5.4.3 Results 61
5.4.4 Discussion 65
5.5 Task 3: The 4+1 System Calculator Application 67
5.5.1 Task Goals 67
5.5.2 Task Procedures 67
5.5.3 Results 68
5.5.4 Discussion 70
6 User Test and Usability Study of the 8+1 Positioning Input System 72
6.1 Introduction 72
6.2 Methods 73
6.2.1 Participants 73
6.2.2 Procedures 73
6.3 Task 1: User Study of the 8+1 System 74
6.3.1 Task Goals 74
6.3.2 Task Procedures 74
6.3.3 Results 76
6.3.4 Discussion 80
6.4 Task 2: Usability Study of the 8+1 System Applications 82
6.4.1 Task Goals 82
6.4.2 Task Procedures 83
6.4.3 Results 84
6.4.4 Discussion 86

7 Conclusions... 87

7.1 Contribution.. 87

7.2 Future Work.. 88

7.3 Final Remarks.. 89

Bibliography.. 90

Appendix A YAUIM Experiments Informed Consent Form.............................. 96

Appendix B Experiment Database Tables.. 98

List of Figures

3.1 The five positions in the 4+1 input system....................................... 29

3.2 Clockwise number formation using five positions. Every two positions represent one number. The formation starts from south and goes clockwise... 30

3.3 The nine common positions 31

3.4 Clockwise letter formation using nine positions. Every two positions represent one letter. The formation starts from south and goes clockwise.... 32

3.5 Clockwise number formation using nine positions. Every two positions represent one number. The formation starts from south and goes clockwise... 33

3.6 A directed graph for the finite state automaton representation of the 8+1 positioning input system. The 8+1 positions input system is depicted as a hierarchical nested finite state automaton................................... 38

4.1 Accelerometer embedded devices that can use the YAUIM technique…….. 42
4.2 Accelerometer readings when the device is held upward at rest. On the average, the Z axis has the value of gravity and values of X and Y axes are close to zero……..………………………………………………………….. 42
4.3 Eight regions seen in the horizontal space when divided evenly…………… 45
4.4 Accelerometer embedded device tilting relative to the up orientation. The angle φ is relative to south and goes clockwise...………………………. 45
4.5 Readings of the three axials and the total acceleration during tilting with and without clicking. The operations were composed of up, north, west, and up without clicking, followed by a sequence of repeated operations with three clicks for each position..……………………………………. 48
5.1 The 90 percent pie chart of 4+1 system experiment. The ranges of south, west, north, and east were calculated based on the 90 percent probabilities…… 56
5.2 The 95 percent pie chart of 4+1 system experiment. The ranges of south, west, north, and east were calculated based on the 90 percent probabilities……...... 57
5.3 The learning curve of the 4+1 system operations of all the users……………. 58
5.4 Mean number input error rates for 4 categories. A10 is for alphabet sized 10. A7 is for alphabet sized 7. Map denotes operations by checking the map. No-map denotes operations without checking the map. Sessions 1 to 4 are illustrated from white to black………………………………….. 62

5.5 Mean performance times of the clicks for the alphabet sized 10. Map denotes operations by checking the map. No-map denotes operations without checking the map. 1st denotes the first click operations. 2nd denotes the second click operations. Sessions 1 to 4 are illustrated from white to black.... 64

5.6 Mean performance times of the clicks for the alphabet sized 7. Map denotes operations by checking the map. No-map denotes operations without checking the map. 1st denotes the first click operations. 2nd denotes the second click operations. Sessions 1 to 4 are illustrated from white to black.............. 64

5.7 Map of the 4+1 system calculator operators. Every two positions represent one operator.. 67

5.8 The learning curve of the calculations 70

5.9 The learning curve of the calculations without including the error corrections... 71

6.1 Response times over the 100 clicks for the self-definition operations 77

6.2 Individual participants' accuracy rates on longitudinal and diagonal orientations. The longitudinal orientations included the south, west, north, and east orientations. The diagonal orientations included the southwest, northwest, northeast, and southeast orientations....................................... 78

6.3 Individual response times on the longitudinal and diagonal orientations. The longitudinal orientations included the south, west, north, and east orientations. The diagonal orientations included the southwest, northwest, northeast, and southeast orientations.. 79

6.4 Comparison of the tilt angle ranges between the 90% probabilities for the operations without feedback and the predefined tilt angle ranges (unit: degree)..81

6.5 Comparison of the tilt angle ranges between the 90% probabilities for the operations with feedback and the predefined tilt angle ranges (unit: degree) .. 82

6.6 Clockwise mapping of numbers and operators for 8+1 system. Every two positions represent one number or one operator.................................. 84

6.7 Average completion times of the 8 questions 85

List of Tables

3.1 The mapping of the lowercase letters, numerical digits, and punctuation symbols…………………………………………………………………….. 35

3.2 The mapping of the uppercase letters, numeric digits, and punctuation symbols……………………………………………………………………. 35

3.3 The mapping of the punctuation symbols and special symbols …..………… 36

3.4 The mapping of the numbers and math operators at calculator mode...………. 36

4.1 The set of values of φ, x, y, z and the corresponding regions...………………. 45

4.2 The set of absolute value of the ratio of x over y, their signs and the corresponding regions ……………………………………………………. 46

4.3 Tilt angle range, signs of x and y, and the corresponding regions…………..….. 47

5.1 Mean error rates of the 5 sessions (unit: percentage)...……………………. 55

5.2 Means of the tilt angles of the primary regions (With standard deviations after the commas, angle unit: degree) …………………………………….. 56

5.3 Results of the linear regression of the logarithm of the response times by the logarithm of the click number …………………….……………… 58

5.4 Proposed region setups of the 8+1 positioning input system (unit: degree) …. 60
5.5 Mean error rates of the number inputs of the 4 categories ………..……. 62
5.6 Mean performance times in seconds of the two clicks in the 4 categories …… 63
5.7 Results of the linear regression of the logarithm of the response times by the logarithm of the session number with alphabet sized 10 ……..….. 66
5.8 Individual participants' time in seconds to complete Task 3 ......……………. 68
5.9 Eight individual participants' performance on individual questions (time unit: second) ……………………………………...…………………. 69
6.1 Average tilt angles and standard deviations for the nine orientations in self-definition operations (unit: degree) ………………………………. 76
6.2 Accuracy rate of the longitudinal and diagonal orientations for the 13 participants……………………………………………………………….. 78
6.3 Average response times in seconds of the longitudinal and the diagonal orientations for the 13 participants ………………………..……………... 79
6.4 Average tilt angles and standard deviations for the nine orientations in feedback operations (unit: degree)…………..…………………………...... 80
6.5 Individual completion times in seconds for the 8 questions …………………. 85

Acknowledgments

The pursuit of a Ph.D. degree is an adventure, a commitment, and an ultimate achievement. Without being associated with a group of great minds, researchers, and mentors, it would be almost impossible. I am blessed to have such great people behind me, who have devotedly supported me to success. At this time of completion, I want to sincerely thank the people around me who have helped me in the Ph.D. pursuit.

First I want to thank my advisors Dr. Robert Pastel and Dr. Jindong Tan. Dr. Pastel's expertise and insights in innovative human computer interface design have greatly inspired me. His devoting efforts in guidance and prompt response to my research questions have set a role model for me in graduate student supervising. I am fortunate and grateful that I have him as my dissertation advisor. Dr. Tan is one of the great researchers and mentors I have ever met. He has contributed in all aspects to my academic endeavors, motivated me to do original research with high standards. I am sincerely grateful for all his efforts in training me into a successful researcher.

I want to give special thanks to my committee members Dr. Steven Carr, Dr. Zhenlin Wang, and Dr. Jianping Dong. Dr. Carr has been the person who brought me to the computer science field. In the past years, he has always been a resource for me to hold on to my pursuit of the Ph.D. degree. Dr. Wang's advice, careful reviews and comments on the dissertation have assisted me in achieving the highest quality. Dr. Dong has provided careful suggestions in the experiments of my Ph.D. research. Her expertise in the design and analysis of experiments has been a strong resource in my final work.

I also want to thank all the colleagues and friends who have helped me in the enduring pursuit of my degree. A special thank-you goes to our Computer Network and System Administration program Chair in the School of Technology. Professor Guy Hembroff has arranged my teaching load carefully so that I have enough time to work on my research. I also give special thanks to my friend Dr. Zhaogong Zhang who has generously demonstrated to me how to use the analysis tools. A special thank-you also goes to Dr. Yu Cai and Dr. Xinli Wang, who have provided me precious advices in pushing my Ph.D. research forward.

Next, I must thank my wife, our parents, our son, our brothers and sisters. They have always been the support and motivation for me to achieve the highest goals. I want to sincerely thank them for being with me. With them, I find the purpose to never give up on achieving dreams. Besides them, I also want to thank all the professors, colleagues, friends who have helped me directly or indirectly. Without them, I could not have come to this day.

In the end, I will give all the praise to the almighty God, for all the great advisors, mentors, family, friends, and great research ideas and insights that have been granted to me!

Abstract

New designs of user input systems have resulted from the developing technologies and specialized user demands. Conventional keyboard and mouse input devices still dominate the input speed, but other input mechanisms are demanded in special application scenarios. Touch screen and stylus input methods have been widely adopted by PDAs and smartphones. Reduced keypads are necessary for mobile phones. A new design trend is exploring the design space in applications requiring single-handed input, even with eyes-free on small mobile devices. This requires as few keys on the input device to make it feasible to operate. But representing many characters with fewer keys can make the input ambiguous. Accelerometers embedded in mobile devices provide opportunities to combine device movements with keys for input signal disambiguation. Recent research has explored its design space for text input.

In this dissertation an accelerometer assisted single key positioning input system is developed. It utilizes input device tilt directions as input signals and maps their sequences to output characters and functions. A generic positioning model is developed as guidelines for designing positioning input systems. A calculator prototype and a text input prototype on the 4+1 (5 positions) positioning input system and the 8+1 (9 positions) positioning input system are implemented using accelerometer readings on a smartphone. Users use one physical key to operate and feedbacks are audible. Controlled experiments are conducted to evaluate the feasibility, learnability, and design space of the accelerometer assisted single key positioning input system. This research can provide inspiration and innovational references for researchers and practitioners in the positioning user input designs, applications of accelerometer readings, and new development of standard machine readable sign languages.

Chapter 1

Introduction

The advances of digital technology have brought pervasive computing devices into people's daily lives. With devices getting smaller and smaller, the display screen and input mechanisms must also shrink. This has presented a dilemma on the interface design. On the one hand, smaller devices are good because users can carry them anywhere and hold them with one hand. On the other hand, the small size of the interface makes user interactions harder. The traditional user input such as the keyboard and mouse is generally impractical for these small devices. On mobile phones, keypads with reduced keys are used to input text. On PDAs and smartphones, gesturing and touch screen techniques are used to input text. On some smaller devices like the music device, the iPod Shuffle, there is no screen and users can only make sequential or random selections by using a few buttons. User input on such small devices is a challenge, and generally the functionality of the device is limited.

Text input is one medium for providing increased functionality on small devices. For example, text input methods would enable the searching of tunes on the iPod Shuffle. One handed input may be advantageous on small devices. If the device does not have a screen then visual feedback is not possible and eyes free input would be advantageous.

One-handed eyes-free text input technique would not only benefit small devices, but it could also benefit users in special environments and users without sight or with other disabilities. For text input using a large wall display (1,2), non-visual feedback during walking (3,4), and applications for users with disabilities (5-8), one-handed eyes-free input could be advantageous. A study (9) showed that most mobile phone users prefer operating the phones with only one hand because the other hand is not available. In some fieldwork or military applications, one-handed eyes-free user input could be required.

In this dissertation, we seek to design a one-handed eyes-free text input technique. We will develop a generic model to assist in designing and evaluating the limited key representation of a larger character set. Using the traditional keyboard, one character is represented by one or a combination of two keys. A mode change permits the same key to represent either the lowercase or uppercase. Consequently, there is no uncertainty of the input for any keystroke using the traditional keyboard, but representing multiple characters with fewer keys ambiguity becomes a problem. The model is based on a modified finite state automaton and helps to resolve the potential problem of ambiguity in character mappings. We will present a positioning input model and implement the text input using only one key and one accelerometer. Also, we eliminate the need for a screen or visible feedback by using audible feedback. The sequence of device tilt positions can be used for both text and command input. We call this single key tilt input technique YAUIM (Yet Another User Input Method) (10).

In this research, data input is via a handheld device with a built-in tri-axial accelerometer. A tri-axial accelerometer detects accelerations in three dimensions in real time. The device tilt positions are derived from the accelerations and mapped to characters. However, the number of tilt positions is limited because tilt orientations could be hard for users to classify accurately. This research limits the tilt directions to up, front, back, left, right and four diagonal directions in the horizontal space. The modified finite state automaton model will assist with the design and analysis of character formations. In our implementation, two consecutive tilt positions represent one character. Another challenge for the technique is that accelerometer readings are noisy. All the vibrations are recorded by accelerometers, so unintentional movements will interfere with the intended input. Tilt reading can also be difficult without delineating the boundaries between consecutive tilts. Our technique uses a single keystroke (pressing a button on the device) to delineate the tilt directions. Another challenge is insuring that the technique is easy to learn and simple to perform, otherwise hand movements can be tiresome and make the mechanism inapplicable. Consequently, we develop the technique on the common tilt directions familiar to users.

This dissertation is in four phases. First, a literary review examines small device input techniques and one handed eyes free input techniques. Second, a modified finite state automaton is developed to represent character input based on limited input positions or keys. Third, accelerometer readings are analyzed in order to implement the automaton model. Fourth, controlled user and usability experiments [IRB approval number M0565] of the implementation space and system performances are presented and analyzed.

Chapter 2

Background and Related Work

2.1 Demands on Alternative Text Input Mechanisms

While people can use conventional keyboard and mouse input for desktop and laptop computers, they may need different input techniques while standing or moving during a presentation, or in fieldwork. Also, special users such as people with disabilities may need unconventional input methods. Research has been conducted to enable user input via hand gestures, head movements, foot steps, voice, etc.

Here are some examples of interface designs for special application scenarios. Mid-air (1) and Soap-mid-air (2) proposed to manipulate objects on a large wall display using a handheld device. In their research, the pointing positions are detected by built-in light sensors and mapped to the screen objects. TwoStick (11) uses a game controller for text input for video games. Touch-wheel (12) designed a one-handed input on a touch-wheel device. Watch-top (13) tried to enable text entry with five keys on watch-like devices by enabling dictionary predictions. DistScroll (14) used an infrared light sensor to detect user-device distance to scroll the menus on the screen. SCURRY (15) designed a hand wearable device with native accelerometers and detected hand and finger movements for inputting. Foot-Step (3) used foot steps to operate the user input system for music menu selection. ToothClick (5) used head movements to enable screen selections for handicapped people. All these input mechanisms have addressed special needs for unconventional user input.

2.2 Research on Text Input with Various Media

Among the novel input designs and developments, hand-held user input devices have been broadly studied (1,2,12-14,16) to support the demands of pervasive computing. Many of these designs have focused on mobile phone applications. Text input on mobile phones is necessary for text messaging. Non-texting users would still need to input text for example while recording names with phone numbers. On the mobile phone keypad, the English letters, numeric digits, and special symbols are designated using only 12 keys. The keys are labeled with four or five characters, e.g., key 2 is labeled with `2', `a', `b', and `c', and key 9 is labeled with `9', `w', `x', `y', `z'. The usual usage, called multi-tap, uses consecutive taps to designate the specific character. Hence each character input requires one to five keystrokes, i.e., the performance is between 1 to 5 KSPC (keystrokes per character). Alternative methods to disambiguate keypad input are TiltText (17) which uses the combination of keys and phone tilt angles and T9 (18) which uses the sequence of keystrokes to predict user input. These techniques have illustrated that fewer key input is possible and performance can be enhanced by using dictionary predictions.

Researchers have designed text input with fewer than 12 keys. Bellman and Mackenzie (19) studied the feasibility of five keys input with various layouts of letters. Evreinova *et al.* (6) presented a four key input design achieving an average of 3-5 WPM (words per minute). MacKenzie (20) studied a three key setup which uses two arrow keys to maneuver a cursor over a linear sequence of characters, and uses one key to select the desired characters. The average KSPC can be up to 10.66. Most of the few key designs are based on navigating followed by selecting. These techniques usually suffer from a high navigation overhead. They can be enhanced by optimizing the layout of the characters so that there is lower KSPC and learning time.

Some small device input designs are based on touch screen interactions. Users select an object either directly on the touch screen or indirectly by a sequence of operations. An example is AppLens and LauchTile (16). It presented a one-handed thumb interaction technique on small mobile devices for a zooming interface that provided multiple views of application data. It utilized thumb gestures on the screen to activate the various functions. Like AppLens and LauchTile, most text input techniques on touch screen use finger gestures. Other techniques use interactions via the accelerometers to select objects or characters on the screen. Unigesture (21) is an input technique based on accelerometer interaction. It uses a sequence of tilt positions to predict words, but does not input individual character or words not in the dictionary.

The involvement of accelerometers and touch screen in user input design brought new opportunities for small mobile device operations. Native accelerometers in small devices can be used to analyze device movements and tilts (10,22), or to observe gestures with the device (4,23). The gestures and tilts have been used for controls in games (24), monitoring physical activity (22), operations on the mobile phone (25), menu selections (3,26), and text input (10,17). Gestures are natural for simulating sports like in Wii sport games. Gestures have limitations when applied to text input because they are less reliable and error prone (23). Tilting inputs are relatively reliable (27), but the number of tilt directions is limited. Some approaches have tried to use multiple keys to combine with the tilt operations (17,18) to use a sequence of operations and checking the dictionary for the meaning of user input (21), or to use multi-level selection (29).

2.3 Development of One-Handed Eyes-Free Text Input

Some recent developments on small device input have explored one-handed eyes-free input techniques (10,30,31). One handed eyes-free input is important on small devices because many mobile phone users use one hand inputs while the other hand is occupied (9). Smaller devices such as iPod Shuffle does not have a screen, consequently an eyes-free input technique is necessary. One-handed eyes-free text input technique would not only benefit small devices, but could also benefit users in special working environments and users without sight. One technique for one-handed eyes-free input is to use audible feedback instead of visual feedback. No-look (30) and VoiceOver (31) both used audio feedback and implemented one-hand eyes-free input on smartphones. VoiceOver built the full QWERTY soft keyboard on screen. When a finger touches a key on the screen the letter is pronounced and a second tap on the screen confirms the selection. No-look pursued a multi-touch method in a multi-level selection technique. VoiceOver achieved an input speed of 0.76 WPM and No-look gained 1.67 WPM (30).

Small devices appear to have lower input speed and lower WPM than what can be achieved on a conventional keyboard input. Many techniques have explored enhancing the input speed (18,32-37). Many techniques use input context to predict the user complete input, for example T9 (18) and LetterWise (32). T9 is a very popular technique for texting on the mobile phone. Potentially users can type a word with only one keystroke for one letter. With the help of preloaded dictionary, T9 provides a guessed word from the keystroke sequences. When two or more words have the same keystroke sequences, the system could offer the user an option to select the desired word. LetterWise uses prefixes of the user input to guess the intended letter. A prefix is formed by the letters preceding the current keystroke. The next letter is suggested based on the probabilities of letter sequences in a language. These speed enhancements are supplementary to many input mechanisms and can be adopted by many input designs.

YAUIM (10) uses a single key combined with tilt positions to input characters and commands. Many applications, which require short text input, phone number dialing, calculation, etc. could use YAUIM input technique. Novice users achieved 2.8 WPM in their first hour of use. In a calculator application, a speed of 33.4 seconds per calculation was achieved by novice users for five digits calculation.

Chapter 3

Single Key Positioning Input Systems

For an input system, the input signals can be any readings from the device operated by user. The output can be a text character, a menu selection, or any other response to the system. The mapping from input signal to output response can be a one-to-one mapping or a many-to-many mapping depending on the number of unique input signals and the number of unique output responses. The full QWERTY keyboard maps each key to a specific English letter or a special symbol depending on the input mode. A mobile phone keypad uses the same key to represent multiple characters. A measure of efficiency for keyboards or mobile keypads is keystrokes per character (KSPC) (38). Lower KSPC implies more efficient inputs. An experimental measurement of performance is the words per minute (WPM). Assuming an average of five characters per word, WPM can be estimated by dividing five into the number of characters inputted in one minute. In this study, tilt positions of the input device are used to input characters. A single key is used to designate a position.

This chapter is organized as follows. First, general input-output mapping methods are described and the corresponding output cost is discussed. Second, the clockwise composition rule for inputting is explained. This rule is applied to two different alphabets, developing the 4+1 and the 8+1 positioning input systems. Third, this chapter explains a modified finite state automaton model of the system which assists the design and the implementation of the input system. Finally, the applications of the automaton model are discussed.

3.1 Input-Output Mappings

Definition 1 *Selection-based mapping: Selection-based mapping is an input-output mapping where output characters are defined by a sequence of selections from a hierarchy of graphical presentations.*

Examples of input systems based on selection-based mappings are Bimanual (39), Tilt-Text (17), MultiTap (28), TiltType (28), GesText (29), No-look (30), VoiceOver (31). The primary operations performed by the user during input are selections from a single level or multiple levels of graphical presentations. For example in VoiceOver, a single level selection-based mapping, users select characters from a layout on the screen. No-look is a two-level design where characters are inputted by selecting a group then selecting a character in the group. In MultiTap a character is inputted by pressing a key multiple times until the desired character appears on the screen.

Definition 2 *Rule-based mapping: Rule-based mapping is an input-output mapping where output characters are defined by rules on combining the sequence of input signals.*

Examples of input systems based on rule-based mappings are like Unigesture (21) and YAUIM (10). Unigesture uses a sequence of keystrokes to index into a dictionary and form words instead of directly forming single characters. YAUIM, the input technique of this dissertation, use rules to represent individual characters.

The mapping from input to output can be represented by:

$$Y = f\,(mode, X1, X2, \ldots, Xi, \ldots, Xn); \qquad (3.1)$$

where Xi (i = 1 to n) is an input signal, Y is an output character, n is the size of the input signal sequence for each output, the function f is the conversion rule, and mode denotes the input mode. The *mode* is used to represent lowercase input, uppercase input, special symbol input, mouse operation, and calculator, etc. The *mode* could have been specified by the user using the input system. In this case, it is defined by a similar formula.

In rule-based mappings, KSPC is the number of keystrokes to specify the mode plus the size of the sequence for an output. In a selection-based mapping, KSPC is the number of keystrokes to specify the mode plus the number of levels that are navigated during the input. The fundamental difference between selection-based mapping and rule-based mapping is how they affect the input devices and operations. In selection-based designs, graphical presentations are required. Users input characters by making selections on the presentations or layouts represented by the graphics. Consequently, the user must be able to see the layouts. In rule-based designs, layouts are not necessary, hence the device does not need a screen and can be small. This research pursues a rule-based approach.

3.2 Clockwise Composition

The Positioning Signals. A positioning input system refers to a technique which uses positions or orientations of the device as input signals. Every position is an input character. The positions can be the tilt orientations or the movement directions of the input device. In this study, we use tilt orientations of the input device as the input signals.

The Alphabet. For illustration, we will only consider the output character set consisting of 26 English letters, 10 numeric digits, 15 punctuation symbols, 14 special symbols, and a few special functions, or a subset of these characters. For example, a phone dialing application will only need the 10 numeric characters and a few function symbols. A simple calculator application will only need the 10 digits and a few operator symbols. A note taking application will use more characters and symbols. We call the set of these characters in a specific application the alphabet of the application. Hence an alphabet is a subset of the output character set for the input system.

To be explicit, the characters and functions in this research are:

1) 26 English letters: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z
2) 10 numeric digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
3) 15 Punctuation symbols: period, comma, semicolon, colon, question mark, exclamation point, underscore, double quotation marks, apostrophe, left and right parentheses, left and right square brackets, left and right curly braces.
4) 14 Special symbols: @, #, $, %, &, +, -, *, /, =, <, >, |.
5) Special functions: lower case letter input mode, upper case letter input mode, mouse operation mode, punctuation and special symbols input mode, calculator mode, backspace, space, selection/enter.

These characters and functions are required for "texting" and note-taking applications. Many of the punctuation and special symbols are infrequently used. We also assume that mouse or pointing operations will not be utilized during eyes-free application scenarios. We consider the 26 English letters, 10 numeric digits, and 4 common punctuation symbols (`,', `.', `?', `!') as the most used characters during text input and design the input system for texting applications using this reduced alphabet.

Clockwise Composition. To enhance the learning of the mapping of the characters, we propose to use the clockwise concept to map the alphabet. The clockwise composition refers to the circular layout of the tilt positions. The characters of the alphabet are then mapped clockwise on this circle of positions based on either a one-to-one or a many-to-many mapping rule.

Clockwise composition rule will be used to compose the output alphabet. For our positioning input system, every two positions will represent one output character. Using the clockwise composition, the 4+1 and 8+1 positioning input systems will be developed in the next section.

3.3 Single Key Positioning Input System Models

3.3.1 The 4+1 Positioning Input System

The 4+1 positioning input system has five positions for inputting characters. Relative to the users, the five positions are backward, left, forward, right and up. They correspond to the directions south, west, north, east, and up, as in a map which is illustrated in Figure 3.1.

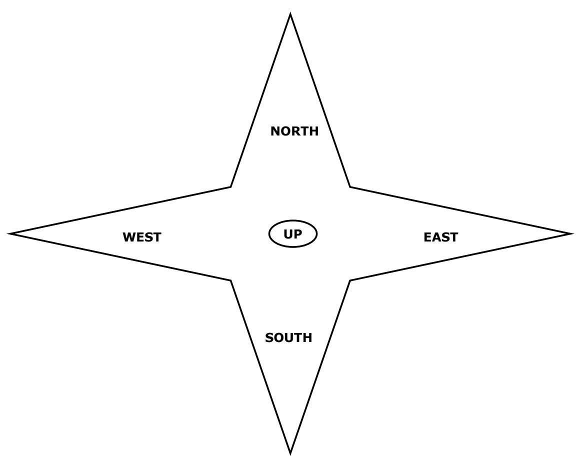

Figure 3.1: The five positions in the 4+1 input system.

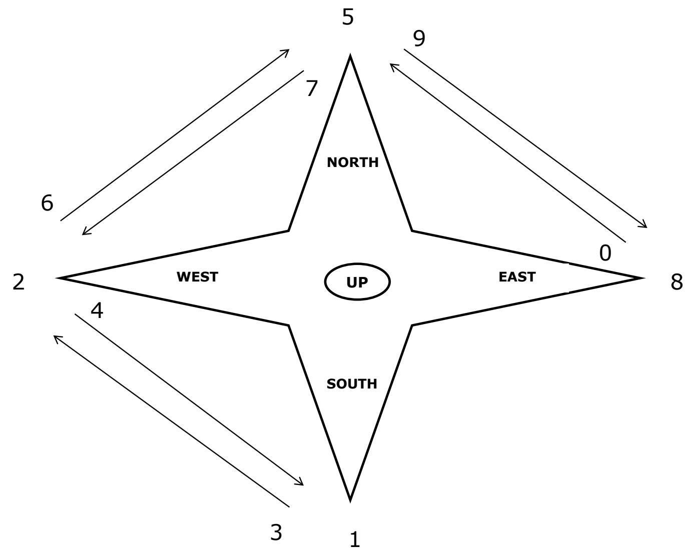

Figure 3.2: Clockwise number formation using five positions. Every two positions represent one number. The formation starts from south and goes clockwise.

Our clockwise composition begins in the south position and maps numbers or characters to two positions. This clockwise composition is a many-to-many mapping. For example, south + south outputs the number 1. Then going clockwise, west + west outputs the number 2. The number 3 is inputted by south + west; number 4 is west + south. The north + north input positions correspond to the number 5. The number 6 is west + north, and the number 7 is north + west. The composition rule continues clockwise until all 10 numbers are defined. All the mappings are illustrated in Figure 3.2. The up position is reserved for special functions and will be discussed in the experimental studies of the application of the 4+1 positioning input system.

3.3.2 The 8+1 Positioning Input System

The 8+1 positioning input system has nine positions for inputting characters. The nine tilt positions relative to the user are illustrated in Figure 3.3, and marked as S(south), SW

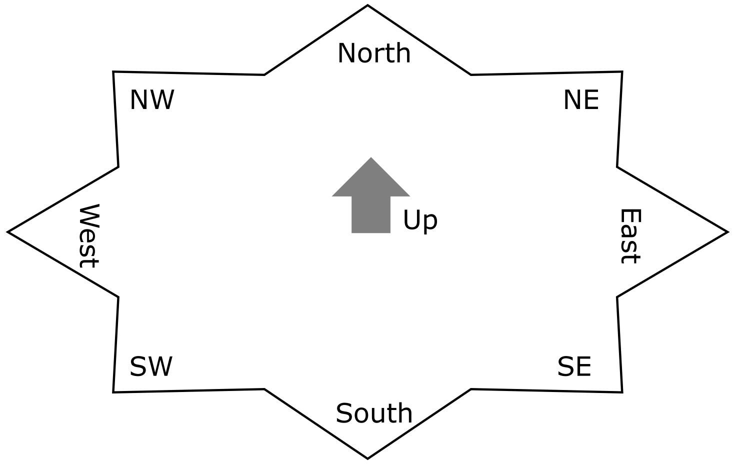

Figure 3.3: The nine common positions

(south-west), W (west), NW (north-west), N (north), NE (north-east), E (east), SE (south-east), and U (up). We divide the input position space into four areas: southwest, northwest, northeast, and southeast areas. The southwest area includes S, SW, and W tilt positions. The northwest area includes W, NW, and N tilt positions. The northeast area includes N, NE, and E tilt positions. The southeast area includes E, SE, and S tilt positions. We assume that the southwest, northwest, and northeast areas are comfortable titling areas for right handed users.

The southwest area is for inputting the letters a, b, c, d, e, f, g, h, and i. The northwest area is for inputting the letters j, k, l, m, n, o, p, and q. The northeast area is for inputting the letters r, s, t, u, v, w, x and y. The southeast area is for inputting the letter z. Every two consecutive tilt positions in an area represent a single letter. Letter input starts from south and goes clockwise. To illustrate this clockwise formation, we present the following examples:

S + S => a

SW + SW => b

S + SW => c

SW + S => d

W + W => e

S + W => f

W + S => g

SW + W => h

W + SW => i

NW + NW => j

W + NW => k

NW + W => l

Etc.

The composition rules are illustrated in Figure 3.4.

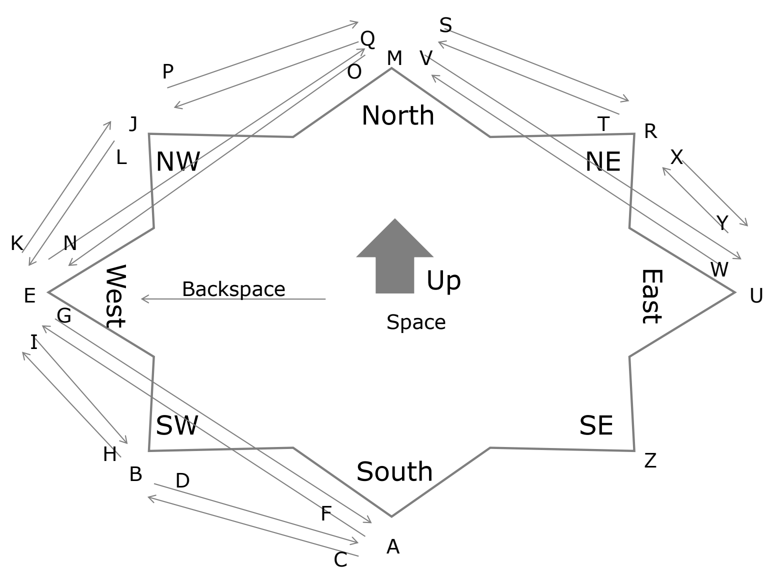

Figure 3.4: Clockwise letter formation using nine positions. Every two positions represent one letter. The formation starts from south and goes clockwise.

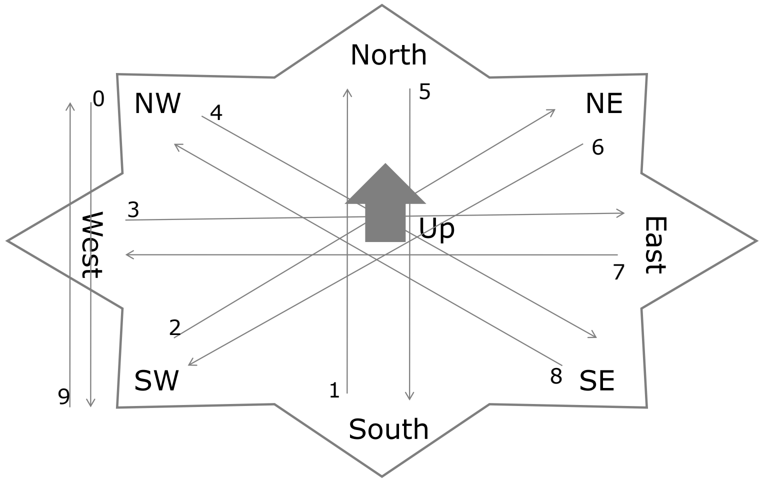

Figure 3.5: Clockwise number formation using nine positions. Every two positions represent one number. The formation starts from south and goes clockwise.

The ten numeric digits are represented by the combinations of opposite and diagonal positions and illustrated in Figure 3.5. It starts from the South and goes clockwise as follows:

S + N => 1

SW + NE => 2

W + E => 3

NW + SE => 4

N + S => 5

NE + SW => 6

E + W => 7

SE + NW => 8

SW + NW => 9

NW + SW => 0

The combination of positions NW, NE and SE form the punctuation symbols: `,', `.', `?', and `!':

NW + NE => ,

NE + NW => .

NE + SE => ?

SE + NE =>

Combining the up position (U) with other tilt positions denotes special functions. South followed by U enables lowercase input mode. West followed by U enables uppercase input mode. North followed by U enables the special symbol input mode. East followed by U enables the calculator mode which is an application for calculation. U followed by west denotes backspace. U followed by east denotes suggested word selection or Enter. Both of lower and upper letter input modes include the numeric digits and the four punctuation symbols input. In the character input mode, two consecutive U positions denote a space character.

The mappings of all the characters are designated in Table 3.1. Uppercase letters and lowercase letters are inputted using separate input modes, so similar tilt combinations are used to represent uppercase letters. The mappings for uppercase letters along with the numeric digits and four punctuation symbols are listed in Table 3.2. The mappings for the special symbols are listed in Table 3.3. The numbers and math operators are listed in Table 3.4.

Table 3.1

The mapping of the lowercase letters, numerical digits, and punctuation symbols

	S	SW	W	NW	N	NE	E	SE	U
S	a	c	f		1				lowercase
SW	d	b	h	9		2			
W	g	i	e	k	n		3		uppercase
NW		0	l	j	p	,		4	
N	5		o	q	m	s	v		symbols
NE		6		.	t	r	x	?	
E			7		w	y	u		calculator
SE				8		!		z	
U			<=						space

Table 3.2

The mapping of the uppercase letters, numeric digits, and punctuation symbols

	S	SW	W	NW	N	NE	E	SE	U
S	A	C	F		1				lowercase
SW	D	B	H	9		2			
W	G	I	E	K	N		3		uppercase
NW		0	L	J	P	,		4	
N	5		O	Q	M	S	V		symbols
NE		6		.	T	R	X	?	
E			7		W	Y	U		calculator
SE				8		!		Z	
U			<=						space

Table 3.3
The mapping of the punctuation symbols and special symbols

	S	SW	W	NW	N	NE	E	SE	U
S	.	;	!		=				lowercase
SW	:	,	“						
W	_		?	)	{				uppercase
NW			(	@	[				
N	|		}	]	#	+	*		symbols
NE					-	$	>		
E					/	<	%		calculator
SE								&	
U			<=						space

Table 3.4
The mapping of the numbers and math operators at calculator mode

	S	SW	W	NW	N	NE	E	SE	U
S	+				1				lowercase
SW				9		2			
W			-				3		uppercase
NW		0						4	
N	5				*				symbols
NE		6				=			
E			7				/		calculator
SE				8					
U			<=						space

3.4 The Modified Finite State Automaton Model

Finite state automaton is a tool to parse regular languages. It is formally defined by a 5-tuple ($Q, \Sigma, \delta, q0, F$) (40). We modify the finite state automaton formalism in order to explain the parsing of our clockwise composition input system. The modified finite state automaton model is represented by a 7-tuple ($Q, \Sigma, \delta, q0, F, f, \Omega$). Σ denotes the input character set. Ω will denote the output character set. In an accept state, the conversion rule, f, is applied to the accepted input characters producing output characters.

Definition 3 *The finite state automaton for the positioning input system is defined as a 7-tuple ($Q, \Sigma, \delta, q0, F, f, \Omega$), where*

1. *Q is the finite state set*
2. *Σ is the input character set*
3. *$\delta: Q \times \Sigma \rightarrow Q$ is the transition function*
4. *$q0$ is the start state*
5. *F is the accept states set*
6. *f is the conversion rule to convert the input character into the output character*
7. *Ω is the output character set*

A finite state automaton can be described by a directed graph $G = (V, E)$, where V are the vertices and E are the edges of the graph G. Every vertex represents a state where the device waits for a new user input. Every edge represents what state the specific user input will lead to. When a user inputs a character, the system is triggered into a new state. We label the output actions by using the notation from the study of Sandnes (41). If an input character `S' triggers an `a' to output, it is denoted as S:a. If there is no output for the input character `S', then it is designated by S:.

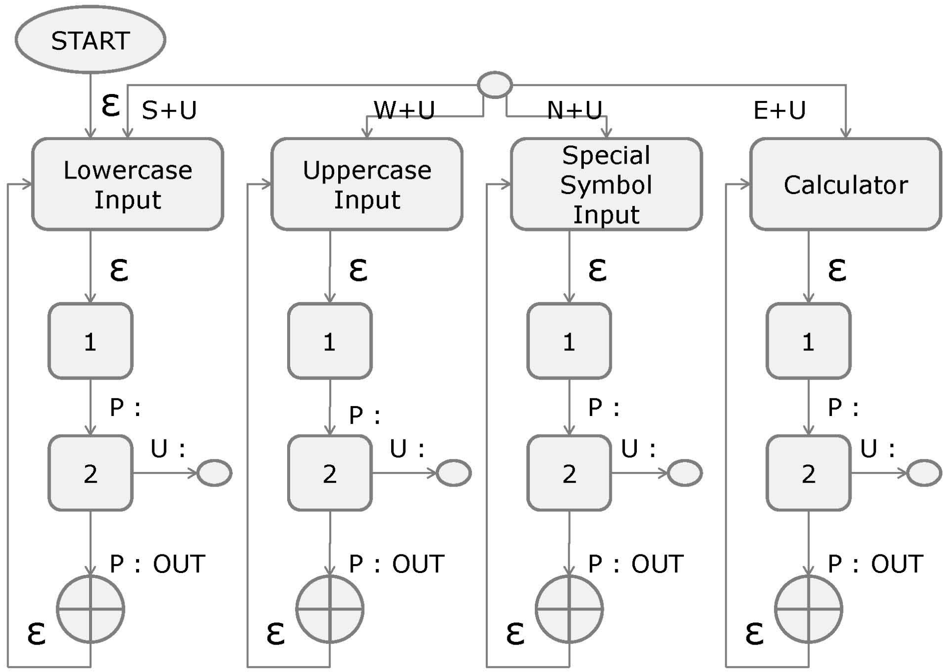

Figure 3.6: A directed graph for the finite state automaton representation of the 8+1 positioning input system. The 8+1 positions input system is depicted as a hierarchical nested finite state automaton.

The 8+1 positioning input system is depicted as a hierarchical nested finite state automaton in Figure 3.6. The START state is the top most node. The second level nodes are represented with large rectangular symbols which are the input modes. The lower level nodes, represented by square symbols, are for the character transition states and ACCEPT states. The crossed circles represent the ACCEPT states. Small circles are equivalent points. The symbol ε denotes transitions without any input. H denotes an input of the horizontal position such as S, SW, W, NW, N, NE, E, and SE. P denotes H and U(up) positions. OUT denotes any output character. Unique to this finite state automaton model is that the ACCEPT state will use conversion tables to translate the sequence of two positions into an output character or an operation.

3.5 Evaluation of the Finite State Automaton Application to the Positioning Input System

The finite state automaton is designed to accept any user input. Hence there is no error state. If the user inputs are parsed as undefined in the conversion table, the automaton will move to the beginning of the input mode. The finite state automaton can help design the positioning input system. First, it can verify the correctness of the design. Second, it can help manage the system input modules. Third, it can assist in designing an efficient system.

Correctness Study. Correctness implies full coverage of the output alphabet. In another word, for every character in the alphabet, there must be a path from START to an ACCEPT state. In order to create the alphabet after the START state, there must be a path from any one state to any other state. In this way, a character could be inputted without going to the START state. The full coverage could be verified by applying a recursive depth-first marking algorithm to traverse the directed graph of the finite state automaton.

Algorithm 1 *Depth-first-marking*

> *Upon input graph G = (V, E) and a vertex v,*
> *Mark v as visited.*
> *For each outgoing edge (v, w) of v do*
> > *if w has not been visited then*
> > *Recursively call Depth-first-marking(G, w).*

Flexibility Study. The finite state automaton can be used to manage the input modules. From the graph representation of Figure 3.6, it is apparent that sub-graphs can be added to the graph. The sub-graphs represent new input modules. The top level states are like the index into the automaton modules and corresponding lower level states form the components of each module. This representation reflects an object oriented software design. Consequently, the finite state automaton model can be used as a guide during the software development.

Efficiency Evaluation. The finite state automaton graph illustrates that two input characters (positions) create one output character, if the input mode does not change. If the input mode changes then the cost for character formation increases. Hence the average character cost in the system is expected to be slightly above two input operations.

Chapter 4

Accelerometer Assisted Single Key Positioning User Input

A static tri-axial accelerometer detects gravity. When the accelerometer is tilted into different positions the total gravity remains the same, but the values from the individual axes change. Consequently, the values of the three axes can determine the tilting direction. This chapter explores capabilities of accelerometers to implement the single key positioning input systems.

Figure 4.1 depicts possible YAUIM devices embedded with accelerometers. The first is a joystick which could be used for regular desktop computing. The second one is a pen computing device with onboard storage and wireless communication. The pen could be used for remote input device during presentation using a large display (11). The third is a small mobile device with embedded accelerometer systems.

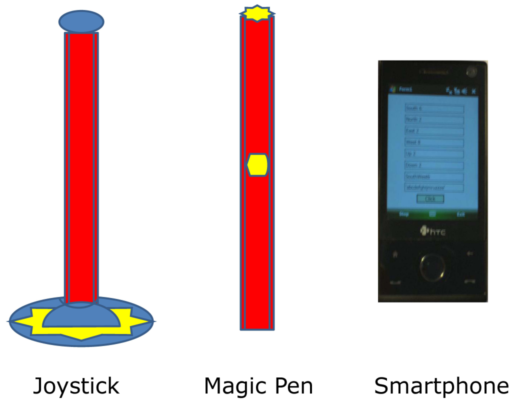

Figure 4.1: Accelerometer embedded devices that can use the YAUIM technique.

4.1 Single Key Triggered Positioning Signal Selection

Figure 4.2 displays the readings of the three axes while the tri-axial accelerometer is held upward during 16 seconds. The Z axis signal varies around 9.8 m/s^2 average value. The X and Y signals vary around their average values close to zero. Filtering the signal can remove the variations in the signal.

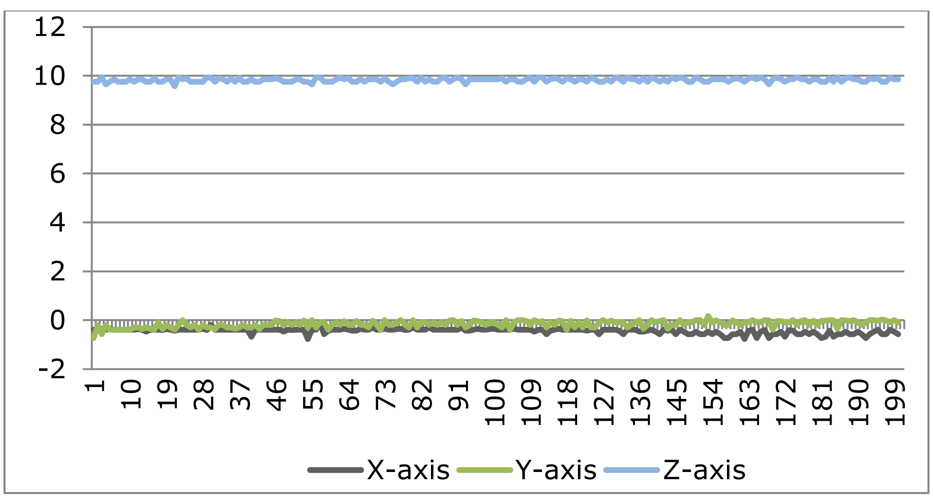

Figure 4.2: Accelerometer readings when the device is held upward at rest. On the average, the Z axis has the value of gravity and values of X and Y axes are close to zero (unit: m/s^2)

The small deviations, noise signals, are uniformly distributed along the 16 seconds. We can average the signal over a short period of time *TICK* to filter the noise and determine valid signal *S*:

$$S = \frac{1}{TICK}\int_{t0}^{t0+TICK} S(t)dt \tag{4.1}$$

where t_0 denotes the moment of the key press. $S(t)$ is the accelerometer reading at time t. S is the mean value in the period of *TICK*, and it will be used for the individual readings of the three axes. The optimal *TICK* is determined by the specific device and determined experimentally.

4.2 Determining Tilt Positions

Generally, relative measurements are more reliable than absolute measurements. Relative measurement from two or more sensors can compensate for instrument bias and environmental effects, while an absolute measurement from a single sensor requires strict control of undesired effects. For example differential amplifiers are far more effective at eliminate noise than single input amplifiers. We prefer to develop an orientation measurement technique that does not rely on the absolute value of gravity, but rather determines the orientation from the ratio of two accelerometer readings.

We denote the value of gravity by g. The readings from three axes are x, y and z. They have the following relationship:

$$g = \sqrt{x^2 + y^2 + z^2}$$

The technique requires that the up direction (z-axis) coincides with the direction of gravity. When the device points to the left (west), and the tilt is 90 degrees, x is equal to g. When the device points forward (north) and the tilt is 90 degrees, y has the value of g. In general, when the device tilts to the left an angle of θ, the value of x is $gsin(\theta)$, z is equal to $gcos(\theta)$ and y is zero. When the device tilts forward an angle θ, y is equal to $gsin(\theta)$, z is equal to $gcos(\theta)$, and x is zero. Similarly, when the device tilts to the right (east) or back (south) at the angle of θ, x and y are equal to $-gsin(\theta)$ respectively, while z is equal to $gcos(\theta)$.

When the device is tilted to the front left (northwest) with an angle θ, z becomes $gcos(\theta)$ and x and y become $gsin(\theta)\sqrt{2}/2$. When the device is tilted to other diagonal directions with the angle θ, x and y will become correspondingly negative but with absolute value $gsin(\theta)\sqrt{2}/2$.

To derive the tilt direction from the readings of three axes, we assume that the tilts angle is no more than 90 degrees. We can ignore the value of z, and conclude that if x is greater than zero while y is zero then the device is tilted to the left. If y is greater than zero and x is zero then the device is tilted forward. If x equals y and both values are positive then the device is tilted to the front left and so on. This suggests an approximation algorithm, when x and y are non-vanishing and unequal. For example, if x is greater than y and both are positive, we can conclude that the device points between the left and front left.

As a special case, we divide the device tilts space evenly into eight horizontal regions. The eight directions are at the center of each region, as illustrated in Figure 4.3. Assume that the device tilts to some region with angle of θ and points with an angle of φ, clockwise away from the south axis as illustrated in Figure 4.4. Table 4.1 shows the expected accelerometers signals and reveals the correspondence between the values of φ, x, y, and the regions.

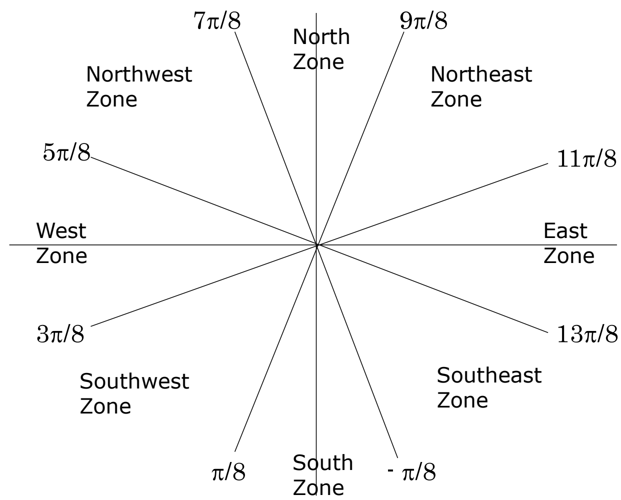

Figure 4.3: Eight regions seen in the horizontal space when divided evenly.

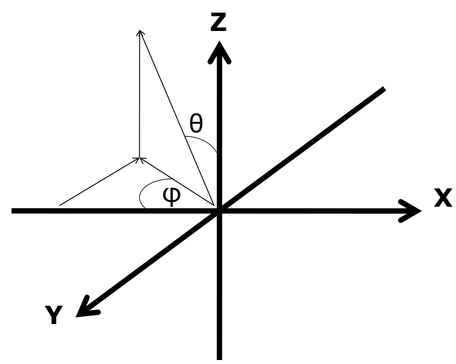

Figure 4.4: Accelerometer embedded device tilting relative to the up orientation. The angle φ is relative to south and goes clockwise.

Table 4.1
The set of values of φ, *x*, *y*, *z* and the corresponding regions

φ	-π/8,π/8	π/8,3π/8	3π/8,5π/8	5π/8,7π/8	7π/8,9π/8	9π/8,11π/8	11π/8,13π/8	13π/8,-π/8	
x	gsinθsinφ	gsinθsinφ	gsinθcosφ	gsinθcosφ	gsinθsinφ	- gsinθsinφ	- gsinθcosφ	- gsinθcosφ	0
y	-gsinθcosφ	- gsinθcosφ	gsinθsinφ	gsinθsinφ	gsinθcosφ	gsinθcosφ	gsinθsinφ	- gsinθsinφ	0
z	gcosθ	gcosθ	gcosθ	gcosθ	gcosθ	gcosθ	gcosθ	gcosθ	g
Region	S	SW	W	NW	N	NE	E	SE	U

Table 4.2

The set of absolute value of the ratio of x over y, their signs and the corresponding regions

φ	-π/8,π/8	π/8,3π/8	3π/8,5π/8	5π/8,7π/8	7π/8,9π/8	9π/8,11π/8	11π/8,13π/8	13π/8,-π/8	
x		+	+	+		-	-	-	0
y	-	-		+	+	+		-	0
Region	S	SW	W	NW	N	NE	E	SE	U

We state that the relative values of *x*, *y* along with their signs define the zone which the device tilts to. The *z* values are the same across horizontal positions for the same tilt angle θ. Therefore the azimuth tilt position is independent of *z*. The ratio of *x* and *y* defines the angle; their signs determine the region. Hence the absolute value of gravity *g* is not a factor because it is eliminated in the ratio of *x* and *y*. The angle φ determines the region. We display the truth values in Table 4.2 to decide the region that the device points to by evaluating the ratio of *x* over *y* and their signs.

In Table 4.2, for the *x* and *y* fields, the '+' or '-' for the x and y fields denote the signs of x and y. A blank space in a field means Don't-care, which could be either \`+', or \`-', or zero.

The truth table requires both *x* and *y* to be zero for the up position, which is not applicable in reality. In order to make the system feasible for the up position, we define a minimum of angle θ' as a valid device tilts. We use the threshold that *x* and *y* greater than $min = g\sin\theta'$ to determine significant tilting away from the vertical. The up position serves as the base position from which all tilts are oriented. This requires us to give special consideration to the size of the up region. If the size of the up region is small then it is easier to tilt to the other positions from the up position. It also reduces the movement tilt amplitude from one tilt position to the next tilt position. The disadvantage of a small up region is that it might be hard to locate, while a large up region would be easy to locate. In the experiment, we will explore the reasonable value of θ' for the up zone.

Table 4.3
Tilt angle ranges, signs of x and y, and the corresponding regions

φ	φ1, φ2	φ3, φ4	φ5, φ6	φ7, φ8	φ9, φ10	φ11, φ12	φ13, φ14	φ15, φ16	
x		+	+	+		-	-	-	0
y	-	-		+	+	+		-	0
Region	S	SW	W	NW	N	NE	E	SE	U

As a conclusion, the following algorithm is used to decide the tilt position using the accelerometer readings.

Algorithm 2 *Determining the tilt positions*

Upon detecting a user input of (x, y):

If y = zero,

then y = min/10.0;

If |x| <= min && |y| <= min,

then sign(x) = 0, sign(y) = 0;

Check the truth table,

Return region(arctangent(x/y), sign(x), sign(y)).

In our implementations, the regions can have different sizes. In that case, only the angular range in the first row of Table 4.1 needs to change, and the values of the first row in Table 4.2 will change accordingly by applying the tangent calculations. In a usability experiment, we will study the varying region sizes. The truth table can be derived from a table like Table 4.3, where φ denotes the tilt angles relative to the south axis.

4.3 Accelerometers' Effects during Clicking

This section inspects the accuracy of the accelerometer tilt measurements during clicking. In Liu *et al.*'s (23) technique the device tilt movement introduced noise and uncertainty during the gesture analysis. In our approach, the device tilt is the valid signal and other movements are noise. The ideal user input would keep the device still and only move during the tilting prior to the click. In reality, this is impossible. Clicking itself can cause movement, and the user might click while the device is not yet still. We investigate these movements and potential source of uncertainties in the accelerometer signal during titling and clicking.

Figure 4.5 shows the reading from 12 seconds of sequential titling operations. The operations were composed of up, north, west, and up without clicking. The sequence of operations was then repeated with three clicks for each position, which is the reason for the long tilt interval in the second 4 orientations. Three axial readings are displayed along with the total which should equal the value of gravity except when the device is accelerated during the transition between tilt orientations and during the click. Comparing the first four tilt movements with the second four with clicking, the clicks caused small surges in the total acceleration but did not drastically change the overall noise level. The large surges occurred during transition between tilt orientations and lasted for a trivial short time. Their effects can be avoided during the analysis.

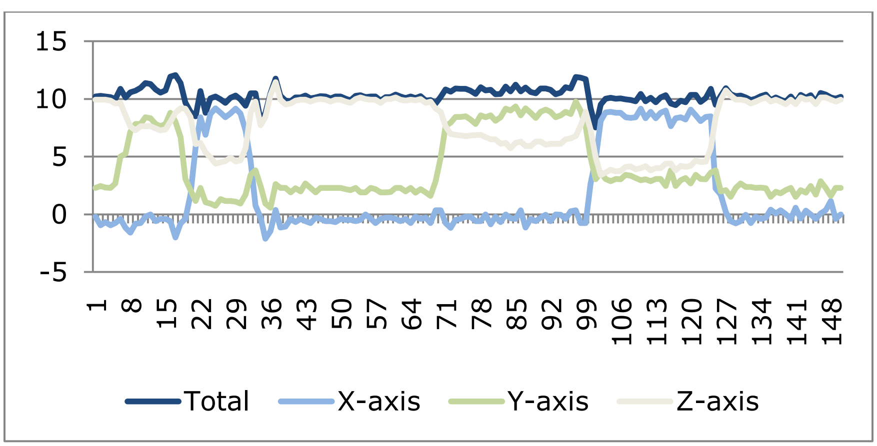

Figure 4.5: Readings of the three axials and the total acceleration during tilting with and without clicking. The operations were composed of up, north, west, and up without clicking, followed by a sequence of repeated operations with three clicks for each position (unit: m/s²)

The YAUIM system identified the 8 operations during the 12 seconds. The parameter *TICK* is set to be zero because of the high precision of the accelerometers in the smartphone. The minimum value for θ is set to 15 degrees. Figure 4.3 also shows that the transition time is much shorter than the stable position time. Consequently, experienced users can input tilt position quickly, and the system ought to be able to keep up with the user input.

Chapter 5

User Test and Usability Study of the 4+1 Positioning Input System

5.1 Introduction

The tilt space of the 4+1 positioning input system is divided into five primary regions: south, west, north, east, and up. This experiment evaluated the 4+1 positioning input system using 3 tasks. Task 1 determined what participants naturally do using the tilt-click operations. The participants were asked to tilt and then click to the 5 primary positions without feedback being provided. Task 2 evaluated clockwise number formations with alphabet sized 5, 7, and 10. The alphabets included numbers 1 to 5, 1 to 7, and 0 to 9 respectively. Task 3 evaluated the calculator application using the 4+1 positioning input system.

5.2 Methods

5.2.1 Participants

The twenty four participants were undergraduate students majoring in computer science, computer engineering, or mathematical science. Their ages ranged from 19 to 24 years, mean of 19.9 years. One participant was female and three were left handed. Twenty one participants reported having experience playing Wii sports or other tilt games on hand-held devices.

5.2.2 Apparatus

Three HTC smartphones with built-in tri-axial accelerometers were used for this experiment. The smartphones ran Windows Mobile 6.1 Professional operating system. They had single processor running at 528MHz, one 2.8 inch LCD touch screen, 192MB of RAM, 4GB of internal storage capacity, and one static tri-axial accelerometer. The weight of the smartphone with battery was 110g. The software was written by the author in the C programming language on a Windows PC and the executables were deployed onto the smartphones. The execution and interactions were solely on the smartphone. The recorded data were transferred and processed on a PC.

5.2.3 Procedures

The participants were asked to sign the consent form in Appendix A before they were allowed to participate in the experiments.

The participants sat in armchairs and held the smartphone with one hand. The smartphone was in upside-down position so that the participants could put their thumb on the round button of the smartphone. They were told to hold the smartphone in a comfortable position either by putting the arm against the chair or in front of the body. The maps of numbers illustrating the corresponding positions for each number were in front of the participants. The participants responded to the phones' audio output of a number by tilting and then clicking the smartphones. The smartphones recorded the accelerometers' values and time when clicked. The systems also recorded the accelerometer values at 20 hertz for analyzing the path of the phone. That analysis is not made for this dissertation. The data was saved in a text file and later transferred to a PC for analysis.

The first three participants composed the pilot study. They completed Task 1 and 2. In Task 1 the participants in the pilot study performed 200 tilts and clicks. In Task 2 one participant performed tilt and clicks with alphabet sized 5, another participant performed tilts with alphabet sized 7, and the third participant with alphabet sized 10. The experiment setup was modified after the pilot study.

The total number of clicks for Task 1 was reduced from 200 to 100, because we suspected the extra tilts and clicks bored the participants. The 100 tilts and clicks were completed within around 3 minutes compared with 7 minutes for 200 tilts and clicks.

Participant 4 through participant 24 performed both Task 1 and Task 2. The alphabet sized 5 was removed from Task 2 because the participants learned the mapping of the 5 numbers immediately. The total number of prompts for both alphabet sized 7 and alphabet sized 10 were also reduced from the pilot study. The total number of prompts for size 7 test was reduced from 360 to 320. The total number of prompts of size 10 test was reduced from 400 to 340. Participant 4 through participant 11 performed Task 2 with alphabet sized 7. Participant 12 through participant 24 performed Task 2 with alphabet sized 10.

Participant 17 through 24 performed Task 3. Task 3 is a prototype of a calculator with 6 functions and symbols: addition, subtraction, multiplication, division, and equal sign. The tilt positions for the functions were illustrated in a second map (operator map). One participant chose to use both the number map and the operator map. All other 7 participants used only the operator map.

5.3 Task 1: Tilt-Click Operations without Feedback

5.3.1 Task Goals

It is important to learn what potential participants can do without practice. Ideally, the tilt ranges should be derived from participants' natural operations using the device. Hence, Task 1 studied the participants' natural performance for the primary regions and the results were used to derive the tilt ranges for both the primary and diagonal orientations in the 8+1 positioning input system.

5.3.2 Task Procedures

The participants were prompted to tilt and click over the 5 primary regions. The participants were first introduced with the five tilt orientations. They were asked to tilt to the smartphone to the prompted orientation after hearing the prompt from the smartphone. They clicked the round button at the upper side of the smartphone after tilting. Every participant responded to 100 prompts and performed 100 tilt-clicks. Successive prompts were made after the participants clicked. The system did not provide feedback to the participants whether the tilt orientations were correct or not. The participants were allowed to ask questions or to make comments before and after the task. They were introduced as follows:

"In this task, you will be prompted audibly by the smartphone to tilt to a position and click. Then you will be prompted to tilt to another position to click. There are five positions you will perform. They are up position, south position, west position, north position, and east position. (A demonstration of the tilt operations was provided.)

"You will tilt the smartphone to the position and click the button. The smart phone will record the position when you click, not the movement about how you tilt. So please just tilt as you feel comfortable and don't wave the smartphone.

"You need to decide the up position. You tilt to a position which is away from up position and to a comfortable angle. Just keep consistent to make yourself feel comfortable and respond in a comfortable speed.

"You will spend about 3 to 10 minutes on this task and there're 100 clicks to operate. After 100 clicks the smartphone will let you know the task is complete. When it's complete, you may exit the program and wait for others to complete.

"Now, what question do you have? If you don't have questions, please start the program."

5.3.3 Results

The 21 participants were divided into two groups based on their error rates: a regular group and an irregular group. The purpose for dividing the participants into two groups was to derive reasonable tilt angle ranges from the regular group for the 8+1 input system.

Table 5.1
Mean error rates of the 5 sessions (unit: percentage)

Sessions	One	Two	Three	Four	Five	Average
Regular group	8.13	5.63	6.25	8.44	6.88	7.07
Irregular group	19.0	29.0	33.0	21.0	27.0	25.8
All 21 users	10.7	11.2	12.6	11.4	11.7	11.5

Error Rates. The error rate was calculated using predefined regions which were set along the diagonal directions 45 degrees to the cardinal directions in the horizontal plane. The up region was defined as a cone with a range of 15 degrees away from the vertical position. The error rates of the 21 participants (participant 4 through participant 24) were (0.01, 0.12, 0.09, 0.02, 0.03, 0.20, 0.28, 0.22, 0.05, 0.35, 0.18, 0.01, 0.0, 0.24, 0.14, 0.11, 0.11, 0.05, 0.07, 0.08, 0.08). Participants 9, 10, 11, 13, and 17 had error rates equal to or larger than 20 percent. Participant 9 made large motion during the task. Participant 10 frequently confused the directions. Participant 11 had small motion and frequently did not perform a tilt. Participants 13 and 17 had no observed abnormal operations. These 5 participants were put in the irregular operation group. The remaining 16 participants were considered to be the regular group which we presumed performed regularly. The mean error rates in percentages divided into 5 sessions (every 20 clicks were a session) were presented in Table 5.1.

Tilt Angle Ranges. The mean tilt angles and their standard deviations were calculated for the five primary regions and reported in Table 5.2. The measurement of angles was relative to south and going clockwise.

The pie charts in Figure 5.1 and Figure 5.2 were for four primary tilt ranges (south, west, north, east) and the diagonal ranges (not utilized in this experiment). They were derived from the regular group and of 90 and 95 percent probabilities region for the tilting orientation after responding to a prompt. The probability here refers to how many percent of tilt operations fell into the specific range. For example, 90 percent of south operations fell into range of the 26 degrees.

Table 5.2

Means of the tilt angles of the primary regions (With standard deviations after the commas, angle unit: degree)

Regions	South	West	North	East	Up
Regular group	1,8	87,13	179,9	278,13	8,4
Irregular group	4,7	84,12	183,14	288,15	7,4
All 21 users	1,8	86,13	180,10	281,14	8,4

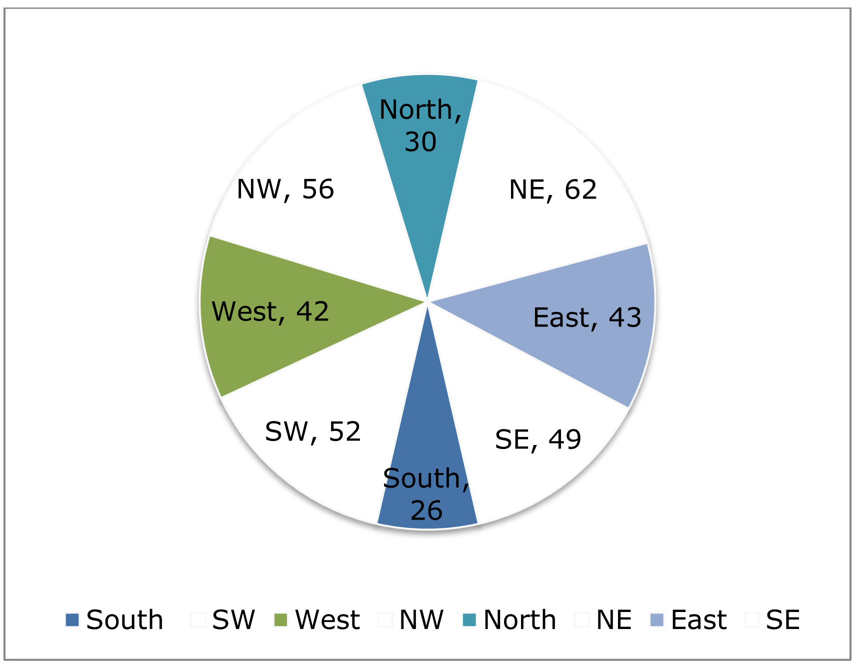

Figure 5.1: The 90 percent pie chart of 4+1 system experiment. The ranges of south, west, north, and east were calculated based on the 90 percent probabilities (unit: degree)

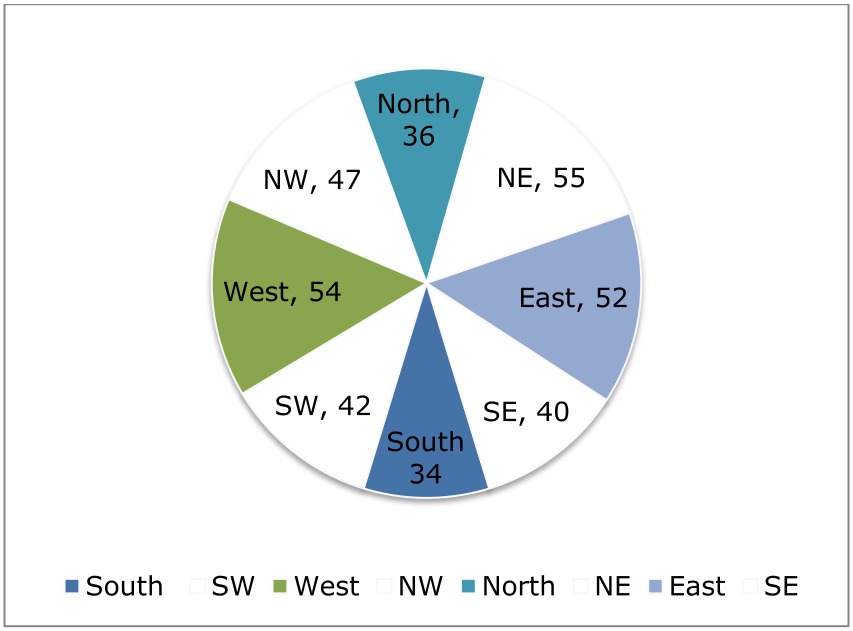

Figure 5.2: The 95 percent pie chart of 4+1 system experiment. The ranges of south, west, north, and east were calculated based on the 95 percent probabilities (unit: degree)

Practice and Performance Time. The response time was determined from the end of the audible prompt to the participant's click. We model practice using the power law of practice (42).

$$RT = aN^{-b} \tag{5.1}$$

Where RT is the response time; N is the click sequence number; a and b are coefficients to be determined; b is called the learning power.

The best fit is determined by linear regression of the logarithm of the response times by the logarithm of the click number on the regular group, the irregular group, and overall. In either category, the intercept and slopes were significant. The results are reported in Table 5.3.

Table 5.3
Results of the linear regression of the logarithm of the response times by the logarithm of the click number

	df	*intercept*	*t*	*p*	*slope*	*t*	*p*	*R*
Regular group	98	0.77	15.16	<2e-16	-0.15	-11.32	<2e-16	0.57
Irregular group	98	0.78	7.46	3.61e-11	-0.13	-4.62	1.17e-05	0.18
All 21 users	98	0.81	15.07	<2e-16	-0.16	-10.95	<2e-16	0.55

The learning power *b* and coefficient *a* in the three groups were (0.15, 2.16), (0.13, 2.18), and (0.16, 2.25) respectively. The learning powers showed the learning in any of the three groups. The model of all 21 participants is plotted in Figure 5.3. The results will be discussed in next section.

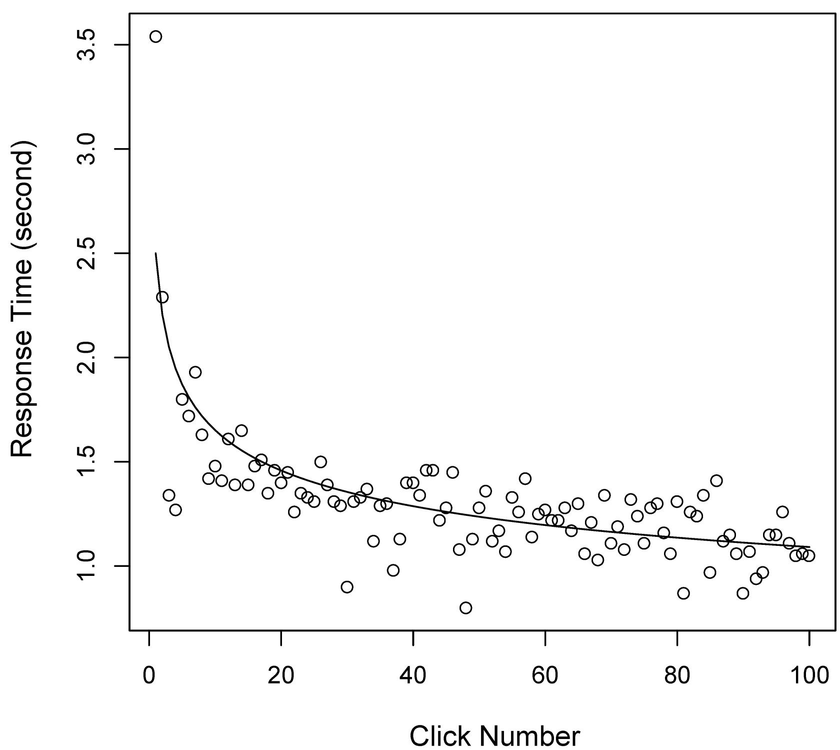

Figure 5.3: The learning curve of the 4+1 system operations of all the users.

5.3.4 Discussion

The participants ideally should be from various demographics and ages of the potential participants, which was not feasible in this study. We could only recruit college students for this study. During the analysis, some participants showed irregular operations which might have been avoided by careful explanations of the tilt mechanism. It is necessary and important to explain the tilt mechanism before the experiment so that the participants can avoid waving movements or other irregular operations.

Without feedback, participants didn't know if their click was in the correct region. The linear regression analysis of the error-rate over click number for the 100 clicks in either of the two groups ($F(1,98) < 1.0$, $p > .05$) showed that error rates were not significant over click numbers. However, the low overall error rate (11.5%) showed that the participants could adapt the tilt-click operations naturally without much training.

The pie charts in Figure 5.1 and Figure 5.2 left room for the diagonal regions that will be used in the 8+1 region experiments. Suspecting that the diagonal regions will be harder to tilt to, we propose to design the 8+1 region ranges based on the 90 percent pie chart. The up operations had mean tilt angle of 7.55 degrees, with standard deviation of 4.0 degrees. Hence the setup of up region of 15 degrees from vertical is reasonable. Table 5.4 lists the proposed region for the 8+1 experiments. Although we used tilt angles of the regular group for the calculations, based on Table 5.2, the paired-*t* tests on both tilt angles ($df = 4$, $p = 0.31$) and standard deviations ($df = 4$, $p = 0.43$) showed the differences between the two groups were in fact not statistically significant. Hence the derivation of the tilt angle ranges based on the regular group will be applicable to all potential users.

The pie charts also showed that the east and west have larger pointing angle ranges than north and south in the regular group. By consulting Table 5.2, the standard deviations of the regular group for east and west were both 13 degrees, while north and south had 9

Table 5.4
Proposed region setups of the 8+1 positioning input system (unit: degree)

South	SW	West	NW	North	NE	East	SE
-12,14	14,66	66,108	108,164	164,194	194,256	256,299	299,348(-12)

degrees and 8 degrees respectively. This confirmed the observation from the pie charts. For the irregular group, north and east had a slight higher standard deviations. In either group, the south had the smallest standard deviations. We could not identify an apparent reason for this observation.

Table 5.3 and Figure 5.3 show that participants got more efficient with the tilt-clicks. The figure also revealed that during the first 20 clicks, the participants learned the most of the operations.

The 4+1 positioning system is feasible for most participants. Within a few minutes of practice the participants reduced the time to perform the operations. The tilt angle ranges derived in this study will serve as the guideline for the design of 8+1 positioning system.

5.4 Task 2: Number Input Operations over the 4+1 System

5.4.1 Task Goals

Task 2 evaluated the clockwise number composition input method for the 4+1 system. Every number was formed by two clicks on the five primary regions as illustrated in the number map (see Figure 3.2). The learnability of the mapping method and feasibility of the tilt-click operations were evaluated by studying the error rates and performance times of the operations.

5.4.2 Task Procedures

The participants were seated in comfortable chairs facing the map for the numbers. The clockwise formation method was explained and the number input was demonstrated. The participants were prompted to input single digit numbers using the map. Then they were asked to operate without using the map. They alternately performed 4 sessions with the map and 4 sessions without the map. The single digit numbers were randomly prompted. There were two alphabets of numbers: alphabet sized 7 included numbers 1 to 7 and alphabet sized 10 included numbers 0 to 9. The participants were randomly assigned to perform one or the other alphabet. The task took less than 30 minutes to complete.

Like in Task 1, the system did not provide feedback to the participants whether the input was correct or not. The next number was prompted audibly after the participants completed an input. The following instructions were read to the participants:

> "In this task, you are prompted to input some numbers randomly. Every number is formed by two consecutive tilt-clicks. The formation rule is demonstrated in the number map in front of you. You tilt and click at one position then tilt and click at the other position to input a number.
>
> "You will use the map first. After some while, the smartphone will tell you to turn over the map and you will do your best without the map. Then it will tell you to use the map again. This will be repeated 4 times altogether. This task takes about 30 minutes."

5.4.3 Results

Eight participants performed Task 2 with alphabet sized 7. Thirteen participants performed the task with alphabet sized 10. We study the data in four categories: alphabet sized 10 with map, alphabet sized 10 without map, alphabet sized 7 with map, and alphabet sized 7 without map. All the 21 participants had finished Task 1 before performing this task.

Error Rates of the Number Inputs. The mean error rate of a session was calculated by the incorrect number of inputs over the total number of prompts in the session. The mean error rates of the sessions in the 4 categories are presented in Table 5.5 and plotted in Figure 5.4.

Table 5.5
Mean error rates of the number inputs of the 4 categories

Sessions	One	Two	Three	Four	Average
Alphabet sized 10 with map	0.10	0.06	0.07	0.07	0.08
Alphabet sized 10 without map	0.32	0.26	0.20	0.18	0.24
Alphabet sized 7 with map	0.04	0.07	0.09	0.07	0.07
Alphabet sized 7 without map	0.23	0.16	0.16	0.08	0.16

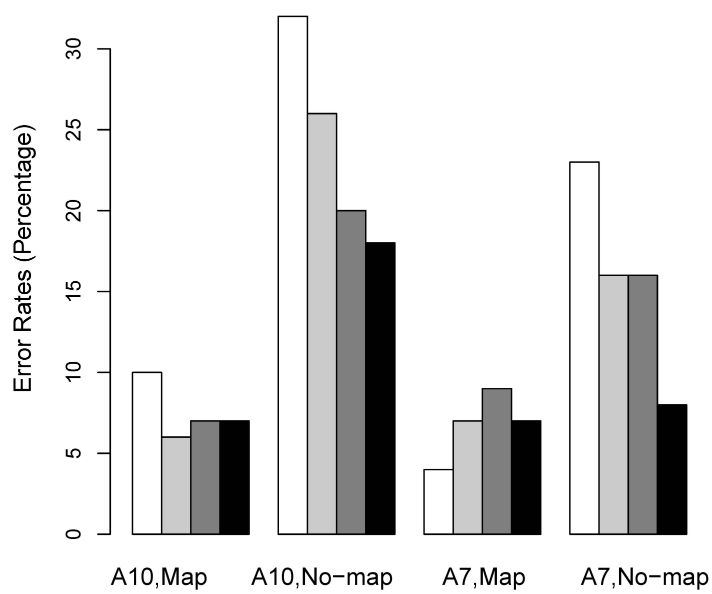

Figure 5.4: Mean number input error rates for 4 categories. A10 is for alphabet sized 10. A7 is for alphabet sized 7. Map denotes operations by checking the map. No-map denotes operations without checking the map. Sessions 1 to 4 are illustrated from white to black.

Table 5.6
Mean performance times in seconds of the two clicks in the 4 categories

Sessions	One	Two	Three	Four	Average
Alphabet sized 10 with map first click	1.67	1.25	1.12	1.01	1.26
Alphabet sized 10 with map second click	0.55	0.41	0.38	0.34	0.42
Alphabet sized 10 without map first click	1.74	1.44	1.39	1.13	1.42
Alphabet sized 10 without map second click	0.53	0.41	0.37	0.39	0.43
Alphabet sized 7 with map first click	1.21	0.96	0.78	0.96	0.98
Alphabet sized 7 with map second click	0.41	0.34	0.25	0.28	0.32
Alphabet sized 7 without map first click	1.44	1.16	1.18	1.12	1.23
Alphabet sized 7 without map second click	0.47	0.33	0.27	0.32	0.35

Practice and Performance Times of the Number Inputs. We examined the two clicks for a number individually. The response time *RT* was the period of time for one click. The response time for the first click started from the end of the audible prompt. The response time for the second click was the time between the first click and the second click.

The mean performance times of the sessions are presented in Table 5.6 and plotted in Figure 5.5 and Figure 5.6.

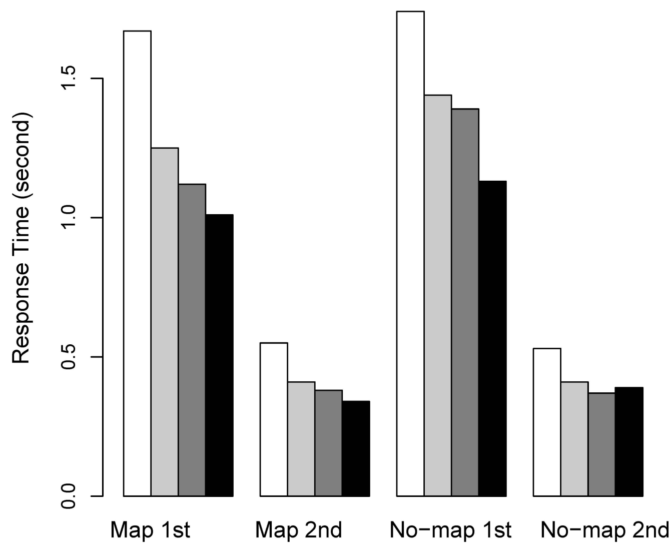

Figure 5.5: Mean performance times of the clicks for the alphabet sized 10. Map denotes operations by checking the map. No-map denotes operations without checking the map. 1st denotes the first click operations. 2nd denotes the second click operations. Sessions 1 to 4 are illustrated from white to black.

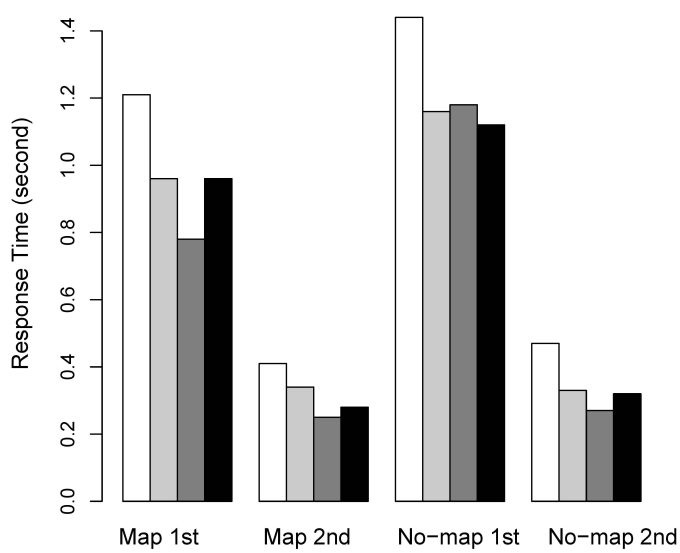

Figure 5.6: Mean performance times of the clicks for the alphabet sized 7. Map denotes operations by checking the map. No-map denotes operations without checking the map. 1st denotes the first click operations. 2nd denotes the second click operations. Sessions 1 to 4 are illustrated from white to black.

5.4.4 Discussion

The alphabet sized 7 involved 3 tilting positions, while the alphabet sized 10 involved 4 tilting positions (see Figure 3.2). By looking at the error rates in Table 5.5, the participants performing alphabet sized 7 showed lower mean error rates without the map (16 percent vs. 24 percent). The difference was statistically significant ($t = 5.74$, $df = 3$, $p = 0.010$). However, it was expected that the participants would perform similarly using the map. This was confirmed by their similar error rates with the map (8 percent vs. 7 percent). The difference was not statistically significant ($t = 0.42$, $df = 3$, $p = 0.705$).

Performing with the map, the participants knew the composition of the numbers, hence we assume that the errors performing with the map were mainly from errors due to improper tilt positions. The errors performing without the map were from both the incorrect positions and incorrect mapping of the numbers. Since no feedback was provided to the participants, they were unaware of the correctness of their tilt and number input. They could not improve on their correctness of the tilt operations without feedback. Hence any improvement of error rates would reflect their learning of the map. Figure 5.4 showed that the participants had improved their error rates in the four sessions performing without the map, but no improvement were presented for the sessions with the map.

The response time for the first click is mainly composed of the thinking time (or the time to check the map when using the map) and the operation time. The response time for the second click was mainly the operation time. Table 5.6 showed the mean response times of the second clicks were lower than those of the first clicks in the corresponding categories. The mean response time for the alphabet sized 7 were lower than those of the alphabet sized 10.

Table 5.7
Results of the linear regression of the logarithm of the response times by the logarithm of the session number with alphabet sized 10

	df	*intercept*	*t*	*p*	*slope*	*t*	*p*	*R*
1st click with map	2	0.50	23.75	0.002	-0.36	-16.20	0.004	0.99
2nd click with map	2	-0.62	-19.22	0.003	-0.34	-10.06	0.010	0.98
1st click without map	2	0.57	9.62	0.010	-0.28	-4.54	0.045	0.91
2nd click without map	2	-0.67	-9.6	0.010	-0.24	-3.34	0.079	0.84

Figure 5.5 and Figure 5.6 indicated that the response times of both the first and second click improved with the practice. The improvement in the first click was mainly due to the participants becoming more familiar with the map. The improvement in the second click was mainly due to their becoming more familiar with the tilt-click operations. Figure 5.5 suggests a smooth curve for the improvement for alphabet sized 10. We model practice for alphabet sized 10 using the power law in Equation 5.1. The best fit is determined by linear regression of the logarithm of the response times by the logarithm of the session number. The results are reported in Table 5.7.

The correlation between response time and the session number of the first click was 0.99 with the map. It was 0.98 for the second click with the map. The learning powers were 0.36 and 0.34 with the map, while without the map they were 0.28 and 0.24 respectively.

This task showed that the clockwise number formation method for the 4+1 system was learned by the participants during the experiment. The participants improved their performance on the number formation and the tilt-click operations during the sessions. Both the performance times and error rates improved in the 30 minutes of practice.

5.5 Task 3: 4+1 System Calculator Application

5.5.1 Task Goals

The purpose of Task 3 was to evaluate the calculator application of the 4+1 positioning input system.

5.5.2 Task Procedures

A calculator prototype was implemented on the 4+1 input system and the participants were asked to solve 8 math questions with this calculator. The calculator prototype used integer numbers and 6 operator symbols: addition, subtraction, multiplication, division, equal sign, and backspace. The operators were formed by the clockwise composition rule as well. They are illustrated in Figure 5.7. For example, south plus up is addition. West plus up is subtraction, etc.

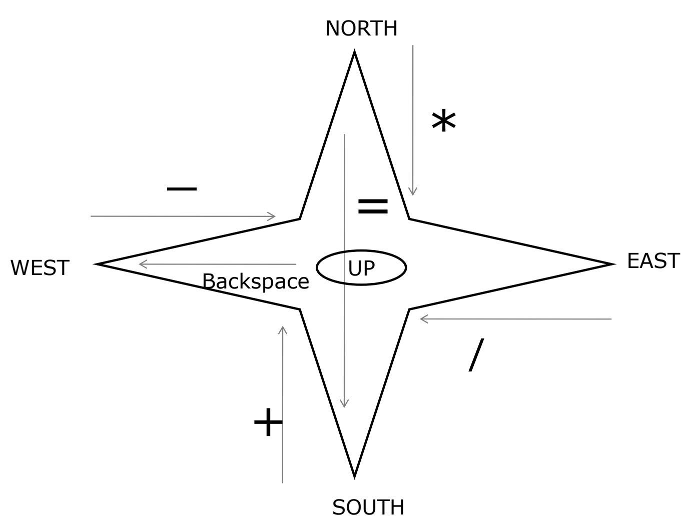

Figure 5.7: Map of the 4+1 system calculator operators. Every two positions represent one operator.

The 8 math questions were:

1) 10 + 562 =
2) 831 - 49 =
3) 687 * 56 =
4) 328 / 74 =
5) 41 + 76 =
6) 509 - 28 =
7) 78 * 243 =
8) 7659 / 37 =

5.5.3 Results

Eight participants, who completed Task 2 with alphabet sized 10, performed this task. After they completed each calculation they wrote down the answer they heard from the phone. During the experiment, the number map and the operators map were available to the participants. However, only 1 participant chose to use the number map. Whenever there was an input error, the participants used the backspace operation to correct the error.

For each question, the completion time was calculated from the first click of the first digit to the second click of the equal symbol. The overall completion time for each participant was the total time for all 8 questions. Tables 5.8 and 5.9 displayed their overall performance time and the performance time on each question.

Table 5.8
Individual participants' time to complete Task 3 (unit: second)

Participant	1	2	3	4	5	6	7	8
Completion time	259	215	318	220	185	227	315	266
Backspaces	5	0	8	0	0	3	5	7

Table 5.9
Eight individual participants' completion time on individual questions (unit: second)

Question	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Participant1	79	26	14	32	34	20	38	16
Participant2	50	30	30	22	20	22	27	14
Participant3	42	71	56	26	21	50	36	16
Participant4	39	31	26	26	21	26	25	26
Participant5	30	30	26	23	17	23	21	15
Participant6	36	33	28	21	43	22	26	18
Participant7	75	55	37	54	19	22	38	15
Participant8	28	28	26	26	24	21	23	90
Average	47.4	38.0	30.4	28.8	24.9	25.8	29.3	26.3

The response time *RT* is the completion time for each question and the trial number *N* is the question number. Assuming the power law equation, the log-log analysis on the average performance times showed ($F(1,6) = 34.06$, $p < .001$, $R = 0.85$). Applying the non-linear least squares analysis using the original power law model for initial estimate, we get

$$RT = 25.94N^{-0.92} + 21.93 \tag{5.2}$$

The plot is presented in Figure 5.8. The decreasing curve indicates that the response time gets improved over the practice.

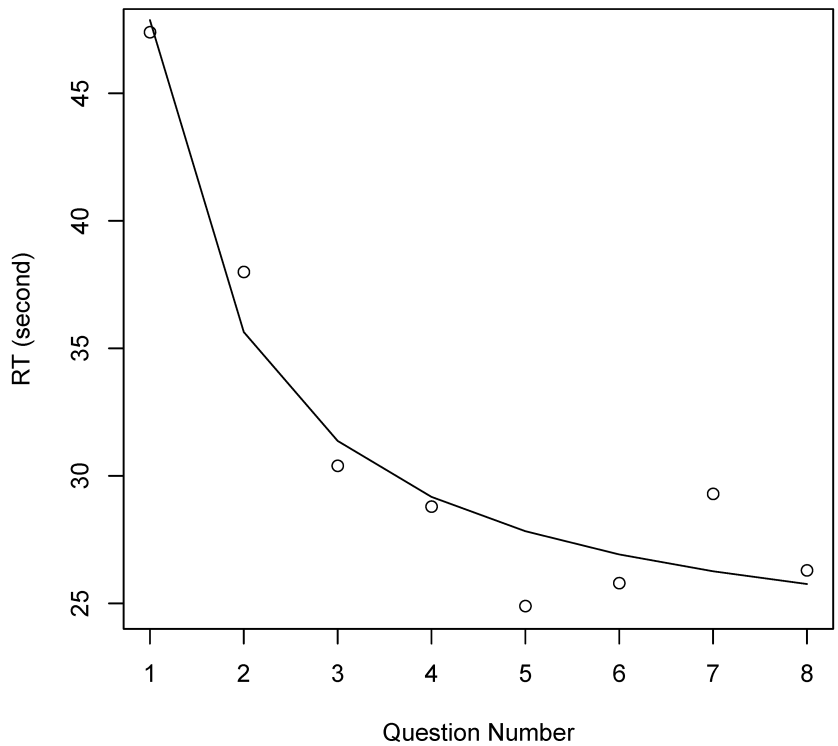

Figure 5.8: The learning curve of the calculations including error correction times

5.5.4 Discussion

Equation 5.2 was based on the completion times with backspace operations included. The equation suggests that eventually the best performance is 21.93 seconds per question including backspace operations. If we do not count the backspace operations (8 values affected within the 64 completion times), the best performance time for each question is 6 seconds by the non-linear least squares analysis. The plot is presented in Figure 5.9. On the average, each question involved 7 characters (numbers and operators). Hence the best performance time for each character is 0.86 seconds.

This task showed that the tilt-click interaction technique and the double click clockwise composition method could be learned in one hour for the calculator application on the 4+1 input system.

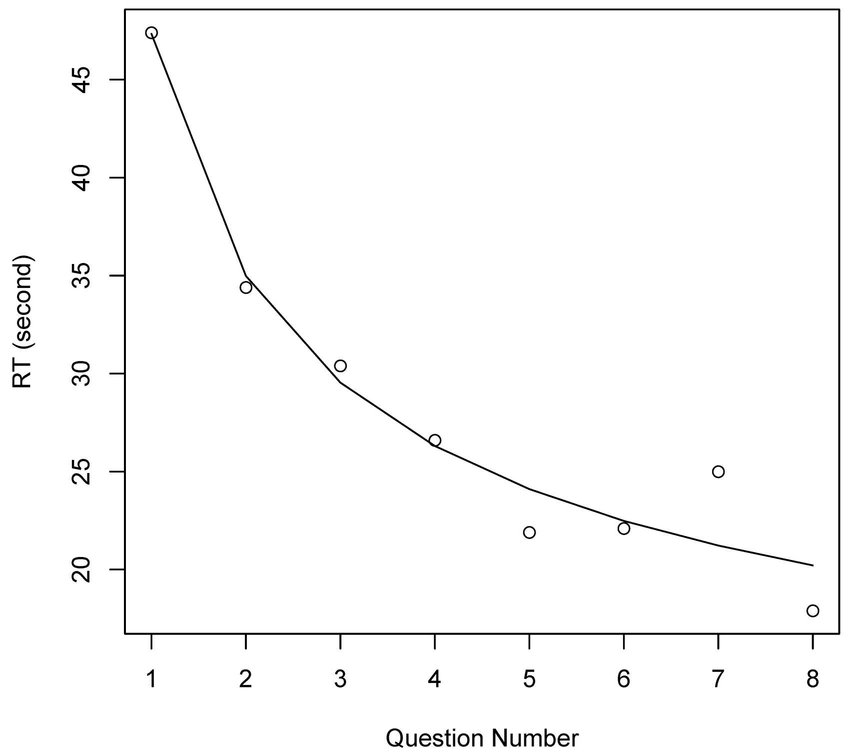

Figure 5.9: The learning curve of the calculations without including the error corrections.

Chapter 6

User Test and Usability Study of the 8+1 Positioning Input System

6.1 Introduction

The 8+1 positioning input system divided the tilt space into eight horizontal regions and one up region. The English letters, numbers, and special operators were mapped onto the tilt space using the clockwise composition rule. Every two positions represented one character. The experiment included two tasks. Task 1 first studied the participants' natural operations on the nine regions. It then examined how the participants adapted to the predefined tilt angle ranges of the 8+1 input system. Task 2 evaluated the text input prototype and the calculator prototype on the 8+1 system. Detailed explanations are provided in the subsections.

6.2 Methods

6.2.1 Participants

Thirteen participants were recruited. They were undergraduate students majoring in computer science, computer engineering, or mathematical science. Their ages ranged from 18 to 37 years, mean of 22.2 years. Four participants were female and two were left handed. Eleven participants reported having experience playing Wii sports or other tilt games on hand-held devices.

6.2.2 Procedures

The 8+1 positioning input system was implemented on the HTC smartphones running the Windows Mobile 6.1 operating system. The predefined tilt angle ranges were set up based on the experiment of the 4+1 positioning input system in Table 5.4. The smartphones recorded the accelerometer values and time when clicked. The systems also recorded the accelerometer values at 20 Hertz for analyzing the path of the phone. That analysis is not made for this dissertation. The data was saved in a text file and later transferred to a PC for analysis.

The participants were asked to sign the consent form in Appendix A before they were allowed to participate in the experiments.

The participants were seated in comfortable chairs. The maps for letters, numbers, and special operators were presented in front of them. The operations were explained and demonstrated before the participants started any task. The whole process took less than one hour.

6.3 Task 1: User Study of the 8+1 System

6.3.1 Task Goals

We wanted to know how the participants defined their natural tilt angle ranges and how they adapted to the predefined tilt angle ranges of the 8+1 positioning input system.

6.3.2 Task Procedures

First, the participants were audibly prompted 100 times to tilt the smartphone to one of the nine regions and click. The prompted tilt orientations were random. No feedback of the orientation was given to the participants. The following instructions were read to the participants:

> "In this task, you will be prompted audibly by the smartphone to tilt to a position and click. Then you will be prompted to tilt to another position to click. There are nine positions you will perform. They are up position, south position, southwest position, west position, northwest position, north position, northeast position, east position, and southeast position. (A demonstration of the tilt operations was provided.)
>
> "You will tilt the smartphone to the position and click the button. The smartphone will record the position when you click, not the movement about how you tilt. So please just tilt as you feel comfortable and don't wave the smartphone.

> "You need to decide the up position first. You tilt to a position which is away from up position and to a comfortable angle. Just keep consistent to make yourself feel comfortable and respond in a comfortable speed.
>
> "You will spend about 3 to 10 minutes on this task and there are 100 clicks to operate. After 100 clicks the smartphone will let you know the task is complete. When it is complete, you may exit the program and wait for others to complete."

Next, the participants explored the 8+1 predefined tilt system. They tilted and clicked in a region and the smartphone audibly responded with the tilt region that the device was orientated in. The participants practiced until they felt comfortable with the input technique. This criterion was determined and self-reported by the participant. We tested the participants' preparation by asking them to input a single character. If they past that test, they continued the experiment. They were told as below:

> "Now, you are provided a system with tilt angle range predefined. You will try to tilt and click to locate all the 9 positions. Once you tilt to a position and click, the smartphone will tell you what position you have tilted to. After you practice for a few minutes and feel confident that you can locate a position that you want, you will be tested if you could tilt to the correct position. If you pass the test, you may go ahead to perform the next procedure."

Finally, the participants performed the feedback operations. They were randomly prompted 100 times by the smartphone to tilt and click in one of the nine regions and feedback was given. If the participants tilted in the correct region, the system would prompt to the next random region. If the participants clicked in an incorrect region, the system would repeat the prompt until the participant tilted in the correct region.

6.3.3 Results

Participants' Self-definition Operations. Six participants performed the tilt and click operations without feedback. The participants' average tilt angles and the standard deviations for the 9 orientations are presented in Table 6.1. Their mean response times over click number are plotted in Figure 6.1. We model practice using the power law in Equation 5.1 by applying linear regression to the logarithm of the response times by the logarithm of click number. The slope in the linear regression or power in the learning model is -0.32 ($F(1,98) = 110.5$, $p < .0001$), and R is 0.53.

Table 6.1
Average tilt angles and standard deviations for the nine orientations in self-definition operations (unit: degree)

Orientation	Mean Tilt Angle	Standard Deviation
UP	10	5
S	12	50
SW	50	31
W	99	49
NW	129	37
N	192	36
NE	230	20
E	263	48
SE	311	54

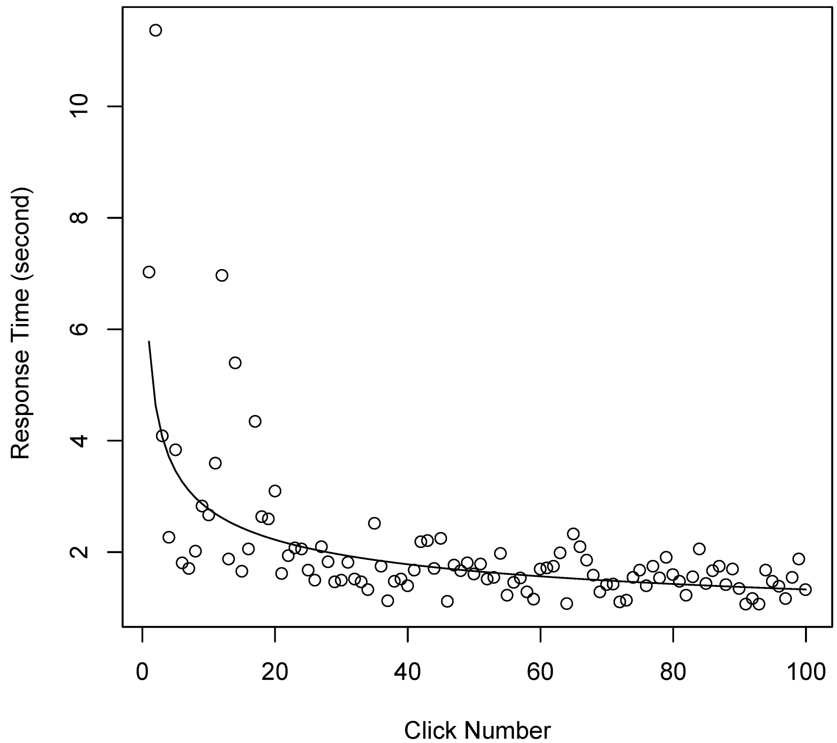

Figure 6.1: Response times over the 100 clicks for the self-definition operations

Participants' Self-explorations and Feedback Operations. Thirteen participants performed the self-exploration for about 5 minutes followed by the 100 prompts with feedback. The recorded data for the feedback operations were used to study the difference between the longitudinal and diagonal orientations. The longitudinal orientations included the south, west, north, and east orientations. The diagonal orientations included the southwest, northwest, northeast, and southeast orientations.

The ratio of the number of correct operations over the number of total operations is the accuracy rate. The accuracy rates for the longitudinal and diagonal orientations for the individual participants are presented in Table 6.2 and plotted in Figure 6.2. The difference between the accuracy rates of the longitudinal and those of the diagonal in Table 6.2 was not statistically significant from paired *t*-test ($t = 1.55$, $df = 12$, $p = 0.15$).

Table 6.2
Accuracy rate of the longitudinal and diagonal orientations for the 13 participants

Participant	1	2	3	4	5	6	7	8	9	10	11	12	13
Longitudinal	0.90	0.88	0.85	0.88	0.77	0.90	0.90	0.92	0.86	0.94	0.77	0.96	0.70
Diagonal	0.96	0.70	0.80	0.63	0.85	0.75	1.0	0.85	0.90	0.85	0.75	0.65	0.79

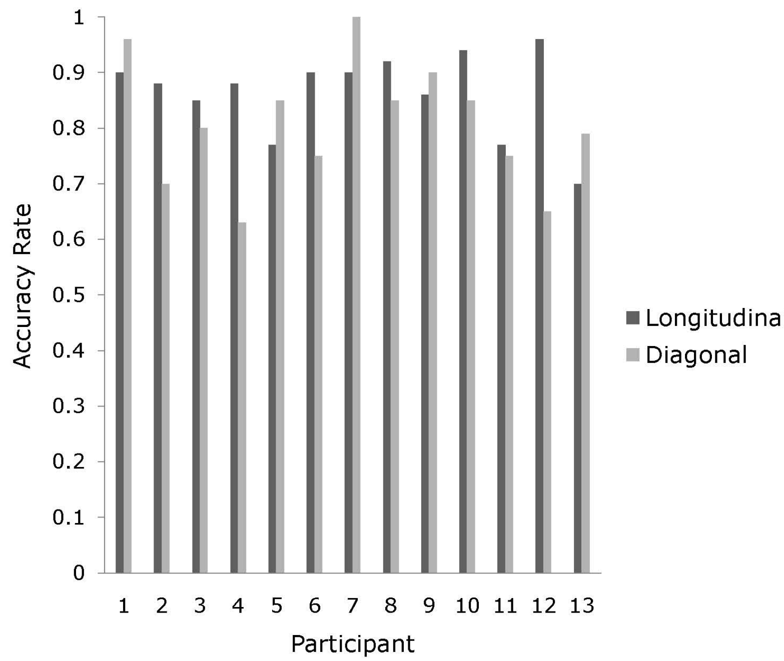

Figure 6.2: Individual participants' accuracy rates on longitudinal and diagonal orientations. The longitudinal orientations included the south, west, north, and east orientations. The diagonal orientations included the southwest, northwest, northeast, and southeast orientations.

The average response times of the operations on the longitudinal and the diagonal orientations of the 13 participants are presented in Table 6.3 and plotted in Figure 6.3. The difference between the response times of longitudinal and those of the diagonal was statistically significant (t = -6.47, df = 12, p < 0.001). The mean values for the longitudinal and those of the diagonal were 1.29 seconds and 1.67 seconds respectively.

Table 6.3
Average response times of the longitudinal and the diagonal orientations for the 13 participants (unit: second)

Participant	1	2	3	4	5	6	7	8	9	10	11	12	13
Longitudinal	1.20	0.96	1.02	0.90	1.42	1.23	1.74	1.01	1.07	1.43	1.95	1.06	1.72
Diagonal	1.43	1.37	1.18	1.20	1.55	1.64	2.28	1.27	1.25	1.74	2.47	1.75	2.55

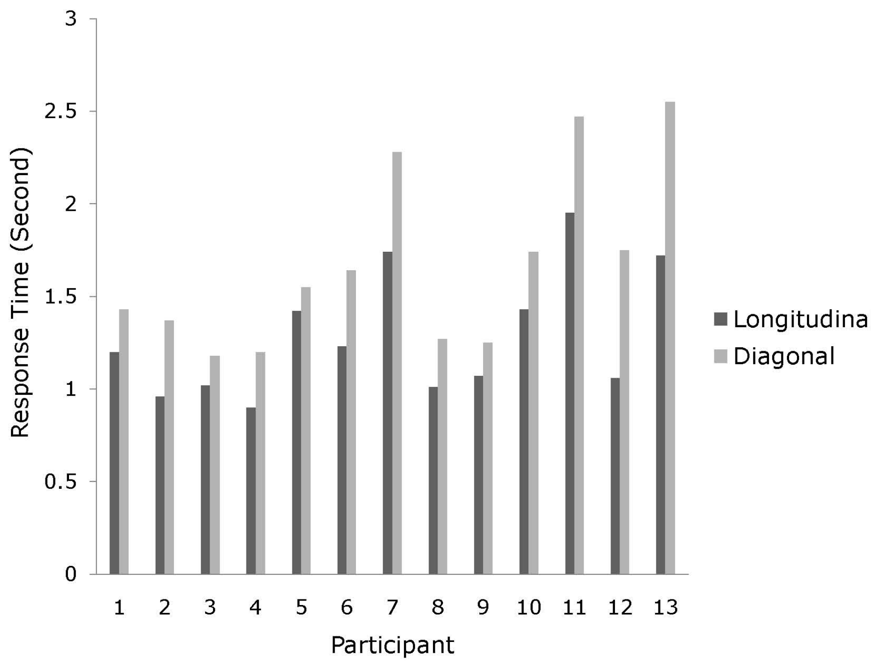

Figure 6.3: Individual response times on the longitudinal and diagonal orientations. The longitudinal orientations included the south, west, north, and east orientations. The diagonal orientations included the southwest, northwest, northeast, and southeast orientations.

The log-log regression analysis showed the response times were not statistically significant over the click number ($F(1,98) = 1.16$, $p = 0.28$).

The average tilt angles and the standard deviations for the 9 orientations are presented in Table 6.4.

Table 6.4
Average tilt angles and standard deviations for the nine orientations in feedback operations (unit: degree)

Orientation	Mean Tilt Angle	Standard Deviation
UP	7	4
S	2	6
SW	39	14
W	87	10
NW	132	13
N	182	6
NE	230	14
E	276	11
SE	323	10

6.3.4 Discussion

Figure 6.1 showed response times improved with practice during the operations without feedback. It also suggests that most of the learning happened during the first 20 clicks.

When the participants were provided with feedback, there was no statistically significant difference (p = 0.15) between the longitudinal and the diagonal accuracy rates. The participants achieved the average accuracy rate of 0.84 (sd = 0.09). The difference between the mean response times of the longitudinal and the diagonal suggests that on average the longitudinal operations were 23% quicker than the diagonal operations. Overall, the mean response time was 1.48 seconds.

Table 6.1 shows that the mean tilt angles were in the predefined regions (see Table 5.4) during the operations without feedback. However, the tilt angles, except for the up orientation, showed large deviations. We plot the tilt angle ranges of the 90% probabilities of tilt operations to compare with the predefined tilt angle ranges in Figure 6.4. The probability here refers to how many percent of tilt operations fell into the specific range.

The large deviations suggest that learning the predefined tilt angle ranges is necessary for the 8+1 positioning input system. Comparison of the mean tilt angles and standard deviations in Table 6.1 and Table 6.4 showed the learning effects on the tilt angles. The mean tilt angles were not statistically different (t = 0.62, p = 0.56), but the decrease in standard deviation was statistically significant (t = 5.03, p = 0.001) after the self-exploration operations. The improvement is confirmed in Figure 6.5.

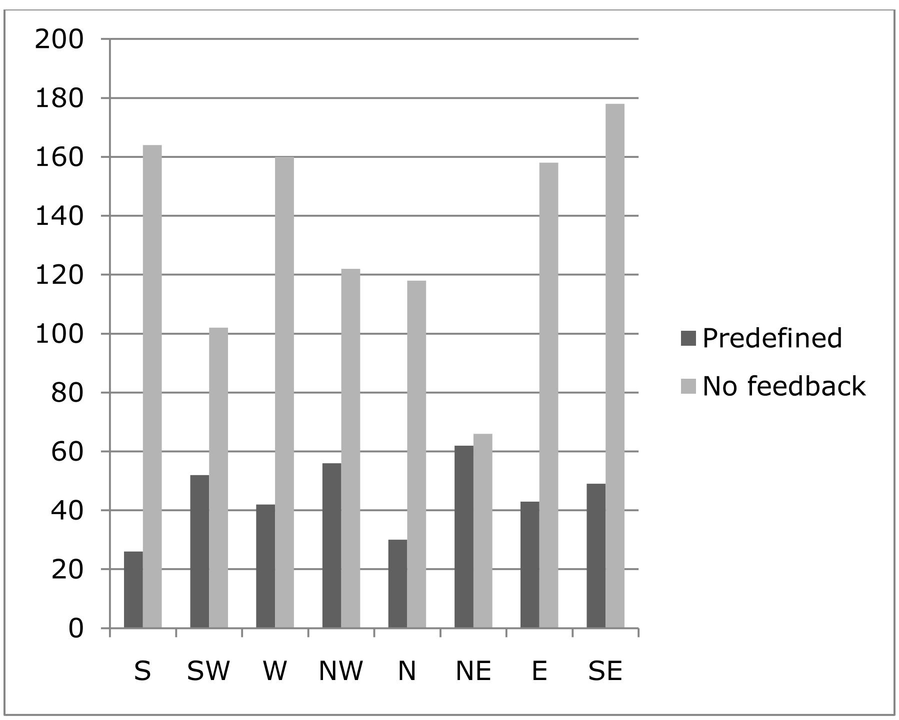

Figure 6.4: Comparison of the tilt angle ranges between the 90% probabilities for the operations without feedback and the predefined tilt angle ranges (unit: degree)

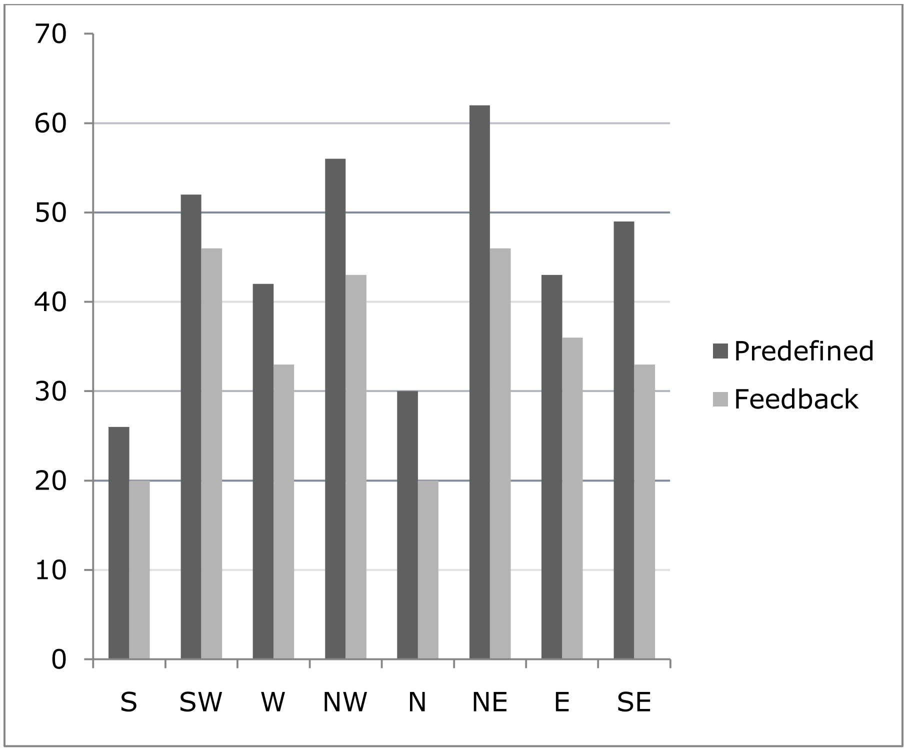

Figure 6.5: Comparison of the tilt angle ranges between the 90% probabilities for the operations with feedback and the predefined tilt angle ranges (unit: degree)

This task showed that the participants could operate the 8+1 system with reasonable accuracy during their first hour of use. It also showed that the operations on the longitudinal regions were more efficient than those of the diagonal regions.

6.4 Task 2: Usability Study of the 8+1 System Applications

6.4.1 Task Goals

Task 2 studied the feasibility of the 8+1 system as a text input for applications. It also examined the performance of the calculator prototype on the 8+1 system.

6.4.2 Procedures

The participants who performed Task 1 were randomly selected to perform either the text input or the calculator application. For the text input, the participants were provided the map for the 26 English letters (see Figure 3.4). The text input method was demonstrated first. Then using the map, they practiced text input by inputting the 26 alphabet letters twice in the alphabet order. Next the participants were asked to input the alphabet without using the map. Finally, while using the map, the participants were asked to input the following phrases: "how are you", "on the web", "dont know", "thank you", "meet here", "be careful", "email jeff", "try later". The participants input a space between phrases. The phrases are from Unigesture (21).

Another group of participants performed the calculator application. They were given a map (Figure 6.6) for the 10 numeric digits and the 6 operators: plus, minus, times, divides, equals, and backspace. The map was created based on the double click clockwise composition method. Using the map they would perform the following 8 calculations:

1) 10 + 562 =
2) 831 - 49 =
3) 687 * 56 =
4) 328 / 74 =
5) 41 + 76 =
6) 509 - 28 =
7) 78 * 243 =
8) 7659 / 37 =

During the process for both text input and calculator application the participants used the backspace operation to correct any input error.

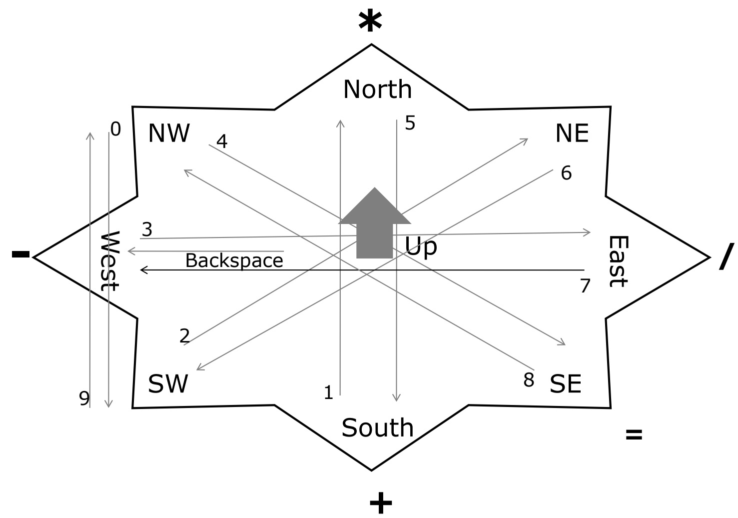

Figure 6.6: Clockwise mapping of numbers and operators for 8+1 system. Every two positions represent one number or one operator.

6.4.3 Results

Five participants performed the text input task. The times to complete the 18 words for the 5 participants were 422, 430, 284, 277, and 385 seconds. The mean value was 360 (SD=74) seconds. The number of backspaces performed were 6, 19, 6, 5, and 8 respectively for each participant, with mean value 9 (SD=6). On average, the 5 participants achieved an input speed of 2.8 WPM (assuming 5 characters per word by following the conventional calculations) within one hour's practice. The overall error rate for all participants was 8.7 percent. The average completion time for the individual characters were not statistically significant over the character sequential number ($F(1,82) = 0.05$, $p = 0.83$).

Five other participants performed the calculator application. The individual completion times for the 8 questions are presented in Table 6.5. We model practice using the power law in Equation 5.1 by applying linear regression to the logarithm of the average completion times by the logarithm of question number. The slope in the linear regression or the power in the learning model is -0.38 ($F(1,6) = 6.16$, $p = 0.048$), and correlation $R = 0.51$. The average completion times for the 8 questions are plotted in Figure 6.7.

Table 6.5
Individual completion times in seconds for the 8 questions (unit: second)

Question	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Participant1	28	65	20	14	11	44	13	21
Participant2	59	34	27	12	31	13	19	31
Participant3	125	58	23	73	16	71	37	28
Participant4	44	134	69	30	22	31	49	40
Participant5	49	87	37	34	39	54	37	55
Average	61	75.6	35.2	32.6	23.8	42.6	31	35
SD	37.5	37.7	20	25	11.3	22.1	14.7	13.1

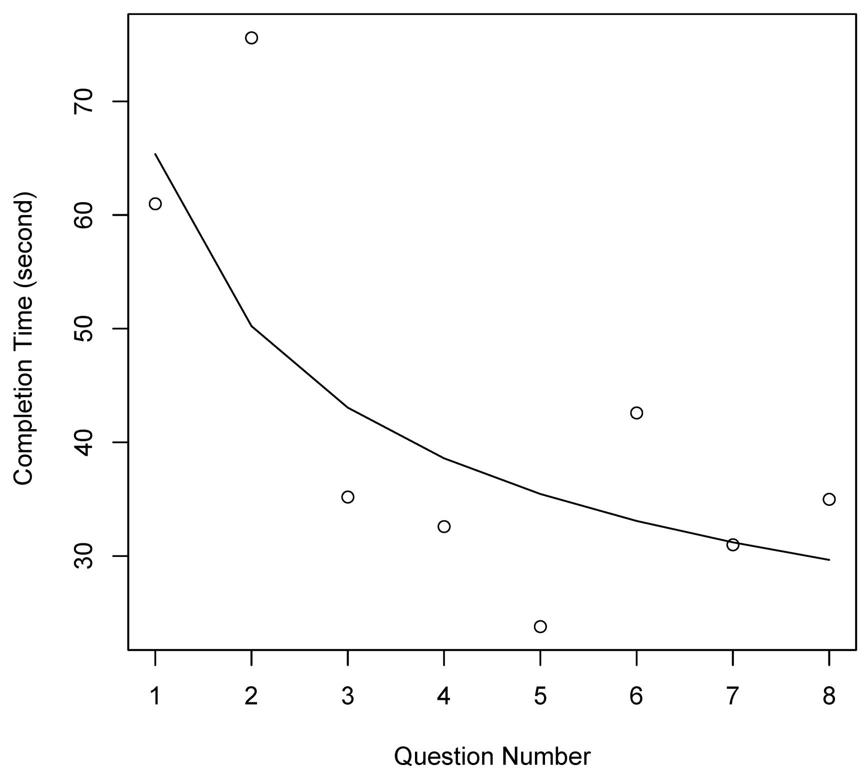

Figure 6.7: Average completion times of the 8 questions

6.4.4 Discussion

The participants practiced only three times the English alphabet before inputting the 18 words. They all completed the input task using the map. They did not show improvement in completion times while inputting the 18 words. On the average, they achieved an input speed of 2.8 WPM with an error rate of 8.7%. The best performance of the 5 participants was 3.7 WPM with an error rate of 5.9%.

Figure 6.7 suggests that there was learning during the calculation tasks. However, after the first two questions, the operation time for each question became stable. The log-log regression analysis showed that the operation time was not statistically significant on question number (p = 0.96). There was no improvement in performance over time after the first two questions. The mean value of the last 6 questions was 33.4 (SD = 6.1) seconds.

The task for the text input and the calculator applications showed that the tilt-click interaction technique and the double click clockwise composition method could be learned in one hour for applications on the 8+1 input system.

Chapter 7

Conclusions

7.1 Contribution

This dissertation studied an accelerometer assisted single key positioning input system. It presented a user input technique which can be operated single handed and eyes-free. The contributions are three-fold.

First, a modified finite state automaton was developed and used to guide the design of the positioning input system. A two position clockwise composition rule was used to map characters to input positions. Two prototypes were created for the study. The finite state automaton was demonstrated for both the 8+1 system and the 4+1 system to input numbers and English letters.

Second, we studied the tri-axial accelerometer readings to represent standard input signals. The prototypes of the accelerometer assisted single key positioning input systems were implemented with and without audio feedback. With audio feedback prototypes were operated single handed and eyes free. We have developed a real-time positioning algorithm which defines device tilt orientations by analyzing the accelerometer readings.

Third, we have conducted controlled experimental studies of the accelerometer assisted single key positioning input system. Thirty seven students participated in experiments studying the 4+1 and 8+1 systems. A calculator prototype was evaluated on the 4+1 system. Both the calculator prototype and the text input prototype were examined on the 8+1 system. The experiments showed that users could operate the 4+1 system without any training. Users needed some practice with the 8+1 system before comfortably conducting operations. On both systems, the application prototypes were welcomed by the participants. Participants achieved an input speed of 2.8 WPM on the text input prototype. An experienced user (the author) achieved the eyes free input speed of 11.2 WPM after two hours of practice in the span of two weeks.

For the experiment, 10,189 lines of codes were programmed and 714,680 data records were collected and analyzed.

7.2 Future Work

The accelerometer assisted single key positioning input system was practical in the experiments on the calculator prototype and text input prototype. However, with the 8+1 system, user practice is still necessary for efficient operation. This is mainly due to difficulties in tilting to narrow regions. To overcome this problem, more training and practices could help, but a technical solution is more favorable. In an input with context, predictions are possible and the system could guess what users want to input. During the user input, the system could adopt a predictive method to help increase the input accuracy. Whenever a tilt falls in the border line area, the predicted result would be provided instead of the result based solely on the accelerometer readings. We believe that this approach will reduce the error rate and result in a flexible error-tolerate system.

The current accelerometer assisted single key positioning system could provide precise character input with audible feedback. This is attractive to some specific application scenarios. However, the input speed is slow compared to normal keyboard input. Although such approaches do not have to compete with the normal keyboard input, a quicker input method would be more favorable. A future study could seek higher input speeds. Natural language input techniques could increase input speeds. If a word is considered as an input unit instead of a letter as one unit, the input speed might be improved. The challenges are that the vocabulary could be too large to map all the vocabulary. On the other hand, even if there is a mapping for each word, it could be hard to practically operate. A system with simple mapping but with full vocabulary coverage would be the ultimate goal.

7.3 Final Remarks

This dissertation presented a general model to aid in the designs of limited input orientations mapping to a larger alphabet. It also implemented a single key positioning input on an accelerometer device. This accelerometer assisted single key positioning system will offer a novel one-handed eyes-free solution. It will also serve as a model for other positioning input system research and practices.

Bibliography

[1] Shoemaker G, Findlater L, Dawson JQ, Booth KS. Mid-air text input techniques for very large wall displays. Proceedings of Graphics Interface. 2009. p. 231-238.

[2] Baudisch P, Sinclair M, Wilson A. Soap: a pointing device that works in mid-air. Proceedings of the 19th annual ACM symposium on User interface software and technology. 2006. p. 43-46.

[3] Yamamoto T, Tsukamoto M, Yoshihisa T. Foot-step input method for operating information devices while jogging. Proceedings of the International Symposium on Applications and the Internet. 2008. p. 173-176.

[4] Kajastila RA, Lokki T. A Gestrue-based and eyes-free control method for mobile devices. Proceedings of the 27th international conference extended abstracts on human factors in computing systems. 2009. p. 3559-3564.

[5] Simpson T, Broughton C, Gauthier MJ, Prochazka A. Tooth-click control of a hands-free computer interface. IEEE Transactions on Biomedical Engineering. 2008;55(8):2050-2056.

[6] Evreinova T, Evreino G, Raisamo R. Four-Key Text Entry for Physically Challenged People. Proceedings of 8th ERCIM workshop, User Interfaces, Adjunct workshop proceedings. 2004. p. 28-29.

[7] Julio MB, Pablo BS. Text entry in the E-Commerce age: Two proposals for the severely handicapped. Journal of Theoretical and Applied Electronic Commerce Research. 2009;4(1):101-112

[8] Belatar M, Poirier F. Text entry for mobile devices and users with severe motor impairments: HandiGlyph, a primitive shapes based on-screen keyboard. Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility. 2008. p. 209-216.

[9] Karlson AK, Bederson BB, Contreras JL. Understanding single-handed mobile device interaction. Human-Computer Interaction Lab, University of Maryland, College Park, HCIL Tech Report 2006-02.

[10] Gao C, Pastel R, Tan J. Yet another user input method: Accelerometer assisted single key input system. Proceedings of the 8th World Congress on Intelligent Control and Automation. 2010. p. 3949-3954.

[11] Költringer T, Isokoski P, Grechenig T. TwoStick: Writing with a game controller. ACM International Conference Proceeding Series. 2007;234:103-110.

[12] Proschowsky M, Schultz N, Jacobsen NE. An intuitive text input method for touch wheels. Proceedings of the SIGCHI conference on Human Factors in computing systems. 2006. p. 467-470.

[13] Dunlop MD. Watch-top text-entry: can phone-style predictive text-entry work with only 5 buttons. Proceedings of the 6th international conference on human computer interaction with mobile devices and services. 2004. p. 342-346.

[14] Kranz M, Holleis P, Schmidt A. DistScroll: A new one-handed interaction device. Proceedings of the Fifth International Workshop on Smart Appliances and Wearable Computing. 2005. p. 499-505.

[15] Kim YS, Soh BS, Lee S. SCURRY: A new wearable input device. IEEE Transactions on Industrial Electronics. 2005;52(6):1490-1499.

[16] Karlson AK, Bederson BB, SanGiovanni J. AppLens and launchTile: Two designs for one-handed thumb use on small devices. Proceedings of the SIGCHI conference on human factors in computing systems. 2005. p. 201-210.

[17] Wigdor D, Balakrishnan R. TiltText: Using tilt for text input to mobile phones. Proceedings of the 16th annual ACM symposium on User interface software and technology. 2003. p. 81-90.

[18] Tegic Communications [Internet]. [Cited 2010 Dec 12]. Available from: http://www.t9.com.

[19] Bellman T, MacKenzie IS. A probabilistic character layout strategy for mobile text entry. Proceedings of the Conference on Graphics Interface. 1998. p. 168-176.

[20] MacKenzie IS. Mobile text entry using three keys. Proceedings of the second Nordic conference on human-computer interaction. 2002. p. 27-34.

[21] Sazawal V, Want R, Borriello G. The Unigesture Approach: one-handed text entry for small devices. Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction. 2002. p. 256-270.

[22] Gao C, Kong F, Tan J. HealthAware: Tackling obesity with health aware smart phone systems. Proceedings of IEEE International Conference on Robotics and Biomimetics. 2009. p. 1549-1554.

[23] Liu J, Wang Z, Zhong L, Wickramasuriya J, Vasudevan V. uWave: Accelerometer based personalized gesture recognition and its applications. Proceedings of IEEE International Conference on Pervasive Computing and Communications. 2009. p. 657-675.

[24] Baek J, Yun B. A sequence-action recognition applying state machine for user interface. IEEE Transactions on Consumer Electronics. 2008;54(2):719-726.

[25] Choi E, Bang W, Cho S, Yang J, Kim D, Kim S. Beatbox music phone: gesturebased interactive mobile phone using a tri-axis accelerometer. Proceedings of the IEEE International Conference on Industrial Technology. 2005. p. 97-102.

[26] Tokoro Y, Tsukamoto M, Muramatsu K, Hosomi S. Pointing with accelerometers for wearable computing. Proceedings of the 11th IEEE International Symposium on Wearable Computers. 2007. p. 111-112.

[27] Touchscreen [Internet]. [Cited 2010 Dec 12]. Available from: http://en.wikipedia.org/wiki/Touchscreen

[28] Partridge K, Chatterjee S, Sazawal V, Borriello G,Want R. TiltType: accelerometer supported text entry for very small devices. Proceedings of the 15th annual ACM symposium on user interface software and technology. 2002. p. 201-204.

[29] Jones E, Alexander J, Andreou A, Irani P, Subramanian S. GesText: accelerometer based gestural text-entry systems. Proceedings of the 28th international conference on Human factors in computing systems. 2010. p. 2173-2182

[30] Bonner MN, Brudvik JT, Abowd GD. Edwards WK. No-Look Notes: Accessible eyes-free multi-touch text entry. To appear in the Proceedings of the eighth International Conference on Pervasive Computing. 2010.

[31] VoiceOver [Internet]. [Cited 2010 Dec 12]. Available from: http://www.apple.com/accessibility/iphone/vision.html

[32] MacKenzie IS, Kober H, Smith D, Jones T, Skepner E. LetterWise: prefix-based disambiguation for mobile text input. Proceedings of the 14th annual ACM symposium on user interface software and technology. 2001. p. 111-120.

[33] How Y, Kan MY. Optimizing predictive text entry for short message service on mobile phones. In: Human computer interfaces international (HCII 05). 2005.

[34] Dunlop MD, Crossan A. Predictive text entry methods for mobile phones. Journal of Personal and Ubiquitous Computing. 2000;4(2):134-143.

[35] Kronlid F, Nilsson V. TreePredict: improving text entry on PDAs. Proceedings of the Conference on Human Factors in Computing Systems. 2001. p. 441-442.

[36] Gong J, Tarasewich P, MacKenzie IS. Improved word list ordering for text entry on ambiguous keypads. Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges. 2008. p. 152-161.

[37] Masui T. POBox: An efficient text input method for handheld and ubiquitous computers. Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing. 1999. p. 288-300.

[38] MacKenzie IS. KSPC (keystrokes per character) as a characteristic of text entry techniques. Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction. 2002. p.195-210.

[39] Wilson AD, Agrawala M. Text entry using a dual joystick game controller. Proceedings of the SIGCHI conference on Human Factors in computing systems. 2006. p. 475-478.

[40] Sipser M. Introduction to the theory of Computation. 1st edition. Boston(MA): PWS Publishing Company; 1996.

[41] Sandnes FE. Evaluating mobile text entry strategies with finite state automata. Proceedings of the 7th international conference on human computer interaction with mobile devices services. 2005. p. 115-121.

[42] Neves DM, Anderson JR. Knowledge compilation: Mechanisms for the automatization of coginitive skills. In: Anderson JR, editors. Cognitive skills and their acquisition. Hillsdale, NJ:Erlbaum; 1981. p. 1-55.

Appendix A

YAUIM Experiments Informed Consent Form

This research has been approved by the Institutional Review Board at Michigan Technological University with approval number M0565. The following is the content of the consent form signed by the participants before they conduct any experiment.

Rights of Research Subjects:

The Michigan Technological University Institutional Review Board has reviewed the request to conduct this project [approval number: M0565]. If you have any concerns about your rights in this study, please contact Joanne Polzien of the Michigan Tech-IRB at 906-487-2902 or email jpolzien@mtu.edu.

If you have any questions regarding this study, please contact Chunming Gao at chgao@mtu.edu or call 906-487-1657.

"I understand the YAUIM project is to experiment on the tilts/wave operations of a smart phone. I have freely volunteered to participate in this experiment. I have been informed in advance what my tasks will be and what procedures will be followed. I have been given the opportunity to ask questions and have had my questions answered to my satisfaction.

“I am aware that the experiment could be uncomfortable to the wrist and I'm free to quit the experiment anytime. I am aware that I have the right to withdraw consent and to discontinue participation at any time.

“My signature below may be taken as affirmation of all the above statements; it was given prior to my participation in this study.”

Name (Print):_____________________________________

Signature: ____

Date: ____

Appendix B

Experiment Database Tables

Table 1: To hold experiments information

1. Table name: experiments
2. Columns:

EXPERIMENT _ID	CHAR(2)
EXPERIMENT _SESSION _ID	CHAR(2)
EXPERIMENT _DESCRIPTION	CHAR(30)
EXPERIMENT _DATE	DATE
EXPERIMENT _USER _ID	CHAR(3)
EXPERIMENT _DEVICE _ID	CHAR(3)
EXPERIMENT _PLACE	CHAR(10)
EXPERIMENT _DURATION	NUMBER(2)

Table 2: To hold experiment data

1. Table name: yauim _experiment _data
2. Columns:

USER _ID	CHAR(3)
DEVICE _ID	CHAR(3)
TASK_ID	CHAR(2)
EVENT_TYPE	CHAR(1)
PROMPT_TYPE	CHAR(2)
TICK_COUNT	NUMBER(8)

RESPONSE	NUMBER(5)
IN_PROCESS	CHAR(1)
CLICK_COUNT	NUMBER(3)
PROMPT_COUNT	NUMBER(3)
PAIR_MARK	CHAR(1)
PROMPT_CHARACTER	CHAR(2)
ACTUAL_CHARACTER	CHAR(2)
PROMPT_ZONE	CHAR(2)
ACTUAL_ZONE	CHAR(2)
TILT_ZONE_ANGLE1	NUMBER(4)
TILT_ZONE_ANGLE360	NUMBER(4)
TILT_AWAY_FROM_TOP	NUMBER(4)
AX	NUMBER
AY	NUMBER
AZ	NUMBER
THE_DATE	CHAR(8)
THE_TIME	CHAR(6)