Interaction Techniques Utilizing Pen Device Characteristics and Various Input Modalities for Pen Computing

Jibin Yin

A dissertation submitted to
Kochi University of Technology
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Graduate School of Engineering
Kochi University of Technology
Kochi, Japan

September 2007

Abstract

Interaction Techniques Utilizing Pen Device Characteristics and Various Input Modalities for Pen Computing

Jibin YIN

Pen computing has attracted a lot of attention as an important access technology, in particular Tablet PCs, PDAs and other pen-based devices are available in the market. However these devices have not carved out a big niche in the market as expected. One of reasons is that current interaction techniques for pen-based systems are mostly only imitations of interaction techniques for mouse devices, i.e., pen interaction model just diverts from WIMP (window, icon, mouse/menu, pull-down menu/pointer) interface. Furthermore, not enough empirical tests have been performed to determine how we can improve pen-input usage and efficiency. Taking advantage of pen devices (mobile, direct pointing/manipulation, user-friendliness, etc.) and utilizing the information of various input modalities sensed from the pen, this thesis, therefore, seek to design and develop easy-of-use interaction techniques that are suitable for pen-based systems, and identifies and quantifies the influential factors that make pen-based interaction techniques more efficient with a view to improving performance and subjective usability for pen-based systems.

This thesis explores fundamental issues on input modalities: pressure and tilt. A series of three experiments were conducted to evaluate human capabilities and limitations in using pen-tip pressure as an additional channel of control information in carrying out trajectory tasks such as drawing, writing, and gesturing. We also explore the human capability of performing discrete target selection tasks by varying the pen tilt angle, with full or partial visual feedback, including different techniques for confirming selection once the target is acquired. A classification of pen tilt based techniques is proposed, along with a series of possible pen tilt technique designs.

Target selections and document navigations are the most common and fundamental tasks. For pen-based OS effective interaction techniques for such selecting and navigating performances play important roles. Therefore, utilizing multiple input modalities and considering pen device characteristics, a series of novel pen-based selection and navigation techniques are proposed (such as ZWPS, Beam Cursor, Adaptive Hybrid Cursor, Line-based Multi-target Selection Techniques and Stroke-based Scroll Techniques). We also designed and conducted experiments to prove their effectiveness.

The major contributions of the thesis can be summarized briefly as: (1) this thesis investigates human performance in relation to couple various input modalities with pen-based interactions by conducting a series of experiments. The highlights of the results of fundamental studies provide guidelines for designs of interaction techniques utilizing various input modalities. (2) This thesis also proposes a variety of interaction techniques for enhancing selection or navigation tasks. We also conducted the experiments to evaluate these techniques, comparing with the current promising techniques or standard techniques in GUIs. The experimental results indicate that these techniques offer significant advantages to pen interface design.

key words Pen computing, pen device characteristic, input modality, human ability, pen interaction technique, selection task, navigation task.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Motivation

This thesis is concerned with pen-based interaction techniques and interrelated design factors that influence human performance. The need for this research has emerged from the development of a variety of pen-based devices (see Fig.1.1) such as PDAs, Tablet PCs and whiteboards. Pen-based interaction is an attractive user-computer interface paradigm. With the advances in hardware technology, off-the-desktop computing in the forms of handheld devices and tablets has made pen-based interfaces increasingly more relevant to mainstream applications. The pen-based interaction is one of natural computer interactions that can enable the ultimate ubiquitous computing. The natural style of pens is in that pen-based interfaces are designed on the Pen-Paper metaphor which is analogous to the user's real working environment; it is natural for people, especially fits for the handwritings. Pen-Paper metaphor is a universal and fundamental way for capturing daily experience, communicating ideas, recording important events, conducting deep thinking and visual descriptions.

Although pen-based interactions have much potential to facilitate computer users there is still dilemma that cumbers advantages of pens. The current pen-based OS (e.g., OS in Tablet PCs released by Microsoft) has relied on a UI that is simply incremental over the existing desktop UI and still remains the WIMP style of mouse-keyboard-based UI. The pen is little more than a mouse pointer in the existing GUI. Therefore, some of

Fig. 1.1 A sample of pen-based devices. (a) small size devices: PDAs; (b) medium size devices: Tablet PCs; (c) big size devices: whiteboards.

the mouse-based interaction techniques may not be suitable for pen-based systems. For example, in traditional GUIs, scrolling tasks are commonly accomplished using Scroll Bars, which are always fixed on the right and bottom sides of windows. For mouse-based systems, it is not difficult to perform scrolling tasks with a mouse because mouse motion is relative movement. However, because pen motion is absolute movement, traditional scrolling tasks performed with a stylus pen make users feel tired because they have to make longer pen-movements than movements made with a mouse for similar tasks. With a pen, users have to switch their attention between working areas and then drag the elevator. To overcome the above drawback, we need to design scrolling techniques which are suitable for pen based systems, e.g. stroke-based scrolling described in Chapter 8.

To solve the problems mentioned above, i.e., finding pen-suitable UI, we begin to address this issue by investigating pen devices characteristics. Pen devices are feature with electronic pens and touch screens. Electronic pens, the input tools of them, originate from physical pens (age-old and traditional writing instruments) so electronic pens inherit the physical properties of real pens, e.g., performances of easily writing and painting for information exchange - taking notes and so on. Electronic pens, as electronic tools, also have their own properties. Electronic pens like mice, are able to use

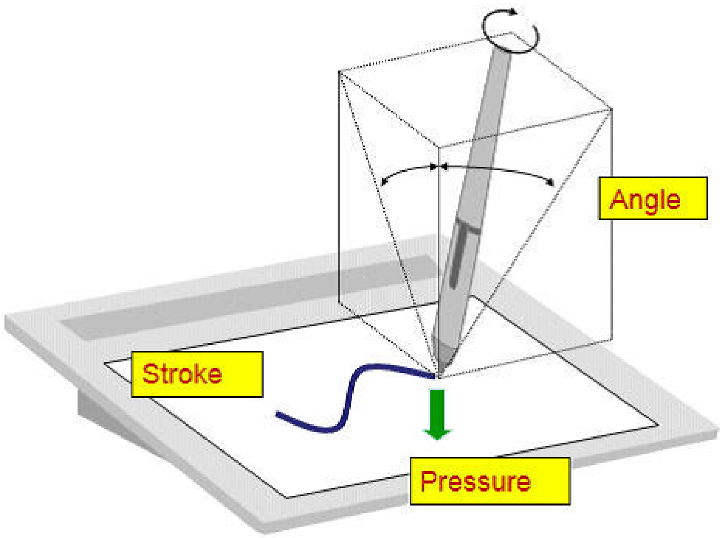

Fig. 1.2 Pen devices characteristics and the additional input modalities sensed from pens: Pressure, Tilt, Azimuth and so on.

the information offered by the two dimensional coordinates x-y. But Electronic pens also offer other potential input modalities which are not available with mice and which can be used to affect a wide variety of interaction techniques in which some may be with minimal user movement. As a direct input device, a unique capability of the pen as an input device would be possible to sense the contact force (e.g. pressure) of the pen tip on the screen thus potentially making it a highly effective input modality for a variety of useful interactions. In a similar way, the pen can also sense the angle (tilt and azimuth) information between the stylus pen and the screen. Pressure reflects users force information and angle reflects position information of users holding pens in pen interaction performances.

In addition to the usual x-y positional cursor control and button clicks that pens are currently used for, one can imagine using the pens' pressure to operate performances that have several discrete states, or to control a continuous variable. These additional input modalities could serve to increase the human-computer communication bandwidth, particularly when pen-based devices are used as pure slates with no keyboard. The current pen-based OS are designed to be operated by pointing devices only with two degrees-of-freedom (coordinates x, y) that map to the x-y position of the cursor, and binary buttons on barrels of pens that enable discrete selection. However, the current pen-based OS lacks of utilizations of pen devices characteristics and additional multiple input modalities (to date, these input modalities have typically only been used by a few drawing, such as stroke thickness or color opacity.), which limits much potential of pens and leads to weakness of The current pen-based OS.

In the research on pen-based user interface software, many researchers have focused on issues about on-line handwriting recognition, and informal user interface (i.e., sketching/drawing-based interface). We emphasize that a successful pen-based application must also include pen-based interaction techniques which are apt to pen characteristics. In views of computer operation pen-based interaction techniques have not been paid much attention on by researchers yet.

Therefore, there is an urgent need to explore pen-based interactions coupling with pen devices characteristics and multiple input modalities, which enable the user to interact with pen-based devices in a natural way.

1.2 Research Goal

With a view to improving performance and subjective usability for pen-based systems, we seek to:

- Investigate the characteristics of stylus pens, and identify and quantify the influential design factors that make pen-based interaction techniques more efficient.
- Design and develop interaction techniques that are suitable for pen-based systems.

As shown in Fig.1.2 the functional potential of stylus pens is far greater than that of the mouse in many situations and that the investigation and exploitation of this

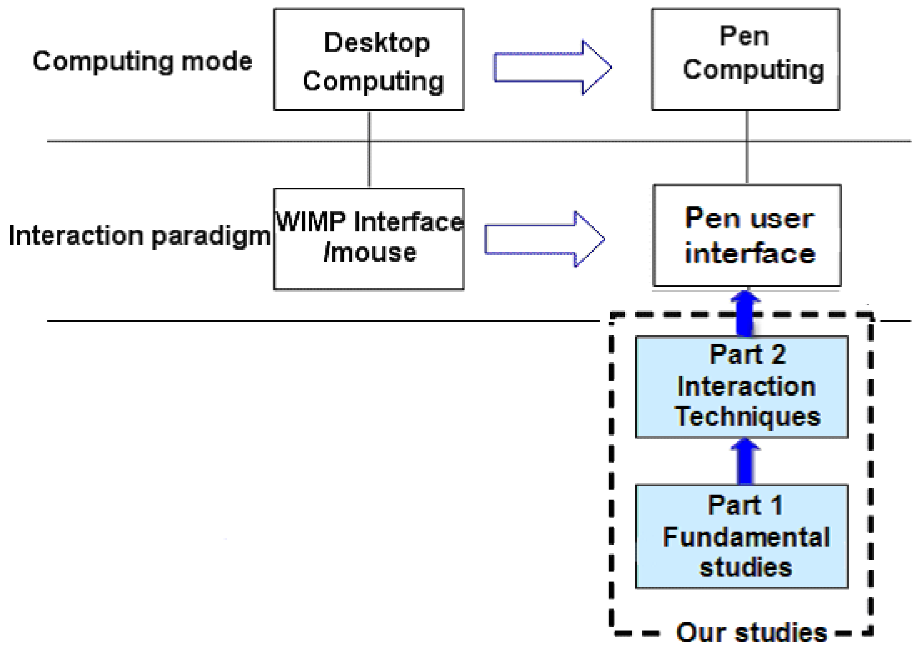

Fig. 1.3 The paradigm shift from desktop computing to pen computing.

potential is still in its infancy. Therefore, the central goal of this thesis is to study pen-based interaction which employs the multiple input modalities available with pens and considers pen devices characteristics. There are two key parts which must be taken to accomplish the main goal:

Part1: We investigate some fundamental issues related to multiple input modalities, i.e. human ability to control and perform one or more input modalities with the stylus pen. Although, at present, a few researchers have applied input modalities to pen-based interaction techniques a coordinated and systematic approach to the study of the characteristics of stylus pens is lacking. Our aim is to establish guidelines for this important area of HCI research and design.

Part2: We develop some novel interaction techniques based on multiple input modalities.

In fact, in research fields of pen computing, there are many voids on investigation to additional input modalities. Such researches are necessary, which can offer guidelines for pen UI design based on additional input modalities sensed from pens. In Part 1, we explore fundamental issues on input modalities: pressure and tilt. Drawing strokes is quite easy for pens to performs, which utilizes coordinates x, y (position information of a pen tip). Such tasks are defined as trajectory-based tasks in HCI (Human computer interaction), which can be modeled by Steering law (section 1.3.2). Pen-tip pressure measurement has been enabled in many of the state-of-the-art pen-based computing devices. Taking advantage of this additional channel of control information in user interface design is an attractive option. The pen as an input device is particularly apt for trajectory tasks, but a scientific study of human ability in pressure control when performing trajectory tasks has not been previously conducted. Our study in Chapter 2 is one systematic attempt at filling such a void. With varying degrees of experimental control, we conducted three experiments, each focusing on a different aspect of the topic. They can provide a body of empirical knowledge as the basis for future research and design of pressure sensitive UIs and applications.

Like pressure, pen tilt input could also serve to further increase the human-computer communication bandwidth. In particular, literature lacks a body of empirical knowledge on the human ability to control pen tilt, which could be used as a guide for designing appropriate pen tilt techniques. Moreover, there are some questions that need to be answered: What target (sector) angle can a user easily select by pen tilt? Whether and how does visual feedback of pen tilt change human performance in pen tilt selection tasks? What mechanisms can be used to indicate the completion of a targets selection when pen tilt is used to select one of a discrete set of targets? These questions are studied in Chapter 3.

Target selections and document navigations are the most common and fundamental tasks. For pen-based OS effective interaction techniques for such selecting and navigating performances play important roles. Therefore, utilizing multiple input modalities and considering pen device characteristics, a series of novel pen-based selection and navigation are proposed. We also designed and conducted experiments to prove their effectiveness. These are studied in Chapter from 4 to 8.

1.3 Background Knowledge

1.3.1 The Development of Pen Computing

The now ubiquitous direct manipulation interface, where visible objects on the screen are directly manipulated with a light-pen, was first demonstrated by Ivan Sutherland in Sketchpad [109], which was his PhD thesis. SketchPad supported the manipulation of objects using a light-pen, including grabbing objects, moving them, changing size, and using constraints. It contained the seeds of myriad important interface ideas. The system was built at Lincoln Labs with support from the Air Force and NSF. William Newman's Reaction Handler [84], created at Imperial College, London (1966-67) provided direct manipulation of graphics, and introduced "Light Handles," a form of graphical potentiometer, that was probably the first "widget." Another early system was AMBIT/G (implemented at MIT's Lincoln Labs, 1968, ARPA funded). It employed, among other interface techniques, iconic representations, gesture recognition, dynamic menus with items selected using a pointing device, selection of icons by pointing, and moded and mode-free styles of interaction. David Canfield Smith coined the term "icons" in his 1975 Stanford PhD thesis on Pygmalion [107] (funded by ARPA and NIMH) and Smith later popularized icons as one of the chief designers of the Xerox Star [105]. Many of the interaction techniques popular in direct manipulation interfaces, such as how objects and text are selected, opened, and manipulated, were researched at Xerox PARC in the 1970's. In particular, the idea of "WYSIWYG" (what you see is what you get) originated there with systems such as the Bravo text editor and the Draw drawing program [39] The concept of direct manipulation interfaces for everyone was envisioned by Alan Kay of Xerox PARC in a 1977 article about the “Dynabook” [53]. The first commercial systems to make extensive use of Direct Manipulation were the Xerox Star (1981) [105], the Apple Lisa (1982) [116] and Macintosh (1984) [117]. Ben Shneiderman at the University of Maryland coined the term “Direct Manipulation” in 1982 and identified the components and gave psychological foundations [102].

Early attempts to use pen input were limited by the touch screen technologies. In the late 1980s, early pen computer systems generated a lot of excitement and there was a time when it was thought they might eventually replace conventional computers with keyboards. Pen computers, as envisioned in the 1980s, were built around handwriting recognition. In the early 1980s, handwriting recognition was seen as an important future technology. In 1991, the pen computing hype was at a peak. The pen was seen as a challenge to the mouse, and pen computers as a replacement for desktops. Microsoft, seeing slates as a potentially serious competition to Windows computers, announced Pen Extensions for Windows 3.1 and called them Windows for Pen Computing. However, pen computers did not sell well. Most of users found pen tablets difficult to use. They also criticized handwriting recognition and said it did not work well.

In fact pen computing has many advantages such as natural operations and mobility. The pen-based input on handheld devices now looks promising. The problems existed in current pen computing are due to the lack of interaction techniques that are quite suitable for pen computing. This is the direct motivation of our research.

1.3.2 Models of Human Performance

A model is a simplification of reality. In human-computer interaction (HCI), models allow metrics of human performance to be determined analytically without undertaking time-consuming and resource-intensive experiments. Predictions of models allow

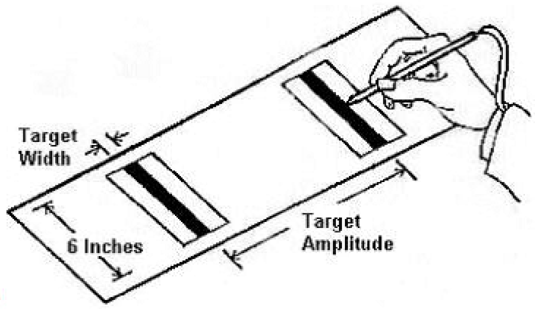

Fig. 1.4 Fitts'law reciprocal pointing paradigm.

a design scenario to be explored hypothetically without implementing a real system and gathering the same performance metrics through direct observation on real users. Therefore, Models help in designing, evaluating, or otherwise providing a basis for understanding the behaviour of a complex artifact such as a computer system. Fitts' law and Steering law are two fundamental models in HCI, which are used in our researches. Now we introduce the two models in the following sections.

Fitts'law

Fitts' law is a model of human psychomotor behavior developed in 1954 [34]. Extending Shannon's theorem in information theory (a formulation of effective information capacity of a communication channel), Fitts discovered a formal relationship that models speed/accuracy tradeoffs in rapid, aimed movement (not drawing or writing). According to Fitts' Law, the time to move and point to a target of width W at a distance A is a logarithmic function of the spatial relative error ($\frac{A}{W}$), that is:

$$MT = a + b\log_2(\frac{A}{W} + 1) \tag{1.1}$$

where, MT is the movement time. a and b are empirically determined constants, that are device dependent. A is the distance (or amplitude) of movement from start to target center. W is the width of the target, which corresponds to "accuracy".

The term $\log_2(\frac{A}{W}+1)$ is called the index of difficulty (ID). It describes the difficulty of the motor tasks. 1/b is also called the index of performance (IP), and measures the information capacity of the human motor system. Mathematically interpreted, Fitts' Law is a linear regression model.

Fitts' law is an effective quantitative method of modeling user performance in rapid, aimed movements, where one appendage (like a hand) starts at a specific start position, and moves to rest within a target area. Card et al. [23] reported the first comparative evaluation of the mouse, and also the first use of Fitts' Law in Human-Computer Interaction. Fitts' Law is an intensively used theory in Human-Computer Interaction. It can be used in assisting interface designs and in interface evaluation.

Steering law

The steering law is a predictive model of how quickly one may navigate, or steer, through a 2-dimensional tunnel. The tunnel can be thought of as a path or trajectory on a plane that has an associated thickness or width, where the width can vary along the tunnel. The goal of a steering task is to navigate from one end of the tunnel to the other as quickly as possible, without touching the boundaries of the tunnel. A real world example that approximates this task is driving a car down a road that may have twists and turns, where the car must navigate the road as quickly as possible without touching the sides of the road. The steering law predicts both the instantaneous speed at which we may navigate the tunnel, and the total time required to navigate the entire tunnel. Within human-computer interaction, the law was rediscovered by Johnny Accot and Shumin Zhai [1], who mathematically derived it in a novel way from Fitts' law using integral calculus, experimentally verified it for a class of tasks, and developed the most general mathematical statement of it. Some researchers within this community have sometimes refer to the law as the Accot-Zhai steering law. In this context, the steering law is a predictive model of human movement, concerining the speed and total time with which a user may steer a pointing device (such as a mouse or stylus) through a 2D tunnel presented on a screen (i.e. with a bird's eye view of the tunnel), where the user must travel from one end of the path to the other as quickly as possible, while staying within the confines of the path. One potential practical application of this law is in modelling a user's performance in navigating a hierarchical cascading menu.

Many researchers in human-computer interaction, including Accot himself, find it surprising or even amazing that the steering law model predicts performance as well as it does, given the almost purely mathematical way in which it was derived. Some consider this a testament to the robustness of Fitts' law.

In its general form, the steering law can be expressed as

$$T = a + b\int_C \frac{ds}{W(s)} \tag{1.2}$$

where T is the average time to navigate through the path, C is the path parameterized by s, $W(s)$ is the width of the path at s, and a and b are experimentally fitted constants. In general, the path may have a complicated curvilinear shape (such as a spiral) with variable thickness $W(s)$.

Simpler paths allow for mathematical simplifications of the general form of the law. For example, if the path is a straight tunnel of constant width W, the equation reduces to

$$T = a + b\frac{A}{W} \tag{1.3}$$

where A is the length of the path. We see, especially in this simplified form, a speed-accuracy tradeoff, somewhat similar to that in Fitts' law.

We can also differentiate both sides of the integral equation with respect to s to obtain the local, or instantaneous, form of the law:

$$\frac{ds}{dT} = \frac{W(s)}{b} \tag{1.4}$$

which says that the instantaneous speed of the user is proportional to the width of the tunnel. This makes intuitive sense if we consider the analogous task of driving a car down a road: the wider the road, the faster we can drive and still stay on the road, even if there are curves in the road.

1.3.3 Research Methodology

The research methodology in this thesis is a combination of experimentation and development of interaction techniques. Experimentation is quite accepted method in HCI research. However, meaningful and significant experimental results depend not only on the execution of the experiment but also on the design of the experimental task and paradigm and, most importantly, on the formulation of the experimental hypotheses, which relies on the theoretical analysis of the issues being investigated.

Interaction techniques are often evaluated in terms of usability. The essential components of usability can be summarized as follows:

- Efficiency: resources expended in relation to the accuracy and completeness with which users achieve aims should be small
- Satisfaction: there should be freedom from discomfort, and positive attitudes towards the use of the techniques should be high
- Learnability: the use of the technique should be easy to learn

To evaluate Efficiency and Learnabilities, accuracy and speed of task completion are usually measured.

Each chapter begins with a research motivation, followed by a literature review. Interfaces that embody the issues being investigated are then designed and implemented. Experiments are then carried out for carefully designed representative tasks. Conclusions are drawn based upon rigorous statistical analysis of the experimental results.

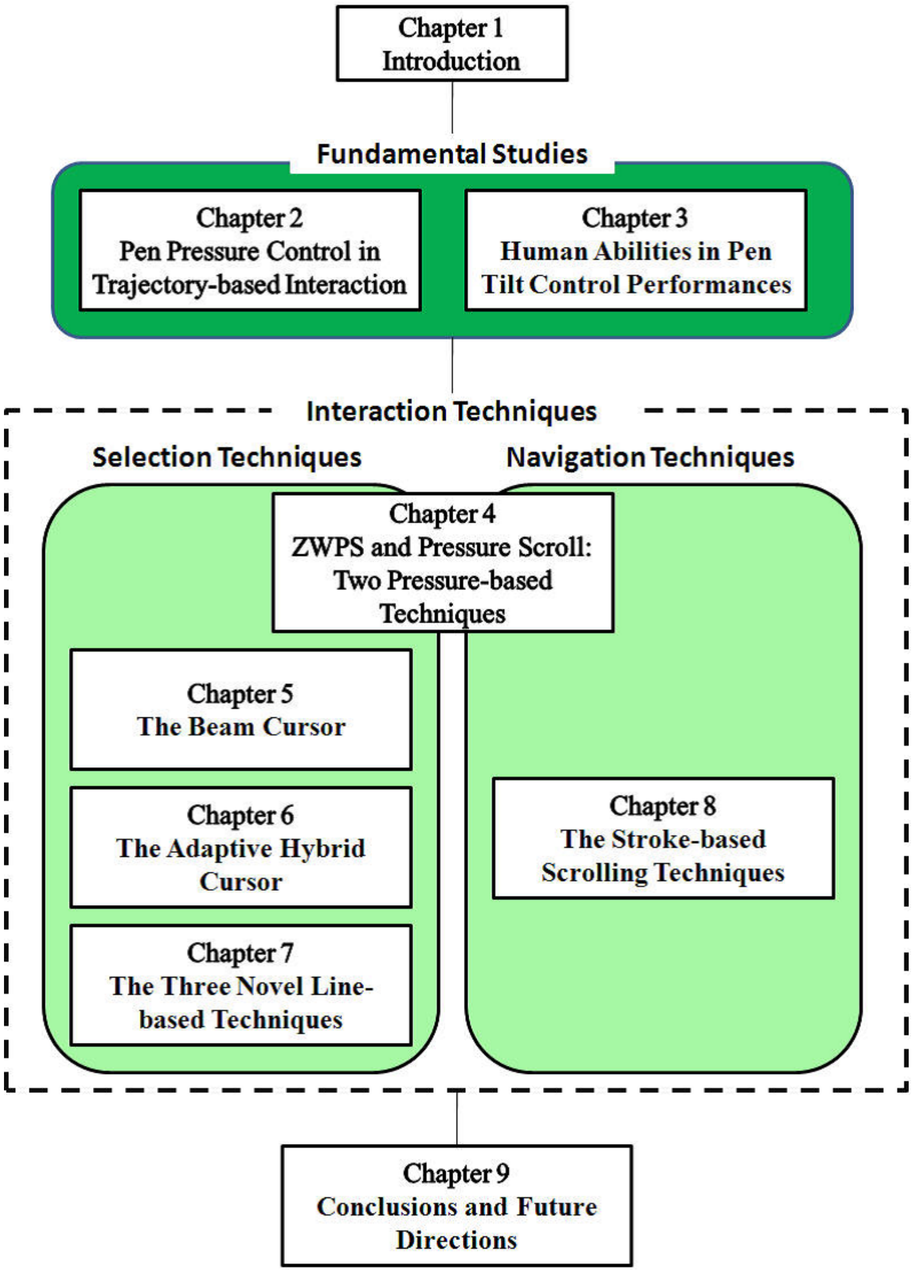

Fig. 1.5 The dissertation structure.

Finally, these results are discussed in relation to the literature. The structure if this dissertation is shown in Fig.1.5.

1.3 Background Knowledge

This is a blank page.

Chapter 2

Pen Pressure Control in Trajectory-based Interaction

2.1 Introduction

As discussed in Chapter 1, there are urgent requirements for the fundamental studies on multiple input modalities sensed from pens. This chapter focuses on human abilities of controlling pressure when performing trajectory-based interaction tasks. We first introduce the motivation of this study.

Pen-based interaction is an attractive user-computer interface paradigm. With the advances in hardware technology, off-the-desktop computing in the forms of handheld devices and tablets has made pen-based interfaces increasingly more relevant to mainstream applications. Consequently, research into pen-based interaction methods has intensified in recent years [9] [47] [59] [66] [67] [99] [124].

A unique capability of an electronic pen as an input device is that it is possible to sense the contact force of the pen tip with the screen (commonly called pen pressure in the UI field although more precisely pressure is defined as force per unit area in physics terminology). Using this third sensing dimension of the pen in addition to the 2D x and y coordinates appeals to both practical software applications (such as in Adobe Photoshop for controlling drawing width or opacity) and HCI research [66] [80] [93] [94] [95]. Using pen pressure, it is possible to increase interaction efficiency since another variable (e.g., width or other drawing properties) can be changed simultaneously while controlling the regular 2D x-y input without resorting to frequent mode switching as in typical mouse-based interactions. Despite such an appealing possibility, pen pressures actual application to date is rather limited. Two types of interrelated research efforts are needed to advance this topic. One is creative design and experimentation, particularly for specific application domains [66] [93] [94] [95], and the other is characterization of the fundamental human capabilities and limitations of using pen pressure [94]. This chapter focuses on the latter, particularly when performing trajectory-based interaction tasks.

To provide further motivations for our study, we first briefly review a few possible designs and applications of pressure-modulated trajectory tasks. These designs and applications illustrate the types of design efforts our study attempts to support at a more basic scientific level.

2.2 Sample Applications of Pressure Trajectory Tasks and Basic Human Performance Issues

Many types of pressure-modulated trajectory tasks can be designed. Here we propose a few examples.

Drawing with different properties: In addition to the previously mentioned width and opacity, it is possible to use pressure to control other properties of drawing, for example, line texture, line type (broken vs. continuous), line color, contrast with background color, etc. Another application is to use pressure to control a third spatial dimension. For example, when drawing with a circle shape, the pen pressure can simultaneously control the height, resulting in a cylinder (see Fig.2.1). *Pressure stroke-based scrolling:* Normally, a pen stroke produces a dragging or a line drawing function.

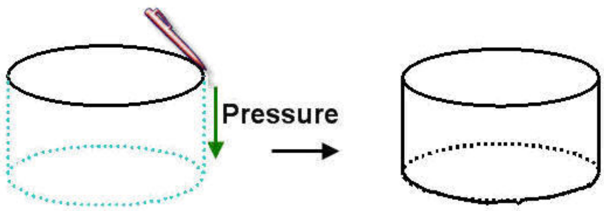

Fig. 2.1 Drawing graphs with pressure as additional dimension.

Fig. 2.2 Activating different commands with varying pressure.

When maintained at a certain (higher) pressure level, a stroke can cause the active document to scroll instead. Furthermore, the scrolling movement ratio (document displacement/stroke length) can be modulated by the pressure level. The higher the pressure, the higher the scrolling ratio. *Pressure stroke gesture commands:* For text editing, drawing a stroke over text usually highlights the characters or words crossed over. With pressure modulation, it is possible to carry different commands when sliding the pen over text. A low pressure level selects the text. A higher level copies the text. A higher still level cuts the text, and so on. Careful design is needed to assign the more "difficult levels to more consequential commands (e.g., delete or cut). See Fig.2.2. Many other similar or completely different pressure-modulated trajectory tasks can be imagined. But, clearly, common basic questions regarding humans ability to use pressure arise. A systematic, theoretical, and empirical basis is needed to design these types of application efficiently without extensive trial and error iteration. For example:

- Without conscious control of force level, what is the natural range of pressure that is typically used when drawing or writing with a pen?

- How does human performance differ at different pressure levels? Whether and how does visual feedback of force change human performance in pressure control?
- What is the human force resolution when controlling pen pressure in trajectory tasks? How many layers (error tolerance interval) of pressure can users discriminate and control? At what error rates?
- Does the steering law [1] [2] [3] apply to "steering under pressure"? How does human ability to control pressure change with the index of difficulty as quantified by the steering law?

The goal of our study is to answer these fundamental and general questions systematically and empirically. In the remainder of this chapter, we first review the literature related to the current work and then report three experiments with increasing complexity and experimental control. Experiment 1 establishes the normal range of force used in drawing with an electronic pen. Experiment 2 examines how well people can maintain pen pressure at different levels, with and without visual feedback. Experiment 3 investigates pressure control in the steering law paradigm. Finally, we discuss and make conclusions of our findings.

2.3 Related Work

Early studies on force-enabled devices include the work of Herot and Weinzapfel [44], who investigated how pressure and torque can be applied to object manipulation on a computer screen. Another early study by Buxton, Hill, and Rowley [21] analyzed the characteristics of touch-sensitive tablets and presented application examples such as using continuous pressure sensing to control the width of a drawing tool.

Recent research on using pen pressure in user interfaces has primarily focused on novel interaction technique design. For example, Ramos and Balakrishnan investigated a video editing application that allows users to annotate video segments using pressure-sensitive techniques [93]. The researchers implemented a pressure widget to assist users in previewing large sequences of video frames. By varying the amount of pressure applied, users were able to view frames in larger or smaller segments. The same researchers later designed Zlider, which allows users to apply pressure to zoom in while performing x-y cursor motion for scrolling or sliding [95]. They tested the widget with different input devices and found that using a stylus for zooming and sliding tasks yielded slightly better performances than using input devices that separate the zoom and slide controls.

Psychophysics, the study of psychological scale (human senses) as a function of physical stimuli, is a relevant discipline when considering a new modality of interaction. One of the main results of psychophysics is Webers law, which models the just-noticeable-difference (JND) of a physical variable as a constant proportion of the reference stimulus value. There is a surprising lack of psychophysical research on human sensitivity to force. One exception is a study conducted by Durlach and colleagues [86] [111]. Their study showed that force, as experienced and tested in an active figure motion paradigm that let participants squeeze two plates separated by a programmable linear motor, does follow Webers law; the average force JND was around 7% to 8% of the reference force. In other words, the greater the operating force is, the larger the JND is. However, pen-tip pressure perception may be very different from perception measured in the active finger motion paradigm since the former is isometric whereas the latter involves motion, among other different characteristics. It is unfortunate that to date no systematic psychophysical studies on pen pressure have been reported that we could base our study on.

The closest empirical study to our current inquiry is one by [94] that systematically evaluated human capability in using pen pressure when performing target selection (Fitts law) tasks. Letting participants push the pen tip to drive a vertical one-dimensional cursor, these researchers were able to establish that pressure control can follow Fitts law (more distant or smaller targets take more time to reach). The investigators found that providing visual feedback of the current pressure level improved performance, to the extent users could reliably control pressure to 6 intervals (bands). While many of these findings are relevant to our current questions, a major departure point is that their study evaluated pressure control while the pen was stationary (for 1D target reaching), while our current interest is human capabilities and limitations in controlling pressure while moving the pen (for trajectory tasks).

The steering law has been proposed by Accot and Zhai [1] [2] [3] as a model and paradigm for studying trajectory-based human-computer interaction tasks, akin to Fitts law for target pointing tasks. The steering law quantifies the difficulty of a trajectory task with an index, and relates path steering time with the index in a linear fashion. The steering law has been verified with several input devices, such as the stylus, mouse, isometric joystick, touchpad, and trackball [2], in different scales [3] and in locomotion [122]. Other researchers have studied and applied the steering law in a variety of settings [30] [43] [89] [110]. For example, recently [89] studied steering through paths with corners, and found that some corner angles were more difficult than others. Clearly, the steering law can potentially offer a systematic evaluation paradigm for studying pen pressure in trajectory-based tasks, which to our knowledge has never been done.

2.4 Experiment 1

Regular use of a pen for drawing or writing inevitably involves a certain range of variable pen-tip contact force. Understanding such a natural range is a necessary starting point of studying human capabilities and limitations in controlling pressure in trajectory tasks. Furthermore, the "weight" of the pen and its holding hand when resting on the screen is probably not zero. To use the pen-tip force as an explicit separate channel of control information, we may need to avoid using force that is below this resting balance point. Our first experiment therefore consisted of two parts. Part 1 measured the distribution of pen-tip force in drawing and writing. Part 2 measured the weight of the pen tip when the user rests the pen tip on the surface of the tablet sensor.

2.4.1 Participants

Ten volunteers (9 males and 1 female) participated in the experiment. The average age was 25.3 years (ranging from 21 to 31). All were right-handed.

2.4.2 Apparatus

The hardware used was a Wacom Cintiq21UX interactive LCD graphics display tablet with a wireless pen, which has a pressure-sensitive isometric tip that is capable of reporting up to 1024 units of force (corresponding to 4N according to the manufacturer's specification). We selected and verified that the digits reported and force applied were linear. The experimental software ran on a 3.2GHz P4 PC with Windows XP Professional operating system. The software for the experiment was developed using Sun Microsystems' Java.

2.4.3 Part 1

Part 1 of the experiment consisted of three tasks. The first task was to draw freehand strokes (arbitrary curves and straight lines) on a blank space in a natural manner, lasting for one minute. The second was to draw basic geometric graphs, such as rectangles and circles, in specified sizes. The third was to write a mixed set of Roman, Japanese, and Chinese (kanji) characters and signatures. Pen-tip pressure was recorded in a 17ms sampling period.

Results

The force distribution in all three tasks combined, and the distributions in each task individually are presented in the histograms in Fig.2.3. There was not much difference in force distribution among the three tasks. Ninety-five percent of the force samples fell within the 210 to 810 units range (corresponding to 0.82 N to 3.16 N) in the combined distribution. As a user drew a stroke with a pen on a screen, the contact force varied, but the force was rarely below 0.82N (3.6% of the total samples) or higher than 3.16N (1.4% of the total samples). Since the participant was not asked to control the pen-tip pressure, 0.82N to 3.16N can be considered a natural and comfortable range of force for most users when performing pen-based trajectory tasks. However, we caution that these measurements were made with, and therefore dependent on, today's state-of-the-art electronic pen technology. It is conceivable that these measurements may change with the mechanical properties of the pen (size, weight, shape) and the screen (friction).

2.4.4 Part 2

Part 2 of this experiment measured the weight of a pen plus its holding hand naturally resting on the screen, with the same 10 participants as in Part 1. They were asked to take up a pen, naturally rest the pen tip on the tablet screen, then lift the pen up and put it back on a desk. This procedure was repeated 10 times for each participant. The steady state value of the pen-tip resting force was recorded.

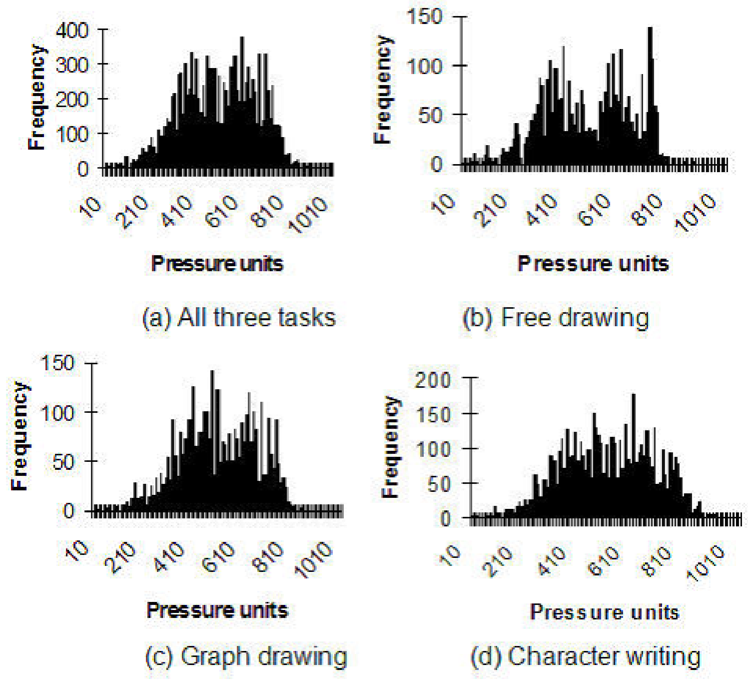

Fig. 2.3 Pressure distribution histograms for three tasks combined and individually.

Results

Fig.2.4 shows the histogram of all recorded resting force values. As shown in Fig.2.4, 90% of the force samples were between 200 units (0.78N) and 400 units (1.58N). The mean resting force was 280 units (1.09N). The significantly non-zero value of the resting force offers one explanation as to why the very low range of pen pressure was less controllable in previous pen-pressure control studies [94].

2.5 Experiment 2

Having established the range of force used in natural drawing and writing, as well as the range of resting forces, our second experiment evaluated people's ability to maintain the pen-tip force within a specified error tolerance interval at different pressure levels while moving the pen. In particular, we were interested in the relative controllability of

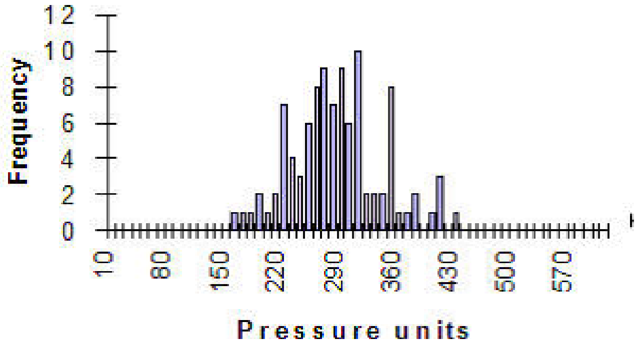

Fig. 2.4 Distribution of "resting force" the weight of a pen and its holding hand.

different levels of pen pressure as well as the impact of visual feedback of pressure.

2.5.1 Design and Procedure

The apparatus used in Experiment 2 was identical to the apparatus used in Experiment 1. Fifteen volunteers (10 males and 5 females) participated in the experiment. The average age was 24.1 years (ranging from 21 to 40). One participant was left-handed, and the others were right-handed.

The task of this experiment was to draw as long a line as possible (toward a series of targets) while maintaining the pen-tip force within a prescribed error tolerance interval. The first independent variable was the level of pressure. We divided the total measurable pressure range of the pen into 6 layers of equal intervals. Previous research [94] using pressure to perform selection tasks has shown that people could reliably separate and control the pen-tip force to such a resolution. However, maintaining a stable pressure in a drawing or steering task could be more difficult than controlling pressure into 6 layers when the pen was stationary.

The second independent variable was visual feedback conditions (with and without feedback). Previous studies have shown that (continuous) visual feedback could

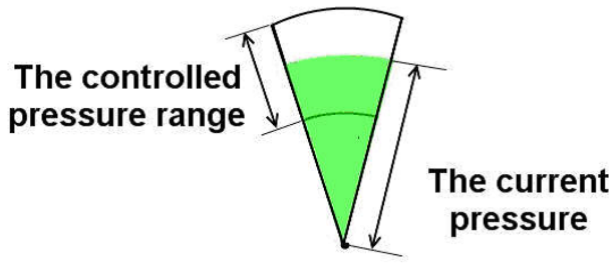

Fig. 2.5 The visual feedback of pressure.

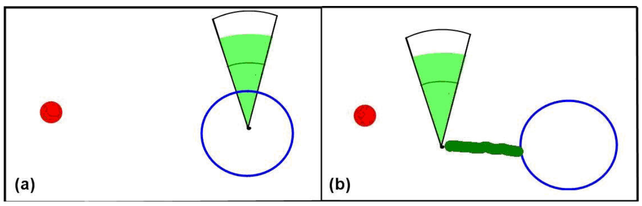

Fig. 2.6 Experiment 2's task. (a) The participant adjusted the pen pressure to within the desired pressure range; (b) kept the desired pressure while dragging the pen-tip toward the red solid circle.

markedly improve pressure control performance in stationary conditions [94]. We expected the same in pressure control during movement in trajectory tasks. In one condition of the experiment, we used a visual display that provided graphical real-time indication of the pressure being applied to the pen tip. Inspired by the pressure cursor design by Ramos and Balakrishnan [95], pen-tip force in the feedback condition was displayed in a wedge-shaped graphical widget (see Fig.2.5and2.6). The green area indicates the current pressure value. Two arc lines define the error tolerance interval that the participants needed to maintain throughout the trial.

The experiment started with an explanation of the task, followed by several warm-up trials in different conditions. The experiment proceeded to three trials of line drawing at each of the 6 (pressure levels) x 2 (with visual feedback and without feedback) conditions. The participants were allowed to rest between trials. For each trial, the participant was asked to draw a stroke toward a marker (solid red circle in Fig.2.6).

The participant first placed the pen tip inside the start circle and adjusted the pen pressure to within the desired pressure interval. They then dragged the pen while maintaining the pen-tip pressure in the interval. As the pen tip left the edge of the start circle, a dark green stroke appeared, showing the pen's trajectory, and the system started to record the distance of the stroke. As soon as the target was reached, the target marker disappeared, and the next target marker appeared. The participant continued drawing toward the next target until the last (the fifth target) was reached. The task was deliberately designed to be almost impossible to finish, so the participant was asked to draw as long a stroke as possible; the stroke length was used as the dependent variable of the experiment. In the actual experiment, 11% of the trials with visual feedback and 4% of the trials without visual feedback reached the last target. The trial terminated whenever the pen pressure was out of the specified error tolerance interval. The direction of each target was random, but the distance between two consecutive targets was kept the same; therefore, the maximum possible length with each trial was a constant (4500 pixels). The participant drew the strokes at a self-chosen speed.

For the visual feedback condition, the visual display of pressure remained throughout the trial. For non-visual feedback, once the pen tip moved out of the start circle, the visual feedback disappeared.

A balanced within-participant factorial design was used.

2.5.2 Results

The main results of Experiment 2 are presented in Fig.2.7. As shown, the participants' ability to maintain pen-tip force within the specified error tolerance interval changed with the pressure level required. They were more successful at the medium (1.30N to 3.26N) and high pressure levels (3.26N to 3.91N) than at the very low pressure levels (0 N to 1.30N). Visual feedback improved performance, particularly for the

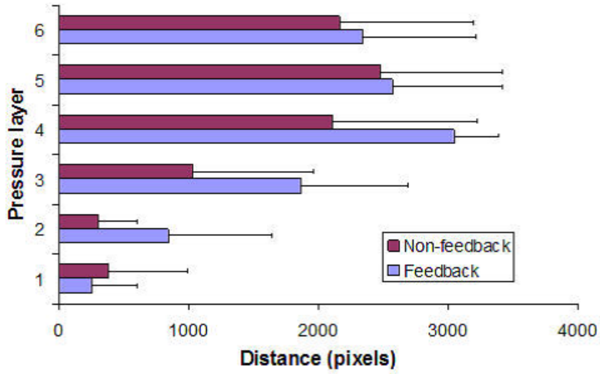

Fig. 2.7 The average movement distance duration while maintaining pen pressure at each of the six levels.

medium-range pressure levels. With visual feedback, the participants were most successful at the 4th level (1.95N to 2.60N).

Statistically, repeated measure variance analysis showed that the pressure level had a significant main effect on the stroke distance, both with visual feedback: $F(5,84)=33.8$, $p<0.0001$, and without visual feedback: $F(5,84)=18.4$, $p<0.0001$. The effect of visual feedback was also significant: $F(1,28) = 9.05$, $p< 0.01$. The distance drawn with visual feedback was significantly longer than that with non-visual feedback. The mean drawn distance was 1821 pixels for visual feedback and 1412 pixels for non-visual feedback.

For the visual feedback trials, post hoc Turkey HSD tests showed that there was no significant difference between the 1st and the 2nd pressure level, and no significant difference among the 3rd, 4th, 5th, and 6th levels. The distances drawn using the 3rd, 4th, 5th, or 6th levels were significantly longer than with the 1st level or 2nd levels of pressure (see Fig.2.7). For non-visual feedback trials, post hoc Turkey HSD tests showed that there was no significant difference among the 1st, 2nd, and 3rd levels and no significant difference among the 4th, 5th, and 6th levels. The stroke distances at the 4th, 5th, and 6th levels were significantly longer than those at the 1st, 2nd, and 3rd levels.

Comparing these results with the results of Experiment 1, as well as the results in the experiment of Ramos et al. in stationary pressure control [94], we concluded that pressure below 1.30 N, which invades the resting force region, is not usable for prolonged pressure control. The user would have to overcome the natural weight of the pen and the holding hand in order to express a desired level of pressure. In contrast, in the higher end of the pressure spectrum tested, although performance appears to begin to degrade at the highest pressure level (Fig.2.7), the levels of pressure that were above those naturally occurring in free drawing and character writing (Experiment 1) were not significantly more difficult to control than the medium range.

2.6 Experiment 3

Built on the first two experiments and with a greater level of experimental control, Experiment 3 evaluated human performance in controlling and maintaining pressure during trajectory tasks in the steering law paradigm. There were multiple goals for this final experiment in our series. First, we wanted to test whether the steering law still held when a certain level of pressure had to be maintained within a tolerance interval while performing the path steering task (called "pressure steering" hereafter). Pressure steering in fact is a 3D task, although the depth dimension is not spatial. Second, if the steering law held for steering with pressure, it would be informative to find out how pressure affects the steering law parameters and how pressure interacts with the steering law index of difficulty. Third, we wanted to further explore human performance (time and error) in pen-pressure control as a function of the number of pressure layer divided. The greater the number of layers the same range of force is divided into, the more precise the pen force has to be controlled to stay within a layer, the greater the number of bits of information one could potentially carry, and the greater difficulty one

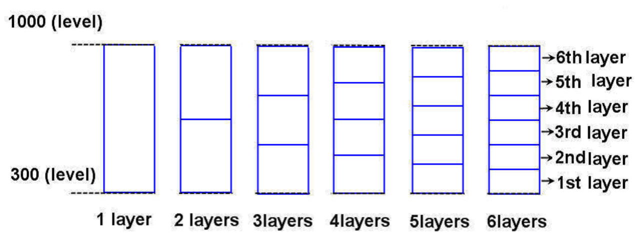

Fig. 2.8 Pressure conditions in Experiment 3.

would encounter. We wanted to reveal the relative pressure steering performance as pressure error tolerance interval changes, hence providing systematic empirical findings to UI designers when considering pen pressure in trajectory tasks.

Since Experiments 1 and 2 had shown that the low pressure range in the resting force region was markedly more difficult to control than other regions, we limited the required pressure to 300 to 1000 units (1.17N to 3.91N). From the results of Experiment 2, it was also clear that visual feedback was critical. To keep the size of the experiment manageable, we limited this experiment to visual feedback conditions only.

2.6.1 Steering Law and The Experimental Tasks

The tasks used in this experiment (Fig.2.9) were based on the steering law. The basic formulation of the steering law is the following:

$$T = a + bID \tag{2.1}$$

$$ID = \int_C \frac{ds}{W(s)} \tag{2.2}$$

where T is the time to successfully steer through path C and $W(s)$ is the path width at s . ID is the index of difficulty. a and b are constants. The current experiment used two types of path: linear and circular. For a linear path with constant width W, (2.2) can be simplified to:

$$ID = \frac{A}{W} \tag{2.3}$$

where A is the path length.

For a circular path with constant width W, (2.2) can be simplified to:

$$ID = \frac{2\pi r}{W} \tag{2.4}$$

where r is the radius of the circular path.

a and b change with the steering device, type of path, etc., and hence can serve as performance indicators.

In this experiment, there were a start segment and an end segment on the two ends of each steering path. The participants first placed a pen tip on the start segment, triggering the visual feedback widget. The participants then adjusted the pen pressure within the required pressure interval. Once adjusted, the entire steering tunnel changed to green to signal that the steering trial could begin. The participants proceeded to draw a red line on the screen, displaying the pen's trajectory. The steering timer started when the pen tip left the start segment and ended when the pen tip entered the end segment. An error was recorded and the trial was abandoned if the pen-tip pressure was out of the specified layer or when the pen tip steered outside of the borders of the path. The participants were asked to minimize errors.

2.6.2 Procedure and Design

The participants and apparatus used in Experiment 3 were identical to those used in Experiments 1 and 2.

A within-participants factorial design with repeated measures was used. The independent variables were the following: path type (linear and circular), the number of layers divided from the 300 to 1000 units of force range (from 1 to 6) (see Fig.2.8), the pressure level maintained (from the lowest layer to the highest layer divided), path length (A=200 and 800 pixels), and path width (W= 30, 40, and 60 pixels). The am-

2.6 Experiment 3

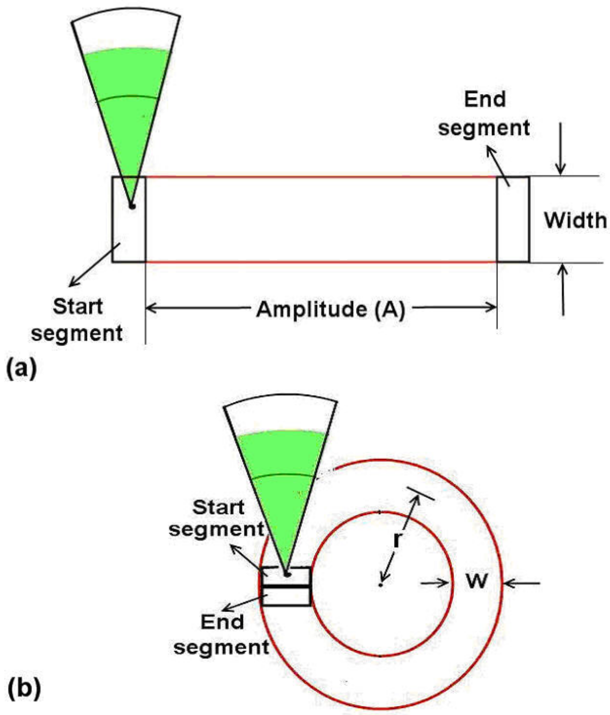

Fig. 2.9 pressure-modulated steering tasks. (a) linear; and (b) circular.

plitudes and widths defined 6 different IDs, ranging from 3.3 to 26.7. The dependent variables were the steering time and error rate. For each combination of the independent variables, two trials were performed. The total number of trials was 2 (types of steering tasks) x 21 (1+2+3+4+5+6 precision conditions) x 2 (length) x 3 (width) x 2 (repetitions) = 504. In the beginning of the experiment, 20 trials ranging from the easiest to most difficult conditions (A, W, and pressure conditions) in both linear and circular paths were run as practice trials. The total duration of the experiment was about 120 minutes. Whenever the participants felt tired, they were allowed to take a break.

2.6.3 Results

The applicability of the steering law, linear path

As the steering law predicted, ID, the steering law's index of difficulty (equation 3) had significant impact on mean successful steering time: $F(5,84)=114.2$, $p<0.0001$. Although pressure steering should be considerably more challenging than regular steering and maintaining pressure might have required greater attention than staying on the path, the steering law still held well. As shown in Fig.2.11, within each and every precision category (the number of pressure layers divided), the mean steering time correlated with the steering law's index of difficulty with 0.90 to 0.95 R^2 values. The steering law coefficients a (intercept) and b (the rate of time change per unit of ID change) both increased as the pressure control precision increased (with one point minor exception). The steering time for each category of precision can be expressed as (in ms):

6 layers: T = 110ID + 678 (R^2 = 0.95)

5 layers: T = 93 ID + 711 (R^2 = 0.93)

4 layers: T = 90 ID + 709 (R^2 = 0.94)

3 layers: T = 81 ID + 729 (R^2 = 0.90)

2 layers: T = 70 ID + 518 (R^2 = 0.91)

1 layer: T = 59 ID + 507 (R^2 = 0.94)

These results indicated that a pressure steering task could still be approximately modeled by the steering law, and the values of the steering law constants a and b indicated the increased task demand as the pressure control precision requirement increased.

The impact of pressure precision, linear path

As expected, as the number of pressure layers increased (hence a narrower band of each pressure error tolerance interval), the steering task took significantly longer

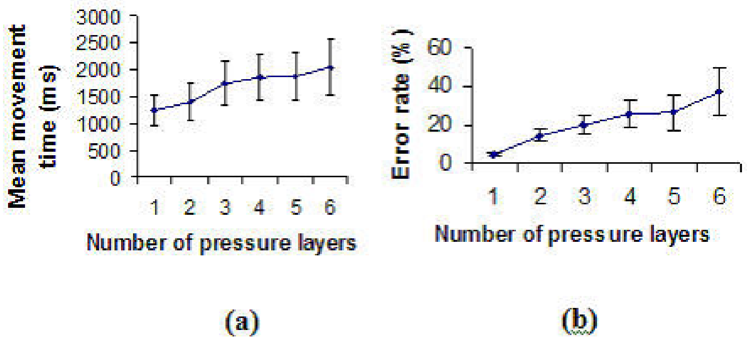

Fig. 2.10 The mean steering time (a) and error rate (b) as a function of the number of layers divided. Larger number of layers required higher pressure control.

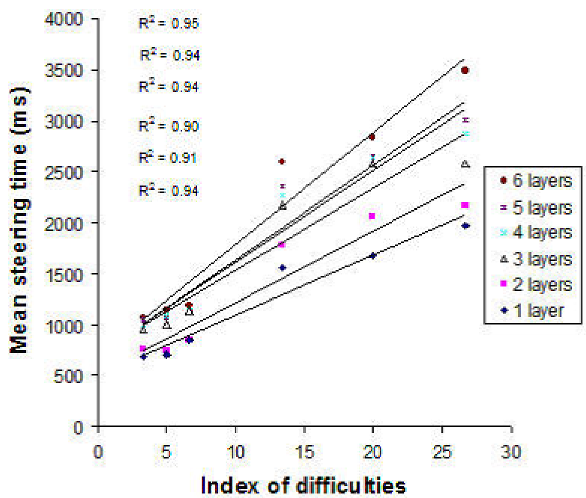

Fig. 2.11 The steering law regression in different pressure precision categories.

time: $F(5,84)=15.5$, $p<0.0001$, with a significantly higher error rate: $F(5,84)=12.2$, $p< 0.0001$. Fig.2.10 illustrates this.

To analyze more quantitatively, we defined σ (called the pressure precision ratio) as the ratio between the error tolerance interval (i.e., the width of each pressure layer) and the total range of force divided. In other words, σ is the size of each pressure layer as a fraction of the total pressure range used. If the number of the pressure layers is N, then

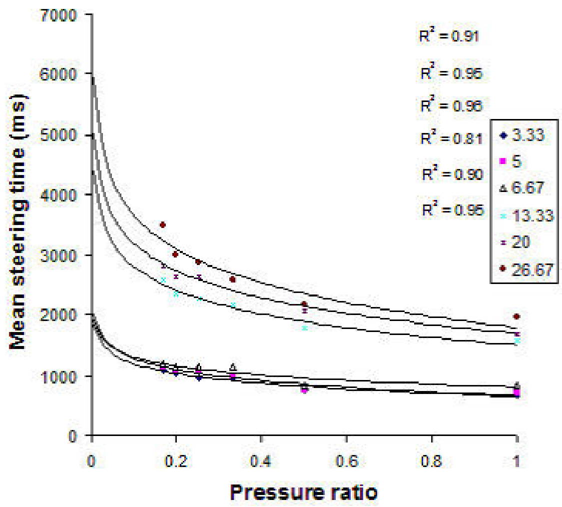

Fig. 2.12 The mean steering time for each ID against different pressure ratios.

$\sigma = (\frac{1}{N} pressure_{max} / pressure_{max}) = \frac{1}{N}$. The regression results of the mean steering time against the pressure precision ratio separated by steering law ID are illustrated in Fig.2.12. A logarithm function could be used to describe the relationship between the mean steering time (in ms) and the pressure precision ratio:

ID=26.6: T = -806 Ln(σ) + 1797 (R^2 = 0.91)

ID=20: T = -643 Ln(σ) + 1701 (R^2 = 0.95)

ID=13.3: T = -561 Ln(σ) + 1504 (R^2 = 0.96)

ID=6.67: T = -217 Ln(σ) + 807 (R^2= 0.81)

ID=5: T = -265 Ln(σ) + 670 (R^2 = 0.90)

ID=3.33: T = -233 Ln(σ) + 657 (R^2 = 0.95)

Fig.2.13 illustrates the mean steering time as a function of both ID and pressure precision σ.

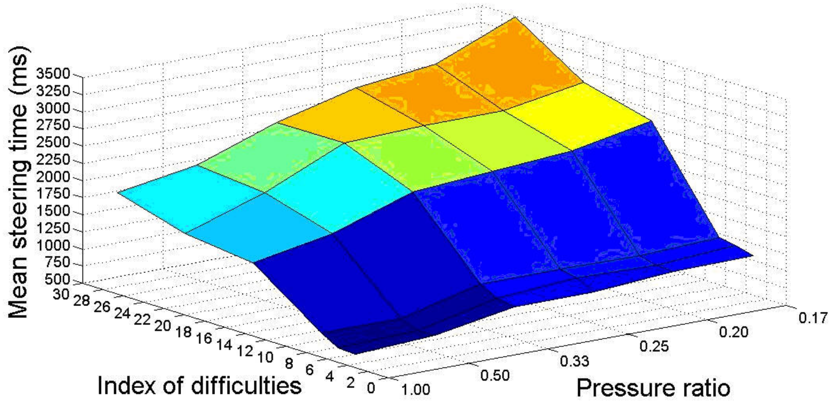

Fig. 2.13 3D perspective illustration of the relationship between mean steering time, ID and Pressure ratio in linear steering.

Error rate, linear path

In Experiment 3, if a trial went out of the boundaries of the path or was out of the pressure layer specified before completion of a trial, an error was recorded and the trial was repeated. For the linear steering tasks the error rates were 4.9%, 14.7%, 20.1%, 26.0%, 26.03%, 26.2%, and 37.0%, respectively, when the 300 to 1000 units (or 1.17N to 3.91N) of force range was divided into 1 to 6 layers (Fig.2.10b). In Accot and Zhai's reports [1] [2] [3] on steering tasks without pressure control, the error rate was in the range of 10% to 25%. Note that both Accot and Zhai's studies and our current study used the most stringent error standard: a correct trial requires the steering trace to be within the steering path 100% of the time. Other researchers have adopted a more relaxed standard. For example, Pastel's study on steering allowed the steering trace to deviate out of the path boundary but by no more than 5% of the path width [89]. The pressure steering tasks were more difficult than simple steering tasks that required no pressure control. Consequently, the error rate increased rapidly from steering while staying in 1 layer of pressure (largest error tolerance interval) to steering while discrim-

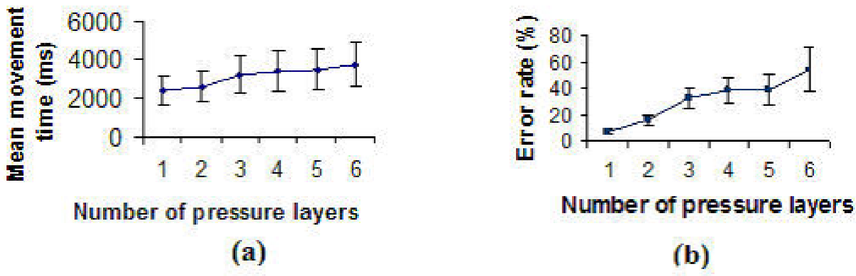

Fig. 2.14 The mean steering time (a) and error rate (b) as a function the number of layers divided in circular steering.

inating, and staying in 1 of the 6 layers of pressure (smallest error tolerance interval) (see Fig.2.10b). The greater the number of pressure layers was divided, the greater the error rate was. The error rate was markedly higher when the precision interval was 1/6 of the total force range. Designers of pen-pressure sensitive interfaces should be informed of these empirical results on error rates. Depending on the application, these error rates may or may not be acceptable.

2.6.4 Circular Path

The results of the circular steering task followed trends similar to those of the linear tasks, as shown in Fig.2.14, 2.15, 2.16 and 2.17. The error rates in circular pressure steering were higher than those in linear steering (compare Fig.2.10b with Fig.2.14b). The statistical variance analysis results for circular steering were qualitatively the same as those of the linear task (hence omitted for brevity).

Specifically, the circular steering times (in ms) as expressed by the steering law for each of the pressure precision categories (number of layers) were:

6 layers: T = 237 ID + 789 ($R^2 = 0.96$)

5 layers: T = 229 ID + 665 ($R^2 = 0.98$)

4 layers: T= 226 ID + 594 ($R^2 = 0.98$)

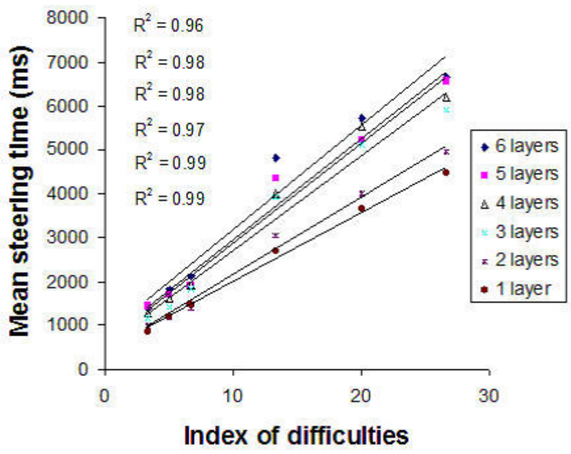

Fig. 2.15 The steering law regression in different pressure precision categories in circular steering.

3 layers: T = 216 ID + 530 (R^2 = 0.98)

2 layers: T = 177 ID + 383 (R^2= 0.99)

1 layer: T = 157 ID + 424 (R^2 = 0.99)

The circular steering times (ms) as a function of the pressure precision ratio for each ID categories were:

ID=26.6: T = -1319 Ln(σ) + 4362 (R^2 = 0.98)

ID=20: T = -1207 Ln(σ) + 3571 (R^2 = 0.91)

ID=13.3: T = -1156 Ln(σ) + 2559 (R^2 = 0.96)

ID=6.67: T = -386 Ln(σ) + 1355 (R^2 = 0.84)

ID=5: T = -368 Ln(σ) + 1091 (R^2 = 0.91)

ID=3.33: T = -343 Ln(σ) + 808 (R^2 = 0.97)

In summary, both the steering law and the pressure precision model held in the circular path tasks as well, and in some case better than (as reflected by the greater R^2 values) in the linear steering tasks.

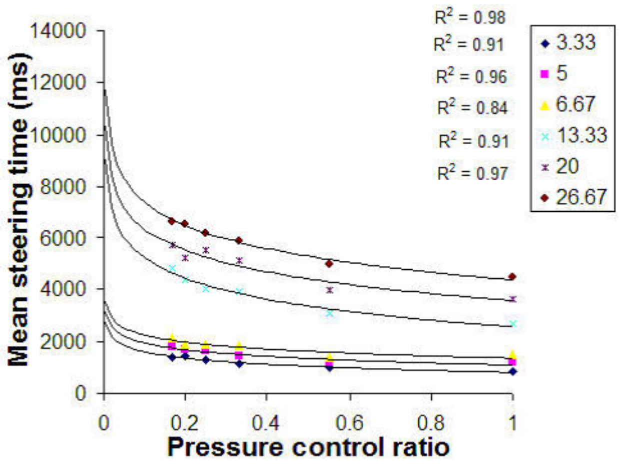

Fig. 2.16 The mean steering time for each ID against different pressure ratios.

2.7 Discussion

At the beginning of this chapter, we posed several basic questions concerning pressure-sensitive trajectory tasks. The results from the three experiments conducted can begin to address these questions.

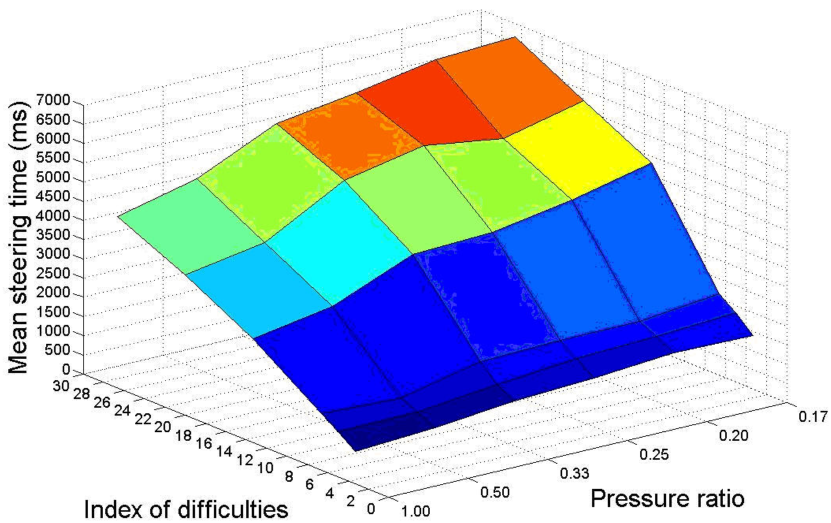

Fig. 2.17 3D perspective illustration of the relationship between mean steering time, ID and Pressure ratio in circular steering.

- What is the natural range of pressure that is typically used when drawing or writing with a pen?

Experiment 1 showed that in regular drawing and writing, pen-tip pressure primarily varied between 0.82N and 3.16N, suggesting pen users could comfortably apply pressure below 3.16N. On the other hand, there was the natural weight (200 to 400 units, 0.78 to 1.58N) of the pen and its holding hand when resting the pen tip on the computer screen (Part 2 of Experiment 1). Controlling and maintaining pressure below the rest force level is considerably more difficult than above it, as shown in Experiment 2.

- How does human performance differ at different pressure levels? How does visual feedback of pressure change human performance in controlling pressure during trajectory tasks?

Experiment 2 introduced pressure level as an experimental control factor. The experiment showed that controlling and maintaining pressure within a fixed precision band was more difficult when operating near or below the rest force level. Although began to degrade at the high end (from the 5th to the 6th level (2.61N to 3.91N) (Figure 7), it was still possible to control and maintain pen pressure above the high end of the natural range of pressure applied in regular drawing and writing (3.16N). Providing visual feedback of the pen-tip pressure significantly improved pressure control performance and widened the levels of pressure controllable, consistent with previous findings in stationary pressure control [94].

- What is the human force resolution when controlling pen pressure in trajectory tasks? How many layers of pressure (or error tolerance interval) can users discriminate and control? At what error rate?

One of the factors controlled in Experiment 3 was the number of layers the same force range was divided into (or the pressure control precision required). While it was possible to complete successfully stringent steering tasks when the force range was divided up to 6 layers (on both linear and circular paths, with steering law ID up to 26), the error rates increased rapidly, from 4.9% to 37% in linear steering and 7.4% to 54.7% in circular steering, as the control precision requirement increased to 1/6 of the force range. The steering time could be modeled by $T = a - bln(\sigma)$, where is the control precision tolerance interval as a fraction of the total force range. a and b are constants. The model can quantitatively predict pressure steering time if we further divide the same force range into more layers.

- Does the steering law apply to pressure steering? How does human ability to control pressure change with the steering law index of difficulty?

Experiment 3 also systematically controlled the steering law's index of difficulty by varying the length and width of the steering path. Results in both linear and circular steering showed that the steering law held quite well for "steering under pressure." The steering law parameters (a and b) changed as the pressure precision requirement changed. The higher the steering ID, the more rapidly the steering time increased with the precision requirement, as is shown by the non-parallel regression lines in Figures 11 and 15.

2.8 Conclusions

Pen-tip pressure measurement has been enabled in many of the state-of-the-art pen-based computing devices. Taking advantage of this additional channel of control information in user interface design is an attractive option. The pen as an input device is particularly apt for trajectory tasks, but a scientific study of human ability in pressure control when performing trajectory tasks has not been previously conducted. The current work is one systematic attempt at filling such a void. With varying degrees of experimental control, we conducted three experiments, each focusing on a different aspect of the topic. Together, they can provide a body of empirical knowledge as the basis for future research and design of pressure sensitive UIs and applications.

We can draw the following conclusions from our results (among many others): The natural range of pressure used in drawing and writing concentrates in the 0.82N to 3.16N range. The resting force of the pen tip on the screen is between 0.78N and 1.58N. Near or below, the resting force is markedly more difficult to use. Visual feedback of pressure improves pressure sensitive trajectory tasks. Up to 6 layers of pressure can be controlled in trajectory tasks, but the error rate is considerably higher for 6 levels. The steering law holds for pressure steering tasks, which enables the systematic prediction of successful steering time for a given path length, width, and pressure precision interval. The steering time can also be modeled as a logarithmic function of pressure control precision ratioσ.

This is a blank page.

Chapter 3

Human Abilities In Pen Tilt Control Performances

3.1 Introduction

Pen-based interaction is an attractive human interface paradigm. With advances in hardware technology, pen computing in the forms of handheld devices and tablets has made pen-based interfaces increasingly relevant to mainstream applications. Consequently, research into pen-based interaction methods has intensified in recent years [47] [67] [99]. Stylus pens, like mice, are able to use the information offered by the two-dimensional coordinates x-y. But stylus pens also offer other potential input modalities (e.g. pressure and tilt). They are not available with mice but they are natural aspects of pen usage which can be used to affect a wide variety of interaction techniques with minimal user movement. Pen pressure has appealed to users of practical software applications (such as Adobe Photoshop for controlling drawing width or opacity) and to HCI researchers [94] [95]. To fully use stylus pen input modalities, appropriate interaction techniques need to be designed which are based on an understanding of the human ability to control these modalities. Ramos et al. [95]have systematically investigated the human ability to control and exploit pen pressure in interaction techniques.

Like pressure, pen tilt input could also serve to further increase the human-computer communication interaction bandwidth. To date, pen tilt input has only been used in a few applications [60] [85]. In particular, literature lacks a body of empirical knowledge on the human ability to control pen tilt which could be used as a guide for designing appropriate pen tilt techniques. This is our main purpose in chapter 3. There are some questions that need to be answered: What target (sector) angle can a user easily select by pen tilt? Whether and how does visual feedback of pen tilt change human performance in pen tilt selection tasks? What mechanisms can be used to indicate the completion of a target's selection when pen tilt is used to select one of a discrete set of targets?

In the following sections we first briefly review previous efforts on this topic. Next, we explore the human ability to control pen tilt by conducting an experiment where participants perform discrete target selection tasks by varying the tilt of a pen, with full or partial visual feedback. The experiment also involves different pen tilt input techniques for confirming selection once the target is acquired. Based on the experimental results, we discuss implications for the design of pen tilt techniques. We also introduce a taxonomy of pen tilt techniques, along with several possible technique designs.

3.2 Related Work

Tilt has been applied in two ways: (1) the tilt of the screen (display) or input devices (regardless of the angle of the pen), we call this device tilt, and (2) the tilt (angle) of the pen in relation to the screen, called pen tilt in this chapter.

Until now, much research has been aimed at device tilt. This group includes many compelling interaction techniques based on the physical manipulation of a small screen device e.g. PDA, including contact, pressure, tilt, and motion directly related to the device itself. Earlier work was done by Fitzmaurice et al. [35] who investigated the use of positions and tilting actions based on the Chameleon system. They explored the potential of the tilting action as a natural way of issuing commands, e.g., scrolling up or down. Rekimoto [97] used both tilt and buttons to build several interaction techniques for navigating menus, maps, and 3-D scenes. During operation, only one hand is required to both hold and control the device, which is especially useful for field workers. Rekimoto and Sciammarella [98] proposed a cordless, multiple degree-of-freedom input device that senses physical manipulation of the device itself, such as rotating, flipping, or tilting. Harrison and colleagues [42] [46], Hinckley et al. [49], Small and Ishii [106], and Bartlett [13] used tilt sensors to scroll through and select information on a handheld device. Eissele et al. [31] used tilt operations to achieve successive scroll and link-step actions. Moreover, a number of researchers [88] [100] [114] have used absolute tilt sensing (i.e. the tilting of the whole device) to enter data. Wigdor and Balakrishnan [114] propose a new technique, TiltText, for entering text into a mobile phone, which tilts a phone in one of four directions to choose which character on a particular key to enter. Similar work has also been done by Partridge et al. [88]and Sazawal et al. [100]. Tilt and orientation have also been used to allow spatially aware display. Fitzmaurice and colleagues [36] studied how artists take advantage of their ability to reorient their work surface while sketching and writing. They introduced and explored many issues relating to Rotating User Interfaces (RUIs), as they called it: applications and toolkits for pen-based computing systems that take into account workplane orientation - the angle of rotation, relative to the user, around the axis perpendicular to the user's work surface.

The studies mentioned above explore the use of tilt sensed by sensors mounted on the screens (or devices). But those studies are quite different to our purpose in this chapter because they do not depend on the characteristics of the pen or on iterations based on the geometrical relationship between the pen and the plane of the screen. Pen tilt is a useful input modality, which reflects posture information of users' holding a pen. With the use of tilt, pen-based systems are able to provide an additional channel to interact with pen-based devices in an intuitive way. Graphics tablets, which are popular commercial isometric input devices, sense and utilize pen tilt information in a variety of meaningful ways. For example, Intuos/Intuos2 Graphics Tablet supports pen tilt, and it has been applied in Photoshop software to vary painting elements such as brush size.

Compared to device tilt, we found that very little literature has been reported on pen tilt input potential. Blasko et al. [20]present two complementary methods to achieve more fine-grained awareness of user-to-device orientation for a hand-held writing surface: one using computer vision techniques and another based on stylus-pose. Oshita [85] developed a pen-based intuitive interface to control a virtual human figure interactively. In this system the tilt of the pen is used to affect the figure's motion. The figure bends and stretches according to the direction and degree of tilt applied by the user. Simple user studies suggest that tilt and other input information from a pen can enhance the interaction of the system significantly. Kuroki and Kawai [60] proposed the use of tilt information for pen interfaces. They observed that users hold three physical tools (pencil, knife, and syringe) differently, and then they implemented drawing software in which the user can do the three operations (copying, pasting, cutting) by changing between three pen tilt angles using a tablet. However, no information is provided about how they determined the range of the three tilt angles of the pen.

In summary, our review indicates that with few exceptions [85] [60] most studies paid attention to the use of tilt sensors for input operations, while, again, little study has been done on the use of pen tilt as an effective input characteristic. In particular, there has not been a systematic exploration into the human ability to control the angle of tilt-sensitive styluses, nor into the design space of tilt angle techniques. This study fills a void in this area.

3.3 Experiment

3.3.1 Aims

The purpose of this study is to explore the human ability to implement discrete selection tasks by controlling the tilt of the input pen. This includes determining what target (sector) angle a user can easily select by pen tilt, and also how visual feedback impacts on human performance. We also compare five techniques for confirming selection after the target has been pre-selected by applying the appropriate pen tilt.

3.3.2 Apparatus

The hardware used was a Wacom Cintiq21UX interactive LCD graphics display tablet with a wireless pen. The pen has a pen tilt sensitive isometric tip that is capable of reporting from 30 degrees up to 90 degrees of pen tilt (in increments of 1 degree). The display was placed horizontally. The experimental software ran on a 3.2GHz P4 PC with the Windows XP Professional operating system. The software for the experiment was developed using Sun Microsystems' Java.

3.3.3 Participants

Twenty volunteers (all male) participated in the experiment. The average age was 21 years (ranging from 20 to 31). All were right-handed. The mean hand size of the participants was 18.7 cm (measured from the tip of the middle finger to the first wrist crease with the subject's hand held flat).

3.3.4 Task

In the experiment a serial target selection task was used (see Fig.3.1). The pen tilt was utilized to control the rotation movement of a pink cursor around a fixed point, either clockwise or anticlockwise. Pen tilt (from 30 to 90 degrees) was mapped uniformly to a circumferential angle (from 30 to 90). A set of equal and consecutive sectors (targets) were drawn by dashed lines around the fixed point i.e. radial lines. The angle of the target was experimentally determined. During each experimental trial, one of the sectors was highlighted in red to show that it was the desired target, and the experimental task was to apply the appropriate amount of pen tilt to rotate the pink cursor into the desired target. When the pink cursor entered the target, the target color changed to green, then a technique was used to confirm the selection.

The selection was confirmed by a confirmatory beep sound. Participants were told to emphasize both accuracy and speed. When a participant correctly selected the target he would hear a beep sound, or else an error had occurred and was recorded.

Two different visual feedback conditions were used in the Experiment: Full-visual-feedback (hereafter referred to as FLV) (Figure 3.1a, 3.1b) and Partial-visual-feedback (hereafter referred to as PRV) (Fig. 3.1c, 3.1d). The FLV condition indicates the target in context with other adjacent targets and it offers continuous visual feedback in the form of the pink cursor that can be rotated around the fixed point by adjusting the pen tilt. In the PRV condition, only the target is represented, and the pen tilt cursor is hidden. When beginning a trial the user has to depend on proprioceptive cues and memory to accurately determine the amount of pen tilt to apply so as to get the hidden cursor into the target. When the degree of tilt controlled the cursor within the desired target, the color of the target changed from red to green. In the FLV condition, visual feedback includes three parts: the pink cursor, the other adjacent targets and the color

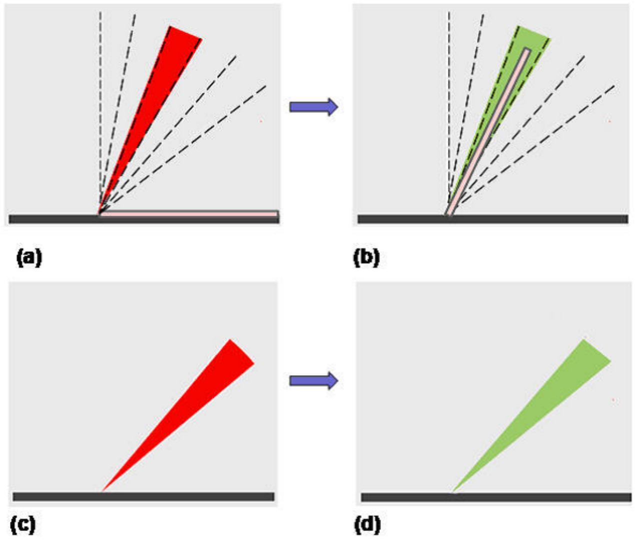

Fig. 3.1 The hardcopy of Experimental interfaces. The targets are represented as sectors marked by dashed lines. The red sector is a desired target. When the pink cursor entered into the desired target its color changed from red to green. The two visual feedback conditions are: Full-visual-feedback (FLV) and Partial-visual-feedback (PRV). The pink cursor indicates the current pen tilt amount. (a)→ (b) shows the selection process in the FLV condition, and (c)→ (d) in the PRV condition.

change from red to green when pre-selecting a desired target. However, in the PRV condition, visual feedback is provided at the final stage of the task (i.e., when the color of the desired target changes). This simulates the condition where expert users may be able to use pen tilt for quick selection in an eyes-free manner, similar to behavior exhibited by expert users of marking menus [64].

3.3.5 Techniques for Confirming Selection

Once the pen tilt cursor is in the target, a mechanism for the user to determine the final selection is required. In traditional graphic user interfaces (GUIs), this is typically implemented by clicking binary buttons on a mouse. Some pens have buttons on their barrels, therefore, an analogous mechanism would be achieved by pressing the barrel buttons. Thus, we use this selection technique in our study. In practice, however, the ergonomics of the pen do not permit good performance using the barrel button because users often rotate the stylus and the button may not always be in a position that facilitates pressing. Additionally, pressing the barrel button often causes inadvertent movement in a certain direction. Such movements are very likely to change the angle of the pen and cause selection confirmation at an inappropriate angle. Furthermore, some pens do not possess such buttons. Therefore, there is a need to explore alternatives to the above typical mechanism. In this study, a total of five selection techniques were investigated. They are described as follows:

Pressing by unpreferred hand

This is a two-handed interactive technique [42] [54]. When a desired target is captured, Pressing by Unpreferred hand uses the unpreferred hand to press a physical key mounted at the corner of a pen-based device to confirm a selection. When the preferred hand adjusts the pen tilt so that it is within the desired tilt range, the unpreferred hand presses the physical key to confirm the selection.

Pressing barrel button

Pressing the button on the pen barrel to confirm a selection is a standard technique in many existing pen-based applications [68]. The barrel button can also serve as a right-click equivalent. In our experiment, when the users have adjusted the pen tilt within a desired target, they press the barrel button to determine the selection.

Holding for a fixed delay

Similar to Pressing barrel button, Holding for a fixed delay is also a standard technique for selection or as a right-click equivalent [68]. In this technique the user is required to press the tip of the pen onto the screen, hold it still for a prescribed delay (in our experiment the prescribed delay was 1 second) to confirm a selection. This technique leverages temporal information to confirm a selection, which is useful on devices where few input channels are available, e.g., a PDA or mobile phone.

Quick Releasing

A selection is confirmed by quickly lifting the stylus from the tablet's screen when a target is highlighted.

Drawing a stroke

The user draws a stroke out of a specified area to confirm a selection while maintaining a desired pen tilt angle. In our experiment the specified area was a circle with a diameter of 20 pixels, the size having been determined by a pilot experiment.

In the rest of the chapter, we will use the abbreviations NonPrefHand, BarrelButton, TimeOut, QuickRelease and DrawStroke to refer to these five techniques, respectively. The last four techniques were also studied in the pressure study [95].

3.3.6 Procedure and Design

A within-participants factorial design with repeated measures was used. The independent variables were the following: visual feedback types V (FLV and PRV), the angle of each target, W (2 degrees, 4 degrees, 5 degrees, 10 degrees and 20 degrees) (see

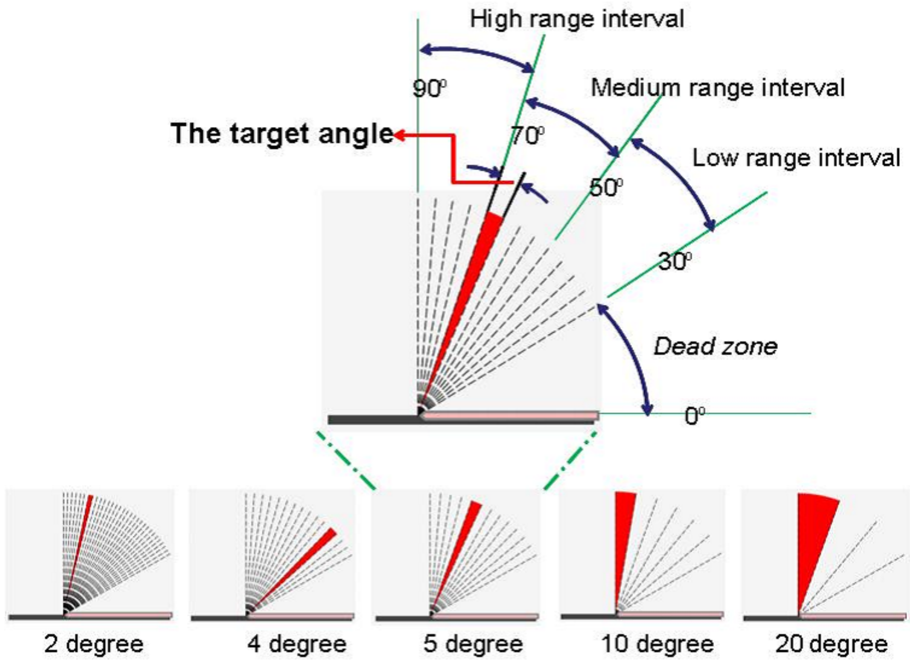

Fig. 3.2 Five target (sector) angles used in the experiment from 2 to 20 degrees. These directly correspond to the actual pen-tilt angles tested for user discrimination sensitivity.

Fig.3.2), selection techniques T (NonPrefHand, BarrelButton, TimeOut, QuickRelease and DrawStroke).

To explore the human ability to control the pen tilt at different levels the whole pen tilt range is divided into three pen tilt range intervals: low range (30-50 degrees), medium range (50-70) and high range (70-90). The three pen-tilt range intervals were determined in a pilot study. 20 participants were asked to take up a pen, naturally rest the pen tip on the tablet screen, then lift the pen up and put it back on a desk. This procedure was repeated 20 times by each participant. The steady state value of the tilt-angle between a pen and a screen was recorded. More than 80% of the pen tilt samples fell within the 50 to 70 degrees range. This was also the natural range of tilt-angle that is typically used when the hand holds a pen naturally resting on the screen, i.e. when the user was not exercising conscious control of the tilt-angle. The range from 30 to 50 and the range from 70 to 90 were seldom used naturally. For each of the combinations of V, W and T three trials were performed, which were selection tasks in low, medium and high pen tilt range intervals respectively. Each trial was repeated three times. Participants were asked to perform two blocks. Each block included all the combinations of V, W and T. The order of techniques was counterbalanced using a 5 x 5 Latin square. In order to familiarize them with the experimental environment, we set a practice session for each technique. When participants understood the task and could perform it correctly, they completed the experiment unassisted. Each participant typically spent 120 minutes finishing all the tasks. In summary, the experiment consisted of:

20 participants

5 angles of targets

5 selection techniques

2 visual feedback

2 blocks

3 types of trials

3 repetitions

= 18000 target selection trials

After they finished testing each technique, the participants were asked to fill in a questionnaire which consisted of three questions regarding "selection difficulty", "fatigue", and "overall usability" on a scale of 1-to-7 (1 = lowest preference, and 7 = highest preference). These questions were made by referring to ISO9241-9 [52].

The dependent variables were selection time (MT), error rate and subjective preference. The selection time was defined as the time from the moment the pen came into contract with the screen surface until the moment the user finished the selection using the appropriate technique. The error rate was defined as the percentage of erroneous selection trials in the total number of selection trials.

3.3.7 Results

An ANOVA (analysis of variance) with repeated measures was used to analyze performance in terms of selection time, error rate and subjective preference. Post hoc analysis was performed with Tukey's Honestly Significant Difference (HSD) test.

Selection techniques

Selection time: In our analysis of selection time we discarded erroneous trials. Participants' performances did not significantly improve during the experiment from block1 to block 2. Analysis showed a significant main effect for selection technique on selection time for the FLV($F(4, 95)=17.2$, $p<.00001$) and PRV ($F(4,95)=15.3$, $p<.00001$) conditions. For both FLV and PRV conditions the post hoc Tukey HSD test showed NonPrefHand, BarrelButton, QuickRelease and DrawStroke were significantly faster than TimeOut. There were no other significant differences across the selection techniques. These four techniques can be grouped according to their selection times. NonPrefHand was the fastest selection technique followed in order by BarrelButton, QuickRelease, DrawStroke and TimeOut (see Fig.3.3).

Error rate: Error rate was also significantly different across selection techniques for both FLV($F(4,95)=19.3$, $p<.00001$) and PRV ($F(4, 95)=5.8$, $p<.0001$) conditions (Fig.3.4). For the FLV condition the post hoc Tukey HSD test showed that TimeOut, NonPrefHand, QuickRelease and BarrelButton were all lower than DrawStroke in error rate (all had lower error rates than DrawStroke). There were no other significant differences across the selection techniques. TimeOut, NonPrefHand, QuickRelease, and BarrelButton can be grouped based on the similarity of their accuracy in confirming se-

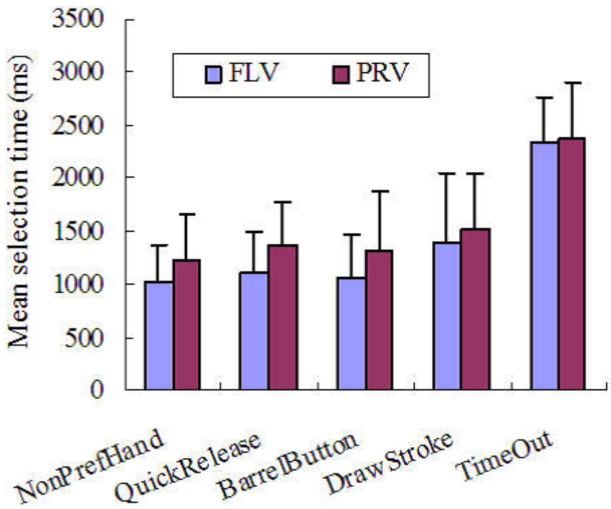

Fig. 3.3 The mean performance time for each selection technique in both FLV and PRV conditions.

lections. TimeOut was the most accurate technique followed, in order, by NonPrefHand, QuickRelease, BarrelButton and DrawStroke (see Figure 4). For the PRV condition the post hoc Tukey HSD test showed that NonPrefHand, TimeOut, QuickRelease and BarrelButton were all lower than DrawStroke in error rate. There were no other significant differences across the selection techniques. NonPrefHand, TimeOut, QuickRelease and BarrelButton can be grouped according to their similarity in accuracy in selection confirmation. NonPrefHand was the most accurate technique followed in older by TimeOut, QuickRelease, BarrelButton and DrawStroke.

Subjective preference: Fig.3.5 shows the subjective ratings for the three techniques. These ratings were based on the average value of the answers given by the subjects to the three questions. Subjective ratings showed that there was a significant difference among the five techniques, $F(4,95)=8.2$, $p<.001$. The post hoc Tukey HSD test showed NonPrefHand, BarrelButton, QuickRelease were rated significantly higher than DrawStroke and TimeOut. There were no other significant differences across the selection

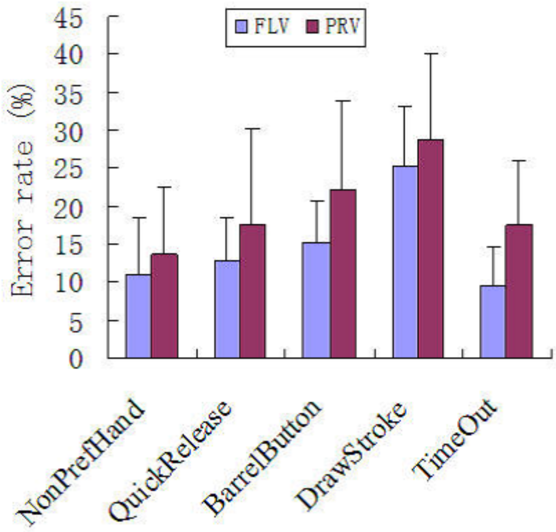

Fig. 3.4 The error rate for each selection technique in both FVF and PRV conditions.

techniques. NonPrefHand was the most preferred followed in order by BarrelButton, QuickRelease, DrawStoke and TimeOut.

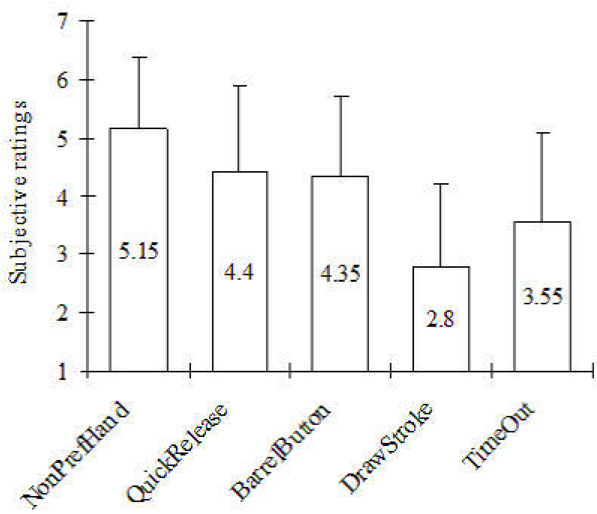

Fig. 3.5 The participants' preferences for each technique.

The target angle to be easily selected

One of the main purposes of this study was to investigate what target angle a user can easily select (See Fig.3.6). Repeated measure variance analysis indicated a significant main effect for different angles of targets on selection time for the FLV (F(4,95)=32.3, p<.00001) and PRV (F(4, 95)=28.9, p<.00001) conditions. As target angles decreased, the selection time (MT) became significantly longer in both FLV and PRV conditions. It is interesting to note that a power function could be used to describe the relationship between the mean selection time (in ms) and the angles of targets (in degree) controlled in FLV and PRV conditions:

For FLV:

NonPrefHand: MT = 2430.8 $\theta^{-0.53}$ ($R^2 = 0.97$)

QuickRelease: MT = 2739.9 $\theta^{-0.56}$ ($R^2 = 0.99$)

BarrelButton: MT = 2595.7 $\theta^{-0.54}$ ($R^2 = 0.99$)

DrawStroke: MT = 3418.5 $\theta^{-0.55}$ ($R^2 = 0.97$)

TimeOut: MT = 5989.9 $\theta^{-0.64}$ ($R^2 = 0.75$)

For PRV:

NonPrefHand: MT = 3221.8 $\theta^{-0.60}$ ($R^2 = 0.98$)

QuickRelease: MT = 3736.1 $\theta^{-0.63}$ ($R^2 = 0.98$)

BarrelButton: MT = 3641.6 $\theta^{-0.64}$ ($R^2 = 0.99$)

DrawStroke: MT = 3991.4 $\theta^{-0.60}$ ($R^2 = 0.98$)

TimeOut: MT = 6262 $\theta^{-0.62}$ ($R^2 = 0.89$)

Error rate was also significantly different across different target angles for the FLV (F(4,95)=44.1, p<.00001) and PRV (F(4,95)=32.0, p<.00001) conditions. Except for DrawStroke, error rates for the FLV condition when the target angles >= 10 ranged from 1 - 6% and were not significantly different within each selection technique, for

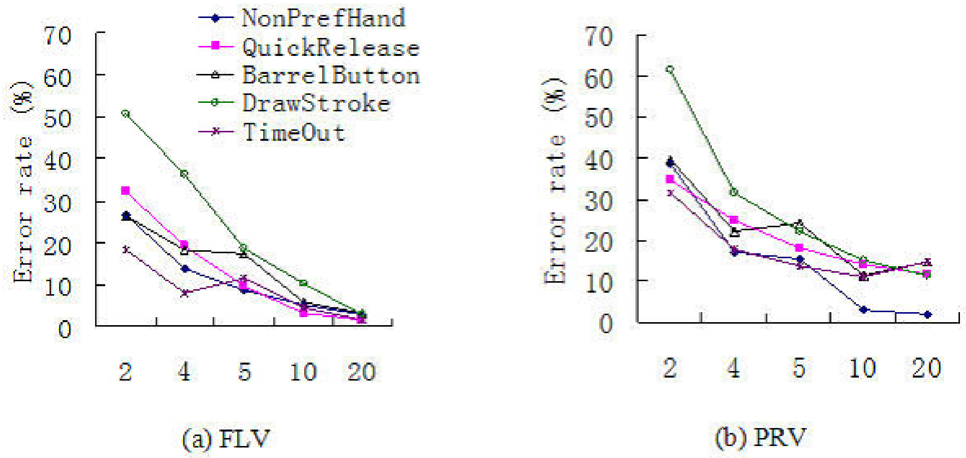

Fig. 3.6 The error rate for each technique at different target angles in both FLV and PRV conditions.

all selection techniques (Fig.3.6a). Except for BarrelButton and DrawStroke, when the target angles >= 10. error rates for the PRV condition ranged from 1 - 12%, and were not significantly different within each selection technique, for all selection techniques (Fig.3.6b). These results indicate to us that when the target angle was >= 10 it could be reliably differentiated with adequate visual feedback.

Effect of visual feedback

In our experiment full visual feedback (FLV) and partial visual feedback (PRV) were included. Analysis showed that no significant difference was found in selection time between the two visual feedback conditions. This is also an interesting result that may be due to the fact that the spatial angle between a pen and a screen could provide inherent (physical) "visual feedback". However, there was still significant difference between the visual feedback conditions in error rate, $F(1,38)=5.9$, $p<.05$. The selection tasks with FLV had lower error rates than those with PRV.

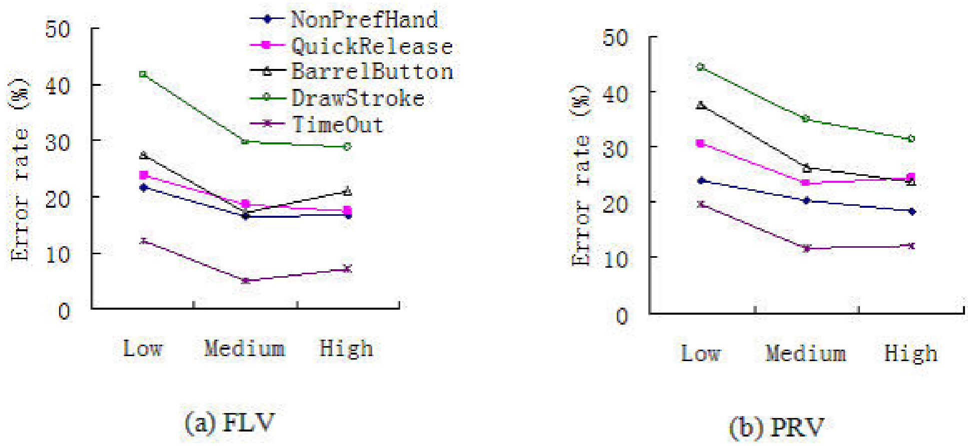

Fig. 3.7 The error rate for each selection technique at different pen tilt range intervals in both FLV and PRV conditions.

Pen tilt range intervals (low, medium and high)

Analysis suggested that no significant difference was found between the low, medium and high range intervals in selection time for each technique in both FLV and PRV conditions. However, a significant difference was found between the three range intervals in error rate for each technique except NonPrefHand in the FLV condition ($p<.01$) on FLV condition (Fig.3.7a). There was a significant difference between the three range intervals in error rate for each technique except NonPrefHand and QuickRelease in the PRV condition ($p<.01$) (Fig.3.7b). The low range interval (30-50) was more difficult to control than the medium (50-70) and high (70-90) range intervals.

3.4 Discussion

At the beginning of this chapter, we posed several basic questions concerning the human ability to control pen tilt. The results from the experiment conducted can begin to address these questions.

- What target angle can a user easily select by pen tilt?

The experimental results showed that as the target angle decreases, the selection time becomes significantly longer in both FLV and PRV conditions. The selection time (MT) could be modeled by

$$MT = a \cdot \theta^b \tag{3.1}$$

where θ is the width of a range of pen tiltand a and b are empirical constants. The model can quantitatively predict selection time if the pen tilt is changed. The experimental results suggest that when the target angle >= 10 degrees it can be reliably differentiated with adequate visual feedback. Investigation of how pen tilt selection tasks could be modeled by Fitts' law goes beyond the scope of this chapter and remains an issue for future research. Analysis suggested that no significant difference was found between the low, medium or high range intervals in selection time in FLV or in PRV conditions. However, more errors occur in the low range interval than in the medium or high range intervals. This indicates that the low range interval is more difficult to control than the medium or high range intervals. As observed in the experiment, the hand holding the pen was often unable to adjust pen tilt in the low range interval, and this lead to inaccurate performances in the low range interval. The cause of bad performance with pressure control in the low pressure range is that pressure is more sensitive in the low pressure range and pressure control is likely to be affected by the weight of the pen, a factor that does not have a significant effect on pen tilt control.

- We asked whether and how does visual feedback of pen tilt change human performance in pen tilt selection

In our experiment, full visual feedback (FLV) and partial visual feedback (PRV) were included. Experimental results suggested that providing FLV of pen tilt significantly improved pen tilt control performance in error rate when compared with PRV.

However, no significant difference was found between FLV and PRV conditions in selection time. Pen tilt between a pen and a screen can be observed by the user's eye, and this constitutes significant visible information. Pen tilt provides physical "visual feedback". This also reflects the fact that the invisible input information of pressure is more dependent on appropriate visual feedback than visible input information of pen tilt.

- What mechanisms can be used to indicate the completion of a target's selection when pen tilt is used to select one of a discrete set of targets?

Experimental results have shown that the different selection techniques have significant effects on the usability of pen tilt for performing discrete selection tasks. During the experiment it was obseved that the process of performing a selection included two steps: paticipants first applied the right amount of pen tilt to move the pen tilt cursor into the target, and then ultilized selection techniques to confirm target selections. Based on experimental analysis in selection time and error rate we found:

TimeOut was the most accurate. This is not surprising because the second step in this technique involves simply waiting for the fixed time delay to pass without any other movements being required. However, this technique was also the slowest technique. This is because, with the TimeOut technique, a fixed delay is required to confirm target selection.

NonPrefHand was subjectively preferred and it was quantitatively confirmed to be the quickest technique in this study. Li et al. [67] also suggest that NonPreHand offers the fastest performance for switching between ink and gesture modes in pen interfaces. NonPrefHand is a two-handed interaction technique. The two-handed technique takes advantage of the cooperation and the division of the task between two hands to reduce task completion time. For the second step of the selection process NonPrefHand also does not require any movement of the preferred hand and it enables greater accuracy. It was closely followed by TimeOut in the FLV condition and it was the most accurate in PRV condition.

QuickRelease, in which selection is confirmed by a quick release, was very similar to NonPreHand in accuracy. However, the requirement for sudden action often makes the user feel nervous, so this technique was not the most preferred by participants. QuickRelease, BarrelButton, DrawStroke and TimeOut are in the same group of one-handed techniques, and QuickRelease ranks first overall. Ramos et al. [94] indicates that QuickRelease shows the best performance of their four pressure selection techniques, while NonPreHand is not involved in any of their technique comparisons.

For BarrelButton, as might be expected from our earlier discussion about the ergonomics of the pen, we observed that pressing of the button in the BarrelButton technique interfered significantly with pen tilt control. Unless the pen's design can be changed significantly, our results indicate that this is not a good technique for pen tilt target selection. This also indicates that alternative selection techniques need to be designed.

DrawStroke was the least accurate selection technique overall. This is because with DrawStroke, the user must draw a stroke while maintaining the pen tilt value within a specified range, which significantly increases performance difficulties. Our results indicate that this is not a good selection technique for pen tilt input.

3.4.1 Implications for UI Design

The results of our experiment suggest several guidelines for the design of pen tilt techniques:

- Reduce interference between spatial x-y movement and pen tilt channels.

Pen tilt control with poor design suffers from interference between spatial movement and pen tilt channels. We observed that participants inadvertently moved the pen tip while trying to perform a pen tilt control task in the experiment, even though participants were instructed that pen tilt was the only factor that affected target acquisition. Many inadvertent movements would decrease accuracy in pen tilt selection tasks. Therefore, a good design for a pen tilt technique should seek to minimize interference of this kind.

- Set proper target angles to be controlled

Experimental results show that the width of range of pen tilt of more than 10 degrees can achieve a good performance while decreasing the width of range of pen tilt drastically degrades performance. Therefore, when designing pen tilt techniques, discriminations for too small a pen tilt range should be avoided.

- Afford real-time and continuous visual feedback

Although controlling pen tilt angle to perform target acquisition is achievable without continuous feedback (PRV condition), pen tilt control was consistently poor in accuracy in that condition. To advance efficiency in pen tilt techniques, full visual feedback (FLV) is necessary.

- Choose a suitable transfer function

Experimental results indicate that pen tilt control in the low range interval significantly degrades performance accuracy, which was also demonstrated by participants during the experiment. To overcome the “poor performance of low pen tilt, an appropriate transfer function is required when incorporating pen tilt into UI designs. For example the low spectrum of pen tilt can be mapped to a “dead zone to prevent its poor performance effect.

3.5 Pen Tilt Techniques

Based on our experimental results a series of interactive techniques are proposed to demonstrate the potential of pen tilt to enhance pen-based interactions. To aid the demonstration, it is useful to analyze key factors relevant to pen tilt control in such techniques.

Pen tilt control manners: In views of operation manners pen tilt can be used to do two types of tasks: discrete or consecutive variants to control attributes of an object correspondingly, e.g., discrete selection, like choosing a color from a list or pie menu, and continuous parameters, like varying brush size.

Pen tilt mapping: In designing pen tilt technique we first consider what visual attribute of a technique the pen tilt channel is to map. In our experiment, the set of sector targets was fixed in space, while pen tilt controlled the pink cursors angle. In other words, the pen tilt channel was coupled to the cursors angle. Generally, pen tilt can be mapped to displacement if the pen tilt is translated into changes in x-y coordinates; angle if it translates into changes in angle or orientation; scale if it is translated into changes in size or scale of cursor or target.

Elements to be controlled: In graphic user interfaces (GUIs), targets and cursors are the distinctive elements in performances. A pen tilt technique needs to operate on these two basic elements: targets and cursors.

According to these key factors: Pen tilt controlling (discrete and consecutive), Pen tilt mapping (displacement, angle and scale) and Elements to be controlled (targets and cursors) a design taxonomy is created, which describes the nature of our proposed pen

Table 3.1 A taxonomy for the design of pen tilt techniques. "D" means pen tilt is coupled to displacement; "A" means pen tilt is coupled to angle; "S" means pen tilt is coupled to scale.

	Tilt control manner	
	Discrete	Consecutive
Target	Angle List Menu (D) Angle Pie Menu (A) 3D Object operation (D+A+S)	Angle Slider (S) 3D Object operation (D+A+S)
Cursor	Angle Marking Menu(S) Magic Pen (S)	Projected Cursor (S)

tilt techniques (see Table3.1).

In designing pen tilt based techniques we use a pen tilt cursor (Fig.3.8), instead of the default cursor found in most GUIs. Our pen tilt cursor provides users with a real-time indicator (the appearance of a red pen) of the pen tilt value they are applying with the input transducer. In pen tilt cursor a series of radial concentric dashed lines show the pen tilt scales. The pen tilt cursor is similar to the setup of our experimental tasks. The rotation direction of the pen tilt indicator follows that of the pen (in other words, in the pen tilt cursor, pen tilt is coupled to cursor angles). Pen tilt cursor is more intuitive because the pen tilt cursor can map the physical stylus naturally. This shows that pen tilt can be one of the S-R (Stimulus-Response) compatible [33] solutions in UI design.

Angle List (or Pie) Menu (Fig.3.9a): the first example is a menu interface. We

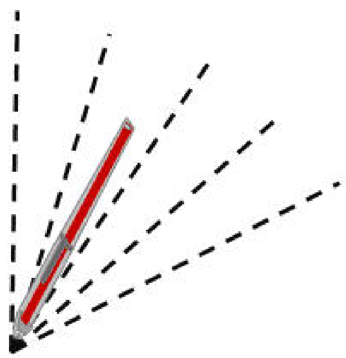

Fig. 3.8 The pen tilt cursor.

implemented two variations, the list menu and the pie menu. The basic interactive mechanism is the same: the user first presses the pen-tip and adjusts pen tilt between the pen and the screen and this moves a menu item. For the list menu, the menu item is moved by the pen tilt; for the pie menu, the menu item is rotated by the pen tilt. When an item is highlighted (pre-selected), a suitable preselected method is used to confirm this selection. With pen tilt, the movement of targets does not require any motion of the pen-tip. This attribute is highly suited to pen-based devices with small screens (e.g., PDAs, and mobile phones).

Angle Slider: Slider is a graphical widget in a GUI with which a continuous value is achieved by dragging a thumb for manipulations such as scrolling or zooming. Angle Slider uses pen tilt to map different scales of a parameter space, while dragging a thumb, to allow highly precise parameter manipulation (see Fig.3.9b). In other words, the scale factor of Angle Slider is adjusted by the tilt of the pen, and the pen-tip x-y position determines the parameters value. There are many scenarios where users may need to fluidly adjust the scale of the parameter space in order to make precise parameter adjustments. For instance, the user may need to achieve navigation tasks (through a map) both at a very large and at very fine scales. This scale can easily be varied by a users adjustment of the pen tilt value.

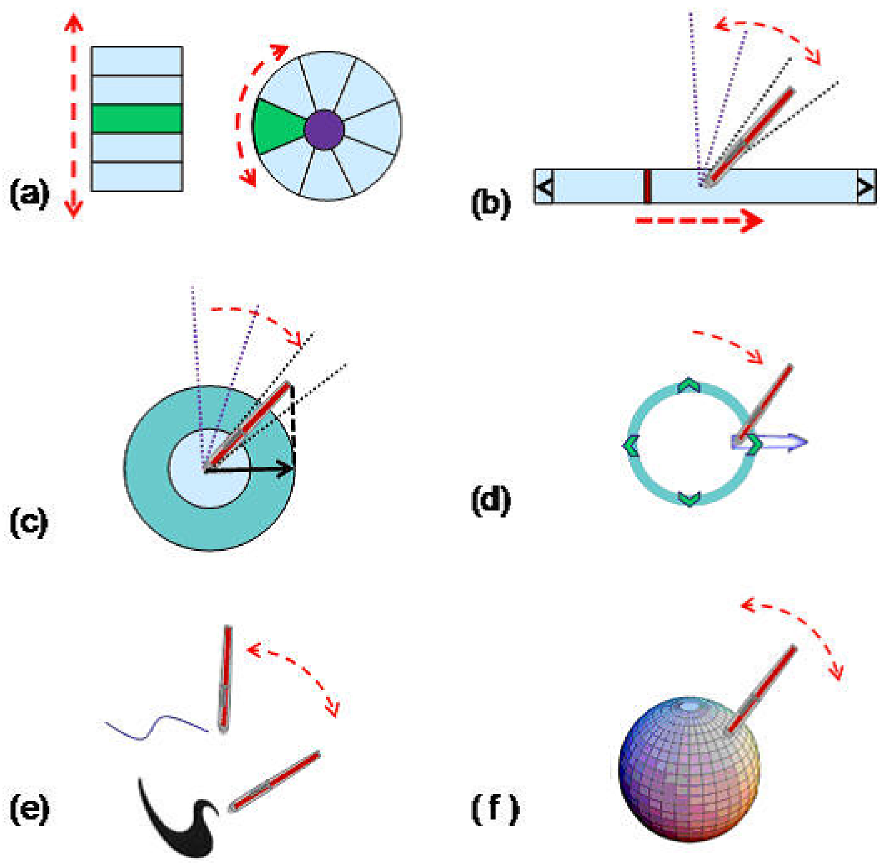

Fig. 3.9 The design for pen tilt techniques. (a) List or Pie Menu. (b) Angle Slider. c) Projection Cursor (Target). (d) Angle Marking Menu. (e) Magic Pen. (f) 3D Operations.

Projected Cursor: Projected Cursor maps pen tilt to its cursor size. The lower the pen tilt value, the bigger the cursor size, which follows the change principle of projection of a pen length (this means when the pen tilt becomes lower the projection of the pen length becomes longer. The cursors size becomes bigger). The cursor with zoomable size can be utilized to select one or multiple targets, or to specify an area. A similar mechanism can be used for target manipulation (see Fig.3.9c).

Angle Marking Menu: this maps pen tilt to different discrete states to extend the number of items available on regular marking menus. Instead of having one option available at a given path, Angle Marking Menu has two or more, depending on the pen tilt difference between the starting and ending points of the marking gesture. Angle Marking Menu has another version, if additionally inducing the azimuth of pens, which couples the azimuth to marks in different paths. This version does not need any sliding movement of the pen-tip (see Fig.3.9d).

Magic Pen: in GUIs some tasks require more than one step to be implemented with different operation methods. If a seamless switch mode is available, two or more techniques can be coupled together. Pen tilt has the potential to be used as a switch mode method. Magic Pen couples two types of painting together (hard pen and soft pen). In high pen tilt, hard pen (where the width of stroke is consistent.) works, while in low pen tilt, the soft pen (where the width of strokes can be changed by pressure) works (see Fig.3.9e).

3D Object Operation: In this example, the user can operate or inspect a 3D object model by controlling pen tilt. While changing pen tilt, the 3D object is rotated. By coupling with azimuth, users can see 3D models from different directions just like inspecting a small object in their hand (see Fig.3.9f).

3.6 Conclusions

Pen-tip pen tilt input has been utilized in some applications of pen-based computing devices. Taking advantage of this additional channel of information control in user interface design is an attractive option. But no empirical study of the human ability to control pen tilt in pen-based interfaces has been conducted previously. The current work is one systematic attempt at filling such a void.

We have presented an experiment that investigated the human ability to use pen tilt to perform discrete target selection tasks, with five different selection techniques. Experimental results show that the NonPrefHand selection technique was preferable overall. With the (target angle) decreasing the selection time significantly increases in both FLV (Full Visual Feedback) and PRV (Partial Visual Feedback) conditions. Selection time could be modeled by a mathematical model (see Equation 1). The width of range of pen tilt of more than 10 degrees can achieve a good performance. Appropriate visual feedback was also found to be beneficial for pen tilt techniques.

Based on the results of our experiment, we have inferred design recommendations and proposed a design taxonomy of pen tilt techniques. Some possible designs of pen tilt techniques were also presented.

3.6 Conclusions

This is a blank page.

Chapter 4

ZWPS and Pressure Scroll: Two Pressure-based Techniques in Pen-based Interfaces

4.1 Introduction

In traditional GUIs many computing performances such as selection tasks and scrolling tasks are commonly driven by coordinate (x, y) input from pointing devices (e.g., Mice), binary buttons or Wheels. Some pen-based input devices provide additional inputs such as pressure, which can obviously widen the input bandwidth. Some fundamental studies have proved that pressure affords significant benefits to computer interactions [21] [44] [94]. However, with few exceptions [93] [94] [95] pressure is seldom coupled with current interaction techniques. Pressure can be typically used to perform interactions by mapping it into several discrete states, or by controlling a continuous variable. Investigation of the potentials of these two use manners of pressure for interaction techniques should be performed. Therefore, this study seeks to look at two novel techniques which incorporate pressure to enhance user performance and preference

First, we propose a hybrid selection technique coupled with a standard Point Cursor and a zoomable technique, using pressure as the switch mode, to enable a quick pixel-level target selection. ZWPS employs the discrete state of pressure to widen the performance of target selection. It has two major advantages: first, it zooms only the specified area to facilitate pixel-level targets, without breaking overview information in interfaces; second, it maintains standard pointing manners for selecting normal or big targets. We also propose a new scrolling technique, Pressure Scroll, which we developed to facilitate scrolling tasks in pen-based interfaces. Pressure Scroll uses the continuous variable of pressure to enhance scrolling performance and utilizes strokes (circular or linear) to control scrolling velocity and variable pressure is utilized to further finely adjust the scrolling velocity.

In the following sections, we will review related work; discuss the design and implementation of ZWPS and Pressure Scroll; evaluate the performance of these two techniques in two experiments; and conclude by discussing implications for user interface design.

4.2 Related Work

The related work to this study includes pen pressure researches, precise selection techniques and scrolling techniques.

4.2.1 Related Work on Pressure

Studies on pressure can be roughly divided into two categories. One category investigates the general capabilities of humans to interact with computers using pressure. For example, Herot and Weinzapfel [44] investigated the human ability of the finger to apply pressure and torque to a computer screen. Buxton [21]studied the use of touch-sensitive technologies and the possibilities for interaction they suggest. Ramos et al. [94]

explored the human ability to vary pen-tip pressure as an additional channel of control information.

The other category of study is where researchers build pressure enabled applications or techniques. For instance, Ramos and Balakrishnan [93] demonstrated a system called LEAN and a set of novel interaction techniques for the fluid navigation, segmentation and annotation of digital video. Ramos and Balakrishnan [95] designed Zlider widget that used pressure to produce different scales to achieve high precision parameter manipulation. Li et al. [67] investigated the use of pressure as a possible means to delimitate the input phases in pen-based interactions. Although these works opened the door to establish pressure as a research avenue, we are unaware of any work which addressed the issue of applying pressure in selection or scrolling techniques. Thus, we attempt to investigate these potentials in this chapter.

4.2.2 Related Work on Precise Selection Techniques

On many occasions there are requirements for precise selection such as searching for a location in a map application. This issue was notably addressed by Sears, Shneiderman and colleagues [90] [101] [103]. Their basic technique, called Take Off, provides a cursor above a pen-tip or the user's finger tip with a fixed offset when touching the screen to achieve precise selection. Zoom pointing is also a typical technique for small target selection, which is currently used in many painting systems. Ren and Moriya [99] investigated different strategies for handling small targets and reported that 1.8 mm (5 pixels) was a crucial limit beyond which special needs arise. Worden et al. [118] proposed an enhanced Area Cursor to alleviate the ambiguity, by including a single point hotspot centered within the area cursor, which took effect when more than one target was within the cursor's bounds. The enhanced Area Cursor performed identically to regular point cursors when targets were close together, and it outperformed point cursors when targets were far apart. Ramos et al. [96] presented pointing lenses techniques to help users easily select targets by offering them an enlarged visual and interaction area. Experimental results showed that Pressure-Activated Lens was the top overall performer and all activation lenses had benefits for selecting targets of less than 5 pixels. Albinsson and Zhai [7] proposed Precision-Handle and Cross-Keys to complement existing techniques for touch screen interaction. We present ZWPS which integrates two techniques (Point Cursor*1 and the zoomable technique) by using pressure as the switch mode to allow both precise and imprecise selections.

4.2.3 Related Work on Scrolling Techniques

Scrolling is also a fundamental task, particularly for devices with small screens. Igarashi and Hinckley [50] proposed a SDAZ (speed dependent automatic zooming) technique. When using it to scroll documents, the documents are automatically zoomed-out as the scroll rate increases. By automatically zooming, the visual flow rate is reduced enabling rapid scrolling without motion blur.

There are efforts that explore navigation techniques in mobile devices with small screens. Baudisch et al. [12] proposed a Collapse-to-Zoom technique which exploits strokes to delete the unimportant contents and highlight the selected ones to enhance navigation performances. MacKay et al. [70] presented a field study comparing software-based navigation techniques (Scroll Bar, tap-and drag, and touch-n-go) on mobile devices. They explored the efficiency and user preferences of these navigation techniques for different levels of mobility (sitting, walking and standing). The works that are more related to ours are the Smith et al. [104]and Moscovich et al [**?**]. Smith et al. presented a radial scrolling tool, which exploits circular gestures in the interface to scroll docu-

*1 The standard regular cursor in GUIs.

ments, advancing or reversing the document by clockwise or counterclockwise circular stroke gestures. Moscovich et al. presented a virtual scroll ring which simulates the hardware scroll ring device that maps circular finger motion into vertical scrolling, to navigate the document. Both of these utilize arc strokes to perform scrolling tasks. Our Pressure Scroll introduces pressure as an additional scale trigger to advance scrolling performance, which is simultaneously controlled by strokes (both arc and line).

4.3 Pressure-based Techniques

Pessure is a continuous scalar value acquired form the interaction of pens and screens. Generally, during a computer interaction a desired object can be controlled by two basic operation modes: using a continuous value to adjust its attributes (e.g., shape, position or color lightness) and using discrete values to change its states (e.g., highlight or not). Correspondingly, there are two fundamental controlled manners for pressure to map the above operation modes respectively: mapping pressure into a consecutive value, or mapping pressure into discrete states in a performance. Based on the above demonstration this study explores the potentials of incorporating pressure with computing interactions by proposing two novel techniques: ZWPS and Pressure Scroll. The following sections describe these two techniques in detail.

4.3.1 Zoom-based Selection Technique with Pressure (ZWPS) for Pixel-level Targets in Pen-based Interfaces

Pen-based devices (e.g., Tablet PCs) can be directly manipulated by a pen, even by a bare finger. When targets are very small, especially at pixel-level, such target selections become very difficult for a pen or a finger to perform. In current applications (e.g., Photoshop) such problems are resolved by zooming. For example, drawing a precise line from one dot (e.g., railway station) to another (e.g., a college) on a digital map requires the user to zoom in and click on one end, zoom out to find the other end, zoom in again to click on the other end, then zoom out again to look at both ends, all of which cause great inconvenience to users. Moreover, after zooming in, the user may lose information in the overview and the detailed windows. The switch between zooming in and zooming out often interrupts the user's attention. This awkward situation becomes our direct motivation to study alternatives to the current technique.

We seek to find a selection technique that should satisfy the following conditions:

- Have a zoomable function to allow a pixel target selection.
- Provide a flexible and seamless switch mode between zooming in and out.
- zoom only a certain part of the display area to prevent the user from losing the information in overview.

Therefore, we present a novel technique called ZWPS to enhance pixel-target selection. ZWPS, in essence, is a hybrid selection technique that includes two selection techniques: the standard Point Cursor and the zoomable technique. The standard pointing selection mode works in selecting common size targets while zoom selection mode works best when selecting small size targets.

To couple the two techniques in ZWPS a suitable switch mode is essential. The properties of pressure (easily producing discrete controlled states) make it competent to seamlessly switch modes in ZWPS. To determine a proper switch threshold, a pilot study with 10 subjects was performed. The task was to draw freehand strokes (arbitrary curves and straight lines), basic geometrical graphs (such as rectangles and circles) and a mixed set of Roman, Japanese, and Chinese (kanji) characters and signatures on a blank space in a natural manner. Pen-tip pressure was recorded in a 17ms sampling periods. Ninety-five percent of the force samples fell within the 210 to 810 units range.

The results showed that pressure levels of more than 810 units were seldom used in a natural manner. Therefore, in ZWPS the threshold value is set at 970 units.

An operation circle (see Fig.4.1a) is also defined to specify a zoomed area. When a user lands a pen tip on a screen surface an operation circle appears. The pen tip is always at the center of the operation circle. When the user slides the pen tip on the screen surface the operation circle follows it. Contents that are enclosed by the circle can be zoomed in by imposing heavy pressure, enough to surpass the switch threshold. The diameter of the operational circle was set at 24mm by pilot studies. An arc with a dotted red line, which is attached on the border of the operational circle, is employed as visual feedback to indicate the current pressure state. When the arc, augmented by pressure from a pen, increases to a complete circle (surpassing the threshold) it means that the zoom technique is activated, Conversely, ZWPS still maintains the standard Point Cursor when effective pressure is not applied. When more than one target is enclosed by the operation circle, they are zoomed in or out together simultaneously. The parts of the zoomed contents that are beyond the border of the operation circle cannot be shown in the operation circle (see Fig.4.1c). If the desired target is not included in the operation circle the user can move the circle to include it in the operation circle by sliding the pen tip. In our current implementations we used a zoom factor of 4. A very small zoom factor value cannot offer significant benefits for small target selections, while a big value will enable little content to be shown in the operation circle. We found through early pilot studies that a value of 4 resulted in acceptable performance.

When selecting a target of normal or big size, the user, imposing light pressure on a pen-tip, taps the target directly to select it in the standard selection mode. When selecting a very small target, the target is first enclosed (Fig.4.1a and 4.1b) by the operational circle, and then the zoom selection mode is activated by heavy pressure (surpassing the specified threshold) to enlarge the target. When the user slides the pen

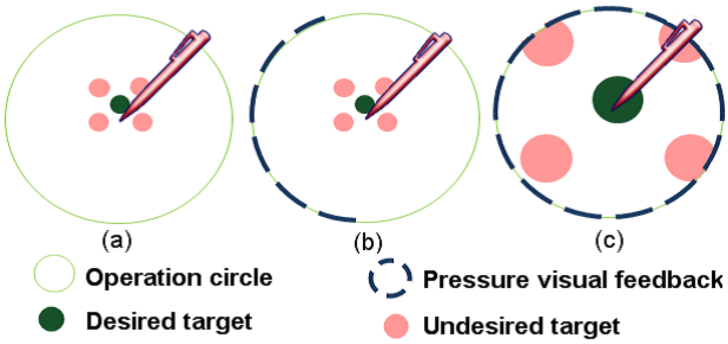

Fig. 4.1 (a) the operation circle appears when the pen tip is landed on the screen. (b) a dotted arc is used to show pressure amount as visual feedback. (c) the zoomable function is activated by pressure that surpasses a specified threshold.

tip into the desired target it is highlighted (preselected) (Fig.4.1c). When the pen tip is lifted from the screen, the target is selected and it recovers to its original size. Note that ZWPS only zooms in the operation circle area around the cursor, not the whole area in the view window, which enables the user to avoid losing the information in overview, even when it is in zoomed status.

4.3.2 Pressure Scroll

Scrolling tasks are also very common operations. A scrolling technique that possesses speed and accuracy (i.e., quickly find the object in a document and accurately relocate it in a desired position) is a good one. In traditional GUIs Scroll Bar is a classic widget which is used to navigate a document. In pen-based systems, Scroll Bar is still used even though some of its characteristics are not suitable for pens. Some researchers propose some new scrolling techniques that use circular stroke to achieve a good scrolling performance for pen-based systems [104] [?]. We also found that a line stroke is good to fluently perform scrolling tasks. In Pressure Scroll we explore the possibility of integrating pressure with the stoke-based scrolling technique. Pressure is utilized as an additional control factor to widen the adjustable range of scrolling velocity (e.g., generating a very fine or a very large scale).

We first illustrate how to achieve scrolling performance only by arc and line strokes. In our implementations, when an arc stroke is drawn to drive scrolling tasks, the coordinates of all points contained in strokes are recorded to determine the direction and the displacement of the scrolled document. A minimum of three points, A, B, and C, are required (as shown in Fig.4.2a). Two vectors indicate the previous and current points passed by the pen. The scrolling direction is determined by the sign of the dot product of the previous vector $\overrightarrow{\mathbf{ab}}$ and the current vector $\overrightarrow{\mathbf{bc}}$. If the sign is positive the direction of rotation is clockwise, and the movement direction is up. The angle difference θ between the two vectors determines the scrolling displacement.

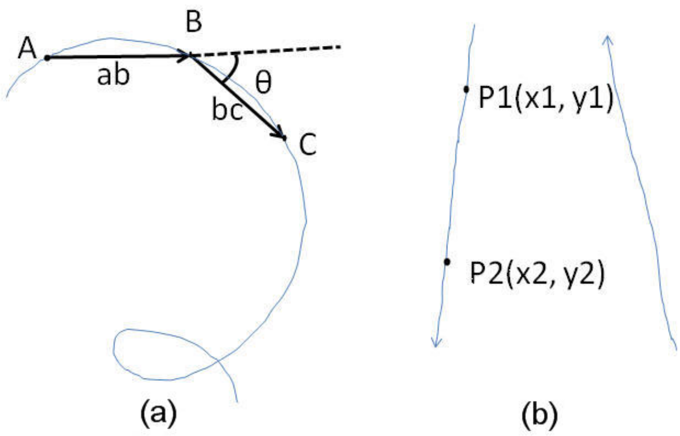

Fig. 4.2 (a) Arc: to scroll a document using an arc stroke; (b) Line: to scroll a document using a line stroke.

When a line stroke is drawn to drive scrolling tasks, as shown in Fig.4.2b, P_1 and P_2 indicate the previous and current points passed by the pen. The sign of $(y_2 - y_1)$ (i.e., direction of the line stroke) determines the scrolling direction of the document: if the direction of the line stroke is up, the document will scroll up, while drawing the stroke in a downwards direction causes the document to scroll down. For vertical scrolling, the vertical difference ($|y_2 - y_1|$) between the previous and current points determines the scrolling displacement; for horizontal scrolling the horizontal difference ($|x_2 - x_1|$) determines the displacement. Moreover, the line stroke is a much more natural manner and has the advantage of visual consistency (i.e., the direction the stroke is consistent with the scrolling direction of the document). We can calculate scrolling velocity produced only by strokes (hereafter referred to as $\mathbf{V}_s$) according to the scrolling displacement and the time that is spent to produce this displacement.

In Pressure Scroll, scrolling velocity (i.e., $\mathbf{V}_{scroll}$) is controlled by two factors: strokes and pressure. As shown in Formula 1, $\mathbf{V}_s$ is the velocity produced only by strokes. We use an exponential function to calculate the scale factor by pressure, where $f(p)$ is a mapping function of the stylus' reported pressure at a particular time.

$$\mathbf{V}_{scroll} = \mathbf{V}_s \cdot e^{f(p)} \tag{4.1}$$

Here, the pressure mapping function $f(p)$ is described. As found by previous work [14], the degree of pen pressure perceived by human users is not consistent with that sensed by a digitizer. For example, at a low spectrum of pen pressure, the sensed pressure value increases much faster than users would expect. Previous work [14] has used a sigmoid transfer function to account for the effects produced by pressure. In our techniques we also employed the sigmoid transfer function. A number of aspects in the application of pressure have been identified: an initial "dead zone", slow response at low pressure levels (where pressure is too sensitive for users to control) smooth change at median pressure levels (where users have good control of pressure), and quick response at high pressure levels (where force applied by the user can produce tremors, causing sudden pressure variations). We employed a piecewise linear function to approximate the pressure mapping function $f(p)$.

4.4 Experiment 1: ZWPS

In Experiment 1, to evaluate the performance of ZWPS, we conducted a quantitative experiment to compare it with Take Off and with the improved Area Cursor. We chose Take-Off, a promising technique, as the baseline, because it is a very common baseline in many other evaluations of techniques for small-target selection reported to date [90] [101] [103]. Area Cursor has a larger than normal activation area. This characteristic enables more efficient target acquisitions. However, there are potential problems associated with the implementation of Area Cursor. For example, when more than one target is within the activation hot spot, it is difficult for Area Cursor to perform selection tasks. Therefore, we employed the improved Area Cursor devised by Worden et al. [118]. As discussed in Section 2.2, even with the enhancement suggested by Worden et al. [118], Area Cursor does not deliver any benefits when targets were close together (Area Cursor simply behaves as a point cursor in this situation). We wondered if ZWPS offers advantages in selecting small targets with the closed distracter targets. We, therefore, employed virtual target width (VW) (as shown in Fig.4.3) and designed the different VW/W (2, 3 and 7) ratios as the setting of the experimental design. The virtual target reflects the amount of void space immediately around a target, which, in our experimental settings, is a square and is determined by four distracter targets placed on four vertexes of the square. In Experiment 1, the operation circle of ZWPS was set at 81 pixels (about 24 mm) and the zoom factor value was set at 4. The width size of area cursor was 12 pixels, which followed Worden et al.'s design on Area Cursor.

4.4.1 Apparatus

The hardware used in Experiment 1 was a Wacom DTI-520 interactive LCD graphics display tablet with a resolution of 1024 x 768 pixels (1 pixel = 0.297 mm), using a wireless stylus that has a pressure sensitive isometric tip (the width of the pen-tip is 1.76mm). It reports 512 levels (ranging from 0 to 1024, the minimum unit is 2) of pressure and has a binary button on its barrel. The experimental software ran on a 3.2GHz P4 PC with the Windows XP Professional operating system.

4.4.2 Participants

Twelve subjects (10 male and 2 female) all with previous experience using computers were tested for the experiment. The average age was 23.9 years. All subjects used the pen in the right hand.

4.4.3 Procedure

The experimental task was a reciprocal pointing task in which subjects were required to select two fixed targets back and forth in succession, but, to simulate a more realistic two dimensional pointing environment, we used a multi-directional reciprocal pointing task which included reciprocal diagonal movements. The targets were drawn as solid circles, and were located at various distances from each other along two directional axes. The goal target, the one intended to be selected, was colored green. When a goal target had been selected, it changed color to red which was an indication that the user now had to select the next goal target. Four red circles were placed around each goal target to control the VW/W ratio (see Fig.4.3).

Subjects were instructed to select the two goal targets alternately. They were told to emphasize both accuracy and speed. When the subject correctly selected the target, he/she heard a beep sound and the targets swapped colors, which was an indication of a new trial. At the start of the each experiment, subjects were given a warm-up block to familiarize themselves with the task and the conditions.

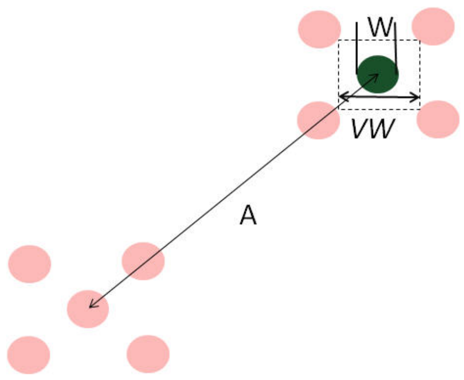

Fig. 4.3 Experimental setup. The green target in the center around four red circles is the goal target. The four red circles around the goal targets are distracters which determined the VW/W ratio.

4.4.4 Design

A within-subject design was used. The independent variables were: selection techniques ST (ZWPS, Take Off and the improved Area Cursor), amplitude A (100, 250, 750 pixels), width W (2, 4, 8 pixels), VW/W ratios (2, 3, 7), and direction DR (2 diagonals). A full crossed design resulted in 162 combinations of ST, A, W, VW/W, and DR. Each combination consisted of 4 selection attempts (i.e., 3 reciprocal movements between the two targets). Each subject performed the experiment in one session lasting approximately forty minutes, depending on each subject's proficiency in selecting the targets. The session was broken up according to selection techniques.

4.4.5 Results

An ANOVA (analysis of variance) with repeated measures was used to analyze performance in terms of selection time, error rate, and subjective preference. Post hoc analysis was performed with Tukey's Honestly Significant Difference (HSD) test.

Selection time

There was a significant difference in the mean selection times among the three selection techniques, F(2,33)=7.95, p<.01. The overall mean selection times were 1589 ms for Area Cursor, 1929 ms for Take Off and 1288 ms for ZWPS. Tukey HSD tests showed that ZWPS was significantly faster than both Area Cursor and Take Off (p<.05). No significant difference was found between Area Cursor and Take Off.

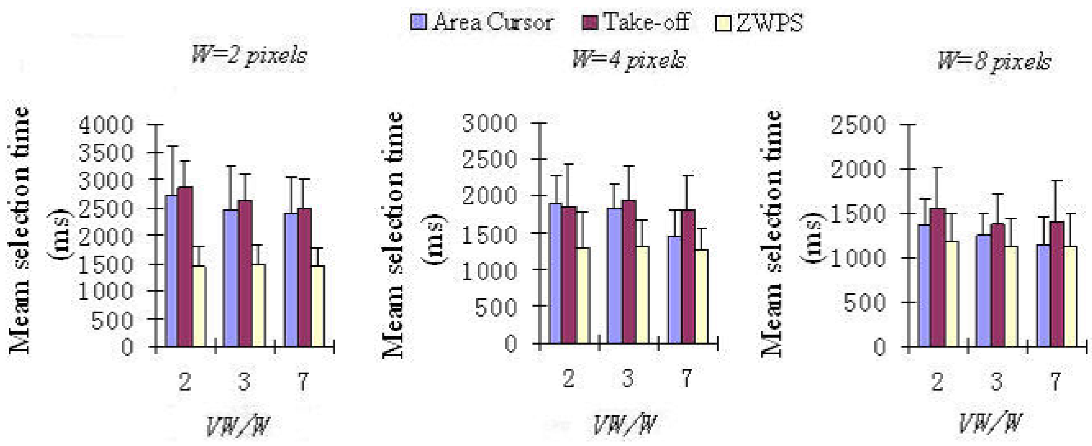

Fig. 4.4 The mean selection time for different sizes of targets at mean selection time for different sizes of targets at each VW/W ratio.

As shown in Fig.4.4, at the width of 2 there was a significant difference in selection time between the three selection techniques, F(2,33)=13.9, 11.8 and 12.9 for the VW/W ratio values of 2, 3 and 7 respectively, all p<.001. For each VW/W ratio value Tukey HSD tests showed ZWPS was significantly faster than Area Cursor and Take Off (p<.001), however, no significant difference was found between Area Cursor and Take Off. At the width of 2, for each VW/W ratio, there was a little void space around the goal target. ZWPS still delivered benefits for such selection tasks by enlarging goal targets, while Area Cursor had no advantages in such situations because it too easily covered more than one target (Area Cursor simply behaves as Point Cursor in this situation) at the same time. Take Off was also worse than ZWPS in selecting very small targets.

At the width of 4 there was a significant difference in selection time between the three selection techniques, F(2,33)=5.89, 9.41 and 6.19 for the VW/W ratio values of 2, 3 and 7 respectively, all p<.001. For VW/W ratio values of 2 and 3, Tukey HSD tests showed ZWPS was significantly faster than Area Cursor and Take Off (p<.05), however, no significant difference was found between Area Cursor and Take Off. The results followed the trends similar to selecting small closed targets (at the width of 2). For the VW/W ratio values of 7, Tukey HSD tests showed ZWPS was significantly faster than Take Off (p<.01), however, there was no other significant difference across selection techniques. Although targets were still small the void space around targets became bigger. Area Cursor began to show its benefits. ZWPS was comparable to Area Cursor.

At the width of 8 there was no significant difference in selection time between the three selection techniques for each VW/W ratio value. When targets and their surrounding void space both became bigger, the three techniques were comparable.

Fitts' law [34] is commonly used to predict the time it takes to move a mouse pointer from one location to another.

$$MT = a + b\log_2(\frac{A}{W} + 1) \tag{4.2}$$

According to Fitts' law, the cursor movement time (MT) increases linearly with the Index of Difficulty (ID), which relies on the logarithm of the distance moved (the amplitude, A) and the width of the target (W). The two constants, a and b, are determined empirically.

We applied Fitts' law to evaluate the experimental results. For ZWPS and Area Cursor we used the effective width (e.g., the effective width= the zoomed width for ZWPS) instead of physical width. As seen in Fig.4.5 ZWPS was modeled by Fitts' law (R^2 =0.98), which indicated that the zoomed size of targets could be utilized fully.

Area Cursor and Take Off were only roughly modeled by Fitts' law, R^2 =0.82 for Area Cursor and R^2 =0.73 for Take Off. The relatively low R^2 for Area Cursor and Take Off may be due to their special selection mechanism and the complex selection tasks in Experiment 1.

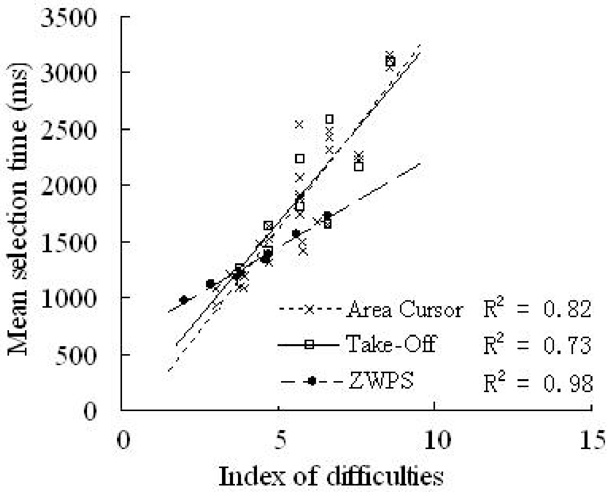

Fig. 4.5 Line regression of index of difficulty against selection time.

Error rate

There was a significant difference in overall mean error rate between the three techniques, F(2,33)=55.7, p<.0001. Tukey HSD tests showed ZWPS was significantly more accurate than both Area Cursor and Take Off (p<.001) in error rate. Take Off was significantly more accurate than Area Cursor (p<.001). Overall error rates were 31.8% for Area Cursor, 12.6% for Take Off, and 5.5% for ZWPS.

At the width of 2 there was a significant difference in error rate between the three selection techniques, F(2,33)=40.4, 63.2 and 45.5 for the VW/W ratio values of 2, 3 and 7 respectively, all p<.0001. For each VW/W ratio value, Tukey HSD tests showed ZWPS was significantly more accurate than both Area Cursor and Take Off (p<.001) in error rate. Take Off was significantly more accurate than Area Cursor (p<.001).

At the width of 4 there was a significant difference in error rate between the three selection techniques, F(2,33)=12.4 and 15.6 for the VW/W ratio values of 2 and 3 respectively, both p<.0001. No significant difference was found for the VW/W ratio values of 7. For the VW/W ratio values of 2 and 3, Tukey HSD tests showed ZWPS and Take Off were both significantly more accurate than Area Cursor (p<.001) in error rate, however, there was no significant difference between ZWPS and Take Off.

At the width of 8 there was no significant difference in error rate between the three selection techniques for each VW/W ratio value.

From accuracy analysis it was shown that ZWPS significantly outperformed Area Cursor and Take Off for small targets close together, which reflected the effectiveness of its zoomable function.

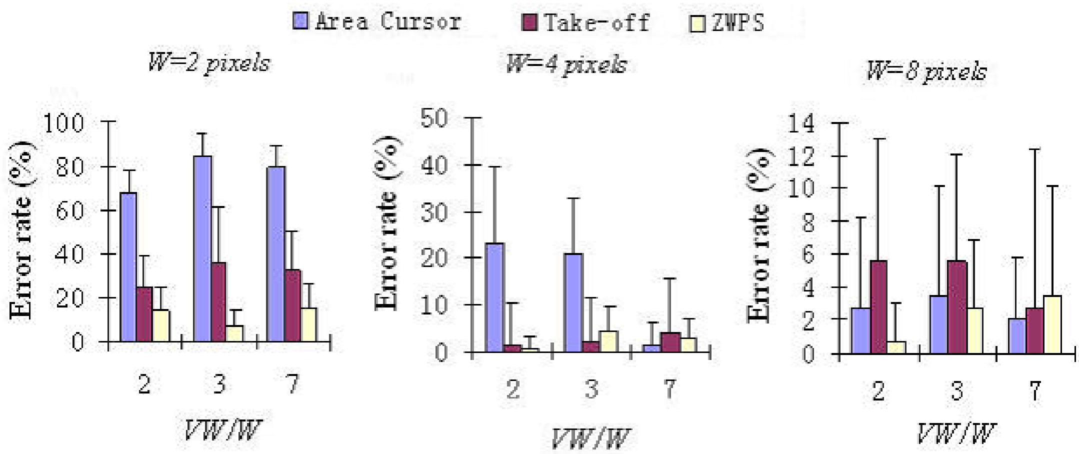

Fig. 4.6 The error rate for different sizes of targets at each VW/W ratio value.

Subjective preference

The three techniques were rated by subjects on "preference" with 1-to-7 scale (1=lowest preference, and 7 =highest preference). There was a significant difference between selection techniques in subjective preference, F(2,33)=31.2, p<.0001. Subjects gave ZWPS (mean = 6.61) a significantly higher rating than Area Cursor (mean = 3.13) and Take-Off (mean = 4.25). Take Off was more preferred than Area Cursor. Subjects preferred ZWPS because ZWPS enables the user to easily select targets by zooming them, especially for very small (pixel level) targets with little void space.

4.5 Experiment 2: Pressure Scroll

The purpose of Experiment 2 was to examine whether pressure, as an additional control parameter, deliver benefits to stroke-based scrolling techniques. The stroke-based techniques with pressure control and without pressure control were compared to achieve the experimental aim. Although Scroll Bar is a standard technique in current GUIs our experiment did not involve this technique. The reason is that Scroll Bar is designed for mice and the result of our pilot studies showed that the stroke-based techniques without pressure control were better than Scroll Bar in pen-based interfaces.

4.5.1 Participants

Ten subjects (9 male, 1 female), all with previous experience with computers were tested for the experiment. The average age was 21.6. All subjects were right handed, and used the pen in the right hand.

4.5.2 Apparatus

The hardware used in Experiment 2 was the same to that in Experiment 1.

4.5.3 Design and Procedure

We conducted the experiment to compare stoke-based techniques with pressure to stroke-based techniques without pressure. a vertical scrolling task was tested in

Experiment 2. Closely following Hinckley et al. [48] we evaluated our technique using a reciprocal framing task. In the vertical scrolling task, subjects scrolled down, then scrolled up, moving back and forth between two lines that are marked “START” or “END” respectively. We colored the “START” target line green and the “END” target line red. In Fig.4.7 the red “frame” at the left of the experimental interface specified the certain width. For each target, subjects scrolled until the target entered the range of the screen identified by the frame. The frame was always centered on the screen. Once the target line was fully within the frame, the subject hit the any key with the left hand. This selection key let the experimental interface know when the user judged the scrolling to be complete. If the target line successfully fell within the frame when the user struck the selection key, there was a short beep sound. If not, the user did not hear any sound, but we had instructed subjects to always continue to the next target (rather than trying to repair the error).

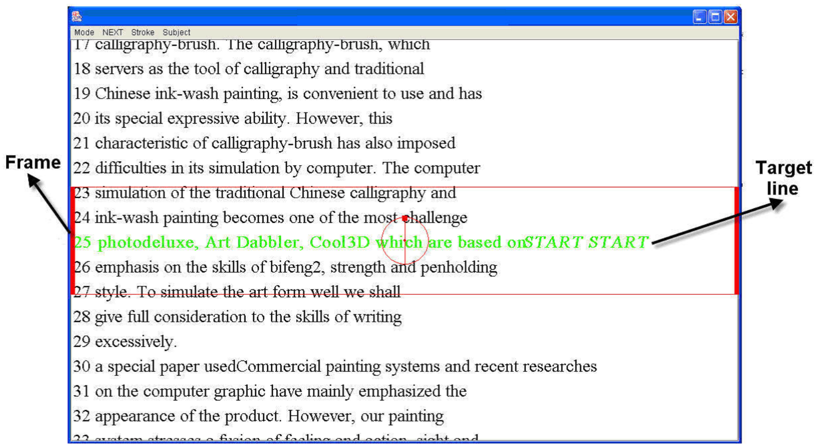

Fig. 4.7 The experiment interface of the reciprocal framing task.

The design of the experiment crossed Method x scrolling Distance (D) x target Width (W). We chose six representative distances: 48, 72, 96, 256, 416 and 570 lines and one frame heights: 3 lines. Each line is about 33 pixels (10mm).

Participants read instructions describing the experimental setup and task. The experimenter then reviewed these instructions with the subjects and introduced the three methods. In order to familiarize them with the experimental environment, we set a practice session for each method. When the experimenter was satisfied that subjects understood the task and could perform it correctly; subjects completed the experiment unassisted. For each method, subjects performed trials for each of the distance-width combinations in a random order. Each trial consisted of 7 phases of reciprocal movement between target lines. Participants typically spent 40 to 50 minutes using the two methods.

4.5.4 Results

There is a line model [8]for movement time MT in scrolling:

$$MT = a + bD \tag{4.3}$$

where a and b are empirically determined constants, and D is the distance (lines) between two target lines. We use this model to analyze the experimental results.

Arc stroke

A repeated measures analysis of variance showed a significant main effect for the pressure means (hereafter referred to as Arc-with-pressure) with non-pressure means (hereafter referred to as Arc-without-pressure), F(1,119)=25.76, p<.01. There was a significant difference between the six distances, F(5,119)=43.11, p<.001. Separate paired samples t-tests were then run for each of the six distances, comparing the scrolling times for the pressure means and the non-pressure means.

For 256, 416 and 570 lines there were significant differences (256: t[9]=3.46, p<.05; 416: t[9]=2.66, p<.05; 570: t[9]=4.11, p<.05). However, for 48, 72 and 96 there was no significant difference (p>.05) (see Fig.4.8). The error rates were pressure means, 3.91%; non-pressure means, 3.0%. There was no significant difference between the two methods in error rate.

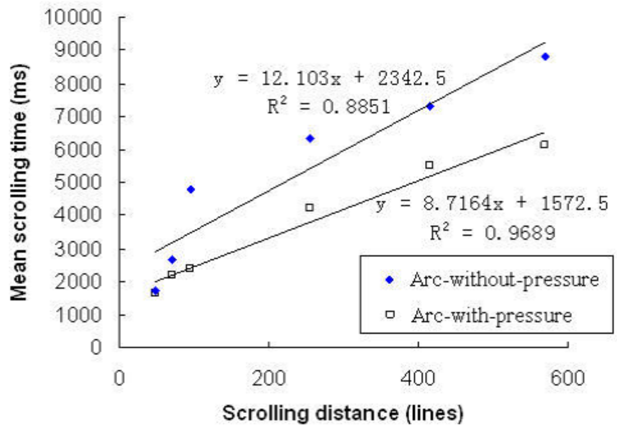

Fig. 4.8 Arc: mean movement time by scrolling distance for pressure mode and non-pressure mode.

Line stroke

A repeated measures analysis of variance showed a significant main effect for the pressure means (hereafter referred to as Line-with-pressure) with non-pressure means (hereafter referred to as Line-without-pressure), $F(1,119)=18.11$, $p<.01$. There was a significant difference between the six distances, $F(5,119)=33.90$, $p<.001$. Separate paired samples t-tests were then run for each of the six distances, comparing the scrolling times for the pressure means and the non-pressure means.

For 96, 256, 416 and 570 lines there were significant differences (96: $t[9]= 2.66$; 256: $t[9]=2.99$, $p<.05$; 416: $t[9]=3.17$, $p<.05$; 570: $t[9]=3.96$, $p<.05$). However, for 48 and 72 there was no significant difference ($p>.05$) (see Fig.4.9). The error rates were pressure means, 3.71%; non-pressure means, 2.83%. There was no significant difference between the two methods in error rate.

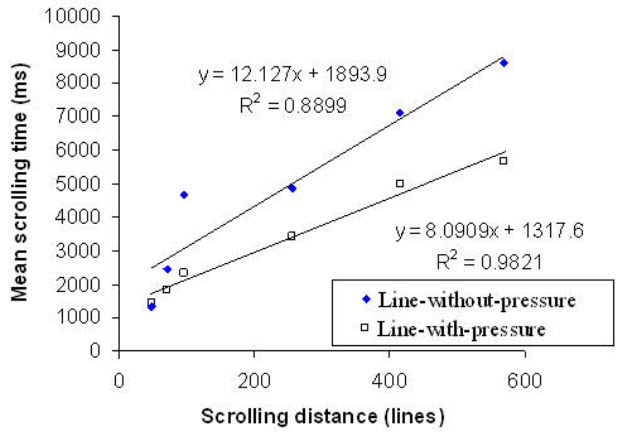

Fig. 4.9 Line: mean movement time by scrolling distance for pressure mode and non-pressure mode.

Subjective evaluation

The methods were rated by subjects on "preference" with 1-to-7 scale (1=lowest preference, and 7 =highest preference). For both arc strokes and line strokes the analysis of the questionnaire showed no significant difference on subject preference between the pressure means and the non-pressure means.

4.6 Discussion and Conclusions

To explore the potentials of incorporating pressure into interactive techniques we designed two novel techniques ZWPS and Pressure Scroll to improve precise selections and scrolling performances in pen-based interfaces.

4.6.1 ZWPS

ZWPS utilizes pressure as a switch mode to couple two selection techniques (Point Cursor and a zoomable technique) to enable it to be competent for both normal and precise selections. An experiment was conducted comparing it with Take Off and the improved Area Cursor. Experimental results indicate that ZWPS outperforms Take Off and the improved Area Cursor in small target selection performances. In particular, ZWPS still achieves good performance with small targets with little void space.

ZWPS provides a zoomable function with enlarged effective target size to easily select very small targets, even with little void space around them. The user tends to zoom in to small targets to reduce selection difficulties and visual burdens before selecting them by ZWPS. Moreover, ZWPS employs the selection manner of “Sliding into targets”, which is significantly more accurate than “Directly landing on targets” [99]. Although the improved Area Cursor has a big hot area, Area Cursor cannot take effect when selecting small targets which are close together. This is because, in such situations, the hot area of Area Cursor tends to cover more than one target and Area Cursor simply behaves as a Point Cursor. The selection manner of “Direct on” also tends to be quite inaccurate in such tasks. Take Off provides a cursor above a pen-tip with a fixed offset when touching the screen to achieve precise selection but it lacks a mechanism to reduce difficulties when selecting very small targets. The pressure activation mechanism enables ZWPS to flexibly and seamlessly switch between the zoomable function and Point Cursor, which allows ZWPS to work like Point Cursor when selecting normal sized or big targets.

Although Pointing Lenses (Pressure-activated lenses) proposed by Ramos et al. [96] is similar to ZWPS in basic mechanism, they only compared Pointing Lenses with three activation manners with Point Cursor by using selection tasks on a small single target. Comparison between Pointing Lenses and other promising cursors (like our experiment) was not conducted. Furthermore, we designed complex selection tasks in which the desired target is surrounded by distracters. Overall, we have proved benefits of ZWPS through our experiment.

In GUIs some tasks require more than one step to be implemented with different operation methods. If a seamless switch mode is available, two or more techniques can be coupled together. ZWPS is such a hybrid technique that adapts to different types of tasks. Good selection performances achieved by ZWPS suggest that pressure should be a good alternative to switch modes.

4.6.2 Pressure Scroll

Pressure Scroll introduces pressure into the stroke-based scrolling techniques as an additional control parameter. The experimental results indicate that for both Arc stroke and Line stroke the pressure manners are better than the non-pressure ones. It also indicates that pressure is a beneficial control parameter in providing fine scale and large scale adjustments of scrolling velocity.

Pressure reflects the information of the user's force imposed on pen-tips when using a pen to interact with screen. Therefore, the use manner of pressure in a performance should not allow contradicting the habits of imposing the force on the pen-tip. For an instance, scrolling performance: when fast scrolling a document, much heavy force is imposed on pen-tips. Pressure Scroll is based on this pre-existing behavior: when pressure is much heavier, the scrolling velocity becomes much higher.

4.6.3 Implications for The Pressure-based UI Design

The results of our experiments suggest several guidelines for the design of pen pressure techniques. First, our experiments have shown that pressure has the potential to advance interaction techniques. Second, Pressure in the *discrete* or *continuous* manners can be used as basic pressure control manners in interactive techniques respectively.

As demonstrated in the experimental results, these two pressure-based techniques deliver some benefits to pen-based interaction. We can develop plug-ins to incorporate them into user interfaces. This will be very appropriate to exploit “selection” mode. We can design an appropriate command to allow switching between them and regular ones. For example, we can set a command button on the taskbar; when the user wants to perform operations by them, he/she would just click the command button to activate them. Or it can be activated by setting this command as an item in the pop-up menu which is also convenient for the user.

This is a blank page.

Chapter 5

The Beam Cursor: A Pen-based Technique for Enhancing Target Acquisition

5.1 Introduction

Ubiquitous computing inspires technique development of pen-based interactions. Currently, pen-based devices have been used widely, such as PDAs, Tablet PCs and whiteboards. In such situation more and more researchers have pay attention to these study issues. Target selection is a very fundamental computing task. As the different display sizes and styles of user interfaces emerges selection tasks become more complex, so that studies aimed at improving target selection have become essential and very significant. Hence, many researchers have proposed various techniques that attempt to enhance target selection. However, most of these new techniques have been designed for mice. But the selection of targets using a stylus pen has some specific and unique characteristics which may not apply to the mouse interface. Two examples are, the lifting and pressing of the pen-tip and the sliding of the pen tip across the screen surface. Obviously, selection techniques that are suitable for mouse interfaces do not necessarily suit the special characteristics of stylus pens. Conversely, it seems equally obvious that the optimal performances of each of these significantly different devices will be achieved by significantly different kinds of operations, techniques and graphic elements.

Therefore, we present the Beam Cursor, a new selection technique which is based on the Slide Touch strategy [99], which is tailored for pen-based interfaces. Ren and Moriya [99]studied several selection task techniques for pen-based interfaces and drew the conclusion that the Slide Touch strategy (Fig.5.1a) outperformed the other five selecting task techniques that they tested. In this technique, the pen-tip initially lands outside a target then slides towards the target; when the pen-tip touches the target the target is selected. The Beam Cursor enhances this technique by adding a virtual target size to the normal physical target size. The virtual target size feature means that the Beam Cursor dynamically updates the effective region of targets. When the pen-tip touches the screen surface, the Beam Cursor uses the initial contact point as a reference point, and divides the total space in which all targets reside into regions, so that there is just one target inside each region. When the pen-tip slides towards a target and enters the effective region of the target, the target is pre-selected and is contained by the beam (transparent red shading) which is “emitted from the cursor. When the pen-tip is taken off the screen surface the target is selected. (Fig.5.1b)

In the following sections, we will review previously published research reports dealing with pointing facilitation. We will then discuss the design and implementation of the Beam Cursor, evaluate the performance of the Beam Cursor in two experiments, and show that the Beam Cursors performance can be modelled by Fitts law. We will conclude by discussing some implications for user interface design and also future work.

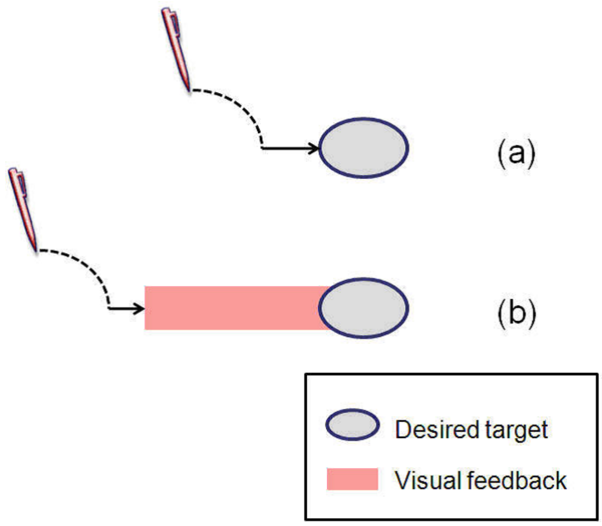

Fig. 5.1 (a) The Slide Touch strategy: the pen-tip initially lands outside the target then slides towards the target; when the pen-tip touches the target the target is selected. (b) The Beam Cursor: the pen-tip lands on the screen surface then slides towards the desired target and the target is pre-selected and contained within a "beam" (shaded area) which is emitted from the cursor in the direction of the target; when the pen-tip is lifted the target is selected. The dot line means the pen movement trace above a screen surface.

5.2 Related Work

5.2.1 Expanding Target or Reducing Distance

Fitts'law [34] is commonly used to predict the time it takes to move a mouse pointer from one location to another.

$$MT = a + b\log_2(\frac{A}{W} + 1) \tag{5.1}$$

According to Fitts'law, the cursor movement time (MT) increases linearly with the Index of Difficulty (ID), which relies on the logarithm of the distance moved (the amplitude, A) and the width of the target (W). The two constants, a and b, are determined empirically and depend on cognition, motor preparation time and on hand-eye coordination. With respect to Fitts' law, there are two simple ways to reduce the difficulty of a pointing task: increasing the target width or reducing the amplitude. Regarding the virtual target size, a target has an effective width which can be defined as the effective area of a target which has been expanded beyond it's physical graphical width. One approach to improving target acquisition is to increase the target's width. McGuffin and Balakrishnan [75] closely examined the degree to which Fitts' Law modeled actions aimed at selecting expanding targets in one-dimensional tasks. They found that Fitts' Law accurately models the performance of such actions, and that movement time is primarily governed by the final expanded size of the target. This result held even when the targets began expanding after most of the movement towards the target (90%) was complete. McGuffin and Balakrishnan's [75] study examined selection of a single object with no surrounding objects, so the influence of distraction due to movement of neighboring objects was not examined. Kabbash and Buxton [55] investigated the use of area cursors. The basic idea is that an area cursor has a larger active region or hotspot for target selection, rather than a single pixel hotspot as in standard cursors. Kabbash and Buxton [55] showed that by setting W to be the width of the area cursor, selection of a single pixel target could be accurately modeled using Fitts' law. Thus, very small targets would have a much lower index of difficulty when selected by an area cursor. However, a problem of ambiguity arises when the desktop environment is densely populated with targets, as multiple targets could fall inside the area cursor at one time. Grossman and Balakrishnan [38] proposed the Bubble Cursor which improves upon area cursors by dynamically resizing its activation area depending on the proximity of surrounding targets, so that only one target is selectable at any time. The evaluation results prove that the Bubble Cursor significantly reduces target acquisition times in both simple and complex multi-target environments. A way to reduce A (the amplitude) is to bring the target closer to the cursor. Bezerianos and Balakrishnan [17] developed the vacuum technique which can bring distance targets closer to the widget's center in the form of proxies that can be manipulated in lieu of the original. This technique is suitable for large screens. An alternative is to jump the cursor to the target. Guiard et al. [41] proposed a selection technique called Object Pointing. In this technique the cursor never visits the empty regions of graphical space. It jumps from one selectable target to another. Object Pointing was found to be considerably faster than regular pointing in a 1D reciprocal pointing task. However in a 2D environment, it was shown that the degree to which object pointing outperformed regular pointing was dependent upon the density of the targets. There have been a number of efforts to facilitate pointing by dynamically adjusting the control display (CD) gain. Worden et al. [118] implemented "Sticky Icons" by decreasing the mouse control-display gain when the cursor enters the icon. Control-display gain determines the mapping between physical mouse movement and resultant cursor movement. In this way, the user must move the mouse further to escape the boundary of the icon, effectively making the icon larger without using extra screen space. Worden et al.'s evaluation showed Sticky Icons to be efficient for selecting small targets. In a technique called semantic pointing, Blanch et al. [19] showed that performance could be predicted using Fitts' Law, based on the resulting larger W and smaller A in motor space. Once again, however, problems arise when multiple targets are presented as the intervening targets will slow the cursor down as it travels to its destination target.

5.2.2 Other Selection Techniques

Beside target selection techniques described above, there are still techniques that based on specific operation manners. Ren and Moriya [99] compared pen-based selection techniques and their characteristics, and proved that the proposed Slide Touch strategy is the best of the six techniques. Slide Touch is where the target is selected at the moment the pen-tip touches the target for the first time after landing on the screen surface. The experimental results show that it is particularly useful in situations where the target is isolated or where targets are arranged sparsely. Guan et al. [40] presented the Zoom Selector which pre-selects, enlarges and relocates the targets covered by a transparent round circle into a large pie sector to enhance the target's acquisition. The evaluation results indicate that the Zoom Selector outperforms the normal click method when used for small targets. It is suitable for small target acquisition or situations where targets are arranged densely.

5.3 Beam Cursor Design and Implementation

The Beam Cursor is an interaction technique that enables quick access to targets on areas of a pen-based display. The Beam Cursor employs the sliding action of the stylus pen and dynamically updates the effective width of the target according to the contact point of the pen-tip and the layout of the surrounding targets, thus enhancing the target's acquisition. When the pen-tip lands on the screen and slides towards a target, the target is included in a "beam" which is "emitted" from the cursor. This means that the target is pre-selected as the stylus approaches and it is selected when the pen-tip is lifted from the screen surface (see fig.5.2).

In designing the Beam Cursor, we explicitly sought to address Slide Touch, which is the inspiration of the Beam Cursor.

Slide Touch [99] is the technique whereby the target is selected at the moment the pen-tip touches the target for the first time after landing on the screen surface. It is a very useful selection technique for pen-based interfaces. However, the technique requires the pen-tip to touch the target which is to be selected before selection can be affected. The Beam Cursor combines the virtual target concept and the Slide Touch strategy to enhance target acquisition. That is, every target is allocated an effective width which is

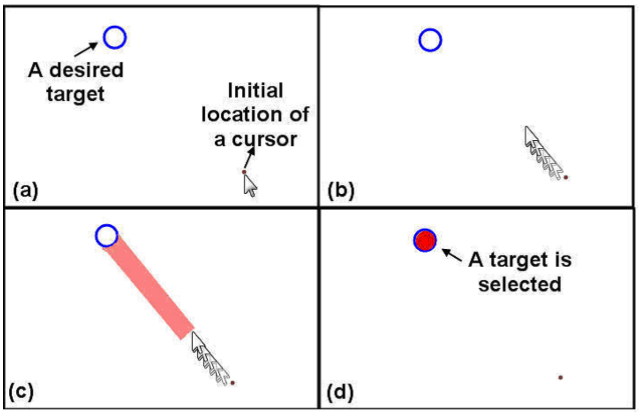

Fig. 5.2 (a) A pen-tip lands on screen surfaces and its initial location is recorded; (b) the pen-tip slides to the desired target; (c) when the cursor enters the effective region of a target the target is contained by a transparent red beam emitting from the cursor; (d) the pen-tip lifts from the screen surface then the target is selected.

bigger than its physical width. The following section discusses how the effective width is allocated to each target.

Regarding virtual target size, every target has an effective width based on its physical width in motor space. During the actual process of target selection, the user can first determine the target that he/she will select. Aiming at the desired target, the Beam Cursor allocates the effective regions of all the targets to enable the desired target to occupy a much bigger effective region. A simple algorithm is used to continuously update the effective regions of targets (see fig.5.3).

- When the pen-tip lands on the screen surface, the contact point is recorded as the reference point.
- Taking that point as the center point, the Beam Cursor divides the total space into n[*1] equal sectors (If the screen is divided into too many sections the error rate is

[*1] There is a closest target to the pen-tip in the screen, which can be determined by Voronoi diagram principle. However, in fact, in each direction area of the screen a possible closest target to the

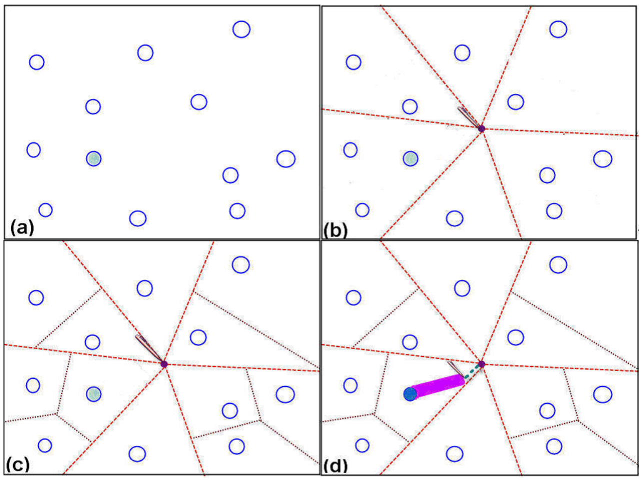

Fig. 5.3 (a) There are many targets in the screen, where the solid blue target is the goal target; (b) When pen lands on screen surface, the initial point is recorded as reference point, which is used as a center point to divide the screen into n sectors (to clear demonstrate the principle, n is set at 6 in the fig.5.3. At fact, the Beam Cursor sets n at 15.). The targets in the same sector constitute a group. (c) Targets in the same group are allocated effective regions according to the Voronoi diagram principle. (d) When a cursor slides into a certain sector the target that is closest to it is pre-selected. Note that all the dot-lines are unseen in the real interfaces.

quite high. The tradeoff of speed and accuracy should be considered. Therefore, based on an informal test n is set at 15). The targets in the same sector constitute a group.

- Targets in the same group are allocated effective regions according to the Voronoi diagram principle. So when a cursor slides into a certain sector the target that is closest to it is pre-selected.

Based on this algorithm the pen-tip can land on any possible position in the vicinity of the target to enable it to have a much bigger effective region. The pen-tip then to

pen-tip exists. To find it the Beam Cursor first divides the whole screen into some sections travel a very short distance to enter the effective region of the desired target. Even if the pen-tip lands on a position imprecisely, a slight movement towards the target affects pre-selection of the desired target. When the cursor enters the effective region of a target the target is contained by a transparent red beam emitting from the cursor. This acts as a reinforcing visual cue to the user, showing that the desired target is indeed pre-selected by the cursor, thus reducing the cognitive load of the user and eliminating any uncertainty about which target will be selected when the pen is removed.

With respect to "abort" of a selection task, the method the Beam Cursor employed is to press an additional button (key) using non-preferred hand [67] to do it, which, in essence, is the mode-switch between selection and non-selection states.

5.4 Experiment 1

The Beam Cursor not only enlarges the effective width of the target but also dynamically updates it, based on the pen-tip's initial landing point on the screen surface. From previous work on expanding targets [75] [124] it was found that users were able to take advantage of the larger expanded target width even when expansion occurred after 90% of the distance to the target had already been traveled. It was also shown that overall performance could be modeled accurately by Fitts' law by setting W to the expanded target width. So we would expect that Fitts' law would hold in situations where the effective width of targets dynamically changes when selecting targets. However, the Beam Cursor has a few specific properties that make it difficult to directly apply Fitts' law to model it.

1. Once the pen-tip lands on the screen surface, the effective width of the target changes.

2. Before capturing a target, the Beam Cursor should be slid towards the target

for a very short distance.

It is important to empirically determine if Fitts' law holds for Beam Cursors. This is the first goal of Experiment 1.

Even if the Fitts' law is shown to model the Beam Cursor performance accurately, this does not necessarily mean that the Beam Cursor provides a significant advantage over Point Cursors. Furthermore, the Beam Cursor enhances target acquisition by enlarging the effective width of targets. And, based on the principle of allocating the effective region to the intended target, the Beam Cursor expands the effective width of the target the user wants to select while shrinking other targets. So we wondered whether the performance is governed by the effective width rather than the actual width of the target. In other words, selecting a target with an actual width W and an effective width EW with a Beam Cursor should be equivalent to selecting a target with an actual width of EW with a regular Point Cursor. Thus, the second goal of Experiment 1 is to determine whether performance is governed by and makes maximum use of the effective width.

To answer these questions in a systematic manner, we begin by studying the Beam Cursor performance in the simplest possible pointing task: 1D target acquisition. We compare the Beam Cursor with the Point Cursor in Experiment 1.

5.4.1 Apparatus

The hardware used in Experiment 1 was the Fujitsu Tablet PC running Microsoft Windows XP. It weighed 1.48kg, and was 210.432 mm (W) x 157.824mm (H). The spatial resolution of the screen was 0.2055 mm/pixel. The software for the experiment was developed using Sun Microsystems Java.

5.4.2 Participants

Eighteen subjects (three females and fifteen males) who had all had previous experience with computers were tested for the experiment. The average age was 22.5. All subjects had normal or "corrected to normal" vision with no color blindness, were right handed, and used the pen in the right hand.

5.4.3 Procedure and Design

The task was a reciprocal 1D pointing task in which subjects were required to select two fixed targets back and forth. The targets were arranged as solid circles, keeping a distance between them along the horizontal axis. The target to be selected was colored green, and the other target was red. In reality, if there were only two targets on the screen, the effective width of targets in a Beam Cursor interface would be very big. The subject would only have to move the stylus pen a very short distance to select the target. Thus, to simulate the realistic target acquisition scenario some distracter targets were placed around both goal targets such that their effective widths (EW) were controlled. Distracters were rendered as blue solid circles (see Fig.5.4). Subjects were instructed to select between the two targets alternately. They were told to emphasize both accuracy and speed. When the subject correctly selected the target he/she would hear a beep sound and the targets would swap colors, which was an indication that the subject had to now move towards and select the other target which was now green.

The design of the experiment was as follows: crossed Cursor Technique (CT) x Amplitude (A) x Width (W) x Effective Width (EW). A full crossed design resulted in 54 combinations of CT (Point Cursor, Beam Cursor), A (288, 576, 864 pixels), W (12, 24, 36 pixels), EW (48, 96, 144 pixels). Each subject had a total of 27 combinations (=3 Amplitudes x 3 target widths x 3 target effective widths) appearing in random order for

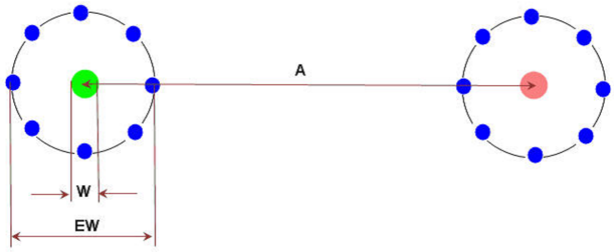

Fig. 5.4 The setup of the 1D reciprocal pointing experiment. The green circle is the target to be selected. The red circle is the next goal target. Blue circles are placed to control the EW/W ratio. Note: EW is an approximate value, which is gotten based on the effective width allocation principle of Beam Cursor.

each technique. Each combination consisted of 5 selection attempts (i.e., four reciprocal movements between the two targets). At the start of the experiment, for each cursor technique, subjects were given a warm-up block of attempts to familiarize them with the task and conditions. Each subject performed the experiment in one session lasting approximately thirty minutes, depending on each subject's proficiency in selecting the targets. The session was broken up according to cursor technique. Whenever the subject felt tired he/she was allowed to take a rest.

5.4.4 Results

An ANOVA (analysis of variance) with repeated measures was used to analyze performance in terms of movement time, error rate and subjective preference. Post hoc analysis was performed with Tukey's honestly significant difference (HSD) test.

Selection time

The analysis showed that there was significant difference between the Point Cursor and the Beam Cursor in selection time, $F(1,34)=21.43$, $p<0.001$. The overall mean selec-

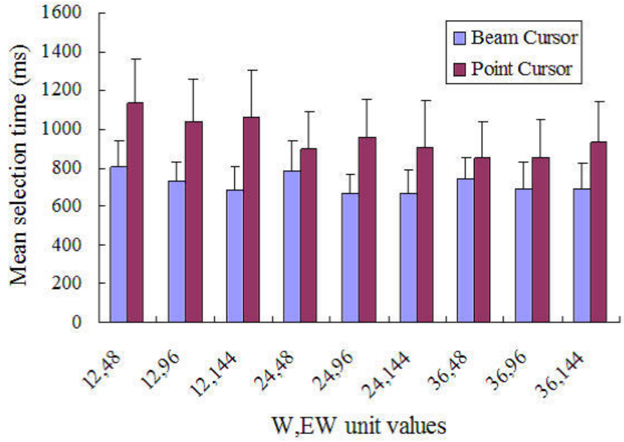

Fig. 5.5 The mean selection time by W, EW values for both cursors.

tion times were 958 milliseconds for the Point Cursor and 718 milliseconds for the Beam Cursor. A repeated measures analysis of variance also showed a significant main effect for W, F(2,105)=23.51, p<0.01; EW, F(2,105)=56.25, p<0.01; and A, F(2,105)=78.54, p<0.01. For each of combinations of W and EW the analysis showed that Beam Cursor was significant faster than the Point Cursor, all at p<.05 level. As Fig.5.5 illustrates, performance of the Beam Cursor is dependent on EW rather than W whereas performance of the Point Cursor depends on W.

Fig.5.6 plots the movement time as a function of the index of difficulty (ID). For the Point Cursor, we define ID as$\log_2(\frac{A}{W}+1)$, while for the Beam Cursor, $\log_2(\frac{A}{W}+1)$. Linear regression analysis showed that the Point Cursor fits the Fitts' law equation with r^2 values 0.9599 and the Beam Cursor fits the Fitts' law equation with r^2 values 0.8727. Here the r^2 value is a little low, which is due to the fact that the effective width of targets is an approximation based on the allocation principle of the effective width in Experiment 1. This means that selection using the Beam Cursor can not only be modeled using Fitts' law, but selection is just as fast as if the target had an actual width of EW and a Point Cursor were being used.

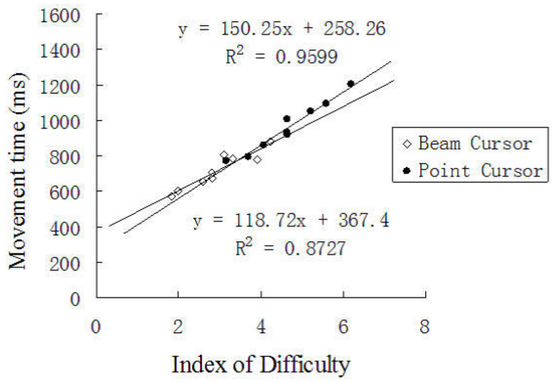

Fig. 5.6 Line regression of target distance against movement time..

Error score

The analysis of mean error score shows that there was no significant difference between the Point Cursor and the Beam Cursor. Overall error rates were 2.78% for the Point Cursor and 3.24% for the Beam Cursor, all well within the typical < 4% range seen in target acquisition studies.

5.5 Experiment 2

Experiment 1 determined that the Fitts' law can model the Beam Cursor and predict the selection time in 1D reciprocal pointing tasks. The experimental results show that Fitts' law can model and predict the Beam Cursor. It also shows that selection performance is governed by the effective width of targets rather than their physical width. In Experiment 1, the Beam Cursor significantly reduced selection time, which further indicates that increasing the effective width of targets does enhance target acquisition.

However, the experiment on 1D targets is an easy and abstract scenario contrasting to actual user interfaces. So we wondered whether the Beam Cursor delivers the same advantages with complex 2D situations. In the second experiment, we explore the Beam Cursor's performance in a more realistic environment with multiple 2D targets with various target widths and layout densities.

If the space surrounding targets is bigger, the effective width of targets will be bigger. In Experiment 2 we will probe this further. And we include the Bubble Cursor [38] which is discussed in the related work section. This technique is perhaps more promising than other existing techniques for improving target acquisition. In the previous work on the Bubble Cursor, it is found that, taking mice as the input device, the Bubble Cursor significantly decreases selection time. So we wondered if the Bubble Cursor offers the same advantage when a stylus pen is the input device? In other words, we wanted to know whether the technique that is suitable for mice is also as suitable for stylus pens. Since the Beam Cursor is a direct extension of the Slide Touch strategy, there is no reason to expect it to perform worse than Slide Touch. This was confirmed in pilot studies, and as such we did not include Slide Touch in our experimental comparison.

5.5.1 Apparatus

The apparatus was the same as in Experiment 1.

5.5.2 Participants

Eighteen subjects (three females and fifteen males) who had all had previous experience in computers were tested for the experiment. The average age was 22.7 years. Twelve of them were test subjects in Experiment 1. All subjects had normal or "corrected to normal" vision with no color blindness, all were right handed, and all used the pen in the right hand.

5.5.3 Procedure and Design

This experiment tested 2 dimensional and multiple target display arrangements. The selection task was serial in contrast to the simple reciprocating movement required for Experiment 1. The target to be selected was green and the others were pale red.

In this experiment, subjects were required to select the green target which appeared randomly among a number of pale red targets on the display. When a selection performance was finished, that green target would change to red and another target would become green indicating that it is the new target. This design required the user to jump in any direction on the screen, not just horizontally as in Experiment 1.

Subjects needed to finish multiple sets of selection tasks. For each set of selection tasks, the number and the width of targets on the screen were different. When the experiment began, the subject saw a green target. The time for the task was recorded from the moment the first green target was selected. Each interval for the two selection actions was recorded. This allowed us to analyze the time of each selection and the total time of all the selection tasks. A successful selection resulted in a beep sound. If no beep sound was heard, it meant that an error had occurred. The feedback we provided was the same as in Experiment 1. When a target was preselected it would be contained by a transparent red beam emitted from the cursor.

The design of the experiment was as follows: crossed Cursor Technique (CT) x Width (W) x Density (D). A full crossed design resulted in 27 combinations of CT (Beam, Point, Bubble), W(12, 24, 36 pixels), D(6, 12, 30). For each of the three techniques, 9 combinations (=3 target widths x 3 target densities) appeared in a random order. Each subject had a total of 144 attempts (3 widths x (6 + 12 + 30) densities). At the start of the experiment, subjects were given a warm-up session for each cursor technique to familiarize them with the task and the conditions. Each subject performed the experiment in approximately forty minutes, depending on individual proficiency. The experiment was broken up according to cursor technique. Whenever the subject felt tired he/she was allowed to take a rest.

5.5.4 Results

Selection time

A repeated measures analysis of variance showed that there was a significant interaction between the three cursor techniques in selection time, $F(2,51)=10.2$, $p<0.001$. The post hoc Tukey (HDS) test showed that the Beam Cursor was faster than both the Point Cursor and the Bubble Cursor($p<0.05$). The Bubble Cursor was significant faster than the Point Cursor in selection time ($p<0.01$). The overall mean selection times were 1196 milliseconds for the Beam Cursor, 1483 milliseconds for the Point Cursor and 1353 milliseconds for the Bubble Cursor (see fig.5.7). The results clearly show that the Beam Cursor can improve target acquisition in complex 2D experimental circumstances. On a pen-based interface and using a stylus pen, the Beam Cursor significantly surpasses the Bubble Cursor [38]. One reason is that the Beam Cursor can endow a much bigger effective region to the desired target than the Bubble Cursor. The other reason is that, in the experimental circumstances of a pen-based interface, the constant lifting of the pen-tip, limits the advantage of the Bubble Cursor. The results also indicate that the selection targets using a stylus pen has its own specific characteristics and that selection techniques that suit mice are not necessarily suitable for stylus pens.

Target width: A significant difference in selection time was observed between the three cursor techniques for each target width, 12, 24 and 36 pixels, $F(2,51)=14.64$, $F(2,51)=9.28$ and $F(2,51)=7.84$, $p<0.001$. This means that significant differences in selection times remained when the target width was varied. As seen in Fig.5.8, with the

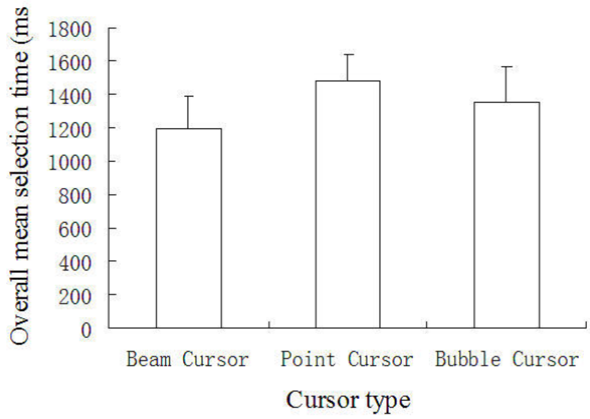

Fig. 5.7 The overall mean selection time for the three cursor techniques.

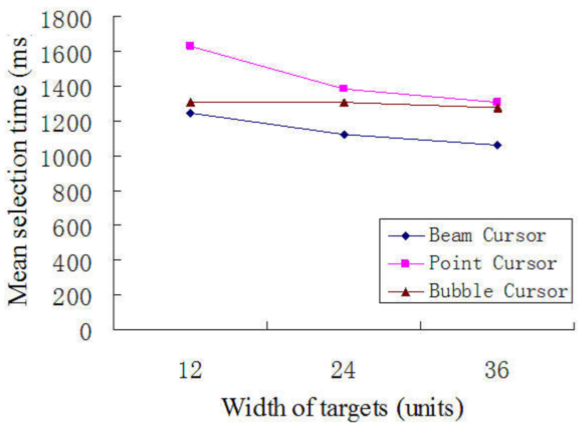

Fig. 5.8 The mean selection time for targets of different target widths.

target width increasing the selection time did not significantly decrease for the Beam Cursor. This is due to the fact that the Beam Cursor is governed by the effective width of targets, not the physical width of the target.

Target density: A significant difference in selection time was observed between the three cursor techniques for each target density, 6, 12 and 30, $F(2,51)=13.05$, $F(2,51)=7.23$ and $F(2,51)=10.32$, $p<0.001$. This means that significant differences in selection times remained when the target density was varied. As seen in Fig.5.9, when the target density was small the difference between the Beam Cursor and the Point

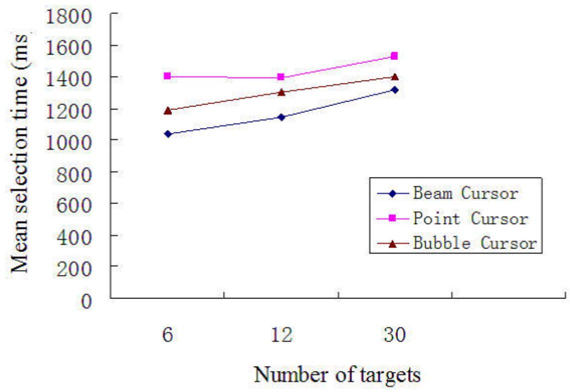

Fig. 5.9 The mean selection time for targets in layouts with different target densities.

Cursor became more significant. This was due to the fact that, when the target density was small, the void space among the targets became wider and the effective width of targets increased. So, for the situation in which targets are placed tightly together, the Beam Cursor probably has a little advantage in selection time.

Error score

The analysis of the mean error score showed that there was no significant difference between the Beam Cursor, the Point Cursor and the Bubble Cursor. Overall error rates were 2.98% for the Beam Cursor, 3.12% for the Point Cursor and 2.45% for the Bubble Cursor, all well within the typical $< 4\%$ range seen in target acquisition studies.

Subjective preference

There was a significant difference between the Beam Cursor, the Point Cursor and the Bubble Cursor in subjective preference, $F(2,51)=18.28$, $p<0.001$. The post hoc Tukey (HSD) test showed that the Beam Cursor was better than both the Bubble Cursor and the Point Cursor, $p<0.05$. The Bubble Cursor was better than the Point

Cursor, $p<0.01$.

5.6 Discussion and Conclusions

The article proposes an interactive technique called Beam Cursor, which enables the quick selection of targets on pen-based interfaces. The Beam Cursor employs the sliding motion and dynamically updates the effective width of targets according to the initial contact point of the pen-tip and the layout of the surrounding targets. The aim is to enhance target acquisition. We then described the methods and results of Experiment 1 and Experiment 2 respectively.

Experiment 1 verified that the Beam Cursor can be modeled and predicted by Fitts' law using the one dimension reciprocal pointing task. We compared the Beam Cursor with the Point Cursor. The evaluation results show that Fitts' law can model the Beam Cursor and predict the selection time. Selection performance is governed mainly by the effective width of targets, not by the physical width of targets. The Beam Cursor outperforms the Point Cursor for the different E/EW ratio.

Experiment 1 is a simple abstract experimental circumstance. Experiment 2 further evaluates the effectiveness of the Beam Cursor on target acquisition. In the second experiment we introduced a current promising selection technique, the Bubble Cursor. We evaluate the three selection techniques under the condition of different target densities and different target widths. The experimental results indicate that the Beam Cursor is better than both the Point Cursor and the Bubble Cursor. There is no significant difference between the Point Cursor and the Bubble Cursor.

Pen devices have specific interactive characteristics: e.g. the lifting and pressing of the pen-tip and the ability to slide the pen-tip on the screen surface. When contrasted with the normal click of a mouse, the unsteadiness of the pen tip can make the press and click action inaccurate. This unsteadiness makes it difficult for users to hit a precise point on a target. According to our observations, it is very common for the pen tip to make contact within a larger range near but outside the target because of the touch screen's slipperiness and the pen tip's vibrations. Therefore, allowing some tolerance in the initial location of the pen-tip and providing a simple means of adjustment via a hand movement which approaches the target would appear to greatly decrease the effect of an imprecise touchdown as well as decreasing the cognitive load of the user. That is why the Beam Cursor exploits the sliding motion.

Virtual targets means that every target has an effective width based on its physical width in its relation to the motor space surrounding it. Actually, selection techniques, such as the area cursor [55] and the Bubble Cursor [38], expand the target width to enhance selection performance. In other words these techniques make full use of the void space around the targets in motor space. The Bubble Cursor employs the Voronoi diagram to increase target size in motor space to the maximum. However, the Beam Cursor allocates the effective region to targets giving priority to the desired target. The goal of this allocation principle is to endow a much bigger effective region to the desired target. Figure 10 shows that the effective region provided by Beam Cursor is more effective than that allocated by Bubble Cursor. As seen from the experimental results, the Beam Cursor indeed reduces the selection time.

We also found the Beam Cursor to be better than the Bubble Cursor, one of the more promising selection techniques in the literature. However, we must be careful before drawing too many conclusions about the relatively poor performance of the Bubble Cursor in our experiment. The Bubble Cursor is able to enhance selection performance where mice are used as input devices. We compared the Beam Cursor with the Bubble Cursor in the pen-based interface, and found that this environment limits the advantage of the Bubble Cursor to some extent. This indicates that selection techniques designed for mice are probably not suitable for stylus pens.

Another characteristic of the Beam Cursor is that it enhances target acquisition without changing the position and size of targets, unlike some other selection techniques [40] [124]. The Beam Cursor dynamically updates the effective width of targets without altering the arrangement of items on the screen.

The effective width determines the selection performance of the Beam Cursor and the void space around targets governs their effective width. So even if a target is very small but its surrounding void space is wide, its effective width is still big and its Index of Difficulty (ID) with regard to selection is quite small. Obviously, the Beam Cursor does not yield any benefit for target selection on a screen where the targets are laid side to side because there is almost no void space among the targets and the effective width of targets is almost equal to their physical width.

The positive results from our experiments suggest that the Beam Cursor could be a beneficial addition to user interfaces. We can develop a plug-in to incorporate the Beam Cursor into user interfaces. This would be very appropriate to exploit ”selection” mode. We can design an appropriate command to allow switching between the Beam Cursor and the Point Cursor. For example, we can set a command button on the taskbar; when the user wants to select targets by Beam Cursor, he/she would just click the command button to activate the Beam Cursor. Or it may be activated by setting this command as an item in the pop-up menu which is also convenient for the user.

Chapter 6

The Adaptive Hybrid Cursor

6.1 Introduction

Target selection via pointing is a fundamental task in graphical user interfaces (GUIs). A large corpus of work has been proposed to improve mouse-based pointing performance by manipulating control display (CD) parameters [19] [41] [38] [55] [118] in desktop environments.

Compare with mouse-based desktop GUIs, pen-based interfaces have a number of different characters. First, pen-based interfaces typically use absolute pointing via a direct input device (i.e., a pen), which is very different from indirect input, such as using a mouse. Second, in addition to the 2D position (x, y) values, many pen-based devices offer additional sensory properties (such as pen pressure values) that can be useful for interaction. Third, many pen-based interfaces have a limited display space and input footprint. As the amount of information displayed on the screen increases, users have to select smaller targets. This is especially obvious in mobile products, such as personal digital assistants (PDAs), pen-based mobile phones, and other mobile pen-based applications. Compared with the extensive studies carried out for mouse-based pointing, more empirical studies are needed to determine how we can improve pen-input usage and efficiency.

Although previous studies have intended to exploit novel pen-based selection techniques, such as Slide Touch [99], Drag-and-pop [15], Bubble Radar [9] and Beam Cursor [120] these techniques were mostly designed for the situations where targets are sparsely, distributed across a display space. When targets are smaller and densely packed, the benefit of these techniques tend to be diminished or become unavailable. To solve such problems, chapter 6 presents the Adaptive Hybrid Cursor, a novel technique that automatically adapts the selection cursor as well as a target space based on pen-pressure. The experimental results indicated that the Adaptive Hybrid Cursor improved selection performance related to high-density, small-target environments.

Recently, an increasing amount of work has explored the use of pen pressure, which is available on pen devices (such as most Tablet PCs or Wacom tablets), as the third input dimension for interaction design [44] [67] [94] [93] [95], in addition to the 2D x-y coordinates. However, little attention has been paid to using pen pressure to improve target selection tasks. This chapter, therefore, investigates the possibility of improving the performance of target acquisition tasks for pen-based environments by taking advantage of pen pressure potentials.

There are three fundamental elements in a selection task: a cursor, a target, and a selection background (including a void space). We explored how the pen pressure can be employed to improve target acquisition tasks by varying these three elements. A background plays an important role in many applications but its use is often overlooked in previous work. For example, numerous functionalities have been designed to associate with the background in Windows and Mac desktops, from basic but important functions such as selecting and deselecting, to re-arranging desktop icons and also to more complex operations such as changing certain properties of applications. A background also serves as a visual storage space for future elements. Furthermore, group selection techniques (such as rectangular or lasso techniques) will be awkward to operate without being able to select an empty space. The famous quote from the ancient Chinese philosopher, Lao Tze, says, "the usefulness of the wheel, cup and house is actually based on their emptiness". Without the ability to select the background, many applications become difficult to use.

This chapter makes the following contributions:

- The Adaptive Hybrid Cursor can be used to select targets that have minimal surrounding space or densely packed small targets;

- The Adaptive Hybrid Cursor improves performance by manipulating all three components of target selection: the background, the target and/or the cursor;

- The Adaptive Hybrid Cursor provides easy cancellation without having to use an extra mode-switch button;

- The Adaptive Hybrid Cursor is the first interaction technique that employs pen pressure for target selection.

In this chapter, we first review the related work. Next we describe the design of our new technique. We then present the evaluation of the Adaptive Hybrid Cursor under various target acquisition conditions. We conclude with a discussion of our results and directions for future work.

6.2 Related Work

In this section, we discuss related work regarding both target selection techniques and pen pressure.

6.2.1 Previous Work on Selection Techniques

Target selection tasks can be modeled by Fitts' law [34] [73]. One common form of Fitts' law is $MT = a + b\log_2(\frac{A}{W}+1)$, which states that the time (MT) to acquire a target with width W and distance (or amplitude) A from the cursor can be predicted (where a and b are empirically determined constants, and the term inside the log function is called Index of Difficulty or ID). Obviously, target acquisition performance can be improved by increasing W, decreasing A, or both.

The width of a target is usually defined by the space it occupies on the screen. The effective target width (EW) may be defined as the analogous size of a target in motor space. In standard pointing, the effective target width matches the visual width. However, the effective width can be increased either for the cursor [38] [55] [118] or the target [26] [75] [124] to achieve the same effect. Most previous studies have shown the effectiveness of their proposal only for single isolated target, while they have not been shown to work well when multiple targets are present in close proximity [26] [41] [75] [124]. The state of the art in this category is Bubble Cursor [38], a mouse-based technique that allows selection of discrete targets by using a Voronoi diagram to associate void space with nearby targets. Bubble Cursor works well even in a normal-density multiple-target environment except for the limitations mentioned in the discussion section of this chapter.

There is also a large body of work that is intended to improve selection performance by decreasing A. They either bring the target much closer to the cursor such as Drag-and-pop developed by Baudisch et al. [15], and 'vacuum filtering' introduced by Bezerianos and Balakrishnan [17], or jump the cursor directly to the target, such as with the object pointing technique [41]. Overall, the performance of techniques aiming to decrease A is largely affected by the number of distracting targets between the starting position and the target. They tend to work well on large displays where targets are further away or in low density environments with few distracting targets. These techniques become less effective with high or normal density environments in regular or smaller size displays such as Tablet PCs or PDAs.

Some have tried to improve pointing and selection by dynamically adjusting the Control Display gain. The gain is increased on the approach to the target and decreased while inside the target thus increasing and decreasing the motor space at critical moments in the selection process. TractorBeam [87] is a hybrid point-touch technique that aids selection by expanding the cursor or the target, or by snapping to the target. Worden et al. [118] implemented 'Sticky Icons' by decreasing the mouse control-display gain when the cursor enters the icon. Blanch et al. [19] showed that performance could be predicted using Fitts' law, based on the resulting larger W and smaller A in the motor space. The common problems for these techniques occur when multiple small targets are presented in close proximity, as the intervening targets will slow the cursor down as it travels to its destination target.

An interesting special case here is a technique which is used on large displays to help reach targets that are beyond the arm's reach [9] [15] [17] [29] [83], e.g., RadarView [83]. However, since RadarView decreases both A and W proportionally, the ID is unchanged. The benefit of RadarView is only demonstrated on larger displays where users can operate on RadarView to save the extra movement required to reach a distant target i.e. one that is beyond arm's reach. Bubble Radar [9] combines RadarView and Bubble Cursor by first placing the objects within reach, and then applying Bubble Cursor to increase selection performance. Bubble Radar also tried to address the background selection problem of Bubble Cursor by using a button switch controlled by the non-dominant hand, however, since Bubble Radar is virtually another Bubble Cursor, its advantage is likely to diminish in a high density environment.

6.2.2 Related Work on Pressure

There has been less work done on pressure than on pointing-based target acquisition characteristics. Studies on pressure can be roughly divided into two categories. One category investigates the general capabilities of humans interacting with computers using pressure. For example, Herot and Weinzapfel [44] investigated the human ability of the finger to apply pressure and torque to a computer screen. Buxton [22] studied the use of touch-sensitive technologies and the possibilities for interaction they suggest. Ramos et al. [94] explored the human ability to vary pen-tip pressure as an additional channel of control information. The other category of study is where researchers build pressure enabled applications or techniques. For instance, Ramos and Balakrishnan [93] demonstrated a system called LEAN and a set of novel interaction techniques for the fluid navigation, segmentation and annotation of digital video. Ramos and Balakrishnan [95] designed Zlider widget. Li et al. [67] investigated using pressure as a possible means to delimitate the input phases in pen-based interactions. Although these works opened the door to establish pressure as a research avenue, we are unaware of any work which addressed the issue of applying pressure into discrete target acquisition. We attempt to investigate this issue in this chapter.

6.3 Adaptive Hybrid Cursor Design

A few previous studies have shown that a reasonable manipulation of targets, cursors and context can enhance target acquisition. However, the tradeoff between "original" state of these three elements and "manipulation" state needs to be considered in technical design. Our approach is to employ pen-pressure which is an available parameter in some pen based devices and can be used to easily produce a continuous value or discrete states. Pen-pressure has the potential to affect selection implementation. Based on this idea we designed the Adaptive Hybrid Cursor technique.

Adaptive Hybrid Cursor includes two states. It first determines whether it should zoom its contexts (target and background) and/or cursor according to the initial location of the cursor and the information regarding the position of targets. If the condition is not suited to the adaptive strategy, Adaptive Hybrid Cursor initiates the Zoom Cursor

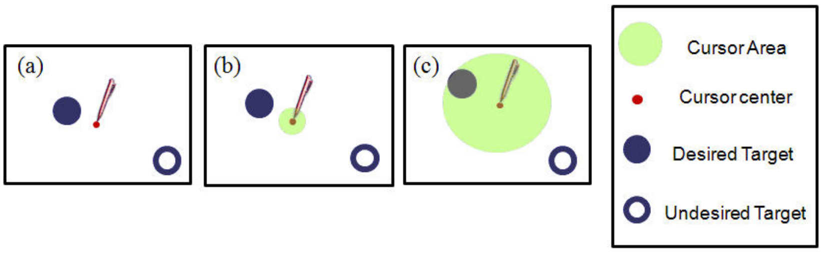

Fig. 6.1 The process of selecting a target with Adaptive Hybrid Cursor in State 1: the adaptive hybrid cursor employs the Zoom Cursor technique which changes the size of the cursor when targets are big in a low density environment. (a) the pen-tip lands on the screen; (b) pressure value is used to zoom the cursor. (c) pressure and location of the cursor are adjusted to make the zoomed cursor interact with the desired target. The desired target is selected by quickly lifting the pen-tip. Note that the same legend is used for Fig.6.2.

technique described in Section 5.1 (see Fig.6.1). If the condition satisfies the adaptive strategy criteria, Adaptive Hybrid Cursor begins to zoom the targets, the cursor and background based on the pressure described in Section 5.2 (see Fig.6.2).

6.3.1 Zoom Cursor Technique (State 1)

One possibly fruitful direction open to the examination of pressure-enhanced target acquisition is to use pen pressure to enlarge the cursor size. Based on this intuition, we designed Zoom Cursor, a technique that allows a user to enlarge the cursor size by pressing the pen tip harder on a tablet or a touch-sensitive screen (see Fig.6.1).

As determined in previous studies [14], the degree of pen pressure perceived by human users is not consistent with that sensed by digital instruments. For example, at a low spectrum of pen pressure, the sensed pressure value increases much faster than users would expect. Previous work has used a sigmoid transfer function to achieve the effects produced by pressure. In our experiments we also employed the sigmoid transfer function. The application of pressure is comprised of an initial "dead zone", slow response at low pressure levels (too sensitive for users to distinguish and control), smooth transitions at median pressure levels and quick responses at high pressure levels (users often confirm pre-selection by imposing heavy pressure on a pen-tip). We employed a piecewise linear function to approximate the pressure mapping.

If pressure causes the cursor to become too large, then more than one target might be included, and this may confuse the user. To overcome this problem, a basic principle should be specified so that when enlarging the cursor, only one target will be included at one time. Therefore, a maximum size for the cursor should be determined according to the current position of the cursor and the layout of targets. This will help to ensure that an enlarged cursor cannot include more than one target. Note that the maximum size of the cursor is dynamically changed based on the proximity of surrounding targets. We follow the algorithm used to set the radius of the cursor in Bubble Cursor. We also use a circular-shaped cursor and we allow only one target to be selected each time.

To describe the algorithm in an environment with targets T1, T2, ..., Tn we used the following definitions:

Minimum Distance i (MinDi): The length of the shortest line connecting the center of the Zoom Cursor and any point on the border of Ti.

Maximum Distance i (MaxDi): The length of the longest line connecting the center of Zoom Cursor and any point on the border of Ti.

A simplified version of the algorithm is as follows:

Calculate the Minimum Distance to each target: MinD1, MinD2,, MinDn

Calculate the Maximum Distance to each target: MaxD1, MaxD2,, MaxDn

Set maximum radius of Pressure Cursor = the second minimum value (MinD1, MinD2,, MinDn, and MaxD1, MaxD2,, MaxDn)

After a desired target is included by the enlarged cursor the target selection is achieved by the quick release manner [94].

6.3.2 Zooming Target, Cursor and Background (State 2)

Using direct pointing, the selection speed has an upper limit due to human limitations such that selecting a 10 cm wide object which is within 10 cm of the human user will take less than a second, while a target which is 10 meters away will take at least several seconds to reach. Thus Bubble Radar uses RadarView to bring the targets within arm's reach so that Bubble Cursor can be subsequently easily applied for actual target selection.

Similarly, if the targets are too small and densely packed, it becomes more difficult for the user to visually locate the target. In such cases, enlarging the workspace has the effect of simultaneously increasing A and W and thus making target acquisition easier. Based on this hypothesis, we decided to enlarge the entire workspace when the target size is smaller than 1.8 mm (about 6 pixels in our experimental setup). (Ren and Moriya's study indicated that 1.80 mm is "the smallest maximum size" [99]), or EW/W value is less than 2 where EW is the effective width. Here, we define EW/W as the density of targets, i.e. the amount of void space immediately surrounding a target. The result of pilot studies showed that the selection technique that zooms cursor, target and background at the same time could not show significant advantages above Bubble Cursor when the value of EW/W is more than 2. We defined an environment where the EW/W ratio was less than or equal to 1.5 as a high density environment, and, when the EW/W ratio was greater than 1.5 and less than or equal to 2, we called it a normal density environment. When the EW/W value was equal to or greater than 3, this was called a low density environment. High density environments are common in today's applications (e.g., a word processor or a monthly calendar viewer). Fig.6.2 is an illustrated walkthrough of the technique in State 2.

The maximum zoom ratio is 3 in our current design. The zoom ratio is controlled

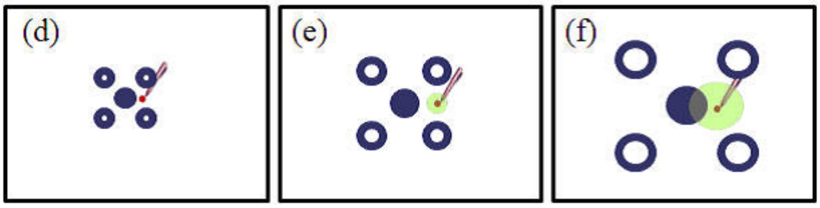

Fig. 6.2 The process of selecting a target with Adaptive Hybrid Cursor in State 2: Adaptive Hybrid Cursor is able to vary the size of targets, cursor and background simultaneously by pressure when approaching small targets and/or small EW/W. (d) the pen-tip lands on the screen; (e) using pressure value to zoom in the targets, the cursor and the background. (f) adjusting pressure and location of the cursor to make the zoomed cursor interact with the desired target. The desired target is selected by quickly lifting the pen-tip.

by the mapped pressure value. At the same time, Adaptive Hybrid Cursor also uses pressure and the "updated" location information of targets to zoom the cursor size according to the principles of Zoom Cursor. When the desired target was interacted by the cursor, the target selection was achieved by the "quick release" motion [94].

The trigger for the enlargement is pen pressure which dynamically adapts the maximum zoom size of the cursor based on the zoomed surroundings, i.e., the cursor should cover no more than one object at a time.

6.4 Experiment

To evaluate the performance of Adaptive Hybrid Cursor, we conducted a quantitative experiment to compare it with Bubble Cursor and with the traditional technique, the regular cursor (the regular pointing selection in graphical user interfaces) as a baseline. First, Bubble Cursor, which is the current state of the art, has been shown to be the fastest desktop pointing technique. Second, Aliakseyeu et al. [9] showed that Bubble Radar combined the benefits of Bubble Cursor in a pen-based situation. However, neither Bubble Radar or Bubble Cursor experiments included very small targets (i.e. less than 1.6 mm). We, therefore, designed the same EW/W (1.33, 2, 3) ratios as for Bubble Cursor but with smaller targets (4 pixels). We wondered if Bubble Cursor offers the same advantage in smaller target situations in pen-based environments. Third, Adaptive Hybrid Cursor also employs the effective width of targets just as with Bubble Cursor, targets being allocated effective regions according to a Voronoi diagram.

6.4.1 Participants

Twelve subjects (11 male and 1 female) all with previous experience using computers were tested for the experiment. The average age was 24.9 years. All subjects used the pen in the right hand. All subjects had normal or a "corrected to normal" vision, with no color blindness.

6.4.2 Apparatus

The experiment was conducted on a Wacom Cintiq21UX, 43.2x32.4cm interactive LCD tablet display with a resolution of 1600 x 1200 pixels (1 pixel = 0.27 mm), using a wireless pen with a pressure sensitive isometric tip. The pen provides 1024 levels of pressure, and has a binary button on its barrel. The tablet's active area was mapped on the display's visual area in an absolute mode. The experimental software ran on a 3.2GHz P4 PC running Windows XP. The experiment software was implemented in Java 1.5.

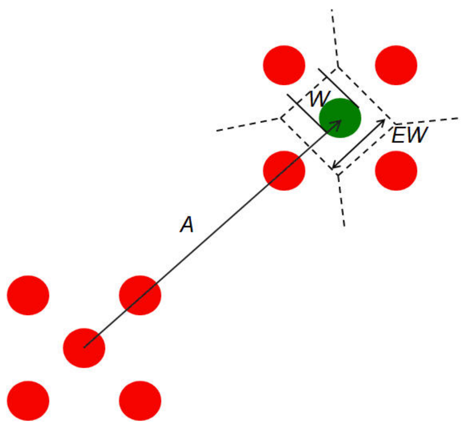

Fig. 6.3 Experimental setup. The red circle in the center around four targets is the start target (as well as one of the two goal targets), the green target is the goal target. The four circles around each of the start and goal targets are distracters which determined the EW/W ratio.

6.4.3 Procedure

Following the protocol [38], we also used a reciprocal pointing task in which subjects were required to select two fixed targets back and forth in succession, but, to simulate a more realistic two dimensional pointing environment, we changed the protocol into a multi-directional reciprocal pointing task which included reciprocal horizontal, vertical and diagonal movements. The targets were drawn as solid circles, and were located at various distances from each other along four directional axes. The goal target, the one intended to be selected, was colored green. When a goal target had been selected, it changed color to red which was an indication that the user now had to select the next goal target. Four red circles were placed around each goal target to control the EW/W ratio (Fig.6.3).

Subjects were instructed to select the two goal targets alternately. They were told to emphasize both accuracy and speed. When the subject correctly selected the target, he/she heard a beep sound and the targets swapped colors, which was an indication of a new trial. At the start of the each experiment, subjects were given a warm-up block to familiarize themselves with the task and the conditions.

6.4.4 Design

A within-subject design was used. The independent variables were: selection techniques ST, amplitude A (288, 576, 864 pixels), width W (4, 6, 12, 36 pixels), EW/W ratios (high = 1.33, normal = 2, low density = 3), and direction DR (horizontal, vertical, 2 diagonals). A full crossed design resulted in 432 combinations of ST, A, W, EW/W, and DR. The order of techniques was counterbalanced using a 3 x 3 Latin-Square. Each participant performed the entire experiment in one session of approximately 60 minutes at one sitting, including breaks corresponding to changes in selection technique. The session consisted of nine blocks of trials completed for each technique. In each block, subjects completed trial sets for each of the 144 combinations of A, W, EW/W, DR appearing in random order. A trial set consisted of 3 effective attempts (4 attempts in total, but the first attempt was the starting point so that it was discarded. Note we had 3 EW/W ratios (high = 1.33, normal = 2, low density = 3), as previously defined in Section 5.2, so we could assess the results from different density environments.

In summary, the design of the experiment was as follows:

12 subjects x

3 techniques (Adaptive Hybrid Cursor, Bubble Cursor, Regular Cursor) x

4 target widths (4, 6, 12, 36 pixels) x

3 amplitudes (288, 576, 864 pixels) x

3 EW/W (high = 1.33, normal = 2, low density = 3) x

4 directions (horizontal, vertical, 2 diagonals)x

3 effective attempts (4 trials total, but the first trial is discarded due to the same starting point) x

3 blocks

= 46656 total effective selection attempts

After they finished testing each technique, the subjects were asked to fill in a questionnaire which consisted of three questions regarding "selection difficulty", "fatigue", and "overall usability" on 1-to-7 scale (1=lowest preference, and 7 =highest preference). These questions were made by referring to ISO9241-9 [52]).

6.4.5 Results

An ANOVA (analysis of variance) with repeated measures was used to analyze performance in terms of selection time, error rate, and subjective preference. Post hoc analysis was performed with Tukey's Honestly Significant Difference (HSD) test.

Selection time

There was a significant difference in the mean selection times among the three selection techniques, $F(2,33)=13.1$, $p<.0001$. The overall mean selection times were 1129 ms for Adaptive Hybrid Cursor, 1177 ms for Bubble Cursor and 1429 ms for Regular Cursor. Tukey HSD tests showed that both Adaptive Hybrid Cursor and Bubble Cursor were significantly faster than Regular Cursor ($p<.001$). No significant difference was found between Adaptive Hybrid Cursor and Bubble Cursor. Significant interaction was not found between selection technique and block number, $F(4,99) = 0.56$, $p = .69$, which indicated the learning improvement did not significantly affect the relative performance of selection techniques.

As shown in Fig.6.4, at the EW/W ratio value of 1.33 there was a significant difference in selection time between the three selection techniques, $F(2,33)=15.1$ and 8.9 for the target sizes of 4 and 6 respectively, all $p<.001$. For target sizes of 4, 6 Tukey HSD

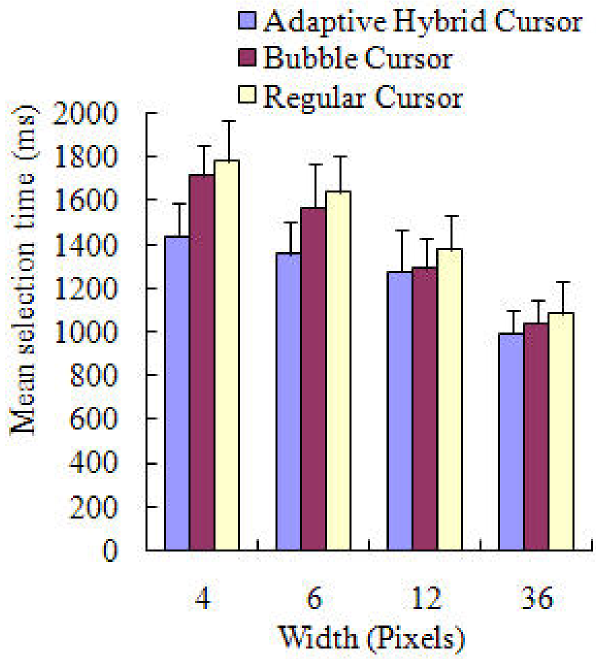

Fig. 6.4 Mean selection times for different sizes of targets at EW/W ratio=1.33.

tests showed Adaptive Hybrid Cursor was significantly faster than Bubble Cursor and Regular Cursor ($p<.01$), however, no significant difference was found between Bubble Cursor and Regular Cursor. No significant differences were found between the three selection techniques for the target sizes of 12 and 36.

At the EW/W ratio values of 2 and 3, both Adaptive Hybrid Cursor and Bubble Cursor were significantly faster than Regular Cursor, $F(2,33)$=8.0, 22,9, 8.8 and 19,6 for EW/W=2; $F(2,33)$=24.2, 14.0, 15.2 and 20.1 for EW/W=3, at target sizes of 4, 6, 12 and 36, all $p<.01$. No significant differences were found between Adaptive Hybrid Cursor and Bubble Cursor in both EW/W ratios.

The perspective brought by Fitts' law in terms of size and distance effects provided a useful framework for our design. However, it is questionable if it is valid to parameterize our results with a Fitts' law model. Adaptive Hybrid Cursor was more complex than a typical single pointing task in Fitts' law studies because it required the user to perform multiple steps, i.e., enlarge the curser and its contents by pressure, confirm the goal target, and select the goal target. Indeed, we obtained a rather poor fit between the Fitts' law model and the actual data collected, with r^2 value at 0.53 for Adaptive Hybrid Cursor, and 0.87, 0.97 for Bubble Cursor, Regular Cursor respectively (we defined ID as $log_2(A/W + 1)$) for Adaptive Hybrid Cursor and Bubble Cursor, while for Regular Cursor $log_2(A/W + 1)$). The r^2 value for Adaptive Hybrid Cursor was much lower than the values for 0.95 or lower than those found in conventional one-step pointing tasks. We also looked at the data of State 1 (i.e. Zoom Cursor) described in Section 5.1. We obtained a better fit with r^2 value at 0.87 for Zoom Cursor but still lower than the values for 0.95. This was due to the fact that users had to control the size of the cursor which they do not have to do in conventional one-step pointing. The r^2 value (0.87) for Bubble Cursor was lower than the values for 0.95. This may have been due to the limitations in pen-based systems mentioned in our discussion section.

Error rate

There was a significant difference in overall mean error rate between the three techniques, $F(2,33)=23.4$, $p<.0001$. Tukey HSD tests showed Adaptive Hybrid Cursor was better than both Bubble Cursor and Regular Cursor ($p<.05$). Bubble Cursor was better than Regular Cursor ($p<.01$). Overall error rates were 4.2% for Adaptive Hybrid Cursor, 5.4% for Bubble Cursor, and 7.3% for Regular Cursor.

As shown in Fig.6.5, at the EW/W ratio value of 1.33, there was a significant difference between the three selection techniques for the sizes of 4 and 6, $F(2,33)=8.1$, 4.2 $p<.05$. For target size of 4 Tukey HSD tests showed Adaptive Hybrid Cursor was better than both Bubble Cursor than Regular Cursor ($p<.05$). No significant difference was found between Bubble Cursor and Regular Cursor. For a target size of 6 Tukey

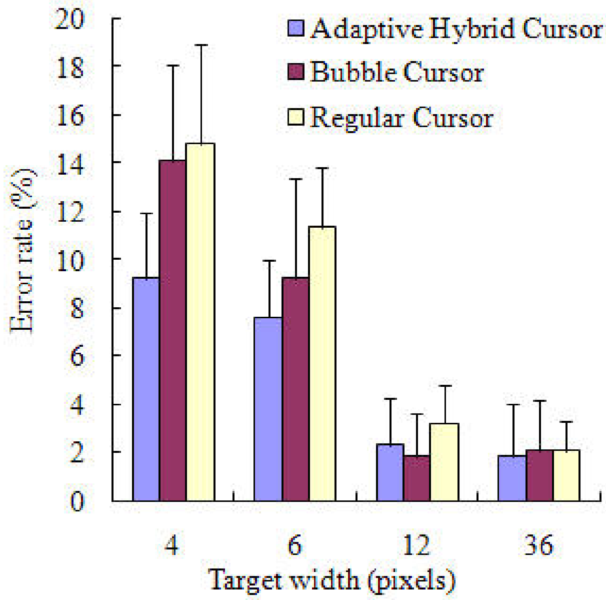

Fig. 6.5 Mean error rates for different sizes of targets at EW/W ratio=1.33.

HSD tests showed Adaptive Hybrid Cursor was better than Regular Cursor ($p<.05$). No other significant differences were found among the three techniques. There was no significant difference in error rate between the three selection techniques for the sizes of 12 and 36.

At the EW/W ratio value of 2, there was a significant difference between the three selection techniques for sizes 4 and 6, $F(2,33)=16.2$, 16.6 $p<.01$. For target sizes of 4 and 6 Tukey HSD tests showed both Adaptive Hybrid Cursor and Bubble Cursor were better than Regular Cursor ($p<.01$). No significant difference was found between Adaptive Hybrid Cursor and Bubble Cursor. There was no significant difference in error rate between the three selection techniques for sizes 12 and 36. The results of the EW/W ratio value of 3 followed trends similar to those of EW/W=2.

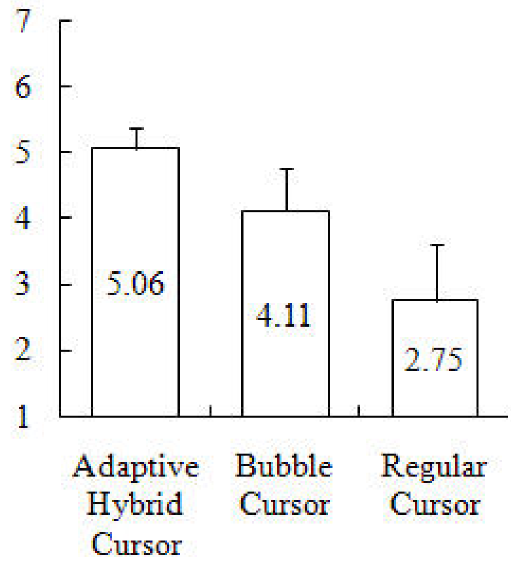

Fig. 6.6 Subjective ratings for the three techniques (1 = lowest preference, 7 = highest preference).

Subjective preference

Fig.6.6 shows the subjective ratings for the three techniques. These ratings were based on the average value of the answers given by the subjects to the three questions. Significant main effects were seen between the three selection techniques, $F(2,33)=38.4$ $p<.001$. Tukey HSD tests showed Adaptive Hybrid Cursor was better than Bubble Cursor, and Bubble Cursor was better than Regular Cursor ($p<.01$). Adaptive Hybrid Cursor was the most preferred (mean = 5.06).

6.5 Discussion

To improve the performance for selecting targets in a dense layout, we designed the Adaptive Hybrid Cursor (including Zoom Cursor), a novel interaction technique for pen-based systems, which enables users to adjust the size of the background, the targets and/or cursor the simultaneously. The Adaptive Hybrid Cursor dynamically adapts the permitted upper boundary of a zoomable selection cursor based on the current index of difficulty of a desired target. As shown in our Experiment, the Adaptive Hybrid Cursor showed advantages over other techniques in performance for small targets in a high density environment. The subjective preferences also showed that the Adaptive Hybrid Cursor was the most preferred technique among the three techniques.

Overall, the Adaptive Hybrid Cursor showed significant improvements in a pen-based selection task. It works well with a pen, and in expanding contexts. At the same time, it remains competitive selection performance without losing the background selection capability, and does not expand the context in groups of big targets, in normal and low density environments. In contrast, many of the other mouse and pen-based interaction techniques have been shown to work well only in low density environments or on isolated targets, even on small targets.

Though Bubble Cursor is comparable to Adaptive Hybrid Cursor in high EW/W ratio or groups of larger targets in a high density environment, it has several limitations compared to our technique, especially in pen-based environments. First, by maximizing utilization of empty screen space, Bubble Cursor trades-off the ability to select an important "target", the background. In contrast, our Adaptive Hybrid Cursor (including Zoom Cursor) allows the user to select the background (by applying lighter pressure). Second, Bubble Cursor lacks the undo function. Our technique provides "natural" cancellation by reversing the pressure value rather than using another mode-switch action like Bubble Radar [9]. Third, Bubble Cursor is not designed for pen-based environments and it does not guarantee continuous, incremental visual feedback of the selection cursor . Though continuous feedback is not assured with the Adaptive Hybrid Cursor either, it can control the size of the cursor well by pen-tip pressure. Fourth, though Bubble Cursor allows denser target placement than many previous approaches, its performance advantage largely degrades when a target is closely surrounded by other objects. In theory, when the target's effective width (EW) approaches its actual width (W), little room can be used to improve the motor space. In fact, it has been shown that as the EW/W ratio changes from 3 to 1.33, the advantage of Bubble Cursor degrades [38]. In contrast, the Adaptive Hybrid Cursor can enlarge the targets, the background, and the cursor, according to the targets' surroundings. Fifth, both Bubble Cursor and Bubble Radar experiments have not included very small targets. To further clarify, we also designed the same EW/W (1.33, 2, 3) ratios but smaller target (4 pixels = 1.08 mm). The experimental results showed that Bubble Cursor suffered from performance limitations in groups of small targets in high density environments.

We varied the essential parameters but we found it necessary to simplify our experimental design in some minor points. First, we set each target in each environment to the same size so that control of the target density parameters could be achieved more easily. Second, we used circular targets so that the distance between start point and destination target was constant in all four directions. Third, in Bubble Cursor's experiment, beside the circles around the target, many black-filled circles were also placed between the starting position and the final target as distracters on the mouse pathway. We omitted intermediate targets (i.e., distracter targets) due to the following reasons. Under indirect pointing environments, these distracters can significantly impact selection performance, since the subjects' selection pathway can't be avoided by the cursor. However, for a direct pointing pen-based environment, the user simply lifts the pen in air to move from the starting position to the goal target where an out-of-range state is possible. This hypothesis was confirmed in pilot studies and in our Experiment. In addition, even though the distracters are placed in between the start and destination targets, visual load will be similar for each of the techniques. Furthermore, the error rate for Bubble Cursor may increase because if the user selects a distracter he/she cannot perform the "undo" task with Bubble Cursor.

6.5 Discussion

We explored the use of pen pressure for improving the performance of target acquisition tasks in pen-based environments. The Adaptive Hybrid Cursor takes advantage of pressure information in pen-based environments. The experimental results have shown that pen pressure can be used to design effective selection techniques for pen-based environments. By using pressure, the Adaptive Hybrid Cursor (or the Zoom Cursor technique) achieves in-place mode switching between background and target selection and requires no additional accessories. This is different from Bubble Radar's approach [9] which uses an additional button to switch states [67]. Our study contributes valuable empirical data for applying pressure for target selection techniques which had not been previously addressed in literature. This chapter also suggests new ways to further improve target acquisition performance for small targets and high density environments. Future work includes applying a combination of strategies found in [9] [67] into the Adaptive Hybrid Cursor for large display environments and group selections.

This is a blank page.

Chapter 7

The Three Novel Line-based Techniques

7.1 Introduction

In current selection tasks, fundamental computing operations include single-selection tasks and multiple-selection tasks. Single target selection is usually accomplished by tapping; multiple target selection is usually accomplished using the rubber-band box. The rubber-band box works like this: the rectangular selection region is specified by extending the diagonal of the rubber band box by dragging; the targets interacted by the rectangular selection region are highlighted for selection. An obvious drawback of the rubber-band box is that it is difficult to select the multiple targets that are not included in the rectangular area. Conversely, it is impossible to exclude unwanted targets from the rectangular area without further clicks, taps or other maneuvers. So, when selecting multiple targets that are arranged irregularly, the user has to implement a variety of selection tasks such as using tapping the "Ctrl" key and the rubber-band box together. In some sense the rubber-band box limits the user's performance in multiple target selections.

Thus we present three novel line-based techniques (Rubber-Line-Sweep, Line-String and Coupling-With-Pressure) to enhance multi-target acquisition (described in Section 7.3).

We conducted experiments to compare Rubber-Line-Sweep, Line-String, Coupling-With-Pressure and Rubber-band box by pen and mouse. In the experiments the selection task was the conjunctive selection of multiple targets (rectangles) in two-dimensional grids varied in terms of the sizes of the targets and the distances between the targets.

In this chapter, we start with related work, followed by introduction and evaluation of the three line-based selection techniques in both pen-based and mouse-based interfaces, and then we give a discussion and conclusion.

7.2 Related Work

For both mouse-based and pen-based interfaces a lot of techniques have been proposed to enhance multi-target acquisition. There are two fundamental strategies to accomplish multi-target selection: one is to specify a selection range (area) to implement such selection tasks [79] [47] [92], the other is to perform them with specific stroke gestures [99] [4] .

Rubber-band box is the standard technique for selecting multiple targets in an area. With this method, the diagonal extent of the drag operation specifies the size of the selection area (rectangle). This method is implemented in most current graphical user interfaces. It provides efficient target selection, but has the limitation that the user can only select multiple targets that are arranged within a rectangular region. When the user wants to select multiple targets in an irregular layout, he/she is required to press another key (such as the Shift key) to select subgroup targets.

Mizobuchi and Yasumura conducted an experimental quantitative analysis of Tapping and Circling selections on pen-based devices [79]. The experiment used a two-dimensional grid varying in terms of the size and the distance of the targets. The results indicate that tapping selection time differed significantly depending on the size and spacing of targets. Conversely, circling selection times differed according to different levels of target cohesiveness and shape complexity. The results do not support the replacement of tapping with circling in pen-based interfaces. Circling requires a long stroke to achieve the selection of multiple targets and this consumes a lot of selection time.

The “lasso” is another approach for area selection that differs form the rubber-band box in tracing around a group of targets. It enables users to select a contiguous set of objects that form an irregular pattern. Hinckley et al. presented Pigtail (a small loop) to delimit lasso selection from making menus [47].

Raisamo evaluated several direct manipulation, time-based and pressure-based area selection techniques in a public information kiosk [92]. The conclusions show that pressure-sensitive techniques provide a possible alternative to other selection techniques, but require careful designing. The time-based technique was the most intuitive and the direct manipulation technique was understood well after an initial learning phase.

Our Rubber-Line-Sweep also exploits rubber-band lines to specify a selection area by a “sweeping” movement. Rubber-Line-Sweep can achieve the selection targets in an irregularly shaped selection area with a short stroke and in less time. Rubber-band lines, which are common graphic elements, have been used by Albinsson and Zhai and Masui et al. in their interaction techniques [7] [74]. Albinsson and Zhai utilizes two rubber-band lines to precisely control the location of a cursor [7]. Masui et al. proposed an interaction technique based on the rubber-band metaphor for manipulating precise data [74].

Repeated operations (e.g., tapping or clicking) can also be used to perform multi-target selection tasks. Ren and Moriya’s “Slide Touch” [99]and Accot and Zhai’s “crossing boundaries” [4] employed strokes to select single target, however, they did not examine the multi-target acquisition. Our Line-String uses a line stroke to "string" multiple targets to select them.

7.3 Novel Multi-target Selection Techniques Design

In designing the novel multi-target selection techniques, we explicitly sought to address drawbacks of the Rubber-band box.

Rubber-band box is the predominant method to perform multi-target selection in GUI. As shown in fig.7.1, goal targets that are not included in a rectangular area are not easily selected with the Rubber-band box technique. The selection of such targets requires extra operations. Such situations often occur in practice.

To overcome the shortcomings of the Rubber-band box method we employ rubber-band lines or line strokes to select multi-targets instead of the rubber-band rectangle (see Fig.7.1).

Rubber-Line-Sweep: it utilizes rubber-band lines to select targets. Rubber-band lines are very common computer graphic elements, the length and direction of which can be easily changed. When the user wants to perform multi-target selection using the Rubber-Line-Sweep he/she first lands the pen-tip on the screen, then moves it to extend a rubber-band line. Targets swept by the rubber-band line are selected. Rubber-Line-Sweep uses "sweep" as the interaction manner of selection. In essence Rubber-Line-Sweep utilizes a rubber-band line to specify a selection area by "sweeping" so it can define irregularly shaped selection areas. To abort selection of a highlighted target the target is swept again by a rubber-band line (see Fig.7.1a).

Line-String: Line-String exploits a line stroke to select targets by stringing them together (see Fig.7.1b). To abort a highlighted target a line stroke is drawn through the target and the highlighted state is aborted.

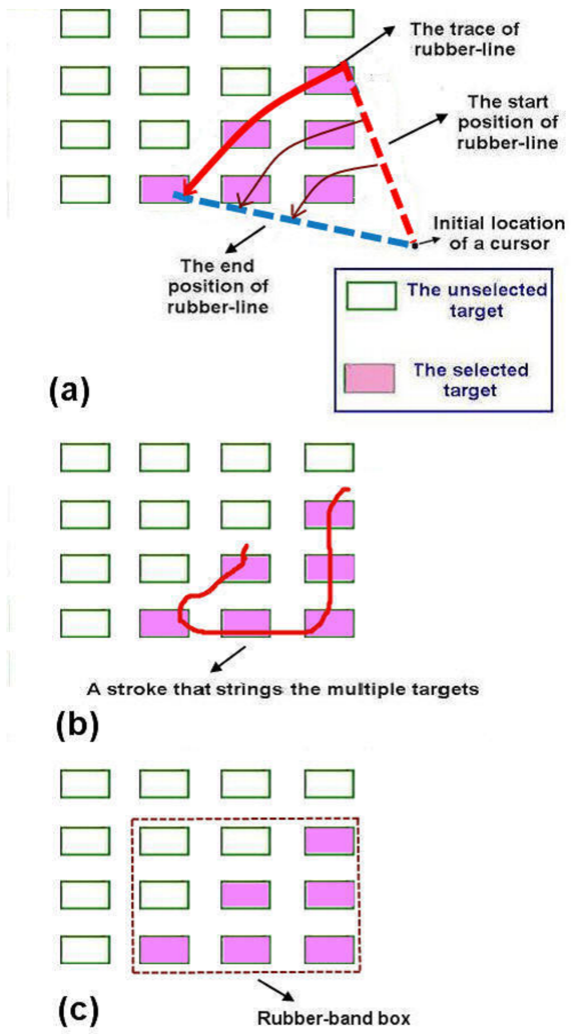

Fig. 7.1 Selection processes of Rubber-Line-Sweep, Line-String and Rubber-band box: (a) Rubber-Line-Sweep: dragging cursor extends a rubber-band line which is used to sweep targets to select them; (b) Line-String: drawing a stroke to string targets to select them.(c) an irregular layout of targets can not be included by a rectangle.

Coupling-With-Pressure: Rubber-line-sweep and Line-string have their own special operational characteristics and advantages. It is even possible to combine the two techniques in special situations, e.g. when they are applied to a pen-based device they can be combined by using pressure as the switch mode between the two techniques, i.e. at light pressure one mode is active while at heavy pressure the other is active. This

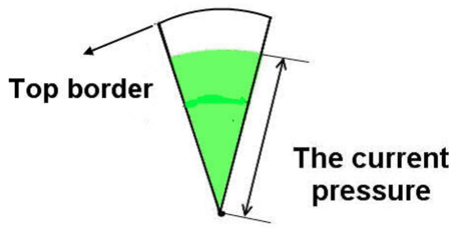

Fig. 7.2 The visual feedback of pressure.

technique is called Coupling-with-pressure. The threshold value is crucial for achieving a free and stable switch between techniques. If the threshold is very low it is quite likely that the user will unintentionally pass the threshold and change from the currently desired method. Therefore, it should be set to a high value which would require a deliberate change in pressure to switch the operational state. To determine a proper switch threshold a pilot experiment with 10 subjects was performed. The task was to draw freehand strokes (arbitrary curves and straight lines), basic geometrical graphs (such as rectangles and circles) and a mixed set of Roman, Japanese, and Chinese characters (kanji) and signatures on a blank space in a natural manner. Pen-tip pressure was recorded in a 17ms sampling periods. Ninety-five percent of the force samples fell within the 210 to 810 units range. The results showed that the pressure level of more than 810 units was seldom used in a natural manner. Therefore, in Coupling-with-pressure the threshold value is set at 900 units.

Visual feedback which denotes the current pressure value is added to Coupling-with-pressure. Ramos et al. (2004) 'study [94] shows that a good visual feedback is needed for pressure-based UI design. Inspired by the pressure cursor design by Ramos and Balakrishnan [95] Coupling-with-pressure pressure was displayed in a wedge-shaped graphical widget (see Fig.7.2). The transparent green area indicates the current pressure value. The top border represents the switch threshold. When pressure is beyond the top border the technique is changed from Rubber-line-sweep to Line-string.

7.4 Experiment 1: Comparison of Four Techniques by Pen

Rubber-Line-Sweep utilizes a rubber-band line to select targets by sweeping. The properties of the line enable Rubber-Line-Sweep to accomplish the selection of a complex layout of a group of targets. Using Line-String, targets that are strung together are highlighted and selectable. Actually, if the number of targets is large, a long trajectory of a drawn stroke to select targets is needed. Coupling-With-Pressure couples Rubber-Line-Sweep and Line-String together using pen pressure as a switch mode. We explored the performance efficiency of the four multi-target selection techniques in different circumstances. The complexity of the layout of the targets and the number of targets were varied.

7.4.1 Participants

Twelve volunteers (ten males and two females) participated in Experiment 1. The average age was 24.3 years (ranging from 21 to 31). All were right-handed.

7.4.2 Apparatus

The hardware used in Experiment 1 was a Wacom DIT-520 interactive LCD graphics display tablet with a wireless stylus that has a pressure sensitive isometric tip. The pen tip reports 512 levels of pressure and has a binary button on its barrel. The experimental software ran on a 3.2GHz P4 PC with the Windows XP Professional operating system. The experimental software was developed using Sun Microsystems Java language.

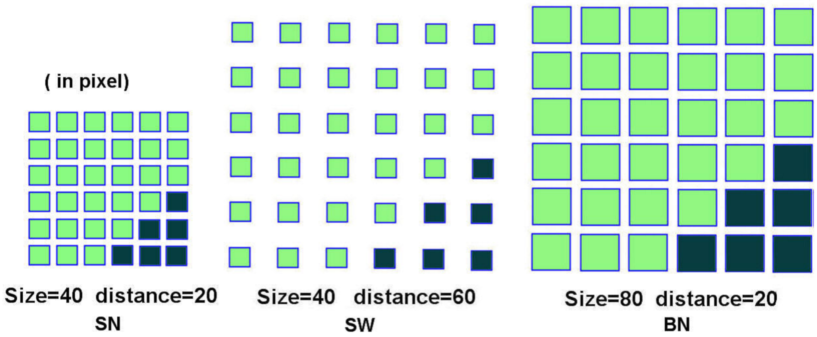

Fig. 7.3 Three combinations of target square size and inter-square distance in two-dimensional (6 x 6) grids.

7.4.3 Task and Stimuli

Generally, the single target or discrete targets are activated by tapping while conjunctive multi-target selections are performed by multi-target selection methods such as the Rubber-band box technique. So conjunctive multi-target selection tasks are prescribed in Experiment 1. For each trial, thirty-six squares were shown on a 6x6 grid. Target squares were green in color, and the other (non-target) squares were white with a blue frame.

In all the tasks, the two-dimensional grids were varied in terms of the size of the squares to be selected, and the distances between the selection squares (i.e., the intra-grid boundaries). Three different combinations of grid layout were used. They were small squares close together (SN), small squares further apart (SW), and large squares close together (BN) (see Fig.7.3).

With respect to the number of targets in a task, there are three types of tasks: four-target tasks, nine-target tasks and sixteen-target tasks. With reference to target layout, we provided three complexity degrees for the tasks: low complexity tasks, medium complexity tasks and high complexity tasks,which were determined by the subjects (descried in Section 4.4). Thirty-six different target layouts were used in Experiment 1.

Fig. 7.4 The layout shapes used in experimental tasks.

These patterns were shown in different positions on the grid and with varying numbers and layouts of squares (see Fig.7.4).

Participants read instructions describing the experimental setup and task. The experimenter then reviewed these instructions with the subjects, introduced the input devices. Then, for each method subjects started with a formal practice session (including all combinations of different conditions). When the experimenter was satisfied that subjects understood the task and could perform it correctly, subjects completed the experiment unassisted.

The subjects were instructed to select the desired targets in each trial as quickly and accurately as possible. They were instructed to press the start button to start a selection task. Then the start button automatically became the stop button. The time was recorded from the moment the pen touched the screen after the user had pressed the start button. When the user finished a trial he tapped any key on the keyboard to finish the task and the end time was recorded (and the stop button was changed back to the start button). When the trial was not correctly accomplished an audible beep would play.

They were observed by the experimenter and encouraged to take frequent breaks. Participants typically spent 30 to 45 minutes using each device. A short questionnaire was completed at the end, to gather subjective opinions.

7.4.4 Design

The design of the experiment crossed Methods (Rubber-Line-Sweep, Line-String, Rubber-band box, Coupling-With-Pressure) x Layouts of grid (SN, SW, BN) x Complexity of targets (Low, Medium, High). Presentation of trials within a block was randomized. In summary, the experiment consisted of:

12 participants x

4 methods x

3 layouts of the grid x

36 selection tasks

= 5184 target selection trials

The complexity reflects the irregular degree of the layout of targets, which can be determined by different ways [18] [24]. In our experiments the complexity of targets layout was determined by the subjective assessments of experimental participants according to their experience. The complexity of these patterns was assessed by a separate sample of 12 participants. Before that the participants were asked to rate each shape in terms of how difficult they thought it would be to implement it, using the following rating scale (1= very easy, 2= easy, 3= medium, 4= difficult, 5= very difficult). Prior to making their judgments, the participants were shown all 36 patterns so that they could get a glimpse of them and calibrate the scale they were using to make a judgment on the complexity of all the patterns. Then participants were shown the patterns one at a time, rating each pattern before the next one was shown. The order of presentation of the patterns was randomized. The ratings across the 10 participants were then averaged to create a scale of performance complexity on which each pattern was located. The resulting scale of complexity was then categorized into 3 levels (low: rated complexity score was below 1.5; medium: rated complexity score was between 1.5 to 3.0; high: rated complexity score was over 3.0).

7.4.5 Results

An ANOVA (analysis of variance) with repeated measures was used to analyze performance in terms of movement time, error rate and subjective preference. Post hoc analysis was performed with Tukey's honestly significant difference (HSD) test.

Selection time

Selection time was the main dependent measure. Repeated measures analysis of variance showed a significant difference in selection time between the four selection methods, $F(3,44)=15.1$, $p<.001$. The overall mean selection times were 2.31 seconds for Rubber-line-sweep, 2.19 seconds for Line-string, 2.33 seconds for Coupling-with-pressure and 2.58 seconds for Rubber-band box. The post hoc Tukey HSD test showed that Rubber-line-sweep, line-string and Coupling-with-pressure were all faster than Rubber-band box ($p<.05$). There were no other significant differences across the selection methods.

There was a main effect in selection time between the target various size-distance conditions. $F(2,33)=11.5$, $p<.01$. For Rubber-line-sweep, Line-string, Coupling-with-pressure and Rubber-band box, selection times differed significantly between the three target size-distance conditions (Rubber-line-sweep: $F(2,33)=14.5$, $p<.01$; Line-sting: $F(2,33)=17.2$, $p<.01$; Coupling-with-pressure: $F(2,33)=9.7$, $p<.01$; Rubber-band box: $F(2,33)=8.3$, $p<.01$). Fig.7.5 shows mean selection times by selection methods and

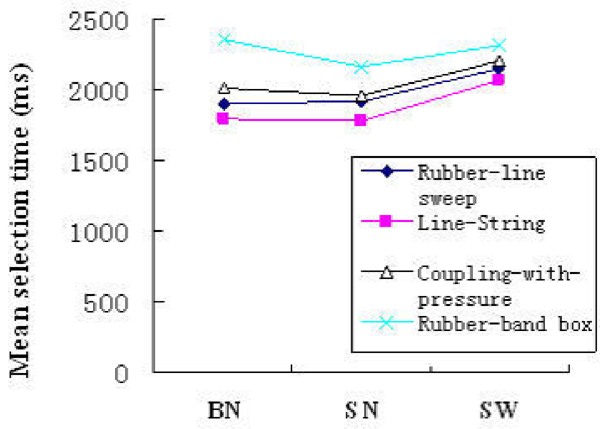

Fig. 7.5 Mean selection time by selection methods and target size-distance conditions

target size-distance conditions.

Fig.7.6 indicates mean selection times by selection methods and the complexity of target layouts at different numbers of targets for the BN condition.

- *4 targets*: For the low complexity layout there was no significant difference in selection time between the four methods. For the medium complexity layout there was a significant difference in selection time between the four methods, $F(3,44)=7.8$, $p<.01$. The post hoc Tukey HSD test showed that Rubber-line-sweep, Line-string and Coupling-with-pressure were all faster than Rubber-band box ($p<.01$) There were no other significant differences across the selection methods. For the high complexity layout there was a significant difference in selection time between the four methods, $F(3,44)=16.9$, $p<.01$. The post hoc Tukey HSD test showed that Rubber-line-sweep, Line-string and Coupling-with-pressure were all faster than Rubber-band box ($p<.01$) There were no other significant differences across the selection methods.
- *9 targets*: For the low and medium complexity layout there was no significant

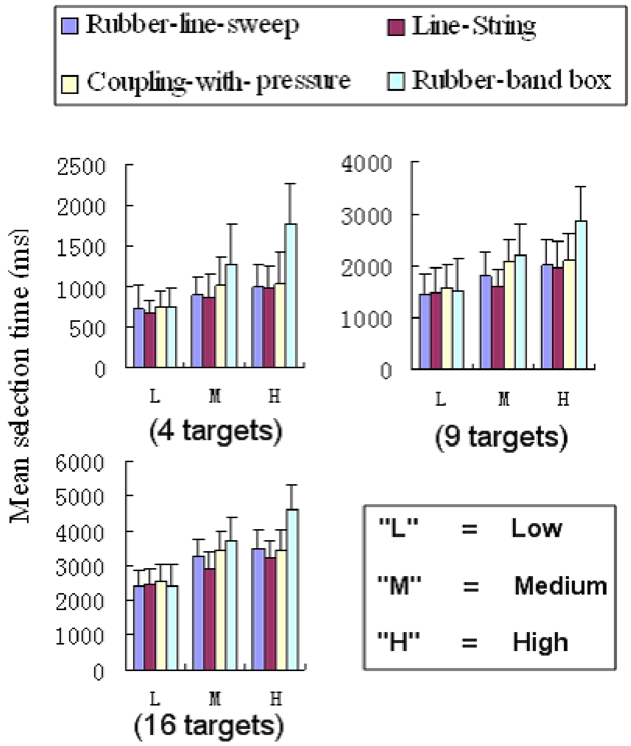

Fig. 7.6 Mean selection times by selection methods and the complexity (Low, Medium and High) of target layouts at different numbers of targets for the BN condition.

difference in selection time between the four methods. For the high complexity layout there was a significant difference in selection time between the four methods, F(3,44)=14.2, p<.01. The post hoc Tukey HSD test showed that Rubber-line-sweep, Line-string and Coupling-with-pressure were all faster than Rubber-band box (p<.01). This was due to that our propopsed techniques enable the user to easily select irregular layout of targets while Rubber-band box had to uses several discrete operations to perform such tasks, which consumed much time. Therefore There were no other significant differences across the selection methods.

- *16 targets*: The results for 16 targets followed trends similar to those of 9 targets, as shown in Fig.7.6 (16 targets).

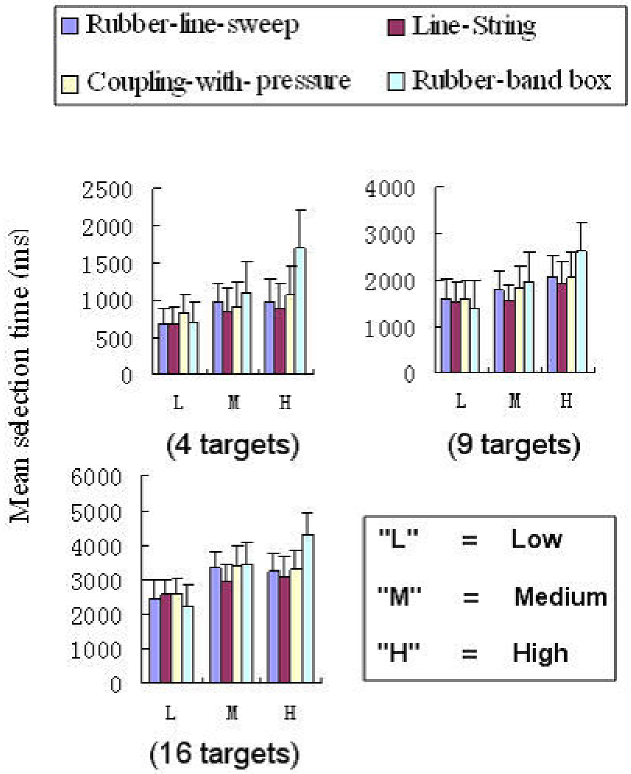

Fig. 7.7 Mean selection times (with SD bars) by selection methods and the complexity (Low, Medium and High) of target layouts at different numbers of targets for the SN condition.

Fig.7.7 shows the results of SN.

- *4 targets*: For the low and medium complexity layout there was no significant difference in selection time between the four methods. For the high complexity layout there was a significant difference in selection time between the four methods, F(3,44)=8.3, p<.01. The post hoc Tukey HSD test showed that Rubber-line-sweep, Line-string and Coupling-with-pressure were all faster than Rubber-band box (p<.01) There were no other significant differences across the selection methods.
- *9 targets*: For the low complexity layout there was a significant difference in selection time between the four methods, F(3,44)=9.7, p<.01. The post hoc Tukey HSD

test showed that Rubber-band box was faster than Rubber-line-sweep, Line-string and Coupling-with-pressure ($p<.05$). This is due to the fact that Rubber-band box can use short distance to produce an area with a box, comparing with other three methods. Therefore, For low complexity layout shape and big number of targets Rubber-band box has its advantage. There were no other significant differences across the selection methods. For the medium complexity layout there was no significant difference in selection time between the four methods. For the high complexity layout there was a significant difference in selection time between the four methods, $F(3,44)=12.5$, $p<.01$. The post hoc Tukey HSD test showed that Rubber-line-sweep, Line-string and Coupling-with-pressure were all faster than Rubber-band box ($p<.01$). There were no other significant differences across the selection methods.

- *16 targets*: The results for 16 targets followed trends similar to those of 9 targets, as shown in Fig.7.8 (16 targets).

The results of the SW condition followed trends similar to those of the SN condition.

Error rate

The analysis of the mean error score showed that there was no significant difference between Rubber-line-sweep, Line-string, Coupling-with-pressure and Rubber-band box. Overall, error rates were 2.31% for Rubber-line-sweep, 2.16% for Line-string and 2.47% for Coupling-with-pressure and 2.55% for Rubber-band box.

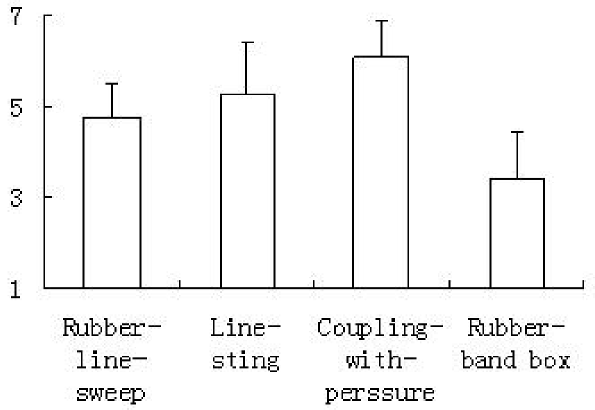

Fig. 7.8 The subjective ratings for the four techniques.

Subjective preference

Subjects were required to rate these techniques on a scale of 1-to-7 (7 for best, and 1 for worst) (see Fig.7.8). There was a significant difference between Rubber-line-sweep, Line-string, Coupling-with-pressure and Rubber-band box. in subjective preference, $F(3,44)=17.2$, $p<.001$. The post hoc Tukey (HSD) test showed that Coupling-with-pressure was better than the three other selection methods, $p<.01$. Rubber-line-sweep and Line-string were both better than Rubber-band box, $p<.01$. There were no other significant differences across the selection methods. Some subjects mentioned that Coupling-with-pressure was more flexible and it enabled the user to choose a proper method (between Rubber-line-sweep and Line-string) depending on the conditions of the selection tasks at any particular time.

7.5 Experiment 2: Comparison of Three Techniques by Mouse

In Experiment 1 we compared Rubber-line-Sweep, Line-string and Coupling-with-pressure with the standard rubber-band box, taking a pen as the input device. Experimental results showed Rubber-line-sweep, Line-string and Coupling-with-pressure brought benefits to multi-target selection performance. We wondered whether they maintain their advantage when taking a mouse as the input device. Therefore, we conducted Experiment 2 to evaluate them with the mouse as the input device.

7.5.1 Participants

Ten volunteers (all males) participated in Experiment 2. The average age was 22.6 years (ranging from 20 to 30). All were right-handed.

7.5.2 Apparatus and Tasks

The input device used in Experiment 2 was a Logitech Optical Wheel Mouse. The other hardware was the same as in Experiment 1.The tasks in Experiment 2 was the same as that in Experiment 1.

7.5.3 Results

Selection time

No significant difference was found in selection time between the four selection methods ($p=.31$).

There was a significant difference in selection time between the target size-distance conditions, $F(2,27)=10.3$, $p<.01$. For Rubber-line-sweep, Line-string and Rubber-band box, selection times differed significantly between the three target size-distance conditions (Rubber-line-sweep: $F(2,27)=7.6$, $p<.05$; Line-string: $F(2,27)=8.2$, $p<.05$; Rubber-band box: $F(2,27)=12.4$, $p<.05$).

Fig.7.9 shows the results of BN.

- *4 targets*: For the low complexity layout there was a significant difference in selection time between the three methods, $F(2,27)=9.7$, $p<.01$. The post hoc Tukey HSD test showed that Rubber-band box was faster than Rubber-line-sweep and Line-string ($p<.01$). No significant difference was found across the selection methods. For medium complexity layouts there was no significant difference in selection time between the three methods. For high complexity layouts there was a significant difference in selection time between the three methods, $F(2,27)=12.8$, $p<.01$. The post hoc Tukey HSD test showed that Rubber-line-sweep and Line-string were both faster than Rubber-band box ($p<.01$). There were no other significant differences across the selection methods.
- *9 targets and 16 targets*: The results for 9 and 16 targets followed trends similar to those for 4 targets.

The results for SN and SW conditions followed trends similar to those for BN conditions. Although Rubber-line-sweep and Line-string showed no benefits when compared with Rubber-band box in overall mean selection time, they still had their advantages for high complex target layouts.

Error rate

The analysis of the mean error score showed that there was no significant difference between Rubber-line-sweep, Line-string and Rubber-band box. Overall error rates were 3.22% for Rubber-line-sweep, 3.11% for Line-string and 3.02% for Rubber-band box.

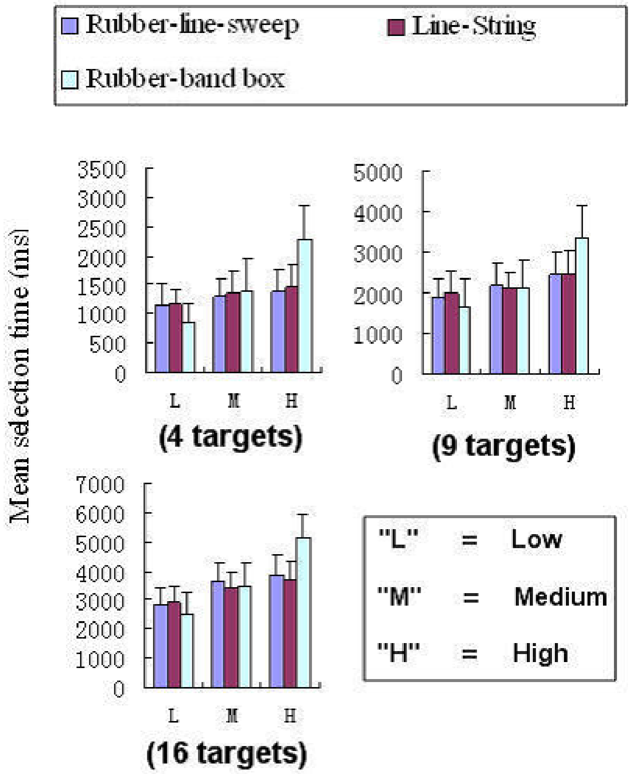

Fig. 7.9 Mean selection times by selection methods and the complexity (Low, Medium and High) of target layouts at different numbers of targets for the BN condition.

Subjective preference

There was no significant difference between Rubber-line-sweep, Line-string and Rubber-band box in subjective preference.

7.6 Discussion and Conclusion

Our main goal in this work is to overcome the weaknesses that the standard rubber-band box has in selecting irregular target layouts. We present three multi-target selection techniques: Rubber-line-sweep, Line-string, and Coupling-with-pressure. We designed the experiments to evaluate and compare them with the Rubber-band box technique. To do so we varied the number of targets, the complexity of the layout shape and the size-distance conditions of targets. The experimental results indicated that with respect to the overall mean selection time, when a pen was the input device, Rubber-line-sweep, Line-string and Coupling-with-pressure were better than Rubber-band box in overall mean selection time and had significant advantages for high complex layout of targets; when a mouse was the input device there were no significant differences between them in overall mean selection time, however, still had advantages for high complex layout of targets. From subjective preferences we found that subjects much prefered Coupling-with-pressure for pens because of easy-to-use switching function between Rubber-line-sweep and Line-string.

Rubber-Line-Sweep and Line-string: Rubber-line-sweep and Line-string are both better than Rubber-band box in selecting complex layout shapes of targets. However, the individual characteristics of each technique determine which is most appropriate in any given task. Rubber-line-sweep is best suited to large numbers of conjunctive targets. Line-string shows more advantages for more complex layout shapes. Some subjects mentioned that they could finish a selection task using a single stroke with this technique. Line-string is less efficient when selecting large numbers of targets or small targets because each target needs to be included in the stroke. The advantages of Rubber-line-sweep and Line-string in selecting targets in complex layout shapes are achieved at the expense of cursor movement distance. So the proposed techniques offer no significant advantages over Rubber-band box when the mouse is used as the input device.

Coupling-with-pressure: This technique couples Rubber-line-sweep and Line-string with pressure and provides two different selection modes so it enables the user to choose a suitable selection method to perform selection tasks according to the layout of the targets. Over half of the subjects thought this was a good way to use the pressure switch i.e. to couple Rubber-line-sweep and Line-string. In our early design this technique did not show obvious advantages in selection time. Adjustments to the pressure threshold value and the pressure visual-feedback significantly improved multi-target acquisition, particularly in the high complexity condition. This implies that proper pressure value and real-time visual-feedback are needed for efficient pressure-based UI design.

Rubber-band box: Experimental results showed taht this method was not suitable for selecting irregular layouts of targets. It showed more disadvantages than Rubber-Line-Sweep, Line-String and Coupling-with-pressure, especially in pen-based interfaces.

The distance the cursor is moved when performing the selection interaction should be discussed. Commonly, a technique that requires less movement time and less distance for the cursor to move could be a more beneficial technique. Rubber-band box extends the diagonal to form a box and thus moves the cursor a comparatively short distance to specify a big selection area but it is efficient only for selecting targets in a very regular layout. For complex target layouts, Rubber-band box requires several discrete operations and thus loses its advantage.

The positive results from our experiments suggest that Rubber-line-sweep, Line-string and Coupling-with-pressure could be a beneficial addition to pen-based interfaces. Our study provides different possible ways to enhance multi-target selections. Pressure, as an additional input parameter, is seldom explored and applied into UI designs. Coupling-with-pressure offers a promising instance of pressure-based applications.

We believe that the results of our work uncovered several basic principles that are applicable directly toward the design of interaction techniques for multi-target acquisition, particularly in pen-based interfaces.

This is a blank page.

Chapter 8

The Stroke-based Scrolling Techniques

8.1 Introduction

Scrolling is a fundamental task in graphical user interfaces. In traditional mouse-based interfaces, scrolling is typically accomplished by rolling the wheel on a mouse or with a scroll bar - dragging the "elevator", clicking on scroll arrows, or clicking in the space between the arrows and elevator.

However, pen-based devices lack scrolling widgets like wheels and pen-based interfaces have different characteristics to mouse-based interfaces, suggesting that an alternative to Scroll Bars might be more appropriate for use with pens. In particular, stroke-oriented gestures are often considered to be more natural in pen-based interfaces. Therefore, in this chapter we examine stroke-based interfaces for scrolling.

Although there are some studies on stroke-based scrolling techniques [82] [104] they all, utilize arc strokes to perform scrolling tasks. We have found line strokes to be very suitable for scrolling performances. The literature lacks a careful, quantitative analysis and comparison between arc and line strokes for scrolling on pen-based devices. Furthermore, studies have not considered the relationship between scrolling speed and the length of documents, the effect of screen size or scrolling directions. Therefore, we look at stoke-based scrolling techniques in order to answer the following questions:

- What are the differences between arc and line strokes to perform scrolling tasks?
- Can the application of speed modes (independent or dependent modes) related to the length of the document offer benefits to stroke-based techniques?
- Do the size of the device screens and the scrolling directions (vertical or horizontal scrolling) have effects on the evaluation result of the three scrolling techniques?

In this chapter, to answer these questions, we first explore the technical characteristics of stroke-based techniques and make a classification according to the modes of control of the scrolling displacement and direction in a standard scrolling task. More specifically, our classification is as follows (see Figure 8.1):

- Arc stroke technique (hereafter referred to as Arc): Drawing a clockwise or counter-clockwise circular stroke to scroll documents vertically or horizontally. The direction is specified by angles in the strokes.
- Line stroke technique (hereafter referred to as Line): Drawing a line stroke to scroll documents up or down. The direction is controlled by pen displacement.

Two experiments were conducted to evaluate stoke-based scrolling techniques, taking Scroll Bar as the baseline. In Experiment 1 these were tested on small, medium and large screens respectively, set in independent control mode. In experiment 2 they were set in dependent control mode so as to be further evaluated. We conclude by discussing implications for user interface design and future lines of work.

8.2 Related Work

Generally, when scrolling through a document to find a target, the user must not only control the final displacement of the document to make the target appear in the viewing window, but also control the speed of the movement, in order to comfortably

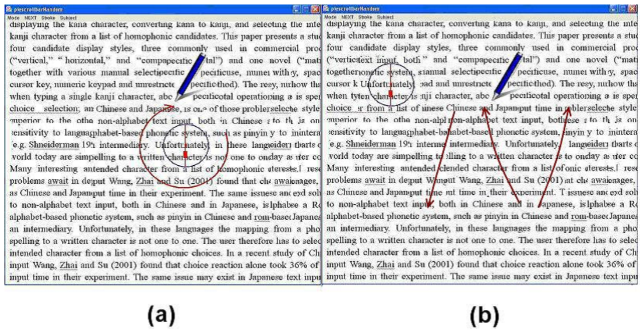

Fig. 8.1 (a) The arc stroke technique: Drawing the clockwise or counterclockwise circular stroke to scroll documents up or down. (b) The line stroke technique: drawing a line stroke gesture to scroll documents up or down.

scan the document to look for the destination. Also, scrolling is usually done in phases: first the user rapidly scrolls to the rough region, then more slowly locates the precise destination. Thus, a good scrolling technique should offer a tradeoff between fast scrolling and precise scrolling. With standard scroll bars, for example, the user can drag the "elevator" to fast scroll a document; or press the "arrow button" or scroll the mouse wheel to precisely scroll a document. Hinckley et al. show that mouse wheel scrolling can be improved by acceleration algorithms [48]. Zhai et al. also investigate using isometric controls that vary the rate of scrolling according to with force applied [125]. However, excessively fast scrolling can cause the information of a document to appear blurred. Cockburn et al. propose a SDAZ (speed dependent automatic zooming) technique [27]. When using it to scroll documents, the documents are automatically zoomed-out as the scroll rate increases. By automatically zooming, the visual flow rate is reduced, enabling rapid scrolling without motion blur.

Baudisch et al and MacKay et al. explore navigation techniques in mobile devices with small screens [12] [70]. Baudisch et al. propose a Collapse-to-Zoom technique which exploits strokes to delete the unimportant contents and highlight the selected ones to enhance navigation performances [12]. MacKay et al. present a field study comparing software-based navigation techniques (Scroll Bar, tap-and drag, and touch-n-go) on mobile devices [70]. They explored the efficiency and user preference of these navigation techniques for different levels of mobility (sitting, walking and standing).

The works that more related to ours are the Smith et al. and Moscovich et al. Smith et al. present a radial scrolling tool and an eyes-free parameter entry for GUI's called curvature dial, which exploit circular gestures in the interface to scroll documents, advancing or reversing the document by a clockwise or counterclockwise circular stroke gestures [104]. Moscovich et al. present a virtual scroll ring which simulates the hardware scroll ring-a device that maps circular finger motion into vertical scrolling, to navigate the document [82]. Both of them initialize kind of arc strokes to perform scrolling tasks. Compared to Scroll bars, these techniques were found to be faster for short distances, but slower for long distances.

The distinct differences between this study and others is that a speed mode related to the length of a document is applied into the stroke-based techniques, speeding up scrolling in long documents. We also explore to use line stokes in scrolling tasks and compare these two types of stroke-based techniques in different scrolling tasks (vertical and horizontal) and screens with different sizes.

8.3 Stroke-based Techniques for Scrolling Tasks

Stroke-based techniques provide two speed modes: one is independent of the length of the scrolled document; the other is proportional to the length of the scrolled document. We present a widget called pie that transparently overlaps at the top of a document. This widget is activated when the display surface is double clicked. The user then draws an arc or line stroke to scroll the document. If the user double clicks elsewhere, the widget is hidden (see Fig.8.2).

8.3.1 Arc Stroke

The user can scroll a document by drawing an arc stroke gesture around the center point of the pie widget. When the pie widget is activated and the point on the screen which is initially tapped by the tip of the stylus becomes the center point. The coordinates of the point are recorded. This information is used, along with history information recorded when the arc stroke is drawn, to determine the direction and the displacement of the scrolled document. We represent the view space in polar coordinates, with the center point as the origin. Two vectors indicate the previous and current points passed by the stylus. The scrolling direction is determined by the sign of the dot product of the previous vector $\mathbf{A}$ and the current vector $\mathbf{B}$. If the sign is positive the direction of rotation is clockwise, and the movement direction is up. The angle difference between the two vectors determines the amount of movement. Equation8.1 shows the calculation of the angle difference between the two vectors. θ is the angle difference.

$$\cos\theta = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}||\mathbf{B}|} \tag{8.1}$$

8.3.2 Line Stroke

The user can scroll a document by drawing a line. The direction of the line stroke determines the scrolling direction of the document: if the direction of the line stroke is up, the document will scroll up, while drawing the stroke in a down causes the document to scroll down. For vertical scrolling, the vertical difference between the previous and current stylus positions determines the scrolling displacement; for horizontal scrolling the horizontal difference determines the displacement.

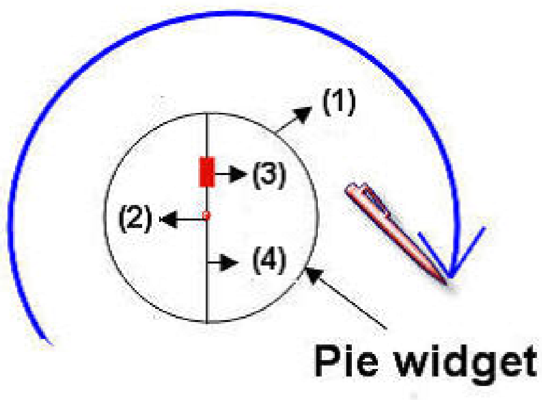

Fig. 8.2 Arc draws arc strokes around a fixed point. The Pie Widget is a transparent overlap on a document. It is composed of four parts: (1) Pie circle, (2) Center point, (3) Position block and, (4) Direction pole

8.3.3 Independent and Dependent Modes

Our length dependent speed mode is inspired by scroll bars, so we first analyze the length dependent speed mode of scroll bars.

When using a scroll bar to scroll a document, the user drags the scroll bar "elevator" to quickly locate the approximate position of a target and then clicks the "arrow" buttons to precisely place it in the view. The speed at which the "elevator" is dragged is dependent on the length of the document. Equation.8.2 denotes the formula for scroll speed. Note that the drag speed and the length of the document together determine the scrolling speed. V_{doc} denotes the movement speed of the document; V_{ele} denotes the dragging speed of the elevator; L denotes the length of the document; D, a constant, denotes the length of the scroll bar; k is a proportional coefficient.

$$V_{doc} = \frac{k \cdot L^2}{D \cdot L - D^2} \cdot V_{ele} \, (L > D) \tag{8.2}$$

On the other hand, the speed at which the arrow button is pressed to scroll the document is independent of the length of the document.

Our stroke-based scrolling uses a document size-dependent speed mode. In Equation.8.3 for Arc, V denotes the angular stroke speed; for Line, it denotes the vertical or horizontal component of the stroke speed. C is a constant; the other coefficients are the same as those in Equation.8.2.

$$V_{doc} = \frac{k \cdot L^2}{C \cdot L - C^2} \cdot V \; (L > C) \tag{8.3}$$

As for the independent speed mode, the speed of scrolling is just determined by the speed of the stroke. In Equation.8.4 for Arc for Line, it denotes the vertical or horizontal component of the speed of the stroke. k denotes the proportional coefficient.

$$V_{doc} = k \cdot V \tag{8.4}$$

In the following two experiments, Arc and Line were set at the independent mode (Equation.8.4) in Experiments 1 , and the dependent mode (Equation.8.3) in Experiment 2.

8.4 Experiment 1: Independent Mode

The goal of Experiment 1 was to evaluate the time efficiency and error rates of the Arc, Line and Scroll Bar techniques on small, medium and large screen devices and to determine whether the screen size affects the experimental results. In most user interfaces vertical scrolling tasks are dominant, however, on some situations horizontal scrolling is required. So we designed both vertical scrolling tasks and horizontal scrolling tasks in Experiment 1.

8.4.1 Participants

Ten subjects (9 males, 1 female) who had all had previous experience with computers were tested for the experiment. The average age was 22.7 years. All subjects were right handed, and used the pen in the right hand. Three of them had had about 1

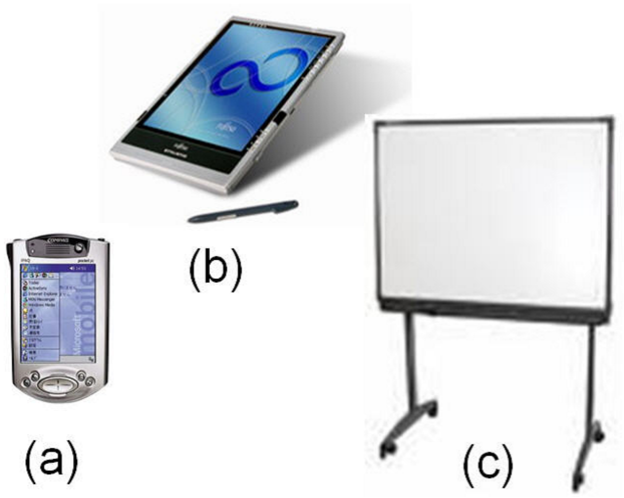

Fig. 8.3 Devices used in experiments: (a) PDAs (small size); (b) Tablet PCs (medium size); (c) Whiteboards (large size).

- 1.5 years of experience in pen-based devices use. The others had no experience with pen-based devices.

8.4.2 Apparatus

We tested the scrolling techniques with three pieces of hardware: a PDA, a Tablet PC, and a digital whiteboard (see Fig.8.3). The PDA was an iPAQ Pocket PC running Windows CE 3.0. It weighed about 190g, and was 84 mm (W) x 16 mm (D) x 134 mm (H). The spatial resolution of the screen was 0.24 mm/pixel. The Tablet PC was a Fujitsu, running Microsoft Windows XP. It weighed about 1.48kg, and was 210.43 mm (W) x 157.82mm (H). The digital whiteboard was a Team board UCHIDA whiteboard that was 1545mm (W) x 1260mm (H) and used a computer with 1.0 GHz Pentium Processor. The software was written in Java.

8.4.3 Design

We conducted the experiment using small, medium and large size pen-based interfaces comparing the Arc and Line techniques with a Scroll Bar, both with vertical and horizontal scrolling. Closely following Hinckley et al. [48], we evaluated our technique using a reciprocal framing task. In the vertical scrolling tasks, subjects scrolled down, then scrolled up, moving back and forth between two lines marked "START" or "END" respectively. We colored the "START" target line green and the "END" target line red. A similar task should also be suitable for horizontal scrolling. In Fig.8.3 the red "frame" at the left and right of the experiment interface specified a certain width. For each target, subjects scrolled until the target entered the range of the screen identified by the frame. The frame was always centered on the screen and its width was 3 lines. Once the target line was fully within the frame, the subject hit the arrow key with the left hand. This selection key let the experiment interface know when the user judged the scrolling to be completed. If the target line successfully fell within the frame when the user struck the selection key, there was a short beep sound. If not, the subject did not hear any sound the subject moved on to the next target. We had previously instructed the subjects to continue to the next target rather than try to repair errors.

The design of the experiment crossed Method (M) x Scrolling distance (D) x Frame height (H) x Direction (Di) (see Table8.1).

The frame height is 3 lines.

8.4.4 Procedure

Participants read instructions describing the experimental setup and tasks. The experimenter then reviewed these instructions with the subjects and introduced the three methods. In order to familiarize them with the experimental environment, we

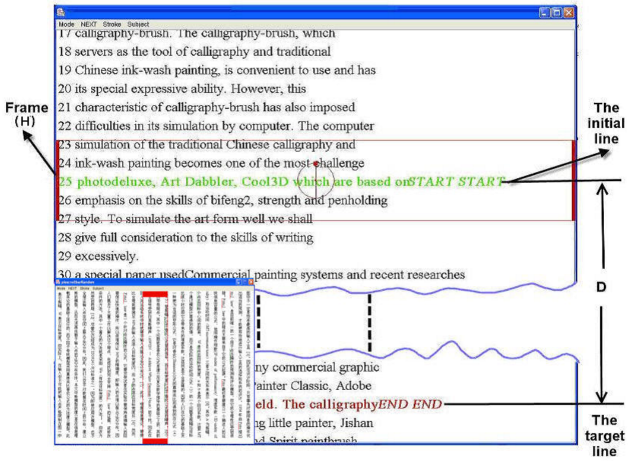

Fig. 8.4 The experimental interface for the reciprocal framing vertical tasks. The insert picture shows the horizontal tasks.

Table 8.1 The scrolling distances of experimental tasks.

Devices	Vertical tasks (lines)	Horizontal tasks (lines)
PDAs	12, 36, 48, 69, 123, 156, 196	12, 36, 48, 69, 123, 156, 196
Tablet PCs	24, 56, 96, 256, 416, 570	24, 68, 122, 256, 416, 570
Whiteboards	24, 56, 96, 256, 416, 570	24, 68, 122, 256, 416, 570

set a practice session for each method. When subjects understood the task and could perform it correctly; subjects completed the experiment unassisted. For each method, subjects performed 7 trials for each of distance-width combinations which appeared in random order. Participants typically spent 40 to 50 minutes using the three methods.

8.4.5 Result

An ANOVA (analysis of variance) with repeated measures was used to analyze performance in terms of scrolling time and error rates. Post hoc analysis was performed with Turkey's honestly significant difference (HSD) test.

To rationally indicate experimental results, based on preliminary analysis, we classified all the tasks into two types of tasks: short distance tasks (less than 150 lines) and long distance tasks (more than 150 lines).

On the Tablet PC

Vertical scrolling task: There was a significant interaction between methods and distances, $F(10,162)=19.2$, $p < 0.01$. There were significant differences between the three methods in the mean completion time for short distance tasks $F(2,27)=5.41$, $p < 0.05$. The post hoc Turkey HSD test showed Line was faster than Arc and Scroll Bar in movement time. Arc was faster than Scroll Bar in movement time. There was a significant difference in the mean completion time for long distance tasks, $F(2,27)=6.53$, $p < 0.05$. The post hoc Turkey HSD test showed Scroll Bar was faster than Arc and Line. There was no significant difference in movement time between Arc and Line (see Fig.8.4). The error rates were Arc, 4.1%; Line, 4.7%; Scroll Bar, 3.9%. There was no significant difference between the three methods in error rate.

Horizontal scrolling task: There was a significant difference between methods and distances, $F(10,162)=23.5$, $p < 0.01$. There was a significant difference between the three methods in the mean completion time for short distance tasks $F(2,27)=4.28$, $p < 0.05$. The post hoc Turkey HSD test showed Line was faster than Arc and Scroll Bar in movement time. Arc was faster than Scroll Bar in movement time. There was a significant difference in the mean completion time for long distance tasks, $F(2,27)=6.38$,

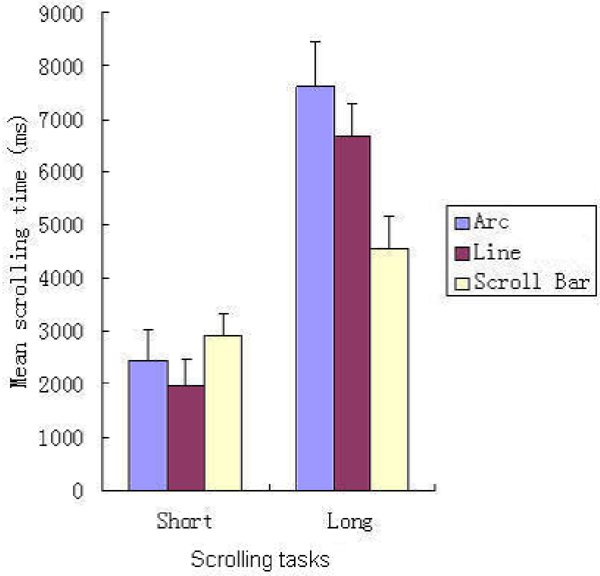

Fig. 8.5 The mean movement time for different types of vertical scrolling tasks.

$p < 0.05$. The post hoc Turkey HSD test showed that Scroll Bar was faster than Arc and Line. There was no significant difference in movement time between Arc and Line (see Fig.8.5). The error rates were Arc, 3.5%; Line, 3.8%; Scroll Bar, 2.9%. There was no significant difference between the three methods in error rate.

On the PDA or Whiteboard

The results of PDA and Whiteboard followed similar trend to that of Tablet PC.

Subjective evaluation

The analysis of the questionnaire showed a significant main effect on subject preferences between Arc-Free, Line and Scroll Bar, $F(2,27)=21.53$, $p<0.01$. Line was the most preferred (mean=1.6)(see Table8.2).

From the results we can conclude that for short distances the stroke technique is

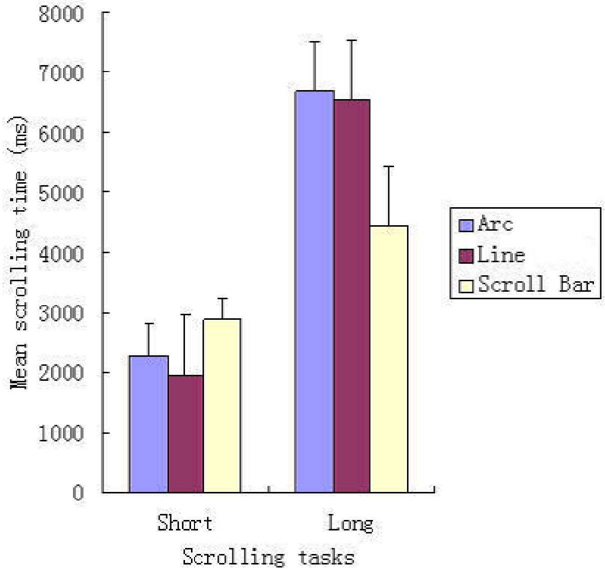

Fig. 8.6 The mean movement time for different types of horizontal scrolling tasks.

Table 8.2 Mean subjective rating, from -3 (most negative) to 3 (most positive). "V" means the vertical task; "H" means the horizontal task.

		Arc	Line	Scroll Bar
PDAs	V	1.2	1.3	-0.2
	H	1.3	1.1	-0.5
Tablet PCs	V	0.7	1.9	-0.4
	H	1.2	1.8	-0.4
Whiteboards	V	1.0	1.8	-0.8
	H	0.9	1.6	-0.6

better than Scroll Bar technique, and the Line technique is better than the Arc technique in movement time; for long distances Scroll Bar is better than the stroke techniques. The size of the screen and the scrolling direction (vertical or horizontal scrolling) have no significant effect on the evaluation result of the three techniques.

8.5 Experiment 2: Dependent Mode

The results of Experiment 1 showed that Scroll Bar was better than Arc and Line for long distances. This is due to the reason that Scroll Bar is a dependent of the length of document technique. So in Experiment 2 the stroke techniques were set at dependent speed mode (see section 3.3). We evaluated the stroke techniques and Scroll Bar again at this setting.

8.5.1 Participants

Ten subjects who had all had previous experience with computers were tested for the experiment. The average age was 22.8 years. Three of them were tested in Experiment 1. All subjects had normal or corrected to normal vision with no color blindness, were right handed, and used the pen in the right hand.

8.5.2 Apparatus

The hardware in Experiment 2 was the tablet PC that was the same as that in Experiment 1.

8.5.3 Procedure and Design

The design of the experiment crossed Method (M) x Scrolling distance (D) x Frame height (H) x Direction (Di). For vertical scrolling tasks we chose six representative short distances: 48, 72, 96, 256, 416 and 570 lines and one frame height: 3 lines. For horizontal scrolling tasks we chose six representative distances: 36, 72, 122, 256, 416 and 570 lines and one frame height: 3 lines. All other factors remained the same as them in Experiment 1.

8.5.4 Results

To better analyze the effects to scrolling time brought document length (lines) a linear model [8] for evaluating scrolling tasks has been introduced:

$$T = a + b \cdot D \tag{8.5}$$

where a and b are empirically determined constants, and D is the distance to target. In our experiment, D was the distance between the initial line and the target line (see fig.8.6). We used this model to analyze the experimental results.

Vertical scrolling task

There was a significant difference between methods and distances, $F(10,162)=21.2$, $p<0.01$. There was no significant difference between the three methods in the mean completion time for short distance tasks. There was a significant difference in the mean completion time for long distance tasks, $F(2,27)=5.42$, $p<0.05$. The post hoc Turkey HSD test showed both Arc and Line were faster than Scroll Bar. There was no significant difference in movement time between Arc and Line (see Fig.8.7). The error rates were Arc, 4.3%; Line, 4.8%; Scroll Bar, 3.8%. There was no significant difference between the three methods in error rate.

Horizontal scrolling task

There was a significant difference between methods and distances, $F(10,162)=25.1$, $p<0.01$. There was no significant difference between the three methods in the mean completion time for short distance tasks. There was a significant difference in the mean completion time for long distance tasks, $F(2,27)=6.19$, $p<0.05$. The post hoc Turkey HSD test showed both Arc and Line were faster than Scroll Bar. There was no significant

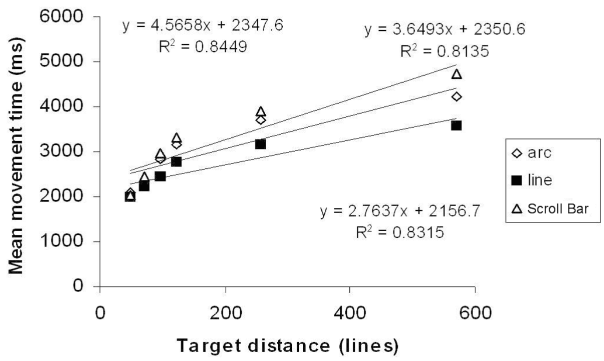

Fig. 8.7 Vertical task: Line regression of target distance against movement time.

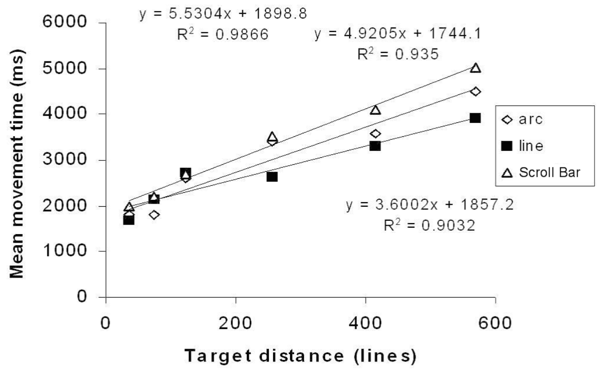

Fig. 8.8 Horizontal task: Line regression of target distance against movement time.

difference in movement time between Arc and Line (see Fig.8.8). The error rates were Arc, 4.1%; Line, 4.9%; Scroll Bar, 3.6%. There was no significant difference between the three methods in error rate.

As shown in Fig.8.5 and Fig.8.6, when stoke-based technqiues was set at independent mode Line and Arc outperformed Scroll Bar for long documents.

Subjective evaluation

The analysis of the questionnaire showed a significant main effect on subject preferences between Arc, Line and Scroll Bar, $F(2,27)=18.11$, $p<0.01$. Line was the most preferred (Table8.3).

Table 8.3 Mean subjective rating, from -3 (most negative) to 3 (most positive). "V" means the vertical task; "H" means the horizontal task.

	Arc	Line	Scroll Bar
V	1.0	2.1	-1.1
H	1.3	1.9	-1.3

8.6 Discussion and Conclusions

We present a quantitative analysis of stroke-based techniques for scrolling tasks in pen-based interfaces. The stroke-based techniques are classified into Arc and Line according to the mode of direction control and scrolling displacement of documents. We designed two experiments to evaluate the stroke-based techniques by comparing them with Scroll Bar.

At the beginning of this chapter, we posed several basic questions concerning stroke-based scrolling techniques. The results from the two experiments conducted can begin to address these questions.

- What is the difference between arc and line strokes in the performance of scrolling tasks?

Experiment 1 showed that for short distances Line was faster than Arc; for long distances there was no significant difference between Line and Arc in performance time.

In Experiment 2 the stroke techniques were set in the dependent speed mode. The results of Experiment 2 indicated that no significant difference was found between Line and Arc. Line is preferred by users.

The Line technique is in accord with the user's visual habit in that the scrolling direction of documents is the same as the direction of the strokes. Line is also easier to master than Arc and scrolling speed is easier to control with the Line technique. Although Line is better than Arc in movement time in some situations and it is preferred by users, Arc still has some specific advantages: the user can trace an arc or circle in a smaller space to manipulate scrolling tasks while with the Line technique, the user has to constantly lift and press the pen-tip.

Interface techniques with consistency and with familiar presentation and gestures can make the user feel more comfortable. Line matches both these requirements. Arc has the advantage of saving space, but lacks operational consistency. Designers should consider these requirements when designing user interfaces.

- Can the application of speed modes (independent or dependent modes) related to the length of the document offer benefits to stroke-based techniques?

Experiment 1 showed that for short distances the stroke techniques were better than Scroll Bar in movement time; for long distances Scroll Bar was faster than the stroke-based techniques in movement time. This is due to the fact that Scroll Bar is a technique that is dependent on the length of a document so, when documents are long its scrolling speed becomes correspondingly faster. In Experiment 2 the stroke techniques were set in the dependent speed mode. The results of Experiment 2 indicated that, for long distances, Scroll Bar was significantly slower than the stroke techniques in movement time. In essence, the dependent mode reflects a concept of adaptive interface, which has the benefit of adapting the technique to the particular computing task. Therefore, Line or Arc set in this mode enhance scrolling performances.

- Do the size of the device screens and the scrolling directions (vertical or horizontal scrolling) have effects on the evaluation result of the three scrolling techniques?

Experiment 1 results suggested that stroke-based techniques with Scroll Bar on medium sized screens followed trends similar to those in small or large screens. For vertical and horizontal tasks the trends were also similar. Therefore, the size of the screen and the scrolling direction (vertical or horizontal) had no significant effect on the evaluation results of the three techniques.

Line is derived from Scroll Bar in that the line that the pen follows when dragging the “elevator” is similar to the line stroke in the Line method. Line is better than Scroll Bar because the user does not have to select the “elevator” but simply draws strokes anywhere in the displayed document to scroll the document. In this way Line reduces the index of difficulty in scrolling performance.

Generally, the operations for documents manipulation include the edit operation and the scrolling operation. Thus, if the user wants to use stroke-based techniques to scroll documents, a switching mechanism is needed. For example, we can use a double-click or a key using the non-dominant hand [**?**] to switch between the two operation modes. In fact, this belongs to the issue on mode-switch in pen-based interfaces. This point warrants in-depth attention and it has been earmarked for our future research.

The stroke-based techniques are better than Scroll Bar in that strokes are a much more natural interactive mode for stylus pens. Furthermore, when Scroll Bar in being used, the user has to switch attention between the document and the scroll bar and this interrupts the user’s workflow.

The positive results from our experiments suggest that the stroke-based techniques could be a beneficial addition to the current stock of user interfaces. We can design a plug-in to incorporate these techniques into user interfaces. This would be very appropriate to adopt "selection" mode. A simple icon or menu item could be used to activate this useful scrolling interface.

Chapter 9

General Conclusions and Future Directions

9.1 General Conclusions

9.1.1 Summary

Development in technology is continuously expanding the design space for pen computing interfaces. In particular, there is a lack of a mature and suitable pen computing OS. It is both theoretically and practically necessary to understand how users' performance relates to multiple input modalities (such as pressure) sensed from pen devices. It is also necessary to study interaction techniques that suit for pen devices. This thesis investigates human abilities of controlling various input modalities and proposes a series of pen-based interaction techniques to advance interaction performances of pen computing.

Two pen input modalities (pressure and tilt) are investigated. Chapter 2 presents a series of three experiments that evaluate human capabilities and limitations in using pen-tip pressure as an additional channel of control information in carrying out trajectory tasks such as drawing, writing, and gesturing. The first experiment measured the natural range of force used in regular drawing and writing tasks. The second experiment tested human performance of maintaining pen-tip pressure at different levels with and without a visual display of the pen pressure. The third experiment, using the steering law paradigm, studied path steering performance as a function of the steering law index of difficulty, steering path type (linear and circular), and pressure precision tolerance interval. The main conclusions of our investigation are the following: The natural range of pressure used in drawing and writing is concentrated in the 0.82N to 3.16N region. The resting force of the pen tip on the screen is between 0.78N and 1.58N. Pressure near or below the resting force is markedly more difficult to control. Visual feedback improves pressure-modulated trajectory tasks. Up to 6 layers of pressure can be controlled in steering tasks, but the error rate changed from 4.9% for one layer of pressure to 35% for six layers. The steering law holds for pressure steering tasks, which enables systematic prediction of successful steering time for a given paths length, width, and pressure precision criterion. The steering time can also be modeled as a logarithmic function of pressure control precision ratio σ. Taken together, the current work provides a systematic body of empirical knowledge as basis for future research and design of pen-pressure applications.

Chapter 3 investigates the use of pen tilt angle as auxiliary in human-computer interaction. An experiment was conducted to explore the human capability of performing discrete target selection tasks by varying the pen tilt angle, with full or partial visual feedback. The experiment included different techniques for confirming selection once the target is acquired. The results suggest that an angle for a radial target (sector) of more than 10 degrees can achieve good performance and that the technique of pressing a button with the non-preferred hand offers the best performance. Based on the experimental results, we discuss implications for the design of pen tilt techniques. A classification of pen tilt based techniques is proposed, along with a series of possible pen tilt technique designs.

The thesis also designs and develops interaction techniques that are suitable for pen-based systems. Chapter 4 investigates the interaction ability of introducing pressure into current basic interactive techniques by developing two novel techniques. A Zoom-based technique with pressure (hereafter referred to as ZWPS) is proposed to improve pixel-target selection. In this technique pressure is used as a switch mode to couple a standard Point Cursor and a zoomable technique together. Pressure Scroll is also presented with a view to advancing scrolling performances by employing arc or line strokes to scroll documents. In this technique pressure is used as an additional control factor to widen the adjustable range of scrolling velocity. We conducted two experiments to examine the effectiveness of ZWPS and Pressure Scroll. The experimental results indicate that they both bring significant benefits to users.

Chapter 5 introduces a novel interaction technique that improves target acquisition in pen-based interfaces. This technique is called Beam Cursor. The Beam Cursor exploits the sliding motion and dynamically updates the effective width of targets on screen according to the original location of the pen-tip, such that even if the pen-tip lands in the vicinity of a target the target can easily be selected. This chapter also provides reports on two controlled experiments which were performed to evaluate the Beam Cursor in both 1D (dimension) and 2D target selection tasks on the pen-based interface. The experimental results indicate that the Beam Cursor is modeled on and predicted by Fitts law and that it is governed by the effective width of the targets.

Chapter 6 presents the Adaptive Hybrid Cursor, a novel target acquisition technique for pen-based interfaces. To assist a user in a target selection task, this technique automatically adapts the size of the cursor and/or its contexts (the target size and the selection background) based on pen pressure input. This chapter systematically evaluated the new technique with various 2D target acquisition tasks. The experimental results indicated that the Adaptive Hybrid Cursor had better selection performance, and was particularly effective for small-target and high-density environments in which the regular cursor and the Bubble Cursor [38] failed to show significant advantages. It is a novel way to improve target acquisition via pressure input, and our study demonstrated its viability and potential for pen-based interfaces.

Chapter 7 presents three selection techniques (called Rubber-ling-sweep, Line-string and Coupling-with-pressure) to enhance multi-target acquisition in GUIs and to overcome the drawback of the standard rubber-band box technique, i.e., the limitation of not being able to select an irregular layout of targets. Rubber-line-sweep utilizes a rubber-band line to select targets by sweeping them. Line-string employs a line stroke to string targets together and select them. Coupling-with-pressure couples these two techniques with pressure as a switch mode. Experiments were conducted to compare these techniques with the standard Rubber-band box, which used a two-dimensional grid which could include varied target sizes, distances and target layouts, and which is applied by using pens as input devices. Experimental results indicate that Rubber-line-sweep, Line-string and Coupling-with-pressure show significant advantages for targets with irregular layouts. Taking performance and subjective ratings together, Coupling-with-pressure outperforms the other three techniques.

Chapter 8 presents a quantitative analysis of stroke-based techniques for scrolling in pen-based interfaces, and compares them with traditional scroll bars. We classified stroke-based techniques into two types: the Arc stroke technique uses angles to determine scrolling direction and the displacement of scrolling, and the Line stroke technique uses distance to determine both the direction and the displacement of scrolling. Experiments were conducted to evaluate stoke-based scrolling techniques in small, medium and large screens respectively. This study also applied a speed mode to the stroke-based techniques. The evaluation results indicate that the Line technique outperforms both the Arc technique and the traditional Scroll Bar. The size of the screen and the scrolling direction (vertical or horizontal scrolling) have no significant effect on the evaluation results of the three techniques.

9.1.2 Contributions

The fundamental studies in this thesis significantly contribute to the understanding of human abilities of controlling various input modalities in pen computing. The highlights of the results of fundamental studies can be summarized briefly as: (1) the physical properties of multiple input modalities should provide rich feedback so that the user can easily feel her control actions proprioceptively. (2) Quantitatively analyzing human abilities of controlling the multiple input modalities in performance tasks. (3) a proper mapping function is essential for coupling the multiple input modalities with current interaction techniques(4) A taxonomy for the design of multiple input modalities-based interaction techniques.

This thesis also proposes a variety of interaction techniques for enhancing selection or navigation tasks. We also conducted the experiments to evaluate these techniques, comparing with the current promising techniques or standard techniques in GUIs. The experimental results indicate that these techniques offer significant advantages to pen interface design.

9.2 Future Directions

This thesis investigates human ability of controlling pressure in trajectory-based tasks. However, there are several directions that can be pursued to extend the current work. As stated in the literature review, there is a lack of psychological studies of force in a manner that can be applied to pen-tip pressure. In particular, it would be interesting to know whether Webers law applies here. If it does, it would make sense to divide the usable force range unequally, giving higher pressure tolerance intervals to higher pressure levels. The results in our experiment do not suggest that Webers law applies to pen-tip pressure since when the force range was divided equally; higher pressure levels (or higher reference values in Webers law terms) did not necessarily perform more poorly than lower levels. In short, the validity of Webers law in the case of pen-tip force and its design implications require further research in the future.

This thesis examines basic issues on pen tilt input. Pens can also provide azimuth input information. Pen azimuth input has also been used in some applications. The study of the human ability to use pen azimuth remains for future work. The future work also includes investigations on the combinations of the different pen input modalities.

This thesis proposes a series of pen-based interaction techniques, which are aimed at the most basic computing performances: selection and navigation tasks. In the future we seek to couple pen devices characteristics and various input modalities with other interaction performances such as basic widgets operations.

Acknowledgement

First, I would like to thank my advisor Professor Xiangshi Ren, for his continuous support in my pursuing the Ph.D. degree. Professor Xiangshi Ren was always there to listen and to give advice. He is responsible for involving me in Human Computer Interaction in the first place. He taught me how to ask questions and express my ideas. He showed me different ways to approach a research problem and the need to be persistent to accomplish any goal. His dedication to work, perfectionism, humour and decency will have a far reaching impact on my life, academic and beyond.

Next, I am also very grateful to Professor Kimura Yoshimasa, Professor Okada Mamoru, Professor Shinomori Keizo and Professor Shimizu Akihiro for their insightful comments on my thesis.

Meanwhile, I would like to thank my family members, particularly my mother for their generous love and support to my life all along.

I'd like to thank members of the Ren lab in the Department of Information Systems Engineering, particularly Xinyong Zhang, Jing Kong, Tsuchida Tomoaki, Tomoki Ooya, Fukutoku Fumiya, Chuanyi Liu, Xiaolei Zhou and Yizhong Xin for their friendship and contributions to my work.

I have received support in the form of scholarships, facilities or travel grant from TAF(Telecommunications Advancement Foundation), IISF(International Information Science Foundation) and Kochi University of Technology. This research would have been impossible without that generous support.

This is a blank page.

References

[1] Accot, J., and Zhai, S. 1997. Beyond Fitts Law: Models for Trajectory-Based HCI Tasks. *In Proc.of CHI97.* pp. 295- 302.

[2] Accot, J. and Zhai, S. 1999. Performance Evaluations of Input Devices in Trajectory-based Tasks: An application of the Steering Law. *In Proc. CHI'99.* 466-472.

[3] Accot, J., and Zhai, S. 2001. Scale Effects in Steering Law Tasks. *In Proc.of CHI2001.* pp.1-8.

[4] Accot, J. and Zhai, S. 2002. More than dotting the i's - foundations for crossing-based interfaces. *In Proc.of CHI2002.*. pp.73 - 80.

[5] Ahlberg, C., and Shneiderman, B.1994. AlphaSlider: A compact and rapid selector,*In Proc. of CHI'94*, pp. 365-371.

[6] Ahlstrom, D. 2005. Modeling and Improving Selection in Cascading Pull-down Menus Using Fitts Law, the Steering Law and Force Fields. *In Proc.of CHI2005.* pp.61-70.

[7] Albinsson, P-A. and Zhai, S. 2003. High Precision Touch Screen Interaction. *In Proc.of CHI2003.* pp. 105-112.

[8] Anderson, T. 2005. A Simple Movement Time Model for Scrolling. *In Proc. of CHI2005.* pp. 1180-1183.

[9] Aliakseyeu, D., Nacenta M.A., Subramanian, S. and Gutwin, C. 2006. Bubble radar: efficient pen-based interaction. *In Proc. the working conference on Advanced visual interfaces.* pp.19-26.

[10] Arvo, J. and Novins, K.. 2000 Fluid sketches: Continuous recognition and morphing of simple hand-drawn shapes, *In Proc. of UIST'00*, pp. 73-80.

[11] Aurenhammer, F. and Klein, R. 2000. Voronoi Diagrams. Chapter 5, in Handbook of computational geometry, J. Sack and J. Urrutia, Editors. North- Holland: Amsterdam, Netherlands, pp. 201-290.

[12] Bandisch, P., Xie, X., Wang, C., and Ma, W. 2004. Collapse-to-Zoom: Viewing Web Pages on Small Screen Devices by Interactively Removing Irrelevant Content. *In Proc. of UIST2004.* pp. 91-94.

[13] Bartlett, J.F. 2000. Rock n Scroll is Here to Stay. *IEEE Computer. Graphics and Applications. 20(3).* pp. 40-50.

[14] Barrett, R., Olyha, J., Robert S., and Rutledge, J. 1996. Graphical User Interface Cursor Positioning Device Having a Negative Inertia Transfer Function. *Patent ♯ 5,570,111, IBM Corp.*

[15] Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A. 2003. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and penoperated systems. *In Proc. of Interact2003.* pp. 57-64.

[16] Baudisch, P., Cutrell, P., Hinckley, K., and Eversole, A. 2005. Snap-and-go: helping users align objects without the modality of traditional snapping. *In Proc. of CHI2005.* pp.301-310.

[17] Bezerianos, A., and Balakrishnan, R. 2005. The Vacuum: Facilitating the Manipulation of Distant Objects. *In proc. of CHI2005.* pp. 361-370.

[18] Brown, D.R., and Owen, D.H. 1967. Metrics of Visual Form: Methodological Dyspepsia. *Psychological Bulletin, 68.* pp.243-259.

[19] Blanch, R., Guiard, Y., and Beaudouin-Lafon, M. 2004. Semantic pointing: improving target acquisition with control-display ratio adaptation. *In proc. of CHI2004.* pp. 519-525.

[20] Blasko, G., Beaver, William., Kamvar, M., and Feiner, Steven. 2004. Workplane-

Orientation Sensing Techniques for Tablet PCs. *In Proc. of UIST2004.* pp. 24-27.

[21] Buxton, W., Hill, R., and Rowley, P. 1985. Issues and techniques in touch sensitive tablet input. *In Proc. of ACM SIGGRAPH Conference*, pp. 215-224.

[22] Buxton, A.S.W. 1990. Three-State Model of Graphical Input in Human-Computer Interaction. *In Proc. of INTERACT'90.* pp.449-456.

[23] Card, S. K., English, W. K., and Burr, B. J. 1978. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics, 21.* pp.601-613.

[24] Chipman, S. F. 1977. Complexity and structure in visual patterns. *Journal of Experimental Psychology: General, 106, 3.* pp. 269-301.

[25] Cockburn, A. and Firth, A. 2003. Improving the acquisition of small targets. *In Proc. of British HCI Conference.* pp. 181-196.

[26] Cockburn, A. and Brock, P. 2006. Human on-line response to visual and motor target expansion. *In Proc. of Graphics Interface 2006.* pp.81-87.

[27] Cockburn, A. and Savage, J. 2003. Comparing Speed- Dependent Automatic Zooming with Traditional Scroll, Pan and Zoom Methods. *In Proc. of BCS HCI '03*, pp. 87-102.

[28] Cockburn, A., Savage, J. and Wallace, A. 2005. Tuning and Testing Scrolling Interfaces that Automatically Zoom, *In Proc. of BCS HCI '05*, pp. 87-102 (2005).

[29] Collomb, M., Hascoet, M., Baudisch, P., and Lee, B. 2005. Improving drag-and-drop on wall-size displays. *In Proc. of Graphics Interface 2005.* pp.25-32.

[30] Dennerlein, J.T., Martin, D.B. and Hasser, C. 2000. Force-feedback improves performance for steering and combined steering-targeting tasks. *In Proc. of CHI2000.* pp.423 - 429.

[31] Eissele, M., Stegmaier, S., Weiskopf, D., and Ertl, T. 2004. Orientation as an additional User Interface in Mixed-Reality Environments. *Workshop GI-Fachgruppe*

AR/VR. pp. 79-90.

[32] Fishkin, K.P., Moran, T.P., and Harrison, B.L.1998. Embodied User Interfaces: Towards invisible user interfaces. *In proc. of Engineering for Human-Computer Interaction: SeventhWorking Conference on Engineering for Human-Computer Interaction.* pp.1C18.

[33] Fitts, P.M. and Seeger, C.M. 1953. S-R compatibility: spatial characteristics of stimulus and response codes. *Journal of Experimental Psychology, 46.* pp.199-210.

[34] Fitts, P.M. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology.* pp. 381-391.

[35] Fitzmaurice, G. W., Zhai, S., and Chignell, M.H. 1993. Virtual Reality for palmtop computers. *ACM Transactions on Information Systems, Vol. 11, No. 3.* pp. 197218.

[36] Fitzmaurice, G. W., Balakrishnan, R., Kurtenbach, G., and Buxton, B. 1999. An exploration into supporting artwork orientation in the user interface. *In Proc. of CHI99.* pp.167-174.

[37] Foley, J.D., Wallace, V., and Chan. P. 1984. The Human Factors of Computer Graphics Interaction Techniques. *IEEE Computer Graphics and Applications.* pp.13-48.

[38] Grossman, T., and Balakrishnan, R. 2005. The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursors Activation Area. *In proc. of CHI2005.* pp. 266C 271.

[39] Goldberg, A., ed. 1988. A History of Personal Workstations. *Addison-Wesley Publishing Company0: New York, NY.* pp.537.

[40] Guan, Z., Ren, X., Li, Y., and Dai, G. 2004. Zoom Selector: A Pen-based Interaction Technique for Small Target Selection *IPSJ Journal.* pp 2087-2097.

[41] Guiard, Y., Blanch, R., and Beaudouin-Lafon, M. 2004. Object pointing: a complement to bitmap pointing in GUIs. *Gaphics Interface.* pp. 9-16.

[42] Guiard, Y. and Ferrand, T. 1987. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behavior 1987. 19(4).* pp. 486C517.

[43] Gutwin, C. and Skopik, A. 2003. Fisheye views are good for large steering tasks. *In Proc. of CHI2003.* 201-208.

[44] Herot, C., and Weinzapfel, G. 1978. One-point touch input of vector information for computer displays. *In Proc. of ACM SIGGRAPH Conference.* pp.210-216.

[45] Guimbretiere, F. and Winograd, T. 2000. Flowmenu: Combining command, text, and data entry, *In Proc. of UIST'00*, pp. 213-216.

[46] Harrison, B.L., Fishkin, K.P., Gujar, A., Mochon, C., and Want, R. 1998. Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces. *In Proc. of CHI '98.* pp.17-24.

[47] Hinckley, K., Baudisch, P., Ramos, G., and Guimbretiere, F. 2005. Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in Scriboli. *In Proc. of CHI2005.* pp. 451-460.

[48] Hinckley, K., Cutrell, E., Bathiche, S., and Muss, T. 2002. Quantitative Analysis of Scrolling Techniques. *In proc. of CHI2002.* pp. 65-72.

[49] Hinckley, K., Pierce, J., Sinclair, M., and Horvitz., E. 2000. Sensing Techniques for Mobile Interaction. *In Proc. of UIST 2000.* pp. 91-100.

[50] Igarashi, T., and Hinckley, K. 2000. Speed-dependent automatic zooming for browsing large documents. *In proc. UIST2000.* pp.139-148.

[51] http://www.crazysquirrel.com/radial-scroll/index.php

[52] ISO ISO9241-9: Ergonomic design for office work with visual display terminals (VDTs)-Part 9: Requirements for non-keyboard input devices. International Standardization Organization, 2000.

[53] Kay, A. 1977. Personal Dynamic Media. *IEEE Computer, 1977. 10(3).* pp. 31-42.

[54] Kabbash, P., Buxton, W., and Sellen, A.G. 1994. Two-Handed Input in a Compound Task. *In Proc. of CHI'94.* pp. 417-423.

[55] Kabbash, P. and Buxton, W. 1995. The "Prince" technique: Fitts' law and selection using area cursors. *In proc. ofCHI'95.* pp. 273-279.

[56] Keyson, D. 1997. Dynamic cursor gain and tactual feedback in the capture of cursor movements. *Ergonomics, Vol.12.* pp. 1287-1298.

[57] Klein, C. and Bederson, B. 2005. Benefits of Animated Scrolling. *In Proc. of CHI'05.* pp. 1965-1968.

[58] Krishnan, S. N., and Moriya, S. 1996. One stroke operations: A new pen-based user interface that can integrate or separate operand specification, menu opening and selection, and action execution, in one or more strokes. *IPSJ Journal, 37.*, pp. 12-026 (1996).

[59] Kristensson, P.-O. and Zhai, S. 2004. SHARK2: A Large Vocabulary Shorthand Writing System for Pen-based Computers. *In Proc. of UIST2004.* pp.43 - 52.

[60] Kuroki, T., and Kawai, Satoru.1999. An interface technology for pen devices using tilt information. *In proc. of Interactive Systems and Software VII.* pp. 1C6.

[61] Kurtenbach, G., and Buxton, W. 1991. Issues in combining marking and direct manipulation techniques, *In Proc. of UIST'91.* pp. 137-144.

[62] Kurtenbach, G., and Buxton, W. 1993. The limits of expert performance using hierarchical marking menus. *In Proc. of CHI93.* pp. 35-42.

[63] Kurtenbach, G., Fitzmaurice, G., and Baudel, T., and Buxton, B. 1997. The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency. *In proc. of CHI'97.* pp. 35-42.

[64] Kurtenbach, G., Sellen, A., Buxton, W. 1993. An empirical evaluation of some articulatory and cognitive aspects of 'marking menus' *Human Computer Interaction 8(1)*, pp. 1-23.

[65] Kurtenbach, G., Moran, T., Buxton, W.: Contextual Animation of Gestural Commands. *In Proc. of GI'94*, pp. 83-90 (1994).

[66] Lai, J., Anderson, R., and Li, Y. 2005. Donuts: A Chinese Input Technique Using Pressure-Sensitive Marking Menus. *In Conference Supplement to UIST2005.*

[67] Li, Y., Hinckley, K., Guan, Z., and Landay, J. A. 2005. Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces. *In Proceedings of CHI2005.* pp. 461- 470.

[68] Lin, J., Newman, M.W., Hong, J.I., and Landay, J.A.2000. DENIM: finding a tighter fit between tools and practice for Web site design. *In Proceedings of CHI2000.* pp. 510C517.

[69] MacKenzie, S., Sellen, A. and Buxton, W. 1991. A Comparison of Input Devices in Elemental Pointing and Dragging Tasks. *In Proc. of CHI'91.* pp.161-166.

[70] Mackay, B., Dearman, D., Inkpen, K. and Watters, C. 2005. Walk .n Scroll: A Comparison of Software-based Navigation Techniques for Different Levels of Mobility. In *proc. of MobileHCI'05.* pp. 183-190.

[71] MacKenzie, S. and Buxton, W. 1992. Extending Fitts Law to Two-dimensional Tasks. *In Proc. of CHI'92.* pp.219-223.

[72] MacKenzie, I., and Riddersma, S. 1994. Effects of Output Display and Control-Display Gain on Human Performance in Interactive Systems. *Behaviour and Information Technology.* pp.328-337.

[73] MacKenzie, S. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human- Computer Interaction, Vol.7.* pp. 91-139.

[74] Masui, T., Kashiwagi, K., and Borden IV. 2002. Elastic Graphical Interfaces for Precise Data Manipulation, *In Proc. of CHI'02.* pp. 143-144.

[75] McGuffin, M. 2002. Fitts' law and expanding targets: an experimental study and applications to user interface design. *M.Sc. Thesis, Department of Computer Sci-*

ence, University of Toronto.

[76] McGuffin, M. and Balakrishnan, R. 2002. Acquisition of expanding targets. *In proc. of CHI2002.* pp. 57-64.

[77] Miller, R., and Myers, B. 2002 Multiple Selections in Smart Text Editing. *In Proc. of IUI'02.* pp. 103-110 (2002).

[78] http://www.microsoft.com/presspass/features/2000/nov00/11-13tabletpc.mspx

[79] Mizobuchi, S., and Yasumura M. 2004. Tapping vs. Circling Selections on Pen-based Devices: Evidence for Different Performance-Shaping Factors, *In Proc. of CHI2004.* pp. 607-614.

[80] Mizobuchi, S., Terasaki, S., Keski-Jaskari, T., Nousiainen, J., Ryynanen, M., and Silfverberg, M. 2005. Making an Impression: Force-Controlled Pen Input for Hand-held Devices. *CHI Extended Abstracts 2005.* 1661-1664.

[81] Murata, A. 1999. Extending effective target width in Fitts' law to a two-dimensional pointing task. *International Journal of human computer interaction, 11(2)* pp.137-152.

[82] Moscovich, T., and Hughes, J. 2004. Navigating Documents with the Virtual Scroll Ring. In *Proc. of UIST2004.* pp. 57-60.

[83] Nacenta, M.A., Aliakseyeu, D., Subramanian, S., and Gutwin, C.A. 2005. A comparison of techniques for multi-display reaching. *Proc. CHI2005.* pp.371-380.

[84] Newman, W.M. 1968. A System for Interactive Graphical Programming. *In AFIPS Spring Joint Computer Conference. 1968. 28.* pp. 47-54.

[85] Oshita, M. 2004. Pen-to-mime: A pen-based interface for interactive control of a human figure. *In EUROGRAPHICS workshop on sketch-based interfaces and modeling.* pp. 43-52.

[86] Pang, X.D., Tan, H.Z. and Durlach, N.I. 1991. Manual discrimination of force using active finger motion. *Perception & Psychophysics, 49(6).* pp.531-540.

[87] Parker, K., Mandryk, R., Nunes, M., Inkpen, K. 2005. TractorBeam Selection Aids: Improving Target Acquisition for Pointing Input on Tabletop Displays. *In Proc. of Interact 2005, Springer.* pp.80-93.

[88] Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., and Want, R. 2002. TiltType: accelerometer-supported text entry for very small devices. *In Proc. of UIST2002.* pp. 201-204.

[89] Pastel, R. 2006. Measuring the Difficulty of Steering Through Corners. *In Proc. of CHI2006.* pp.1087-1096.

[90] Potter, R.L., Weldon, L.J.,and Shneiderman, B. 1988. Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proc. of CHI'88.* pp. 27-32.

[91] Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. 1996. The Go-Go Interaction Technique: Non- Linear Mapping for Direct Manipulation in VR. *In Proc. of UIST'96.* pp. 79-80.

[92] Raisamo, R. 1999. Evaluating different touched-based interaction techniques in a public information kiosk. *Conference of the CHI Special Interest Group of the Ergonomics Society of Australia.* pp.169-171.

[93] Ramos, G., and Balakrishnan, R. 2003. Fluid interaction techniques for the control and annotation of digital video. In Proc. of UIST2003. pp.105-114.

[94] Ramos, G., Boulos, M., and Balakrishnan. 2004. Pressure Widgets. *In Proc. of CHI2004.* pp.487-494.

[95] Ramos, G., and Balakrishnan, R. 2005. Zliding: Fluid Zooming and Sliding for High Precision Parameter Manipulation. *In Proc. of UIST2005.* pp.143-152.

[96] Gonzalo Ramos, Andy Cockburn, Ravin Balakrishnan, Michel Beaudoiun-Lafon. 2007. Pointing lenses: facilitating stylus input through visual-and motor-space magnification. In *Proc. CHI 2007.* pp. 757-766.

[97] Rekimoto, J. 1996. Tilting Operations for Small Screen Interfaces. *In Proc. of UIST96.* pp. 167-168.

[98] Rekimoto, J., and Sciammarella, E. 2000. ToolStone: effective use of the physical manipulation vocabularies of input devices. *In Proc. of UIST2000.* pp.109C117.

[99] Ren, X. and Moriya, S. 2000. Improving selection performance on pen-based systems: A study of pen-input interaction for selection tasks. *ACM Transactions on Computer Human Interaction (ToCHI), 7(3).* pp.384-416.

[100] Sazawal, V., Want, R., and Borriello, G. 2002. The Unigesture Approach. One-Handed Text Entry for Small Devices. *In Proc. of Mobile HCI2002.* pp. 256-270.

[101] Sears, A. and Shneiderman, B. 1991. High Precision Touchscreens: Design Strategies and Comparison with a Mouse. *International Journal of Man-Machine Studies 43(4)*, pp. 593-613.

[102] Shneiderman, B., 1983. Direct Manipulation: A Step Beyond Programming Languages. *IEEE Computer, 1983. 16(8):* pp. 57-69.

[103] Shneiderman, B. 1991. Touch screens now offer compelling uses. *IEEE Software 8(2).* pp. 93-94.

[104] Smith, G. and Schraefel, M. 2004. The Radial Scroll Tool: Scrolling Support for Stylus-or Touch-based Document Navigation. *In Proc. UIST2004.* pp. 53-57.

[105] Smith, D. C., Harslem, E. F., Irby, C. H., and Kimball,R. B. 1982. The Star User Interface: an Overview. *In Proc. of the 1982 National Computer Conference.* pp. 515-528.

[106] Small, D., and Ishii, H. 1997. Design of Spatially Aware Graspable Displays. *In Proc. of CHI'97.* pp. 367-368.

[107] Smith, D.C.1977 Pygmalion: A Computer Program to Model and Stimulate Creative Thought. *Basel, Stuttgart: Birkhauser Verlag. PhD Thesis, Stanford University Computer Science Department.*

[108] Smith, R., and Taivalsaari A.1999. Gerneralized and Stationary Scrolling, *In Proc. of UIST'99.* pp. 1-9.

[109] Sutherland, I.E.1963. SketchPad: A Man-Machine Graphical Communication System. In AFIPS Spring Joint Computer Conference. pp. 329-346.

[110] Tudoreanu, M., and Kraemer, E. 2002. A Study of the Performance of Steering Tasks under Spatial Transformation of Input. *In submission to Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization.*

[111] Tan, H.Z., Pang, X.D. and Durlach, N.I. 1992. Manual resolution of length, force, and compliance. *In Kazerooni, H. (ed). DSC-Vol. 42, Adavances in Robotics, The American Society of Mechnical Engineers.*

[112] Tan, H., Srinivasan, M., Eberman, B. and Chang, B. 1994. Human factors for the design of force-reflecting haptic interfaces. In *In Proc. of the ASME Winter Annual Meeting.* pp. 353-359.

[113] Wherry, E. 2003. Scroll ring performance evaluation. *In Proc. of HCI'03* pp. 758-759.

[114] Wigdor, D., and Balakrishnan, R. 2003. TiltText: Using tilt for text input to mobile phones. *In Proc. of UIST2003.* pp.81-90.

[115] Wills, G. 1996. Selection: 524,288 Ways to say 'This is Interesting', *In Proc. of IEEE InfoVis '96.* pp.54-60.

[116] Williams, G. 1983.The Lisa Computer System. *Byte Magazine, 1983. 8(2).* pp. 33-50.

[117] Williams, G.. 1984. The Apple Macintosh Computer. *Byte, 1984. 9(2):* pp. 30-54.

[118] Worden, A., Walker, N., Bharat, K., and Hudson, S. 1997. Making computers easier for older adults to use: area cursors and sticky icons. *In proc. CHI' 97.* pp. 266-271.

[119] Yin, J. Ren, X., and Ding, H., 2005. HUA: An Interactive Calligraphy and Ink-

Wash Painting System. *In Proc. of CIT 2005*. 989-995.

[120] Yin, J. and Ren, X. 2006. The Beam Cursor: A Pen-based Technique for Enhancing Target Acquisition. *In Proc. of HCI 2006*. pp.119-134.

[121] Zhai, S. and Kristensson, P.-O. 2003. Shorthand Writing on Stylus Keyboard. *In Proc. of CHI2003, CHI Letters 5(1)*. pp.97-104.

[122] Zhai, S., Accot, J. and Woltjer, R. 2004. Human Action Laws in Electronic Virtual Worlds - an empirical study pf path steering performance in VR. *Presence, 13, 2 (2004)*. pp.113-127.

[123] Zhai, S., Buxton, W., and Milgram, P. 1994. The "Silk Cursor": Investigating transparency for 3D target acquisition. *In proc. of CHI94*. pp. 459-464.

[124] Zhai, S., Conversy, S., Beaudouin-Lafon, M., and Guiard, Y. 2003. Human on-line response to target expansion. *In proc. of CHI2003*. pp.177-184.

[125] Zhai, S., Smith, B., Selker, T. 1997. Improving Browsing Performance: A Study of Four Input Devices for Scrolling and Pointing Tasks. *In Proc. of INTERACT'97* pp. 286-295.

[126] Zhai, S. 1995. Human performance in six degree-of freedom input control. *Ph.D. Thesis. University of Toronto.*

[127] Zhao, S., and Balakrishnan, R. 2004. Simple vs. Compound Mark Hierarchical Marking Menus, *In Proc. of UIST2004*. pp. 33-42.

Appendix A

Publications

A.1 Articles in or submitted refereed journals

1. Yin, J., and Ren, X. 2007. ZWPS and Pressure Scroll: Two Pressure-based Techniques in Pen-based Interfaces. *To appear in Journal of Information Processing Society of Japan (IPSJ)*, Vol.48 No.12..

2. Yin, J., and Ren, X. 2007. The Beam Cursor: A Pen-based Technique for Enhancing Target Acquisition. *Conditionally accepted by Int. J. Interacting with Computers.*

3. Yin, J., Ren, X., and Zhai, S. 2007. Human Performance of Pen Pressure Control in Trajectory Tasks. *Submitted to Int. J. Behavior & Information Technology.*

A.2 Articles in full paper refereed international conference proceedings

4. Yin, J., Ren, X., Ding, H. 2005. HUA: An Interactive Calligraphy and Ink-Wash Painting System. *In Proceeding of CIT2005.* pp. 989-995.

5. Yin, J., Ren, X. 2006. The Beam Cursor: A Pen-based Technique for Enhancing Target Acquisition.*In Proceedings of the 20th BCS HCI Group conference in co-operation with ACM (London, England, 11-15 September, 2006), Springer.* pp.119-134.

6. Yin, J., Ren, X., and Liu, Z. 2006. Circular-gesture and Double-ellipse: novel software-based techniques for generating extra input states in pen-based interfaces. *In Proceedings of APCHI2006: 6th Asia Pacific Conference on Computer Human Interaction (Taipei, China, October 11 - 14, 2006), 10 pages, Springer.* (**Best student paper award**)

7. Tsuchida, T., Ren, X., and Yin, J. 2006. A Novel Scrolling Technique for Pen-based System. *In Proceedings of APCHI2006: 6th Asia Pacific Conference on Computer Human Interaction (Taipei, China, October 11 - 14, 2006), 10 pages, Springer.*

8. Ren, X., Yin, J., Li, Y., and Zhao, S. 2007. The Adaptive Hybrid Cursor: A Pressure-based Target Selection Technique for Pen-based User Interfaces. *In proceedings of INTERACT 2007.*

9. Yin, J., Ren, X. 2007. ZWPS: A Hybrid Selection Technique for Small Target Acquisition in Pen-based Interfaces.*In Proceedings of INTERACT 2007.*

10. Yin, J., Ren, X. 2007. Investigation to Line-based Techniques for Multi-target Selection. *In Proceedings of INTERACT 2007.*

11. Ren, X., Mizobuchi,S., Yin, J. and Ooya, T. 2007. Establishing User Discriminated Pressure Levels and the Effects of Posture on Pressure Input. *In Proceedings of IEEE ICICIC 2007.*

A.3 Articles in abstract refereed international conference proceedings

12. Yin, J. and Ren, X. 2005. The study of Stroke-based Scrolling Techniques in pen-based interfaces. *In Proceedings of NEINE'05 (the International Conference on Next Era Information Networking, Shanghai, China, September, 2005).*

13. Ren, X. and Yin, J. 2006. Zoom-based technique with pressure as switch

for pixel-level targets in pen-based interfaces. *Proceedings of NEINE'06 (the International Conference on Next Era Information Networking, Kochi, Japan, 17-19 September, 2006).* pp.313-314.

14. Yin, J. and Ren, X. 2006. Pressure Cursor: a novel technique for target acquisition in pen-based interface. *In Proceedings of NEINE'06 (the International Conference on Next Era Information Networking, Kochi, Japan, 17-19 September, 2006)* pp.317-319.

15. Ooya, T., Ren, X. and Yin, J. 2006. The Effects of Gender Difference: An Experiment on a Force Control Device. *In Proceedings of NEINE'06 (the International Conference on Next Era Information Networking, Kochi, Japan, 17-19 September, 2006).* pp.309-312.

16. Ren, X., and Yin, J. 2007. Empirical study of multi-target acquisition techniques in pen-based interfaces. *In Proceedings of NEINE'07.*

A.4 Articles in refereed local conference proceedings

17. Yin, J., and Ren, X. 2005. The study of the stroke-based techniques for scrolling task in pen-based interface. *In Proceedings of SJCIEE2005.*

18. Yin, J., and Ren, X. 2006. Pen User Interface Based on Stroke-driven and Pressure-driven modes. *In Proceedings of SJCIEE2006.*

19. Ooya, T., Ren, X., and Yin, J. 2006. An Experimental Usability of Human Abilities on Force Control Device. *In Proceedings of SJCIEE2006.*