

Pressure-Sensitive Pen Interactions

by

Gonzalo Alberto Ramos

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy

Graduate Department of Computer Science
University of Toronto

Copyright © by Gonzalo A. Ramos 2008

Pressure-Sensitive Pen Interactions

Gonzalo Alberto Ramos

Doctor of Philosophy

Graduate Department of Computer Science
University of Toronto

2008

Abstract

Pen-based computers bring the promise of tapping into people's expressiveness with pen and paper and producing a platform that feels familiar while providing new functionalities only possible within an electronic medium. To this day, pen computers' success is marginal because their interfaces mainly replicate keyboard and mouse ones. Maximizing the potential of pen computers requires redesigning their interfaces so that they are sensitive to the pen's input modalities and expressiveness. In particular, pressure is an important and expressive, yet underutilized, pen input modality.

This dissertation advances our knowledge about pressure-aware, pen-based interactions and how people use these techniques. We systematically explore their design by first investigating how pressure can affect pen interactions. We propose novel techniques that take advantage of the pressure modality of a pen to control, link, and annotate digital video.

We then study people's performance using pressure to navigate through a set of elements and find that they can discriminate a minimum of six different pressure regions. We introduce the concept of *Pressure Widgets* and suggest visual and interaction properties for their design.

We later explore pressure’s use to enhance the adjustment of continuous parameters and propose *Zliding*, a technique in which users vary pressure to adjust the scale of the parameter space, while sliding their pen to perform parameter manipulations. We study *Zliding* and find it a viable technique, which is capable of enabling arbitrarily precise parameter adjustments.

We finally present a novel interaction technique defined by the concurrent variation in pressure applied while dragging a pen. We study these *pressure marks* and find that they are a compact, orientation-independent, full interaction phrase that can be 30% faster than a state-of-the-art selection-action interaction phrase.

This dissertation also makes a number of key contributions throughout the design and study of novel interaction techniques:

- It identifies important design issues for the development of pressure-sensitive, pen-operated widgets and interactions,
- It provides design guidelines for interaction techniques and interface elements utilizing pressure-enabled input devices,
- It presents empirical data on people’s ability to control pressure, and
- It charts a visual design space of pressure-sensitive, pen-based interactions.

Acknowledgements

I am deeply grateful to many people whom have been by my side all these years. Firstly, I am grateful to my committee members, a fantastic blend that honored me by lending their expertise, candid advice and support. Professor Ronald Baecker provided me with an invaluable pool of experience on seminal aspects of HCI and best practices of on User-Centered Design. I remain in awe of his enthusiasm for the field and for his ability to defuse, single-handed, tense situations with hotel officials while securing a whole parking lot to have an after-hours party. Professor Mark Chignell has inspired me to seek rigor in experiment design and analysis, and has been an incredible resource in areas of Human Factors. I regret not seeking out to his generous Kiwi mentorship more often. Professor Karan Singh's nimble intellect and knowledge has helped me to find and fill gaps of my own. His talent to combine effortlessly art and science has been a source of inspiration that will stay always with me. He is a man truly outside of time, since all the clocks in his house give a different time. My advisor, Professor Ravin Balakrishnan has constantly challenged both my intellect and tolerance for capsaicin and saturated oils: good scientific knowledge comes from constantly questioning what we think we know, and good beef is better deep-fried, extra spicy and extra crispy. Ravin's mentorship has helped me to learn to sail the often-tempestuous waters of academia and allowed me to grown professionally. Thank you, Ravin.

To my circle of UofT friends: You are family. To my family in Argentina: You are my roots and you travel with me wherever I go. My love for you is beyond measure. To Buenos Aires "La Reina de Plata": Being a Porteño is awesome.

Last, but not least, to Rebecca: you make all possible. I love you.

Table of Contents

Abstract	ii
Acknowledgements	iv
Table of Contents.....	v
List of Tables.....	x
List of Figures	xi
<u>1</u> Introduction	1
1.1 Pressure as an Additional Input Modality	3
1.2 Thesis Organization.....	4
1.3 Research Methods.....	6
1.4 Contributions	7
2 Background	9
2.1 The Perception of Pressure	9
2.1.1 Cutaneous and Subcutaneous Mechanoreceptors.....	9
2.1.2 Muscle and Skeletal Mechanoreceptors	12
2.1.3 Psychophysics and the Study of Variations in Signal Intensity	14
2.1.4 Implications for Design	16
2.2 Pressure Input in the GUI: Prior Art.....	17
3 Pen Interactions Incorporating Pressure as an Additional Input Channel: a Case Study.	25
3.1 Video as Media.....	25
3.2 Traditional Video/Film Practices.....	26
3.3 Related Systems and Techniques.....	29
3.3.1 Pen-based Interaction Systems and Techniques	29
3.3.2 Video Annotation and Navigation Systems.....	33
3.4 Overview and Design Philosophy of LEAN	36
3.5 Gestures, Commands, and Scribbling.....	38
3.6 Pressure and Pressure Widgets	40
3.7 Video Control	41
3.7.1 Position+Velocity Sliders	42

3.7.2	Twist-Lens Sliders	44
3.7.3	Video Segments	46
3.8	Annotations and Links	47
3.8.1	Link Navigation and Manipulation	50
3.9	Discussion and User Feedback	51
3.10	Summary	54
4	Pressure Widgets	55
4.1	Introduction	55
4.2	Experiment	56
4.2.1	Apparatus	56
4.2.2	Task and Stimuli	57
4.3	Participants	59
4.4	Procedure and Design	59
4.5	Performance Measures	62
4.6	Results	62
4.6.1	Selection Techniques	63
4.6.2	Discernable Number of Pressure Levels	66
4.6.3	Effect of Visual Feedback	68
4.6.4	Conformity with Fitts' Law	69
4.6.5	Control at Different Pressure Levels	70
4.7	Summary	71
4.8	Implications for Design	73
4.9	Factors for the Design of Pressure Widgets	74
4.9.1	Interference	75
4.9.2	Visuals Elements and Behavior	76
4.10	A Study on Visual Features and Interference	78
4.10.1	Apparatus	81
4.10.2	Participants	81
4.10.3	Procedure and Design	81
4.10.4	Performance Measures	82
4.10.5	Results	82
4.10.6	Discussion	86

4.11	Summary	87
5	Zooming and Sliding for High-Precision Parameter Manipulation	89
5.1	Related Work	90
5.2	Motivation and Goals	95
5.3	The Zlider	96
5.3.1	Pressure Cursor	97
5.3.2	Integrated Zoom & Slide Control	98
5.3.3	Clutching the Zoom Level	100
5.3.4	The Selection Mechanism	101
5.3.5	Scrolling.....	105
5.4	Alternatives for Decoupled Zoom Control	106
5.4.1	Force Button	106
5.4.2	Discrete Keys.....	107
5.5	Experiment.....	108
5.5.1	Apparatus.....	108
5.5.2	Participants	110
5.5.3	Task and Stimuli	110
5.5.4	Procedure and Design	111
5.6	Results	113
5.6.1	Selection Time	113
5.6.2	Crossings	115
5.6.3	Qualitative Results: User Preferences	117
5.7	Discussion.....	118
5.7.1	Other Designs: The Zliding Wheel.....	119
5.7.2	Other Technologies, Other Directions.....	121
5.8	Summary	122
6	Pressure Marks	124
6.1	Previous Work	126
6.2	Pressure Marking.....	129
6.2.1	Browsing Through Pressure Marks	132
6.2.2	Pressure Marks' Anatomy: Reduction and Parsing.....	134
6.3	User Study #1	138

6.3.1	Apparatus.....	138
6.3.2	Participants	138
6.3.3	Task and Stimuli.....	139
6.3.4	Procedure and Design.....	140
6.3.5	Results	141
6.3.6	Summary.....	145
6.4	User Study #2	145
6.4.1	Apparatus and Participants	146
6.4.2	Task and Stimuli	146
6.4.3	Procedure and Design.....	147
6.4.4	Results	148
6.5	On The Use of Pressure Marks.....	151
6.5.1	Pressure Marking Menus.....	152
6.5.2	Simple Pressure Marks	153
6.5.3	Pressure Tails.....	155
6.5.4	Pressure Fanning.....	155
6.6	Summary.....	156
7	Conclusions and Future Research.....	158
7.1	Design Issues & Guidelines.....	159
7.1.1	Interaction Duration.....	159
7.1.2	Visual Feedback	160
7.1.3	Pressure Taming: Noise.....	160
7.1.4	Pressure Taming: Control.....	162
7.1.5	Avoiding Task's Disruptions.....	162
7.1.6	Performance Limits in Discrete Pressure Widgets	164
7.2	Charting the Design Space of Pressure Widgets	164
7.2.1	An Atom-Inspired Description of Pressure Widgets	166
7.3	Future Directions	168
7.3.1	Revisiting Digital Video Interaction.....	168
7.3.2	Learning Experience	171
7.3.3	Pressure as a Measure of Intention	172
7.3.4	A Concert of Expression Channels.....	172

7.3.5	Other Devices	173
7.3.6	Place in the Interface Ecology	173
7.4	Final Remarks.....	174
8	Bibliography	175
9	Appendix A: Pressure Widgets' Study Survey Forms	190
10	Appendix B: Zlider's Study Survey Form	197
11	Appendix C: Pressure Marks' Study Survey Form	201

List of Tables

Table 2-1: Main muscular and skeletal proprioceptors	14
Table 3-1: Gesture grid that shows the basic set of ... the <i>LEAN</i> system.	40
Table 4-1: Distribution of outlier samples.....	63
Table 4-2: Distribution of outlier samples.....	83
Table 4-3: Average number of crossings (NC) per tracking level.....	84
Table 5-1: Distribution of outlier data points across main experimental conditions.....	113
Table 7-1: Design matrix for the design of pressure widgets	165

List of Figures

Figure 1-1: Basic expression channels for a pen.	2
Figure 2-1: Cross section of skin showing main cutaneous tactile mechanoreceptors.....	11
Figure 2-2: Muscle spindle	12
Figure 2-3: Golgi organ	13
Figure 2-4: Generic psychometric function.....	15
Figure 2-5: Reproduction of photographs of Herot and Weinzapfel's system.	18
Figure 2-6: Main display for paint program – from (Buxton et al., 1985)	18
Figure 2-7: Spaceball used as a reference isometric device in Zhai's experiments	19
Figure 2-8: Experimental setup for measuring pressure perception... ..	19
Figure 2-9: Pinch grip used on Bao & Silverman's study.....	20
Figure 2-10: Block diagram of the Experimental setup used by Srinivasan and Chen.	21
Figure 2-11: Device used by Lecuyer in one of his experiments	21
Figure 2-12: The Geozui3d interface.....	22
Figure 2-13: Examples of force sensing hardware	23
Figure 2-14: Examples of force sensing hardware	23
Figure 2-15: Pop-through mouse and an its three-state button.....	24
Figure 3-1: Animator Lynn Smith using the Genesys system.....	29
Figure 3-2: Screenshots of the Tivoli and the Flatland environment.	30
Figure 3-3: Screenshots of the Electronic Cocktail Napkin and Xlibris prototype	31
Figure 3-4: Marking menu technique; Flowmenu widget; and a color <i>toolglass</i>	32
Figure 3-5: CrossY system ... & Hover widgets.....	33
Figure 3-6: SILVER's video editing interface & Photofinder's interface.....	35
Figure 3-7: Videotater's interface consists of three main regions [...].....	36
Figure 3-8: The LEAN system running on a Tablet PC	38

Figure 3-9: Pressure Widgets.....	41
Figure 3-10: A typical video player with a VCR-like media control widget	42
Figure 3-11: The PVslider widget and features.....	43
Figure 3-12: This partial view of the <i>TLSlider</i> shows [...]	45
Figure 3-13: Twist Lens Slider.....	46
Figure 3-14: A sequence demonstrating the contextual visualization of an annotation.	49
Figure 3-15: Frames connected to a note are visualized as thumbnails [...].....	50
Figure 3-16: An example of a typical session with <i>LEAN</i>	51
Figure 4-1: Visual feedback conditions.....	58
Figure 4-2: Signatures for the selection methods	59
Figure 4-3: Subdivisions of pressure space used for the study.....	61
Figure 4-4: Effect of selection on <i>Mean Time</i> (left), and Error Rate (right).	64
Figure 4-5: Effect of selection on <i>Number of Crossings</i> throughout experimental blocks.	65
Figure 4-6: Effect of the number of pressure levels on <i>Error Rate</i> [...]	67
Figure 4-7: Effect of the number of levelson <i>Number of Crossings</i> [...].....	67
Figure 4-8: Effect of the number of levelson <i>Mean Time</i> for the Full Visual condition.	68
Figure 4-9: Linear regression of Index of Difficulty versus <i>Mean Time</i> [...].....	70
Figure 4-10: Effect of distance (amplitude) on pressure level on <i>Number of Crossings</i>	71
Figure 4-11: Example of visual elements on a traditional linear menu.....	77
Figure 4-12: Concept designs for different pressure widgets.....	78
Figure 4-13: [...] behavior of different discrete pressure widget designs	80
Figure 4-14: Average distance traveled by the pen's tip [...]	83
Figure 4-15: Average % error rate (+/- SE) per widget design.	84
Figure 4-16: Average Navigation time (seconds +/- SE) per widget.	85
Figure 5-1: Drag and Pop interaction technique.....	91
Figure 5-2: [...] window where its buttons are weighted according to their semantics.	92

Figure 5-3: The Alphasl原因er [...] & the FineSlider widget(s)	92
Figure 5-4: Diagram of the take-off technique.	93
Figure 5-5: Precision handle.	93
Figure 5-6: Igarashi and Hinckley's implementaion of speed dependent zooming.	94
Figure 5-7: Zliding on the Zlider widget.	97
Figure 5-8: Pressure cursor.	98
Figure 5-9: [...] parabolic-sigmoid transfer function used to preprocess the pressure signal..	99
Figure 5-10: Effect of the signal's stabilization and filtering techniques.	100
Figure 5-11: Clutching the zoom level [...]	101
Figure 5-12: Zlider's state-transition diagram.	103
Figure 5-13: Air Clutching in action.	105
Figure 5-14: Wacom CintiQ interactive display [...]	109
Figure 5-15: Phidgets pressure transducer used in our experiment.	109
Figure 5-16: Elements in the experimental setup.	111
Figure 5-17: Average selection time per technique and width.	114
Figure 5-18: Average selection time per block and scale-adjusting technique.	114
Figure 5-19: Average crossings per distance*width.....	115
Figure 5-20: Average crossings per target's width and scale-adjusting technique.	116
Figure 5-21: Average number of crossings per experimental block [...]	116
Figure 5-22: Zliding Wheels.....	120
Figure 6-1: A pressure mark is used to select and copy [...].....	124
Figure 6-2: Flowmenu can integrate selection and direct manipulation [...]	127
Figure 6-3: This image illustrates how a pigtail delimiter [...]	127
Figure 6-4: Marquee menu's four types of commands.....	128
Figure 6-5: Profiles of the four proposed pressure signatures.....	131
Figure 6-6: Browsing through pressure marks.	133

Figure 6-7: Pressure vs. traveled distance.	135
Figure 6-8: An example of experimental stimuli [...]	139
Figure 6-9: Average errors per block for straight lines and lassos.	142
Figure 6-10: Average errors per pressure signature for lines and lassos.	143
Figure 6-11: Average Trial Time per block for lines and lassos.	143
Figure 6-12: Average Trial Time per pressure signature for lines and lassos.	144
Figure 6-13: Example of an experimental trial [...]	147
Figure 6-14: Average trial time (+/- SE) per technique and selection type.	149
Figure 6-15: Average trial time per block. Power regression lines are shown.	150
Figure 6-16: Average error per technique and selection type.	150
Figure 6-17: Average error per block per technique. Power regression lines are shown.	151
Figure 6-18: Pressure marking menu design.	152
Figure 6-19: Different simple pressure mark flicks.	154
Figure 6-20: Ripple feedback.	154
Figure 6-21: Pressure tails example.	155
Figure 6-22: Example of pressure fanning.	156
Figure 6-23: A map annotation in a multi-user scenario [...]	157
Figure 6-24: Example of compound pressure marks.	157
Figure 7-1: Example of a hysteresis function for navigating [...]	161
Figure 7-2: Frequency of use for pressure locking and air clutching [...]	162
Figure 7-3: An atomic representation of a simple pressure widget.	166
Figure 7-4: Atom diagrams for the <i>Flag</i> widget.	167
Figure 7-5: The atomic diagram of the Pressure Cursor, which uses two atoms.	167
Figure 7-6: The atomic diagram for Pressure Marks, which uses two atoms.	168
Figure 7-7: Profile of a pulse gesture	169
Figure 7-8: Mockup diagram of the PVZlider	170

List of Videos^{*}

Accompanying video for Chapter 3 <http://www.youtube.com/watch?v=c-4tjto6Akw>

Accompanying video for Chapter 4 <http://www.youtube.com/watch?v=n3Ybz8KiB68>

Accompanying video for Chapter 5 <http://www.youtube.com/watch?v=EcE3XBytN-U>

Accompanying video for Chapter 6 <http://www.youtube.com/watch?v=qR2mKwkATpk>

^{*} Videos are also available through the University of Toronto T-Space: <https://tspace.library.utoronto.ca>

List of Appendices

Appendix A: Pressure Widgets' Study Survey Forms	190
Appendix B: Zlider's Study Survey Form	197
Appendix C: Pressure Marks' Study Survey Form	201

1 Introduction

Much work has been done by researchers in the last 40 years (Accot & Zhai, 2002; Baecker, 1969; Baecker, 1974; Gross & Do, 1996; Guimbretière, Stone, & Winograd, 2001; Kurtenbach & Buxton, 1993; Pederson, McCall, Moran, & Halasz, 1993; Schilit, Golovchinsky, & Price, 1998; Sutherland, 1963; Sutherland, 1966) in an effort to make pen computing both a powerful and a usable platform. Despite these efforts, pen-based computers have yet to realize the same degree of success as the more widely used keyboard- and mouse-driven computers.

This lack of success is arguably largely due to the almost direct application of the standard point-and-click keyboard/mouse-based interface to pen-based computers. The imposition of this type of interface creates situations where the flow of a user's task is disrupted because (a) user-interface elements are located away from the focus of the interaction (e.g., tool bars), (b) too many modes of operation are present (e.g., select, draw, manipulate) and (c) the user-interface elements or interactions (e.g., the double click) were designed for a different pointing device (e.g., the mouse).

In addition to the above issues, elemental tasks on a mouse-driven GUI such as pointing and clicking can become challenging to perform if using a pen. Pen movements can be noisy – i.e., the pen or the interactive surface can be subject to tremors from different sources. Also, visual parallax between the stylus and the input surface, and occlusion from the user's hand can make targeting tasks difficult by misleading users as to the true location of the computer system's cursor. Further, the unique affordances of pen input technologies have yet to be

INTRODUCTION

fully utilized, resulting in pen-based computers that are little more than regular computers with the mouse replaced by a pen.

We argue that maximizing the potential of pen-based computers requires the careful design and implementation of the interfaces and the interactions they depend on – both of which significantly differ from keyboard- and mouse-based techniques. Furthermore, these new interactions need to be sensitive to the stylus' idiosyncrasies and the way users handle such a unique input device. A pen not only can indicate a position in X/Y space, but it also allows the specification of a number of often ignored parameters. Figure 1-1 shows a representative set of these parameters such as hover, tilt, rotation and pressure. In particular, the pressure users can apply through a pen's tip is an often overlooked expression channel with the potential to expand the vocabulary of available interactions in novel, unexplored ways.

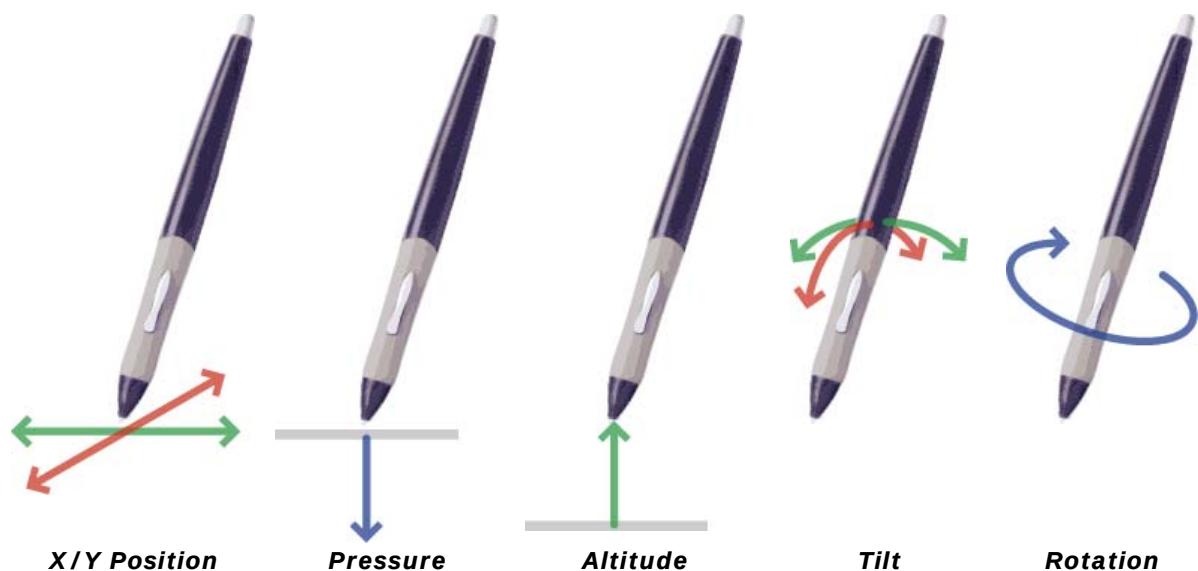


Figure 1-1: Basic expression channels for a pen. *Position* indicates the x/y location of the pen's tip relative to some sensing surface. *Pressure* indicates the amount of force applied through the pen's tip onto the sensing surface. *Altitude* indicates the distance between the pen's tip and the sensing surface. Most current hardware only reports if the pen's tip is within certain distance from the sensing surface or not. *Tilt* indicates the angles that determine the inclination between the sensing surface and the pen's main axis. *Rotation* indicates the absolute angle of the pen's barrel in relation to the sensing surface's north.

1.1 Pressure as an Additional Input Modality

Through this thesis we seek a deeper understanding of an input modality not extensively used in current pen-based user interfaces: the pressure-sensing capability of the pen. We also want to obtain a greater comprehension of the design space of pen-based interaction that incorporates this extra input modality. Pressure is one of the most promising additional expression channels a pen can provide, as users typically have substantial prior experience in varying the levels of pressure exerted when using traditional pens and pencils on paper.

In the context of the written word, people’s ability to control pressure contributes to an incredibly rich and often beautiful mechanism of expression. The different levels of pressure one applies during the act of writing let us not only exert control as to the speed of one’s handwriting (e.g., one can break or decelerate by pressing harder), but also embed our intention and emphasis into what one writes. Pressure is considered by graphologists as the third dimension of handwriting and it is often used as an indication of a someone’s personality: “The writing zones carry movement across, as well as up and down on the paper. That’s the first and second dimension. Pressure moves *into* the paper and has been labeled the depth component. It reflects, in a very literal way, the impression the writer makes on the world.” (Lowe, 1999).

Pressure is also a key expression channel in many human-to-human communication processes. People can express affection or frustration through touch and the right amount of pressure in the right place can cause extreme sensations such as paralyzing pain or exhilarating pleasure. Also, pressure can be a crucial element for many human-to-machine interactions such as during a musical performance. Mastering a particular instrument such as a string one generally involves the skilful control of pressure by the performer’s fingers. In addition to the

INTRODUCTION

above, pressure allows us to probe the environment surrounding us and be aware of its properties. A trivial example of this occurs when someone examines the ripeness of a fruit or the soundness of a physical structure.

All these examples speak of a highly expressive and interesting input modality and motivate us to consider using the pen's pressure on current tablet computers to increase the human-computer communication bandwidth and usability, particularly when tablet computers are used as pure slates without keyboards. This benefit is not exclusive of pen-based systems and can also extend to everyday devices such as personal digital assistants, cell phones, and watches, where pressure sensing could be easily incorporated. For example, simple uses of the pressure magnitude can be envisioned either to operate user-interface elements that have several discrete states, or to control the value of a continuous variable.

However, to foster the use of pressure in tablets and other devices, appropriate user-interface elements and interaction techniques must be designed. Such designs should be rooted in a thorough understanding of the user's ability to control pressure through a pen. One of the major contributions of this thesis is the design and implementation of novel interaction techniques, which leverage the pressure applied through a stylus as an additional expression channel. These new ways of interacting with pen-based devices also have the potential to improve the usability of input devices with sensing capabilities similar to the pen's.

1.2 Thesis Organization

In order to achieve our vision of accomplishing a deeper understanding of the pressure input modality in the context of pen-based user interfaces and achieving greater comprehension of the design space of pressure-enabled, pen-based interaction, we seek to investigate:

INTRODUCTION

- How does the use of (pen) pressure affect the GUI? What new interactions does it enable?
- What are the dimensions of a design space for the design of pressure-sensitive user interface elements?
- What are the characteristics of human performance when interacting with such an input modality?
- Can pressure be considered as a discrete parameter space? Can pressure be considered as a continuous parameter space?
- Can pressure be used in concert with other input modalities?

The following chapters of this thesis follow this research agenda. This dissertation first presents in chapter 2 relevant background information regarding the physiology of pressure perception and the existing literature that has investigated the use of pressure in the context of Graphical User Interfaces (GUIs), as well as relevant previous work that has investigated instances of pen-based interactions.

In chapter 3, we initially explore the design space of pen-based interactions by proposing a set of techniques to control, manipulate, link, and annotate digital video. As part of this exploration, we implemented an interactive prototype, which includes several novel pressure-aware, user-interface elements.

In chapter 4, we formally introduce *Pressure Widgets* – user-interface elements that exploit the capabilities of pressure-sensing technology. This chapter also presents the results of an experiment that investigates people’s performances when controlling the pressure applied through a pen under different visual feedback conditions. Chapter 4 continues by exploring how changes in key visual design dimensions of pressure widgets affect their usability. Based on our findings up to this point, we suggest attributes defining a design space for pressure

INTRODUCTION

widgets and propose recommendations for the design of discrete pressure widgets along with developing alternate widget designs.

In chapter 5, we explore the use of pressure as a continuous parameter and we present a novel interaction technique for variable high-precision, parameter manipulation: *Zliding*. We instantiate and study *Zliding* through a novel selector widget called *Zlider*, which users scrub in order to adjust a parameter at a granularity that depends on the pressure applied through the pen.

Chapter 6 explores the use of pressure in concert with movements in x-y space and presents a new type of pressure-sensitive interactive technique, *Pressure Marks*, which are pen strokes where variations in pressure make it possible to indicate both a selection and an action simultaneously. We study the viability of *Pressure Marks* and compare their performance against a state-of-the-art sequential selection-action mechanism.

Chapter 7 integrates ideas and results from previous chapters and presents a statement of principles and recommendations for the design and use of pressure-sensitive pen interactions. We conclude our dissertation by discussing the implications of our work and future research directions.

1.3 Research Methods

Throughout the research in this thesis, we apply a user-centered methodology to the design and implementation of user-interface elements and interactions. Through free-form and semi-structured interviews, we obtain feedback from potential users during all stages of the design process. We also evaluate our new interaction techniques under several experimental conditions such as type of task, input mechanism, type of visual feedback, etc. In these cases, we collect quantitative data of both user performance and preferences and use statistical analysis

INTRODUCTION

in order to (a) identify trends in the collected data and (b) to detect significant effects when experimental conditions such as the input technology, the type of technique, or a technique's attributes are varied.

1.4 Contributions

Our research helps to eliminate the obstacles currently preventing pen-based computing from gaining wider acceptance by expanding our knowledge of how people interact with both pen-based devices and interaction techniques that take advantage of pressure-sensing pens. In particular this dissertation makes a number of contributions:

1. It identifies important design issues for the development of pressure-sensitive, pen-operated widgets and interactions. These issues include the interaction's duration, the need for real-time visual feedback and the necessity of filtering the pressure's signal.
2. It provides design guidelines and usage principles for interaction techniques and interface elements that utilize pressure-enabled input devices. For example, we suggest forms the visual design of pressure widgets can take, as well as scenarios where the use of pen pressure can serve as an effective mechanism for mode switching (Chapter 4), parameter control (Chapter 5) or command specification (Chapter 6).
3. It presents new widgets and user-interaction techniques, which include the Position-Velocity Slider, Twist-Lenses, Sequential Icons, the Flag, Pie, and Bulls-eye Discrete Pressure Widgets, the Pressure Cursor, the Zliding, Zlider bar and wheel, Pressure Marks, Pressure Fanning, Simple Pressure Marks and Pressure Tails.
4. It presents empirical data on users' ability to control pressure, in both discrete and continuous tasks. In particular our qualitative studies identify learning effects as well as how different experimental parameters and conditions affect users' performance.
5. It helps charting a visual design space of pressure-sensitive, pen-based interactions. We do this by presenting a set of three dimensions that can guide designers in describing a par-

INTRODUCTION

ticular pressure-enabled user-interface element, and comparing existing ones and imagining new ones.

We expect this thesis to contribute to the unlocking of pen-based computing's potential for rich interactions, which are currently not possible with the traditional point-and-click paradigm. Not only is this research applicable to the proper design of new and effective ways to interact with pen-based devices, but it will also improve the usability of input devices with similar pressure-sensing capabilities.

2 Background

The purpose of this chapter is twofold. We first present a snapshot of those mechanisms that enable humans to perceive pressure. In the context of this thesis, pressure is directly related to the force one applies to a surface with the tip of a pen or stylus. Having a basic understanding of the human pressure-sensory apparatus is important, as it provides us with insight into what is reasonable to expect from people when they interact with and through pressure. In the second part this chapter we will present literature investigating the use of pressure in the context of the GUI.

2.1 The Perception of Pressure

Pressure can be formally characterized as the amount of force applied per unit area of a surface. For example, a person standing still applies more pressure on the floor through the tip of a stiletto heel than through the heel of a running shoe. However, in the context of this thesis we make the term pressure equivalent with force. Our choice of words should ring true, because in the case of a pen, one applies force through its tip, which has a very small, point-sized contact area. Regardless of the way we describe pressure, it is important to pay attention to the way in which we perceive it, because this perception can directly affect how we interact with these magnitudes.

2.1.1 Cutaneous and Subcutaneous Mechanoreceptors

Humans possess sensory receptors that allow them to perceive information about their surrounding environment, such as sound, light, temperature, touch, pressure, vibration, and cutaneous tension. These receptors are part of a large system known as the somatic sensory sys-

BACKGROUND

tem, which is responsible in humans for the senses of touch, position (proprioceptive sensation), pain and temperature. In this dissertation, we will briefly focus our attention on a specialized group of these receptors that fire signals under the presence of mechanical stimuli, such as pressure. These receptors are specialized neurons known as *mechanoreceptors*, which, when deformed by mechanical forces, allow us to become aware of an object's properties such as shape, texture, and weight. At the same time, mechanoreceptors let us perceive both those forces we apply and that are applied upon us.

The sensory experiences or stimuli we feel through mechanoreceptors can be described through different attributes such as place, spatial extent, quality, intensity and temporal duration. Place or location indicates where in our bodies the stimuli occur. Spatial extent indicates how much of an area a sensation can affect. The success by which we perceive the place and extent of a sensation correlates with the density of a particular sensor in a particular part of our bodies. A stimuli's quality denotes the type of energy it carries, such as temperature, sound, mechanical, etc. Intensity reflects how much energy the stimuli carry. Finally, temporal duration indicates how long-lived the stimuli are. This last magnitude has a direct impact on the *rate of adaptation* of a sensor. In the case of mechanoreceptors, this rate tells us how long it takes for the signal produced by the mechanoreceptor to disappear, if the mechanical stimulus remains constant.

The adaptation rate of mechanoreceptors is an important phenomenon that informs us as to the extent to which people are good at sensing absolute magnitudes of pressure, sensing changes in pressure values or sensing both. This information is crucial for the design of interactions that rely on both the application and perception of force. For example, it is not desirable to have an interaction design that takes too long and allows person's perception of pres-

BACKGROUND

sure to adapt, i.e., a person might believe that is applying a low level of pressure when in fact he or she might be applying a high one.

The importance of this adaptation rate is exemplified by the neuroscience literature (Kandel, Schwartz, & Jessell, 1991), which often classifies mechanoreceptors in our skin as fast-, and slow-adapting receptors. The mechanoreceptors of most interest to us are located in our fingertips, which interface with the environment through glabrous (i.e., smooth, non-hairy) skin. The main mechanoreceptors in this area are the Pacinian Corpuscles, Meissner's Corpuscles, Merkel's discs, and Ruffini corpuscles. Figure 2-1 summarizes the properties of these receptors.

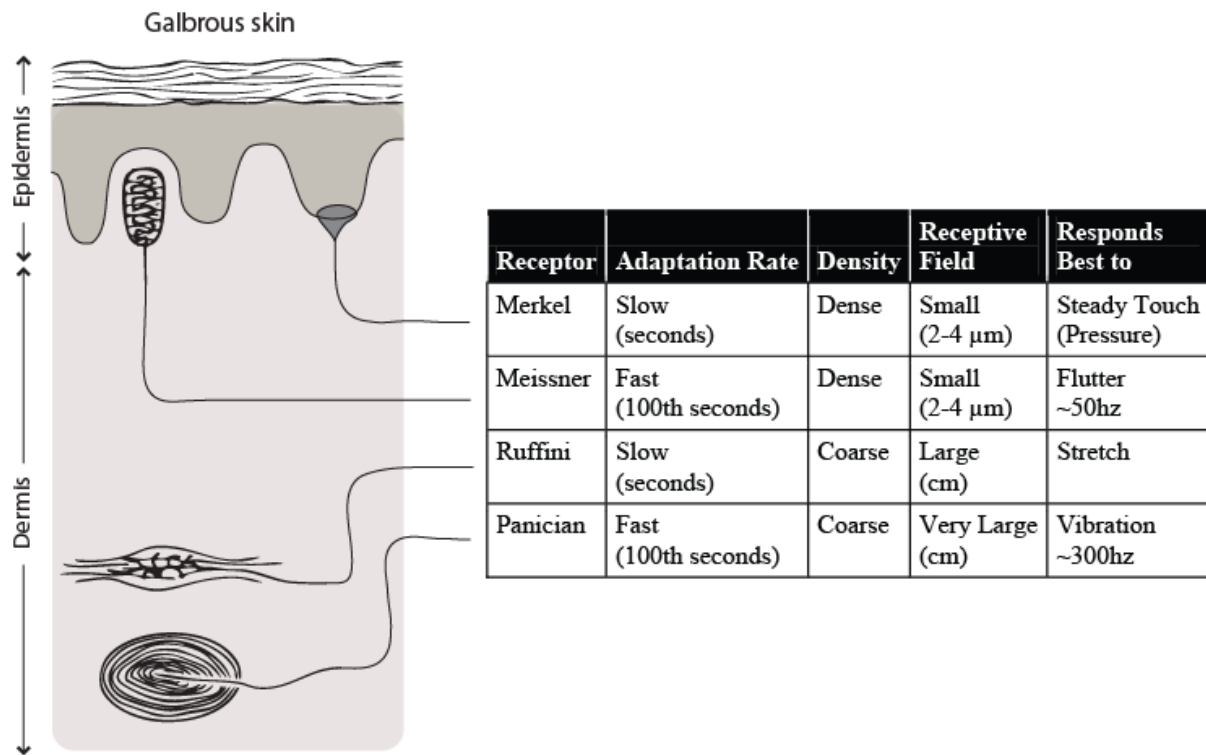


Figure 2-1: Cross section of skin showing main cutaneous tactile mechanoreceptors – adapted from (Goldstein, 2002) and (Kandel et al., 1991).

BACKGROUND

2.1.2 Muscle and Skeletal Mechanoreceptors

In addition to cutaneous and subcutaneous mechanoreceptors, there is another major class of mechanoreceptors, which are an integral part of the human somatic sensory system. These receptors are referred as *proprioceptors* or “receptors for self” and continuously provide information about the position and location of an organism’s muscle-skeletal structure – e.g., limbs. They can provide information not only about a limb’s position, but also about its movement. While not directly linked to the sense of touch or force, proprioceptors are a very important component of the human somatic sensory system. As such, it would be premature to argue that they have no influence the overall sense of touch. We will therefore mention them here.

There are three main types of proprioceptors: joint capsules, muscle spindles and Golgi organs. Muscle spindles are bundles of thin, specialized muscle fibres (Figure 2-2) that run parallel along striated muscle fibres and are attached at both ends to the muscle’s connective tissue. Spindles respond to muscle length by firing signals at different rates that depend on how elongated a muscle bundle is. As such, spindles provide the brain with continuous information about a muscle’s length.

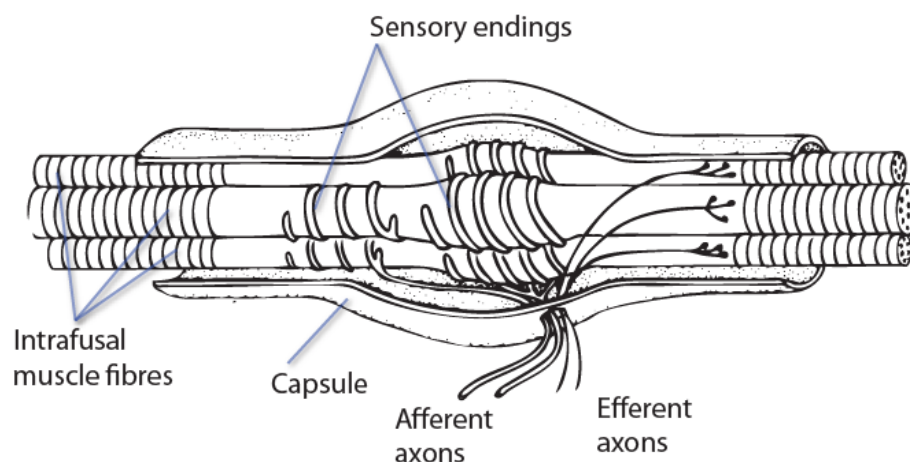


Figure 2-2: Muscle spindle – adapted from (Kandel et al., 1991).

BACKGROUND

In similar numbers as spindles, Golgi organs (or Golgi tendons) are an encapsulated structure located between tendon and muscle. This organ (Figure 2-3) reacts to changes in a muscle's contraction and fires signals to the central nervous system at a rate proportional to the amount a muscle is flexed. Golgi tendons complement the information provided by the spindles: together they provide accurate information about the mechanical state of a muscle at any given time.

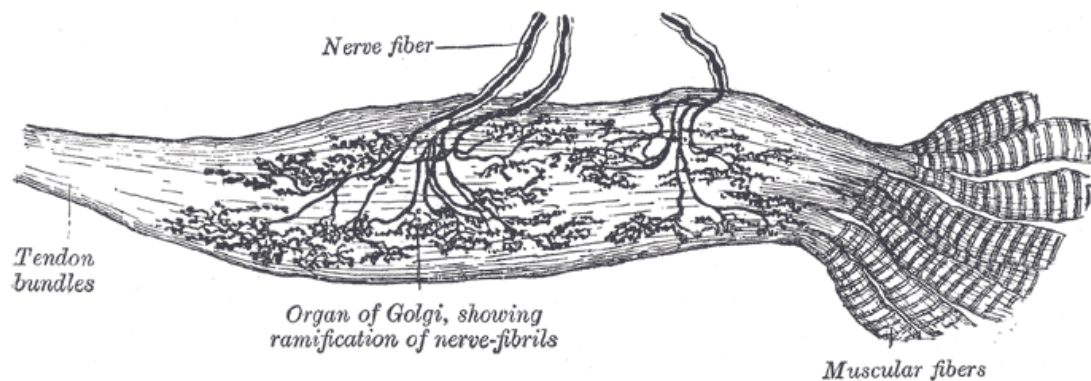


Figure 2-3: Golgi organ – from (Gray & Lewis, 1918)

Afferent (nerve) fibres innervating joint capsules also act as proprioceptors, in particular these fibres end in one of three different encapsulated structures that resemble Ruffini, Paninian and Golgi organs. Because subjects with artificial joints still have a good sense of static position (Kandel et al., 1991), it is believed that joint capsules do not play a dominant role in the perception of static position. However, evidence suggests that joint capsules are sensitive to extreme joint angles. Table 2-1 summarizes general information about proprioceptors.

BACKGROUND

Table 2-1: Main muscular and skeletal proprioceptors

Receptor	Sensation	Adaptation
Muscle Spindle	Muscle stretch	Rapid initial transient and slow sustained
Muscle: Golgi Tendon Organ	Muscle tension	Slow
Joint: Panician	Joint Movement	Rapid
Joint: Ruffini	Joint pressure	Slow
Joint: Golgi Organ	Joint torque	Slow

2.1.3 Psychophysics and the Study of Variations in Signal Intensity

The control of a physical magnitude, such as force or pressure, is intimately linked to the accuracy in which we perceive this physical magnitude. (Kandel et al., 1991) tell us that the ability to extract information about the intensity of external stimuli is important because: a) it allows us to distinguish between two stimuli that differ only in intensity, and b) it allows us to evaluate a stimulus' intensity over a range of values. The ability of an individual to perceive a particular stimulus is characterized by a psychometric function, which is obtained by presenting a subject with a series of stimuli of increasing strength. Figure 2-4 illustrates a typical example of such a function.

This function reveals that the sensitivity of a sensory system varies with the intensity of a stimulus. For example, it is easy to tell 1kg from 2kg, yet it is difficult to perceive the same difference between 50kg and 51kg. This type phenomenon is one of the backbones of the field of *psychophysics*, a discipline that studies the relationships between physical stimuli and its subjective perception. A number of formulations try to model this relationship between perception and a particular stimulus.

BACKGROUND

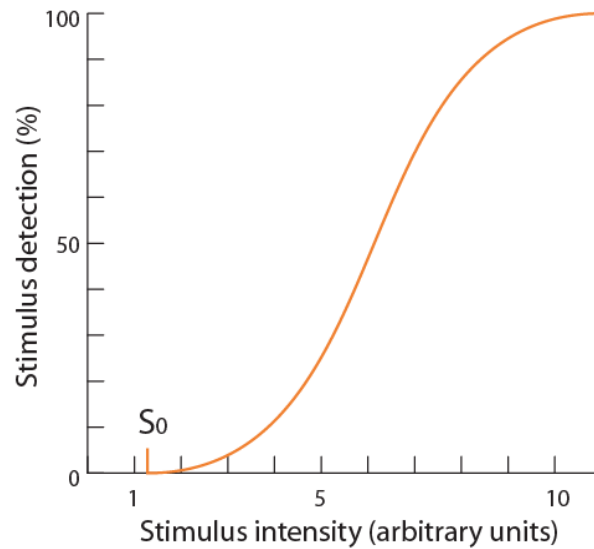


Figure 2-4: Generic psychometric function.

In 1834, Ernst H. Webber proposed the relationship $\Delta S = K \times S$ where ΔS denotes the smallest detectable difference (or just noticeable difference, JND) from a reference stimulus S is proportional, through a constant K , to the stimulus S . Yet, Webber did not provide a way to measure sensation – i.e., his result helped understand when a certain intensity will or will not be detected, but not how much stronger a sensation was when compared to another. In 1860, Fechner extended Webber’s work by proposing that whenever a JND is added to a stimulus, the perceived sensation will increase by a jump of constant size. This way of thinking about units of sensation resulted in a logarithmic formulation of perceived intensity I given a stimulus S :

$I = K \times \log \frac{S}{S_0}$. This formula is known as the Webber-Fechner law, where S_0 represents

the stimulus value below which nothing is perceived (Figure 2-4). Fechner’s formulation remained the prevailing view among researchers (Smelser & Baltes, 2001) until 1956, when Stevens proposed a different, formulation that better explained experimental observations (Stevens, 1975). His formulation describes how sensation intensity I grows as a power func-

BACKGROUND

tion of a stimulus intensity S : $I = K \times (S - S_0)^n$, and is referred to as *Steven's Power law*, where n depends on the type (modality) of the stimulus.

These formulations have proven useful for considering various tasks, including the perception of weight. As such, they are important to consider in the design of interactions that respond to a person's use of pressure. Still, the types of interactions we envision are active and thus differ from the tasks and scenarios used to formulate these psychophysics models – i.e., measurements involving subjects being presented with a particular stimulus as opposed to subjects exerting the stimulus. This difference makes previous results not directly applicable to the type of active interactions we envision and will explore in this thesis. Nevertheless, psychophysics principles can provide valuable guidance for designing appropriate interactions based on the exertion and perception of pressure.

2.1.4 Implications for Design

The design of interaction techniques sensitive to the amount of force we apply through a pen should be guided by our knowledge of the human somatic sensory system. While models of stimulus response have been proposed, they are not universally applicable to all scenarios. For example, there has been no study that can say with certainty that isometric stylus force tasks can be modeled using the previously discussed laws. Some areas remain to be explored, as previous research does not directly apply to the scenarios where our interactions of interest will take place (active vs. passive). Still, the basic properties of mechanoreceptors help us identify two main implications for the design of interactions that make use of pressure:

- Avoid scenarios that lead to sensory adaptation. Mechanoreceptors stop providing precise feedback after a few seconds of constant stimulus (less than 10 seconds). We argue that is desirable to design interactions that last no more than the time it takes a mechanoreceptor

BACKGROUND

to adapt or that give mechanoreceptors no time to adapt. This design principle gives users' perception systems the opportunity to capture accurately pressure information for the duration of a particular interaction.

- Interactions that depend on the difference between forces should account for differences in perception. Psychophysics literature tells us that humans' ability to perceive differences in force intensities varies depending on the intensity of the forces. For example, an interaction that measures the transition between a soft and a hard press should not rely on a fixed, absolute threshold value to be crossed. Instead, it would be better to measure the level and magnitude where a differential of pressure occurs, e.g. an absolute threshold can be crossed accidentally while a relative difference in pressure would not.

2.2 Pressure Input in the GUI: Prior Art

While there have been several efforts that seek to bring forth novel interaction techniques that are sensitive to the unique idiosyncrasies of pens, there is a conspicuous absence of the systematic study of pressure applied through a pen in modern GUIs. Still, the use of pressure in the GUI has been investigated sporadically through different approaches: in case studies, which are anecdotal at best, and in the human perceptual literature that for the most part does not consider the implication of pressure control for GUIs, particularly for pen computing. This section presents the related work prior to our own research.

Among the earlier investigations of the use of pressure in user interfaces is work by Herot and Weinzapfel. They explored the ability of the human finger to apply pressure and torque to a computer screen (Figure 2-5). They implemented and informally tested five interaction techniques that allowed users to control a cursor's position and speed, as well as to push, pull, disperse, and rotate objects displayed on a computer screen. Their conclusions are based on

BACKGROUND

informal feedback, which highlight the benefits that direct manipulation has for the learning process with new interaction methods. Herot and Weinzapfel also observe that accuracy in the input methods that make use of contact forces against a screen is achievable if the system provides continuous, real-time, visual feedback (Herot & Weinzapfel, 1978).

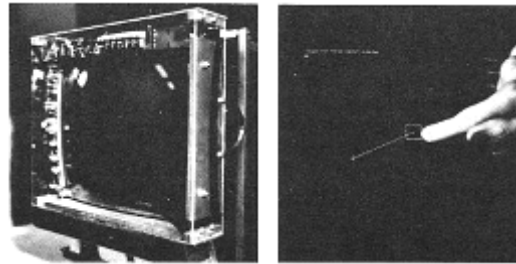


Figure 2-5: Reproduction of photographs of Herot and Weinzapfel's system.

Buxton et al. (Buxton, Hill, & Rowley, 1985) explored touch-sensitive tablet input and suggested that even though pressure control can be difficult in the absence of button clicks or other similar tactile feedback, it still remains a “ripe area of research.” Furthermore, they present an example of a painting application that employs continuous pressure sensing to control the width of the tool used (Figure 2-6). This is perhaps the most commonly used application of pressure input today, and is found in artistic applications such as Adobe's Photoshop or ArtRage.

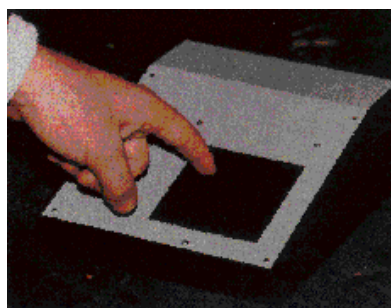


Figure 2-6: Main display for paint program – from (Buxton et al., 1985). Touch tablet used in his demonstrations.

More recently, Zhai conducted a series of experiments quantifying the effects of varying various dimensions of six-degrees-of-freedom input devices on six-degrees-of-freedom manipula-

BACKGROUND

tion and tracking tasks. Of his many observations, the most relevant to our present work was that isotonic² devices perform best when used for position or zero-order control, while isometric³ devices, such as the one illustrated in Figure 2-7, are best suited for rate or first-order control (Zhai, 1995). Still, these results stem from an experiment where the device used is dramatically different from a pen.



Figure 2-7: Spaceball used as a reference isometric device in Zhai's experiments

In other related research, Tan et al. measured several perceptual metrics such as force sensing, pressure perception, position sensing resolution, stiffness, and force control (Figure 2-8). While their results do not apply to the case of pen-based interaction techniques, their methods are relevant to this thesis, since they emphasize how knowledge of human biomechanical, sensorimotor, and cognitive abilities can guide the design of force-reflecting, haptic interfaces (Tan, Srinivasan, Eberman, & Chang, 1994).

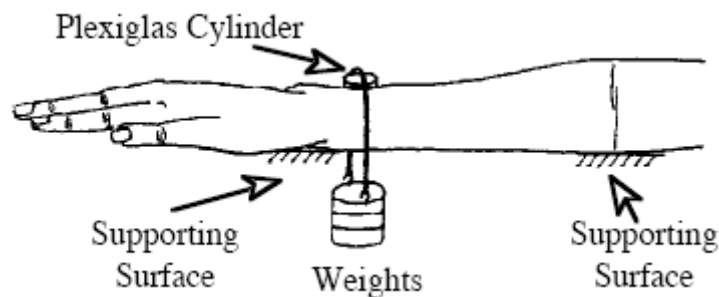


Figure 2-8: Experimental setup for measuring pressure perception at different point on a subject's arm: elbow, and wrist

² The term *isotonic* denotes here an input device where applies force and the encountered resistance is such that the participating muscle's length changes. This interaction usually results in the input device moving.

³ The term *isometric* qualifies here a device where one applies force and the encountered resistance is such that the involved muscle's length remains the same. This interaction results in the input device not moving.

BACKGROUND

More recently Bao and Silverstein (Bao & Silverstein, 2005) collected normative data on pinch and hand grip strengths using a digital dynamometer. In addition to capturing upper bounds on grasping strength, their study shows evidence that people were able to match applied forces. In other words, people were able to apply a level of force similar to one they applied before. The hand pinch is a similar position to the one a person uses when holding a pen (Figure 2-9), thus the results from this article are of relevance to us. However, Bao and Silverman's study do not capture the force applied through what would be the tip of a pen.



Figure 2-9: Pinch grip used on Bao & Silverman's study.

Srinivasan and Chen conducted a controlled study in which users were asked to follow different time-profiles of forces (constant, sinusoid, and linear ramps) that were displayed on a computer screen. Participants controlled the force applied to a pressure sensor using their index finger pad (Figure 2-10), under a number of different experimental conditions (normal vs. anesthetized fingertip). The experiment sought to measure human ability to control contact force against a rigid object, while determining the impact of different sensory feedback (presence vs. absence of visual feedback). While their results shed some light on human performance for that particular task, their conclusions cannot be easily extrapolated to produce significant design recommendations about the number of pressure levels a human can reasonably

BACKGROUND

discriminate, the learning effects that may occur, or the impact that different types of visual feedback may have (Srinivasan & Chen, 1993).

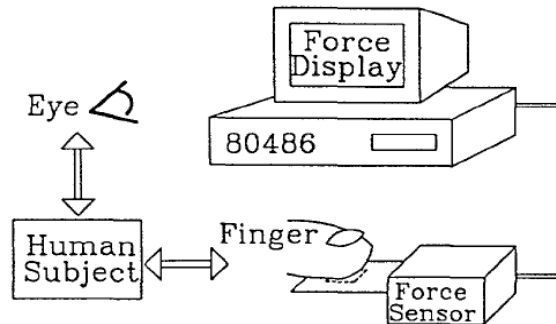


Figure 2-10: Block diagram of the Experimental setup used by Srinivasan and Chen.

Visual feedback seems to be crucial in order to control pressure, and under the right circumstances, it can be very valuable. Lécuyer et al. carried out a series of experiments that compared the stiffness discrimination between a virtual spring and an equivalent actual spring using a SpaceBall isometric device (Figure 2-11). Their findings reveal that, with appropriate visual feedback, an isometric device can be used to simulate haptic information, thus offering the user the illusion of using a non-isometric device (Lécuyer, Coquillart, Kheddar, Richard, & Coiffet, 2000).

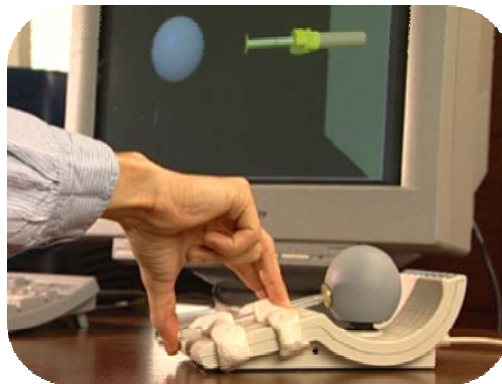


Figure 2-11: Device used by Lecuyer in one of his experiments

The importance of proper visual feedback is a recurrent theme in the related literature. Within the framework of their GeoZui3D visualization system, Komerska et al. developed a haptic widget, which controls the viewpoint of a large 3D data space. Users interact with this widget

BACKGROUND

using a Phantom haptic device (<http://www.sensable.com>). The principles employed in designing this haptic widget emphasize visual (Figure 2-12) and haptic feedback in order to provide users not only with the current state of the input device, but also with indicators that suggest what possible interactions are available (Komerska, Ware, & Plumlee, 2002).

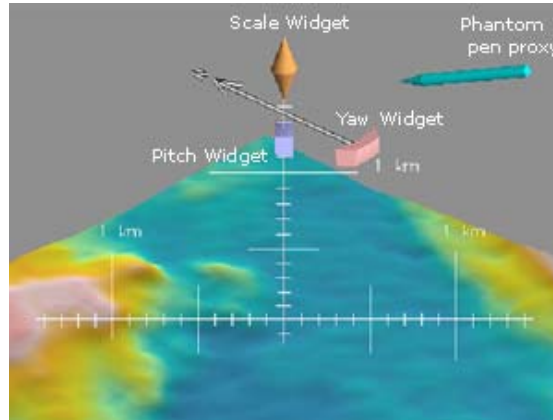


Figure 2-12: The Geozui3d interface

Raisamo evaluated one direct manipulation and four pressure-based area selection techniques for an information kiosk with a pressure-sensitive screen. In the study, users changed the radius of a selection circle by changing the amount of pressure applied to the screen. Different transfer functions that mapped pressure to the circle's radius were used for three of the selection techniques, while the remaining technique incrementally increased the circle's radius based on a pressure threshold. The study reports that even though users had difficulty controlling two of the pressure-based methods, they still ranked the slowest pressure-selection technique almost as highly as the direct manipulation technique. Their overall results indicate that appropriately designed pressure-sensitive interaction techniques could be a practical alternative to standard isotonic methods (Raisamo, 1999).

Currently, many commercial isometric input devices sense and utilize pressure and force information in some meaningful way. Examples of these devices include IBM's Trackpoint joystick, which enables 2D scrolling to be performed independently of the x-y position of the

BACKGROUND

mouse; the DualShock2 controller for the Sony PS2 gaming console, whose buttons translate the pressure the user applies into actions in a game; and pens on digitizing tablets such as the Wacom CintiQ, which have typically been used by artists to vary brush characteristics in drawing and painting programs. There are also exciting emerging technologies that are capable of sensing, to some degree, the force that a user applies. Examples of these are the SmartSkin (Rekimoto, 2002), and Mitsubishi's Diamond-Touch table (Dietz & Leigh, 2001). Figure 2-13 and Figure 2-14 illustrates these aforementioned technologies.



Figure 2-13: Examples of force sensing hardware. (Left) IBM's Trackpoint joystick. (Right) Sony's PS2 DualShock2 controller.

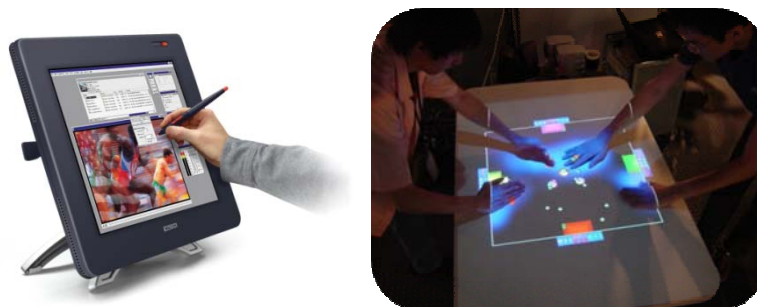


Figure 2-14: Examples of force sensing hardware. (Left) WACOM's CintiQ interactive display. (Right) Rekimoto's SmartSkin can sense the area pressed against its surface, an indirect measure of pressure.

There are also research prototypes that explore devices that augment traditional pointing devices on desktop computers. Zeleznik et al.'s present pop-through buttons as an alternative to the binary off-on switches commonly found on mice (Zeleznik, Miller, & Forsberg, 2001). Figure 2-15 illustrates how with pop-through buttons users press the button lightly to activate its first state (click), and harder to activate its second state (pop). This additional state enables

BACKGROUND

a novel set of interactions. The pop-through concept is a step towards having a continuous transducer, instead of a simple button, but it has not been explored or developed sufficiently.



Figure 2-15: Pop-through mouse and an its three-state button.

In summary, our review indicates that while there is significant number of research efforts in literature on the use of pressure in the user interface, there has not been a systematic investigation into human's ability to control pressure-sensitive pens, nor into the design space of pressure-sensitive widgets. In the following chapters, we present our research efforts aimed at gaining an understanding of both human capabilities to control pressure and of the design space of interactions that make use of a pen's pressure channel of expression.

3 Pen Interactions Incorporating Pressure as an Additional Input Channel: a Case Study.

This chapter describes a case study originating from our desire to confront the design issues and challenges present in the context of pen-based interactions. We also take the opportunity to include in our designs extra channels of expression that a stylus can provide – in particular pressure. We start exploring the design space of pen-based interactions by proposing a set of techniques to control, manipulate, link, and annotate digital video that departs from the traditional, VCR-like metaphor followed by non-linear editors and video players. We implemented and informally tested these new interaction techniques in a concept prototype, dubbed *LEAN*. *LEAN* serves as an exploratory platform for new visualization and interaction techniques that take advantage of the pressure modality of the stylus⁴.

3.1 Video as Media

Video is a data type that has only recently moved to digital form. The increasing availability, and ever-decreasing cost, of digital video capture equipment has resulted in the creation of videos moving beyond the realm of specialists such as filmmakers and TV producers into the broader consumer market. While the ability to capture raw digital video footage has become easy, affordable, and a popular pastime for many, the software applications for navigating and manipulating the resulting hours of footage remain relatively difficult to use – even more so on pen-based platforms. Currently available video manipulation and editing software tends to

⁴ Much of the material presented in this chapter was previously published in the Proceedings of UIST '03 Symposium on User Interface Software and Technology (Ramos & Balakrishnan, 2003).

have user interfaces that mimic the style of old analog editing suites, with all their accompanying idiosyncrasies. However, additional functionality afforded by the non-linear digital form is often buried within layers of menus, and many tasks often involve modal dialogues that disrupt the flow of the user's thoughts and actions. As a result, accomplishing even the simplest of tasks can take inordinate amounts of time and be rather frustrating. In addition, current tools do not easily allow for videos to be annotated or segments to be quickly linked to each other or to other data types.

While these problems are not unique to video, much work has already gone into mitigating them for data types such as text and images, whereas comparatively little research exists on user interfaces for video. Moreover, unlike text or still images, video sets the pace at which it must be experienced. This presents unique interaction and visualization challenges given video's nature as an object existing not only in space, but also in time.

Some of these challenges are worth addressing by leveraging the casual interaction style of a pen, e.g., one can annotate by directly writing on top of the workspace or video frames. Still, there are things to be learned from practitioners that work with video in their work.

3.2 Traditional Video/Film Practices

During the design process of the interaction techniques we developed, we conducted a number of interviews of five professionals from the Cinema and Visual Arts program at University of Toronto and Ryerson University's film school, each professional using video for very different purposes. These uses included the study and critique of film as an art form, the academic use of film/video as a record keeping medium, and the creation and editing of video in a production setting. We interviewed these professionals in their workplace. We solicited feedback on their methods, tools, and current practices. We also either demonstrated early

PEN INTERACTIONS... A CASE STUDY

versions of *LEAN* running on a TabletPC, or played a series of videos that demonstrated the interaction techniques afforded by the system.

Our observations provided us with insight into the current tools and techniques used for interacting with video. They also enabled us to develop and refine our interaction techniques such that they leverage current best practices. People involved in film and video production want to narrate a story. They manipulate and rearrange large quantities of film/video clips in order to arrive at the desired final product. When film is in digital form, *Non Linear Editors* (NLE) like Adobe Premiere or Final Cut Pro are the tools commonly used to cut, paste, and compose movie segments. Digital video allows for the reversible manipulation of its contents, and provides access to an assortment of compositing effects. However, NLEs do not offer the directness in manipulations and interactions that are typical of physical film. For example, interviewees used to working with actual film appreciated being able to simply hold a film strip in both hands and to quickly move it back and forth in order to preview a segment. They are also used to holding it up to the light in order to view the contents of a single frame. In addition, these practitioners are accustomed to using a grease pen to make annotations directly on the film.

Scholars and students who study film as an art form analyze, critique, and communicate their views about a movie's context, history, features, and techniques. Interestingly enough, however, publications and articles in this field exist exclusively in the printed form. As a result, concepts and information relevant to those who study film are transmitted solely with the aid of static images, or at best a sequence of thumbnails accompanied by a textual explanation or transcript. Professors of film studies expressed their dissatisfaction with both the limitations of printed material and with the authoring tools at their disposal. They emphasized the need to

be able to portray the dynamic nature of a particular movie scene, along with its relationship both to other scenes, and to the movie as a whole. Film students face challenges when they need to access and navigate a heterogeneous set of artifacts that includes film, tape, and digital media.

For the non-technically savvy user, having to utilize different tools for media manipulation is a common source of frustration. It is common for practitioners in this area to transcribe a movie clip into text or a log. Once in this form, the transcript becomes a representation of a movie that can be accessed and manipulated using a set of tools (e.g., word processors) with which users are generally more familiar. Ethnographers are particularly concerned with the study and systematic recording of human cultures, and often use video to collect their observations and to analyze them later. The analysis of these videos can involve tasks such as annotating portions of a clip, tagging frames, and organizing the scenes and data into collections. Our observations and interviews strongly suggest that all the aforementioned practitioners would certainly benefit from tools that support casual annotation, linking, control, and dissection of one or more video streams. Furthermore, these tools should be as unobtrusive as possible, allowing users to perform their tasks without a surfeit of user interface widgets cluttering their data space.

All interviewees expressed an intense interest in the early versions of *LEAN*. Even at the almost marginally interactive rates provided currently by the TabletPC hardware it was demonstrated on, one of the interviewees stated that ‘...*I could use a system such as this right now*’. Many of the interviewees shared this positive assessment toward the system as a whole, without emphasizing a particular feature or functionality.

3.3 Related Systems and Techniques

There are a number of pieces of related work that address the areas of non-intrusive interactions, navigation of video streams, and annotations, all of which have influenced our work.

3.3.1 Pen-based Interaction Systems and Techniques

Pen-based computer systems date back to the days of the MIT's Lincoln Labs TX series of computers the late fifties and sixties (Buxton, 2005). Pioneering work in this laboratory made use of a light pen as well as a digitizer tablet in interactive computer systems that enable tasks such as the graphical specification of animations (Baecker, 1969; Baecker, 1974), computer procedures (Sutherland, 1966) or vector-based diagrams (Sutherland, 1963).

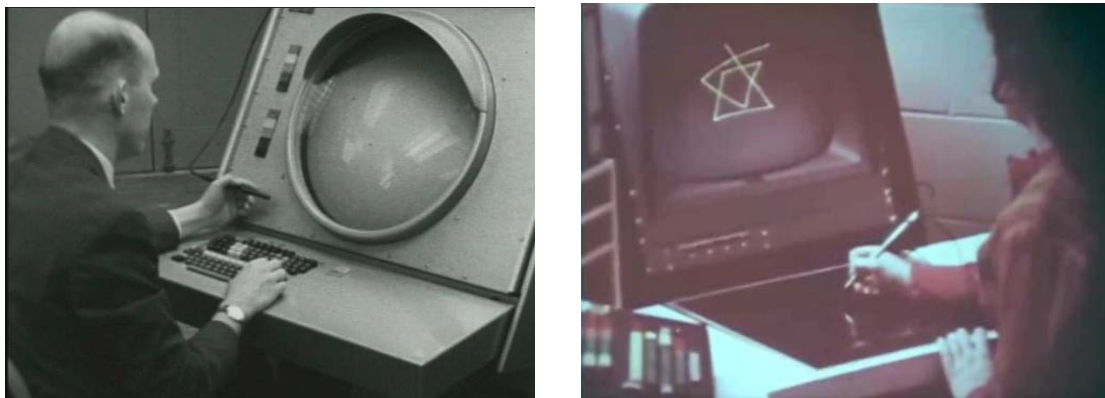


Figure 3-1: (Left) Bert Sutherland using the TX-2 computer terminal. (Right) Animator Lynn Smith using the Genesys system (Baecker, 1974).

It was not until the late 80s that pen-based interfaces resurface in the form of the Wang Freestyle system (Levine & Ehrlich, 1995). The Freestyle was a sophisticated multi-media communication system that used pen-based input to both accept gestural commands and annotate the images of electronic documents including e-mails, scanned papers and faxes.

Interactions using a pen as an input device are frequently showcased in whiteboard interfaces such as in Tivoli (Pederson et al., 1993) and Flatland (Levine & Ehrlich, 1995; Mynatt, Igara-

shi, Edwards, & LaMarca, 1999), or in the work done on large displays by Guimbretière et al. (Guimbretière et al., 2001). Figure 3-2 illustrates screenshots of these two systems.

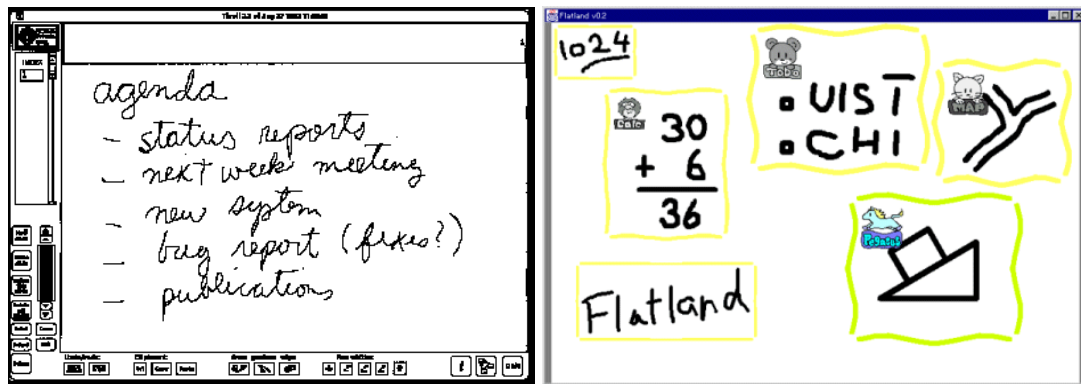


Figure 3-2: (left) Screenshot of the Tivoli system. (Right) Screenshot of the Flatland environment.

These types of interfaces elicit a casual style of interaction, which is not exclusive of large interactive surfaces. The Electronic Cocktail Napkin (Gross & Do, 1996) is a pen-based environment that supports the abstraction, imprecision, and ambiguity of freehand diagrams made by users. The system (Figure 3-3 left) parses the ink drawings and is able to recognize and disambiguate shapes, based on the drawing's context and structure. The XLibris system (Schilit et al., 1998) imitates paper by using a high-resolution pen tablet display that provides users with some of the affordances of paper. With XLibris (Figure 3-3r ight), users can annotate and highlight pages of documents with an ease approaching that of printed materials. XLibris departs from the traditional WIMP interface and follows the design principles of a transparent, minimalist user interface and modeless interaction.

PEN INTERACTIONS... A CASE STUDY

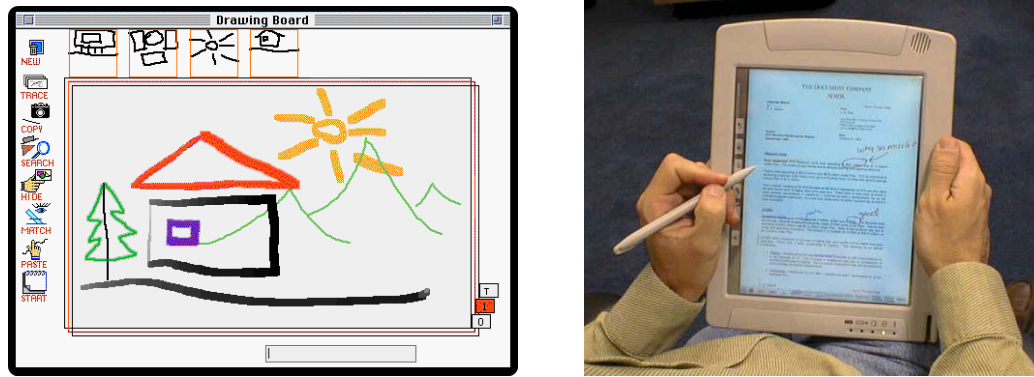


Figure 3-3: (left) Screenshot of the Gross' Electronic Cocktail Napkin. (Right) Photo of the Xerox's Xlibris prototype.

One of the virtues of a minimalistic interface is that of not providing unnecessary distractions for users during their work. To a certain degree, contextual, in-situ tools can also provide minimal interference during a task. Toolglasses (Bier, Stone, Pier, Buxton, & DeRose, 1993) provide users with a bimanual, nonintrusive tool that does not distract their attention from the tasks at hand (Figure 3-4 right). Another non-intrusive technique is Marking Menus (Kurtenbach & Buxton, 1993). Marking Menus are transient widgets that allow users to have access to commands in a fluid manner. With Marking Menus (Figure 3-4 left), novice users can take advantage of a hierarchical radial menu structure, while advanced users can access commands by making a mark, or gesture, without having to wait for the menu to appear. FlowMenus (Guimbretière & Winograd, 2000) (Figure 3-4 centre), FaST sliders (McGuffin, Burtnyk, & Kurtenbach, 2002), and Control Menus (Pook, Lecolinet, Vaysseix, & Barillot, 2000) present quick, easy to learn, and transient controls that combine menu selection and the adjustment of continuous values. In addition, FaST sliders allow users to switch quickly between different scale granularities when adjusting parameter values.

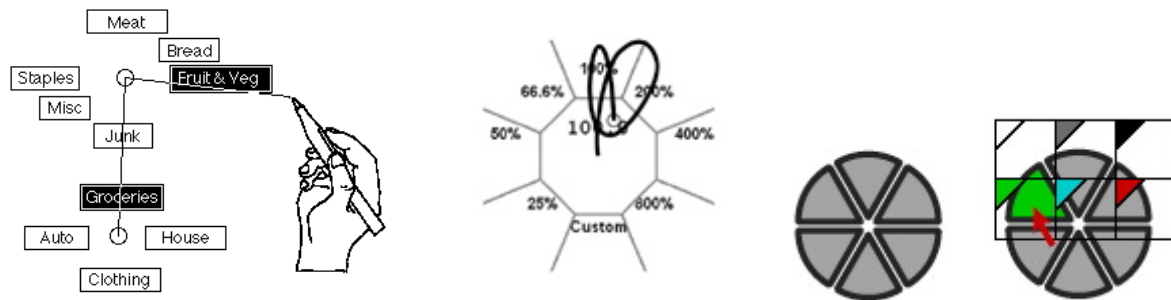


Figure 3-4:(Left) Marking menu technique. (Centre) Flowmenu widget allows for the fluid and continuous adjustment of a parameter. (Right) A color toolglass allow for the color specification of a graphical object.

Since the implementation and publication of our initial research on pen interaction techniques (Ramos & Balakrishnan, 2003), a number of system and interactions embody a significant number of contributions in the area. For example, CrossY (Apitz & Guimbretière, 2004) is a pen-based drawing program that explores what shape controls could take if they could be driven by crossings instead of clicks and arguably allows for a more fluid interaction workflow (Figure 3-5 left). Scriboli (Hinckley, Baudisch, Ramos, & Guimbretiere, 2005) is another pen-based system that explores the role of delimiters in pen gestures. Delimiters are interaction separators used to determine the structure of interactive phrases and can break the flow of an interaction. In particular, the Scriboli system served as test bed of many pen-based interaction techniques, such as the use of pigtail delimiters. Zeleznik et al.'s Fluid Inking proposed a different syntax-based approach where prefix clicking and postfix punctuations can disambiguate inking from gestures (Zeleznik & Miller, 2006) .

It is often the case that pen-based systems posses more than one mode of operation, e.g., marking, drawing, manipulating, etc. Li et al. investigate different mode switching techniques for pen-based interfaces and finds that a switch activated by a person's non-dominant hand is the best mechanism for choosing between two modes (Li, Hinckley, Guan, & Landay, 2005). When a tablet is in slate mode, the availability and appropriateness of such a secondary button

might be in question because there is no magic location that can guarantee equally accessibility to different people and holding grips. Hover widgets (Grossman, Hinckley, Baudisch, Agrawala, & Balakrishnan, 2006) provide a solution to this problem by giving users access to different modes by performing short gestures while a pen is within tracking (hover) range (Figure 3-5 right).

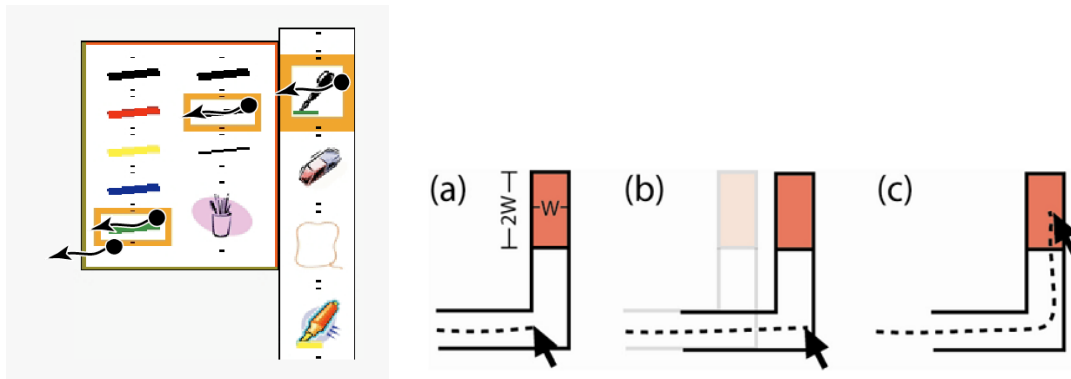


Figure 3-5:(left) In the CrossY system users cross elements in the interface to perform actions, in this case choosing a line's color. (Right) Diagram exemplifying a hover widget interaction where as a cursor moves through a widget's tunnel (a, b,) to eventually reach its activation area in red (c).

3.3.2 Video Annotation and Navigation Systems

Snibbe et al. explore interactions techniques for digital media such as video through haptically actuated controls, such as knobs and sliders (Snibbe et al., 2001). Haptic feedback also allowed them to present points of interest, such as bookmarks, annotations or transitions, as bumps, which allowed users to use their sense of touch to perceive them. Based on active elements, this research presents a compelling use of non-traditional input channels in the computer interface.

SILVER (Casares, Long, Myers, Stevens, & Corbett, 2002) is a video-editing tool that introduced a number of interaction and visualization techniques. Of particular interest to us is the system's Timeline View, which displays an explicit 3-level hierarchy that is defined when the user zooms down into a video segment (Figure 3-6 left). This hierarchy is useful for navigat-

ing through the time-line of the video. Users can also add text annotations that span a portion of a video segment. Our *LEAN* system is similar in the way it handles the visualization of video segment relationships, but it does not have the limitation of allowing only a 3-level hierarchy.

The VANNA system (Harrison & Baecker, 1992) allows people to manipulate and annotate temporal information. The system supports a variety of input devices, e.g. mouse, keyboard, touch screen, and pen, all of which can be used to capture either on-line or off-line notes. The MAD system (Baecker, Rosenthal, Friedlander, Smith, & Cohen, 1996) for authoring digital video as hierarchically structured multimedia documents also allow for user-defined annotations of time-based media. These multi-media documents (consisting of a combination of text, images, audio and video) could be annotated during the authoring process with elements such as text or voice.

The PhotoFinder system (Shneiderman & Kang, 2000) addresses the complexity of a large collection of annotated images by allowing users to drag-and-drop labels from a scrolling list of attribute values to a particular place on a photo (Figure 3-6 right). The Boom Chameleon (Tsang, Fitzmaurice, Kurtenbach, Khan, & Buxton, 2002) introduces a specialized input and output device that allows users to navigate and annotate a 3-D environment. Users make annotations on this system by drawing directly on the surface of a virtual object, or by taking 2-D snapshots that capture the user's point of view at a given point in time.

PEN INTERACTIONS... A CASE STUDY

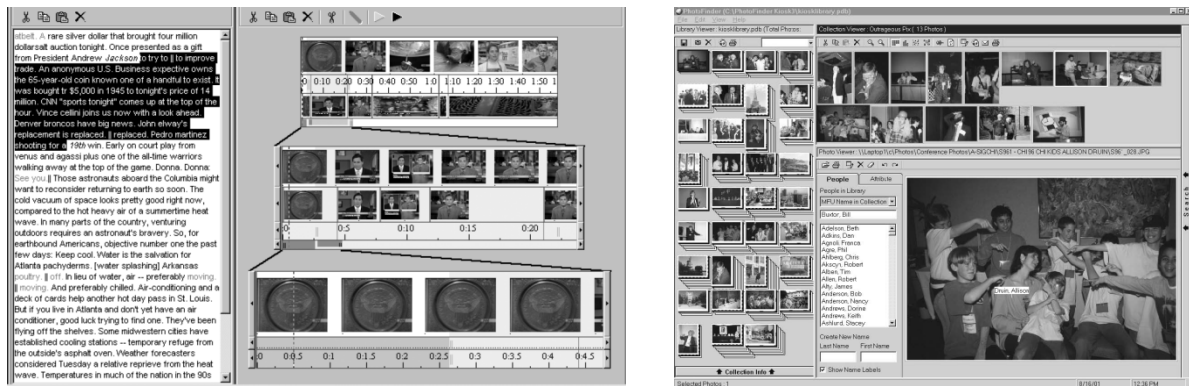


Figure 3-6: (left) SILVER's video editing and annotation interface is divided in two regions: one for annotations (e.g. dubbing) and other where a video stream can be visualized at up to three levels of detail. (Right) Photofinder's interface consists of three main areas: one where different piles or folders with photos exist, another that displays the contents of a pile/folder, and another where a particular photo can be inspected and annotated by simply dragging and dropping labels from a nearby list.

In short, our review of the literature indicates that while many of the issues with which we are concerned – video, annotations, linking, casual interactions, and uncluttered workspaces facilitated by transient widgets – have been explored individually by various researchers, they have yet to be explored in combination. Also, since the original research (Ramos & Balakrishnan, 2003) presented in this chapter, a number of significant research efforts have appeared.

The Family Video Archive (Abowd, Gauger, & Lachenmann, 2003) is a system that allows for the browsing, annotation and filtering of a person's collection of family movies. The system aids users in detecting and defining a scene's boundaries as well as allowing for the tagging of video scenes with textual metadata. Through a zoomable user interface (Bederson & Hollan, 1995; Perlin & Fox, 1993) the system permits users to browse through a collection of scenes from potentially many different video sources. Finally, the archive lets users search through its contents, thanks to user-defined meta-data embedded into the video scenes.

The Family Video Archive offers powerful functionality, but it requires the use of a mouse and keyboard configuration. In contrast to this, the M4Note system (Goularte, Camacho-

Guerrero, Valter R. Inacio, Cattelan, & Pimentel, 2004; Goularte, Cattelan, Camacho-Guerrero, Jr., & Maria da, 2004) annotates multimedia content using a pen-based system. These annotations are used for indexing, semantic processing and querying of a video stream. Another system that leverages the capabilities of pen computers is Videotater (Diakopoulos & Essa, 2006), which allows for the navigation, selection, segmentation, and tagging of digital video. Videotater's interface is divided in three main sections, each devoted to a particular task: tag creation and selection, video playback, and timeline navigation and manipulation (Figure 3-7). This system draws from our research and makes use of the pressure applied through a pen to differentiate between selection and tagging (scene splitting/merging) modalities. This system presents an interesting evolution from concepts we introduce in this chapter. Still, there are issues of scaling such, as video length or screen size, which remain the subject of further studies.

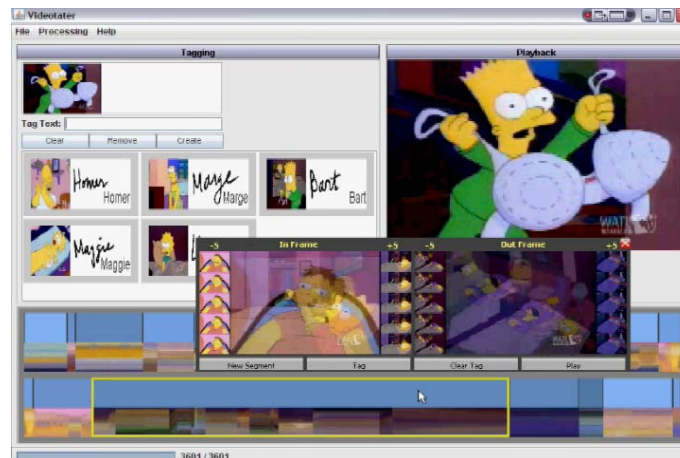


Figure 3-7: Videotater's interface consists of three main regions: tag creation (top left), timeline segments (bottom), and playback window (top right). As a user hover over a timeline segment, a detailed popup frame for the segment appears

3.4 Overview and Design Philosophy of LEAN

We developed a system called LEAN (Figure 3-8) that serves as an exploratory platform for new visualization and interaction techniques for navigating and controlling digital video. Our system targets the casual user, and in addition to various editing operations, allows for casual

PEN INTERACTIONS... A CASE STUDY

annotation and cross-linking of video streams. Its primary interface is a digitizer tablet with a pressure-sensitive pen. Our intention is to leverage users' familiarity with pen-based interactions in the physical world, and the emerging tablet-based computers.

LEAN allows for the manipulation of a video stream by using a small set of gestures that lets users start, stop, and travel to any arbitrary point in time in the stream. Also, by using only simple gestures, users are able to select intervals, or segments, from the video. Besides allowing users to manipulate the video stream, the system also permits users to attach annotations – easily created by scribbling on the working area or over the video image – to video frames and segments. By connecting an annotation to a desired element on the working area, the user can provide it with a positional and temporal context. In addition, users can trigger at will visualizations that correspond to a complete video segment and that allow for both the quick navigation of the video stream and the speedy location of the annotations situated within. In designing LEAN, we were particularly interested in creating techniques to enable users to navigate and annotate digital video with fluidity and ease similar to navigating and making annotations on printed material using physical tools such as pens and post-it notes. Another goal was the design of appropriate visualizations for the subsequent retrieval and viewing of those annotations. In our design, we strove for a minimalist approach to the interface, both in the gesture set used, and in the visual aspects of the design, believing that an excess of visual decorations introduces noise to the task at hand and only serves to make the user acutely aware of the intrusive presence of the computer.

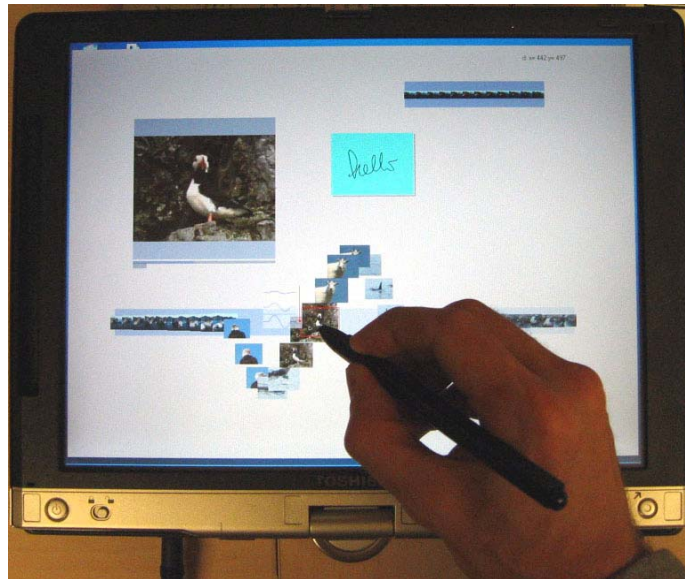


Figure 3-8: The LEAN system running on a Tablet PC

3.5 Gestures, Commands, and Scribbling

Systems that use a pen as an input device for both commands and data input have to contend with the ambiguity that often results when interpreting the user's input actions. For example, an input stroke could have several meanings: a gesture intended to invoke a command, a simple scribble, or a simple pointer movement. Previous research systems have adopted different approaches to address these ambiguities. For example, Flatland (Mynatt et al., 1999) uses a button on the pen to divide the user's input into two modes: drawings and meta-strokes, and a tap gesture to invoke a pie-menu for command entry. DENIM (Lin, Newman, Hong, & Landay, 2000) separates scribbles and commands by using a button on the pen, and also by using a tap gesture to invoke a pie-menu that then provides users with further commands. Guimbretière et al. use a button on the pen to invoke a FlowMenu for command input (Guimbretière & Winograd, 2000). Another approach is to interpret the input strokes and classify them into either command gestures or raw scribbles.

PEN INTERACTIONS... A CASE STUDY

We use a combination of these approaches – e.g., a mode switch and parsing. A small set of gestures is interpreted by parsing single-stroke inputs using Rubine’s features (Rubine, 1991). The effect a gesture has depends on the context in which it is made, i.e. the object(s) upon which it is made. Table 3-1 summarizes this gesture set. We will explain the various gestures and their interpretations in detail as we proceed through this chapter. With the exception of ‘selecting’ objects, we found that for the purposes of our initial research, it sufficed that the system distinguishes between scribbles and commands by a simple algorithm that tests a stroke’s features such as space, time, speed, and pressure. Objects in LEAN are ‘selected’ in the working area by using the pen’s button, all without even having to touch the tablet’s surface. Chosen objects reveal their links, and can be later moved over the workspace by moving the pen over the tablet’s surface while simply keeping the pen’s button pressed.

Our system also uses menus and widgets that are invoked by *Tapping-And-Holding* the pen on the tablet’s surface for a small period of time, after which the control appears or becomes active. This is similar to the way marking menus were originally invoked (Kurtenbach & Buxton, 1993). An animated icon, similar to the one found in the Apple Newton or in Windows Mobile platform provides users with feedback regarding the initiation and completion of the tap-and-hold gesture.

Table 3-1: Gesture grid that shows the basic set of gestures recognized by the *LEAN* system. The top row shows the object that gestures can be applied upon, while the leftmost column enumerates the basic set of gestures. A grid's cell describes the effect of a particular gesture on a certain object. The *TLSlider* and *PVSlider* are elements of the interface we will discuss in the following sections.

	Video Frame 	Video Segment (VS) 	Note 	Link 	Empty space
Flick Right 	Invoke main <i>TLSlider</i>	n/a	Draw line	n/a	Create Note
Flick Left 	Hide main <i>TLSlider</i>	Hide / Minimize <i>TLSlider</i> (1)	Delete Note	n/a	
Interval 	Annotate frame	Create new Video Segment / <i>TLSlider</i>	Add scribble to Note	n/a	
Tap-And-Hold + Scrub	Control video flow	Navigate through timeline	Navigate through links	n/a	n/a
Tap-And-Hold + Pressure	Select beginning / end of new VS	Vary amplitude of Twist Lens	Pin / Unpin Note	Grab	n/a
Scribble 	Annotate frame	n/a	Add scribble to Note	n/a	Create Note
Tap	Start / Pause Video	Jump to Frame	n/a	n/a	n/a
Pen's Button	Select Object / Reveal Relationships				
Pen's Button + Move	Move Object				

3.6 Pressure and Pressure Widgets

Our system uses the pressure information from the pen to expand the set of directly available commands to the user. A pen's pressure is sometimes used in image manipulation programs like Adobe Photoshop to control some continuous parameters of a drawing tool, such as the thickness of a pencil or the opacity of a brush. However, traditional WIMP interfaces assume that a user's pointing device can only produce spatial x-y position coordinates and discrete clicks as input to a system. As such, their widgets are designed only for these two input types and do not take full advantage of the pen's pressure modality. To leverage the capabilities of the pressure-sensitive pen, we developed visual Pressure Widgets that help users become

aware of the amount of pressure applied, and the consequences of varying the pen's pressure (Figure 3-9).

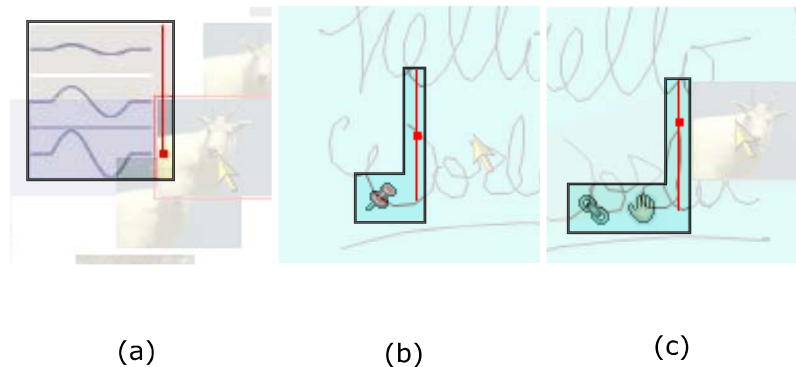


Figure 3-9: Pressure Widgets (background of this figure has been altered in order to emphasize the widget's appearance). a) Continuous control of the amplitude of the Twist-Lens. b) Discrete control for pinning a note to the workspace. The pinning action occurs after the pressure exceeds the displayed threshold. c) Discrete control for grabbing a link. A sequential icon indicates the action of grabbing and the item to be grabbed which is a link. A video displaying this pressure widgets can be found at <http://youtube.com/watch?v=c-4tjto6Akw> – timecode 2:05.

Discrete pressure widgets activate an action once a certain pressure threshold is exceeded, while continuous pressure widgets map pressure to the control of a continuous parameter. The key element of pressure widgets is the visual display of the effects of the changing pressure. For continuous pressure widgets, we use a series of icons that reflect the consequences of the user's actions (Figure 3-9a). For discrete pressure widgets, we use a single icon (Figure 3-9b), or set of icons (Figure 3-9c), displayed at the appropriate pressure threshold. Instead of employing complex icons to describe compound actions, we chose a small, simple set of icons that can be combined in what we call sequential icons (Figure 3-9c). Sequential icons are likely to be simpler to learn than composite ones.

3.7 Video Control

The control of a video stream in most software is carried out using a VCR-like interface (Figure 3-10), with different buttons or widgets that play, pause, fast forward, or rewind the video. In addition, clicking on the timeline often directly positions the video at a particular

point in time. Such an interface produces a separation between the video data with which users are engaged, and the widgets necessary to control it. This strategy of separating the controls from the data works with text documents and other types of non-temporal material, because of their static nature. In these cases, we expect (and are usually not disappointed) that a small switch in our attention from the document to the control and back will return us to the same view of the document. The same cannot be said about video – a media that changes as time passes, when engaged. In video, this separation between controls and data forces users to play a ‘game’ of target acquisition, which is unnecessary and quite avoidable in a properly designed video control interface.

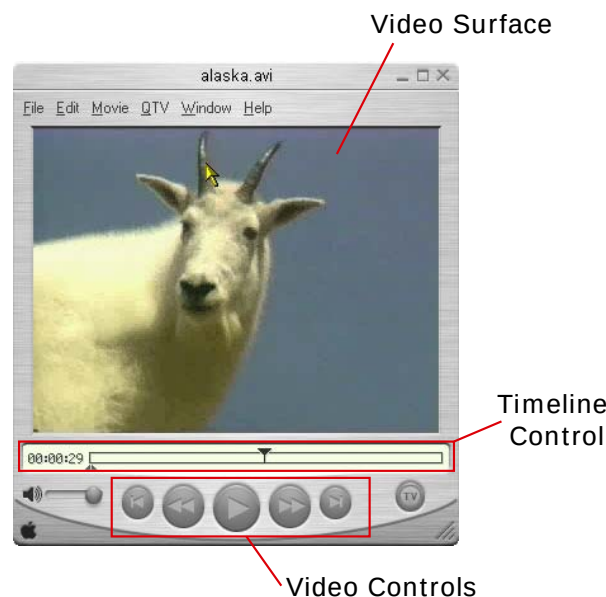


Figure 3-10: A typical video player with a VCR-like media control widget. This interface separates the document (Video Surface) and the widgets that control it (Timeline and Video Controls).

3.7.1 Position+Velocity Sliders

We incorporated a number of interaction techniques into a ‘one-stop shopping’ solution for the non-intrusive control of a video stream. Users can start and stop a video by tapping on the video surface. Users perform fast forward and rewind functions by using a novel, unobtrusive transient position+velocity slider widget, called the PVslider. The PVslider (Figure 3-11) is a

PEN INTERACTIONS... A CASE STUDY

hybrid position+velocity control that allows users to drag across the tablet's surface in order to move within the vicinity of the current frame using position control, or to move forwards or backwards in the stream at a variable rate using velocity control. The PVslider is invoked when the user taps and holds over the video, a gesture that defines the point of origin (PO) of the control. The control looks like a horizontal line segment, which follows the pointer in the vertical dimension and remains connected to PO with a line, or 'rubber-band', linking the pen's position and PO (Figure 3-11b).

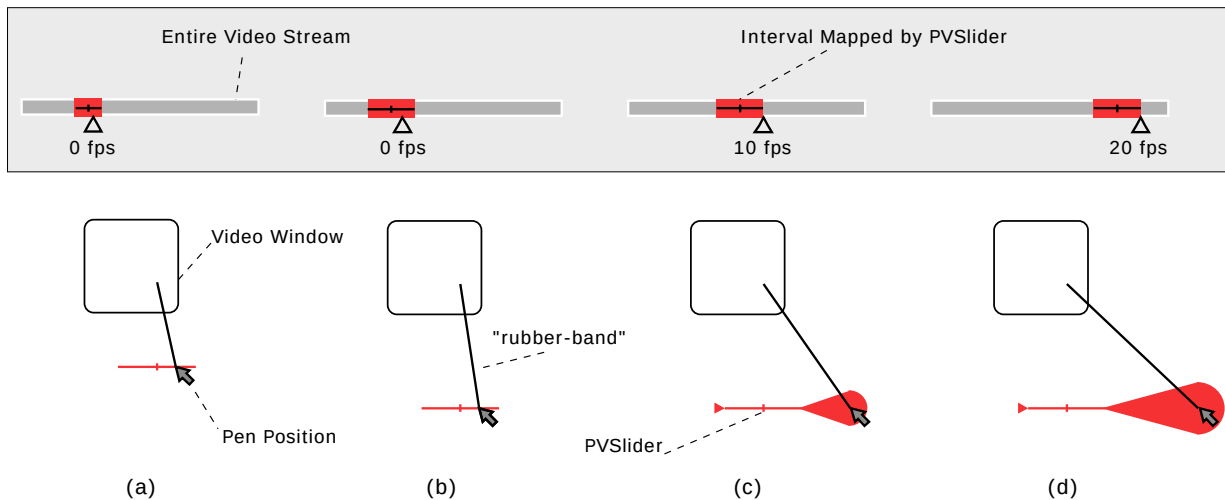


Figure 3-11: The PVslider widget and features. a) The PVslider is connected to the point of origin (PO), and mapped to an interval of the video stream. Note: the grey box above it is not part of the interface; it is here for illustrative purposes. Also the frames-per-second (fps) values are illustrative and do not correspond to real data. b) As the pen's vertical distance to PO changes, the size of the interval mapped changes. c, d) Moving the pen beyond the Position Region takes it into the Velocity Region. The farther away the pen is from the starting point in the horizontal direction, the faster the users move through the video stream. The size of the Velocity Region cone provides visual feedback on the magnitude of the current speed. A video illustrating this widget can be found at <http://youtube.com/watch?v=c-4tjt06Akw> – timestamp 0:28.

The PVslider is divided into a *Position Region* and a *Velocity Region*. The *Position Region* of the PVSlider is the horizontal line the user sees (Figure 3-11a). This line is mapped to an interval on the video stream centered around the frame where the control was invoked. The size of this interval is directly proportional to the vertical distance between PO and the current pen's position. As such, the interval's size can be changed by moving the pen in the vertical

direction (Figure 3-11a, b). Moving the pen in the horizontal direction within the boundaries of the *Position Region* allows the user to scrub through the frames in the given interval. The user enters the *Velocity Region* by dragging the pen horizontally beyond the ends of the *Position Region* (Figure 3-11c-d). Here the PVslider acts as a velocity control allowing the user to move through the video stream at a velocity proportional to the length of the rubber-band – i.e., the farther away the pen moves from PO, the faster the user moves across the video stream in that direction. Thus, users can fast forward or rewind the video by dragging to the right or left of PO. Note that the transition from position to velocity control is seamless, with no explicit mode switch. Rather, the switch is implicit, based simply on the distance of the cursor from the PO in the horizontal direction. In addition, the PVslider constantly provides visual feedback indicating its current status as either a position or velocity control, along with the magnitude of the speed at which the user moves through the video stream (Figure 3-11c, d).

3.7.2 Twist-Lens Sliders

Although the PVslider offers users an absolute position control, this region does not map to the whole length of the video stream the same way slider controls on VCR-like interfaces do. With this in mind, we developed a novel interaction and visualization technique based on fish-eye lenses called the Twist Lens slider or TLslider. Using a flick right gesture (Table 3-1), a user invokes the TLslider, which provides a visualization of the complete video stream as a sequence of thumbnails. Once a user taps and holds on the TLslider, it acts as an absolute position control for the portion of the video stream to which it is mapped. When the TLslider becomes active, the user can drag across the control with the pen and the result is that the fish-eye view expands the area centered at the location of the pointer. While the visualization

of the TLslider enables the frames of interest to be expanded visually, our design does not expand the targets in the motor domain because of the issues regarding target acquisition that have been studied in detail by (McGuffin & Balakrishnan, 2002). In this study, a widget with multiple targets expanding in the motor domain, the motor location of the targets typically shifts as the targets change size, making them difficult to acquire. Such an effect is visible in the ‘dock’ in the Mac OS X interface. Instead, we keep the mapping between the video frames and the space defined by the TLslider constant. However, this design choice presents another challenge: the frames visually expanded by the fish-eye view partially occlude their neighbors, or context (Figure 3-12).

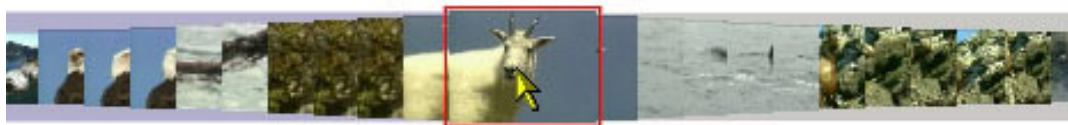


Figure 3-12: This partial view of the *TLslider* shows how a regular fish-eye approach that keeps a fixed target size may present occlusion problems in the vicinity of the focus. A video displaying this widget can be found at <http://youtube.com/watch?v=c-4tjto6Akw> – timecode 1:35.

We overcome this problem in two ways. First, the thumbnail that is the focus of attention shows not an enlarged version of the closest key frame, but the actual frame corresponding to that particular point in time. Second, we morph the linear layout to an s-shape (which gives this technique its name) that depends on the pressure applied by the user’s pen on the tablet’s surface. Figure 3-13 illustrates this. The figure shows from top to bottom how the amplitude of the lens changes with the magnitude of the applied pressure through the pen, displayed on the right. A continuous pressure widget (Figure 3-9a) provides a visual preview of the results of varying the pressure. By showing the precise frame at a particular point in time, instead of a static thumbnail representing an interval, we allow users to preview accurately their moving through the timeline. By smoothly morphing the slider into a sinusoidal shape, we create sufficient space to eliminate occlusion among thumbnails. We found that this distortion tech-

nique has the added bonus of providing a visualization that is not occluded by the user's hand as is often the case in devices that integrate display and digitizer (e.g. Wacom CintiQ or TabletPC), and that can also accommodate, by mirroring its shape, both right-handed or left-handed users (Figure 3-8).

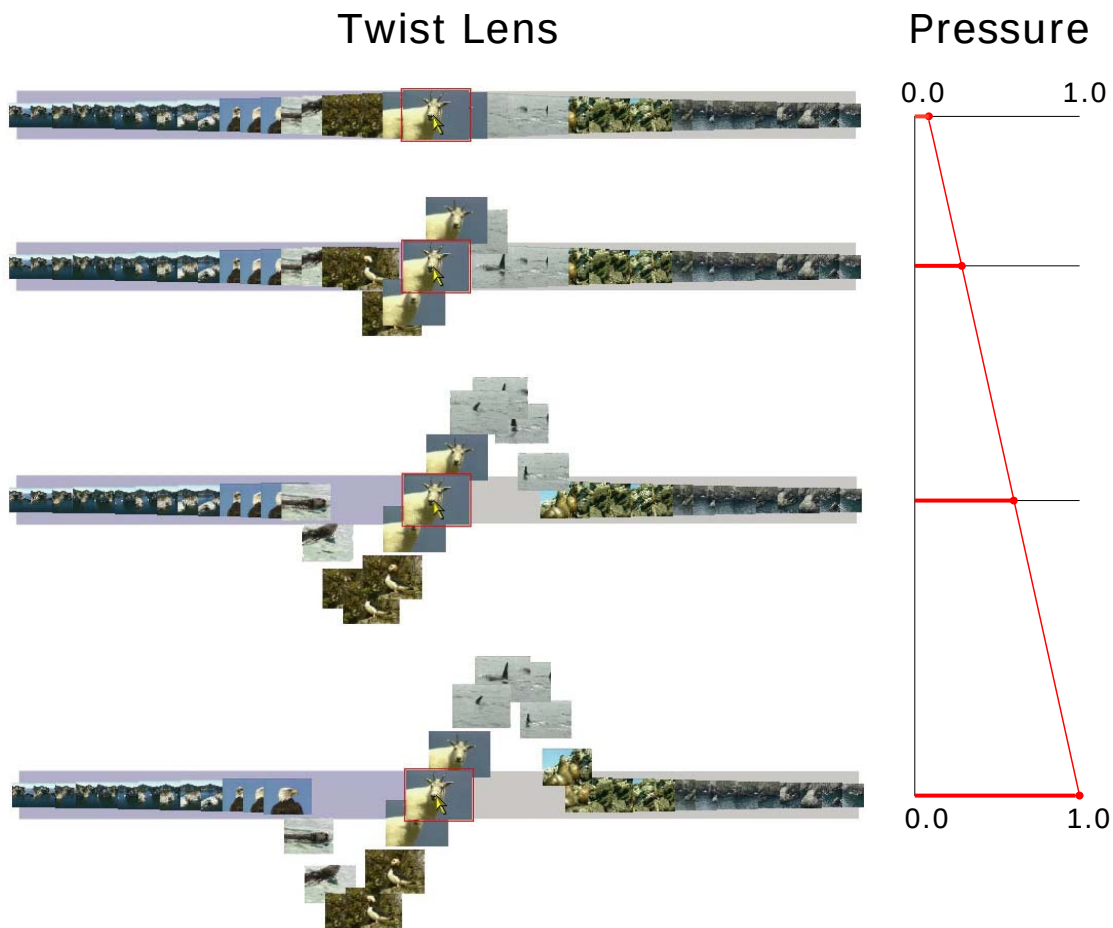


Figure 3-13: Twist Lens Slider. The figure shows from top to bottom how the amplitude of the lens changes with the pen's pressure, which is displayed on the right. A video displaying this pressure widgets can be found at <http://youtube.com/watch?v=c-4tjto6Akw> – timecode 1:35.

3.7.3 Video Segments

In our system, the TLslider is also a particular instance of a more generic object, a Video Segment. Video Segments are sections of the video stream that the user can define simply by selecting an initial and final frame, or by using a gesture to select an interval from an existing Video Segment. Video Segments also indicate the progress of the video stream, by changing

over time the color of its background border from grey to blue as the video plays. Unlike typical progress bars found in most video players, which are spatially separate from the associated video stream, ours does not divide the user's attention. This feature allows users to see at a glance if the segment has been already played, if it is currently being played, or if it has not been played yet.

In order to unclutter the workspace users can collapse a Video Segment into an iconic representation with a simple flick gesture (Table 3-1). We also support the user's need to see relationships – for example, if a Video Segment is fully or partially contained in another. When a user grabs a segment, the system automatically displays its immediate relationships to other segments via a series of semi-transparent 'large-base' arrows, as shown in Figure 3-16. Video Segments can be used to structure a video stream into different pieces that can then be used to support tasks such as the analysis of film and the navigation through a video stream. In a sense, this is analogous to the traditional practice of using a pair of scissors to cut film into strips that we observed during our user interviews and task analysis.

3.8 Annotations and Links

Apart from providing controls for video navigation and segmentation, another primary goal of our work was to research techniques for annotating video. Because of its widespread use and undeniable fluidity (a process where the stream of the task at hand is not broken), active reading on paper is used as a model to study, and from which to generalize, the practice of annotation (Marshall, 1997), or as a metaphor for systems and interface design (Schilit et al., 1998). To a certain extent, we follow this approach and let users create explicit annotations by writing directly into the empty area of the screen. They can then connect the resulting note to a

movie frame or a Video Segment. Users can also scribble on top of a video frame in order to leave ‘in-place’ markings on a particular frame.

From (Marshall, 1997) we learn that annotations have both form and function. One of the most significant attributes of an annotation’s form is its location. For example, a note on the margin of a book has a location near some printed text that is likely related to what was hand-written. In addition, the portion of a photograph where a circle was drawn, or the moment at which a voice comment was made, also demonstrates the importance of an annotation’s location, regardless of the type of media. An annotation only becomes useful because of its location and its relationship with the surrounding context. When dealing with printed material, a mere visual inspection can reveal both the annotation and its context. However, this is not the case with a video stream, where the context can be not only space, but also time. When the context of an annotation is temporal, a person must experience the media through time until the moment when the annotation was actually made occurs. The nature of temporal context does not allow us to experience the previous and future moments that surround an annotation’s place with a quick glance, unlike the way we experience spatial context. In order to provide the user with a similar type of contextual awareness as occurs with annotations made in space, we have developed an approach that visually blends a linked annotation smoothly in and out of the environment, as the moment (or time interval in the case of a Video Segment) when the annotation was made approaches (Figure 3-14a, b, c), and then passes (Figure 3-14c, d, e).

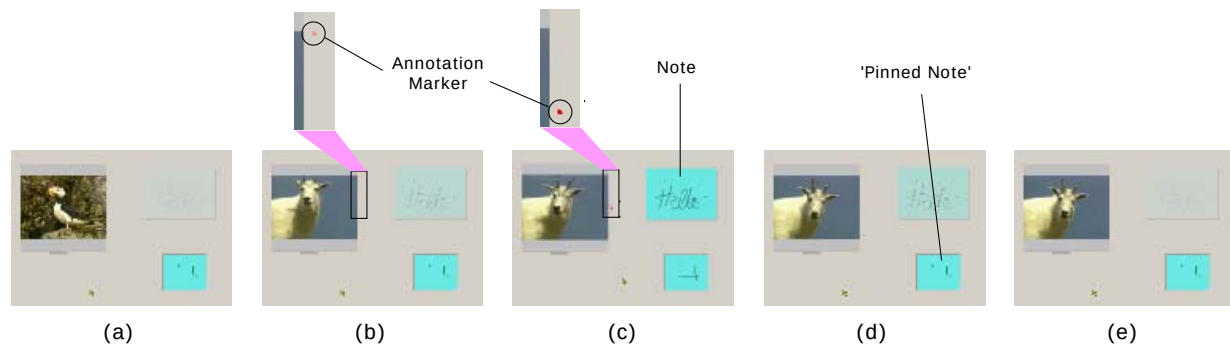


Figure 3-14: A sequence demonstrating the contextual visualization of an annotation. From a) - c) A note fades into the workspace, while an annotation marker -- zoomed in b) and c) -- provides further information. From c) - e) A note fades out of the workspace, while the annotation marker keeps providing information. a) through e) A pinned note remains visible at all times, regardless of the current frame being displayed. A video displaying this pressure widgets can be found at <http://youtube.com/watch?v=c-4tjto6Akq> – timecode 3:05.

This is similar to the techniques used in HyperVideo (Sawhney, Balcom, & Smith, 1996), where hypervideolinks or ‘opportunities’ fade in and out of a running video sequence. However, while the aforementioned work in HyperVideo separates creators and users, ours blurs the distinction between ‘readers’ and ‘writers’ of an annotated video stream. Other visual cues are provided in the form of animated markers on the side of the video frame being played. These markers have a size and position directly related to both the number of annotations and the moment a particular annotation was made. Users also have the ability to ‘pin’ a note into the workspace using a discrete pressure widget, making it visible at all times. Notes connected to a frame have an associated thumbnail that can be seen on all Video Segment containing the annotation’s temporal context (Figure 3-16). Notes made directly over a frame have an associated mark also seen on the relevant Video Segments. These thumbnails and marks can be used as visual landmarks or bookmarks that help users to navigate the video stream to reach defined points of interest. A note attached to Video Segments has the same behavior, except that its thumbnail is displayed on the right of the segment (Figure 3-8).

3.8.1 Link Navigation and Manipulation

Our system regards annotations as links between two data objects, links that can be traveled in any direction. If an annotation is visible, a user is able to find quickly the two objects participating in it. In general, and as was described in the case of Video Segments, selecting an object on the workspace reveals the object's direct relationships with other entities on the workspace (Figure 3-8).

For example, selecting a visible note reveals the links (annotations) in which the note participates. The user can then tap-and-hold the note to reveal a set of thumbnails that corresponds to the frames to which the note is connected. These thumbnails also function as a menu from which the user can select a frame (i.e., a point in time) to be visited (Figure 3-15). Users can also grab these thumbnails in order to unlink a note from a frame (deleting the link), or in order to move the link's endpoint to another note.

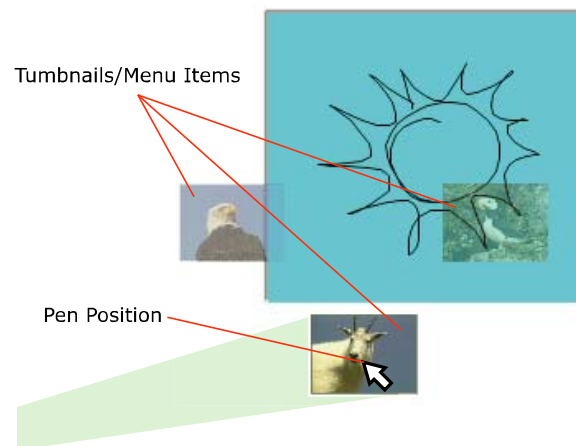


Figure 3-15: Frames connected to a Note are visualized as thumbnails that can be used as a menu to visit these annotated frames. The thumbnail under the pen is emphasized and an indication of relationship connects it to the point in the video stream where it can be found. A video displaying this pressure widgets can be found at <http://youtube.com/watch?v=c-4tjt06Akw> – timecode 3:05.

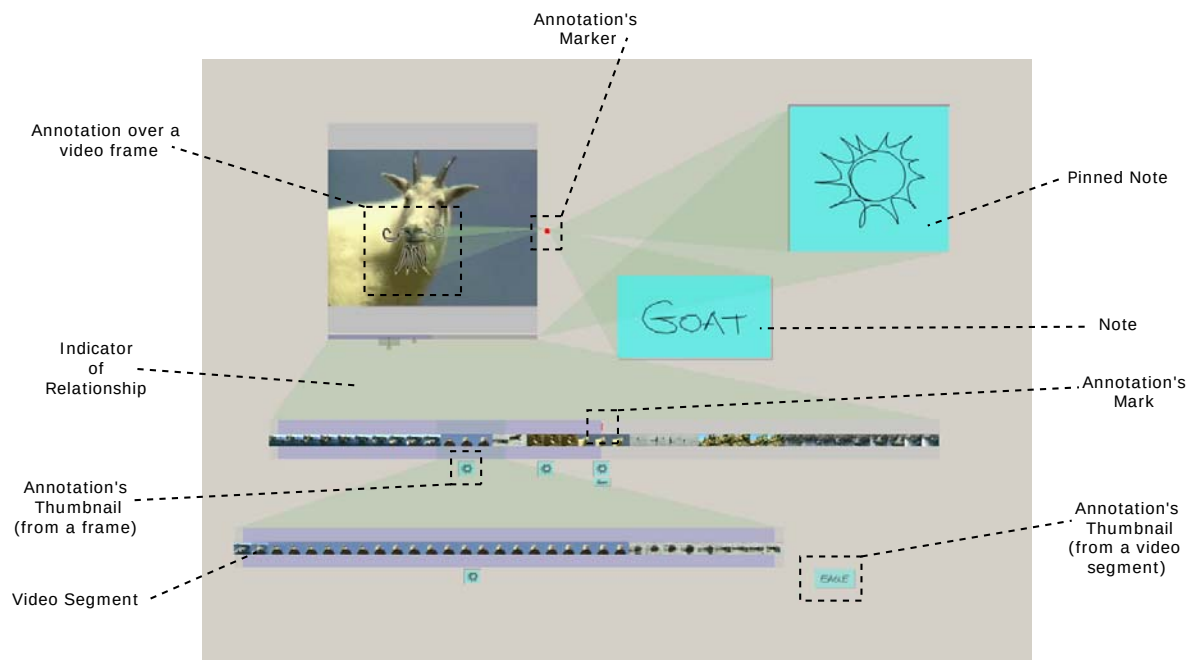


Figure 3-16: An example of a typical session with *LEAN*. The figure identifies the different elements on the screen.

3.9 Discussion and User Feedback

In developing *LEAN*, we strove to follow a simple set of design rules and interaction principles, including maintaining a minimalist interface without a surfeit of decorative elements, unobtrusive visualizations and interactions, and a small easily understood set of meaningful gestures. Through our design process, however, we found that tradeoffs between these principles needed to be considered. For example, there is the tension between the desire to have a minimalist interface and the nature of the available input / output devices. When there are no explicit widgets or controls available, an object should provide the affordances that suggest how it should be operated upon. In the physical world, people can use sight and touch to scan quickly for an object's affordances. However, with objects behind the glass of a computer screen this task is not so easily accomplished. This lack of direct access is one of the causes of the abundance and (mis)use of controls and decorations in many graphical user interfaces. The techniques demonstrated in *LEAN* have provided examples of how to achieve such

minimalist interfaces. Six users (three of the professional and faculty we initially interviewed and three graduate students at our research laboratory) informally tried LEAN on a desktop platform. After a 5-minute guided tour of the system, we asked them to explore the system freely and encouraged them to engage in tasks that involved navigating and annotating a video clip. Only some of these users had previous experience with pressure sensitive digitizer tablets, and all of them considered themselves novice or inexperienced users of video editing systems. Although not a formal study, observing these users provided us with the opportunity to gather valuable feedback that helped us to fine-tune the interaction techniques presented in this chapter. We can summarize our observations as follows:

- *Pressure Control:* When using the TLslider, people initially exhibited difficulty in controlling the amount of pressure they were applying with the pen. However, we also observed that after a few minutes of practice, they became aware of the consequences of varying levels of pressure and then developed better pressure control. Users also consistently reported that the pressure widgets provided useful feedback when they were using the pen.
- *Tap-And-Hold Gesture:* Users' responses to the tap-and-hold gesture were mixed. While some were comfortable with a delay of $\frac{3}{4}$ of a second, others found this waiting time excessive and referred a $\frac{1}{2}$ second delay instead. This last group made frequent use of the navigation controls and found it unacceptable to have to wait for their operation to be started. Regardless of their timing preferences, all users found the animated feedback provided while performing the tap-and-hold gesture useful.
- *Mode Errors:* It was common for users to try to use the PVslider directly, without first making a tap-and-hold gesture. This behavior revealed a mode error in which users scrib-

bled on top of the video frame instead of moving through its timeline. In a sense, this observation helps to demonstrate that the PVslider provides an intuitive and useful media control that users liked. On the other hand, our observations may indicate that users did not perceive the gesture as a whole, but rather as two separate phrases. Buxton's work on 'chunking and phrasing' (Buxton, 1986) suggests that it could be possible to abandon the tap-and-hold gesture in favor of one that leverages the user's kinesthetic tension (i.e., the pen's pressure) instead of time. By doing so, we can create a continuous 'statement' that combines the invocation and use of a control that itself incorporates both kinesthetic (pressure) and visual (rubber-band) tension (McGuffin et al., 2002).

- *Unforeseen Functionality:* After 15 minutes of use, all users easily became familiar with the features of the LEAN system, and even used it in ways that we had not previously anticipated. For example, one person started using the system as if it were a storyboarding authoring tool by making notes appear and disappear while a video was played. Furthermore, this user seemed more interested in the dynamic nature of the notes, than in the contents of the video. In general, users during their first session were able to create what can be best described as 'pop-up videos' with surprising ease.
- *Scaling:* At this point our system only handles videos in the order of a few minutes in length. It is not hard to imagine that the workspace in a system such as LEAN's may become over populated with annotations that were made over a long video stream. Because of this, it still remains to be studied how the visualization and interaction techniques we presented in this chapter scale in the presence of both a large number of annotations and Video Segments.

Our preliminary user observations provide us with encouraging evidence in favor of the interaction designs we produced. Still, in addition to having users in the field utilize the LEAN system in a holistic way, it would be of interest to researchers to have quantitative data gathered through formal studies in order to evaluate if the different interaction techniques contained in LEAN present a significant improvement over traditional methods of video navigation, control and annotation. These types of studies are intended future work, but fall outside of the scope of this dissertation.

3.10 Summary

This chapter describes how we used the *LEAN* interactive prototype to explore a number of design issues and challenges in the context of pen-based interactions. While performing this exploration, we purposely considered and utilized pressure as an additional input channel in our interaction designs. *LEAN* not only serves as a launch pad for presenting a set of novel interaction techniques for the navigation, segmentation, and annotation of digital video, but also shows how pressure-based interaction techniques have the potential to expand the vocabulary of interactions techniques available in pen-driven systems.

The richness and success of this vocabulary is dependent of both the effectiveness and usability of the interaction techniques that implement its phrases. Because of this, it is important that we understand what are the usability and design factors involved in the conception of user-interface elements that respond to pressure. In the next chapter we start investigating some of these issues.

4 Pressure Widgets

The *LEAN* system we presented in Chapter 3 explores the area of pen-based, pressure-aware interactions. In this chapter, we investigate in more detail human performance when using a pen's pressure as an additional input modality. To do so, we perform a quantitative study that investigates human ability to perform discrete selection tasks by controlling pen pressure and formally introduce and elaborate on the concept of *Pressure Widgets* as new, user-interface elements that exploit the capabilities of pressure-sensing technology⁵.

4.1 Introduction

Traditional user interfaces are designed mainly to be operated by pointing devices with two degrees-of-freedom that map to the x-y position of the cursor, and binary buttons that enable discrete selections. In the case of pen-based systems, there are additional degrees-of-freedom such as tilt, altitude, rotation and pressure. To date, this pressure input has typically only been used by a few drawing and image manipulation programs, like Adobe Photoshop, to modulate parameters of the active brush, such as stroke thickness or colour opacity. As tablet computers become more prevalent, it would seem advantageous to utilize this pressure sensing capability of pens throughout the user interface. In addition to the usual x-y positional cursor control and button clicks that the pen is currently used for, one can imagine using the pen's pressure to operate widgets that have several discrete states, or to control a continuous variable. This additional input modality could serve to increase the human-computer communication band-

⁵ Much of the material presented in this chapter was previously published in the Proceedings of the CHI '04 Conference on Human Factors in Computing Systems (Ramos, Boulos, & Balakrishnan, 2004).

PRESSURE WIDGETS

width, particularly when tablets are used as pure slates with no keyboard. To increase the use of pen pressure, appropriate widgets need to be designed. In turn, these designs will need to be guided by a thorough understanding of the user's ability to control pressure using a pen. Questions to be answered include, between how many discrete levels of pressure can a user easily discriminate? What is the impact of visual feedback and what form should it take? Can users with sufficient practice apply different levels of pressure without any visual feedback? What mechanisms can be used to indicate completion when pressure is used to acquire one of a discrete set of targets in a widget?

4.2 Experiment

The objective of this study is to investigate human ability to perform discrete selection tasks by controlling pen pressure. This includes determining the number of levels of pressure a user can comfortably discriminate between when using a pen, and the impact of visual feedback. We also compare four techniques for confirming selection after the target is located by applying the required pressure.

4.2.1 Apparatus

We used a Wacom Intuos 9x12 tablet with a wireless pen with a pressure sensitive isometric tip that does not provide any distinguishable haptic feedback. The pen provides 1024 levels of pressure⁶, and has a binary button on its barrel. The tablet's active area was mapped onto the display's visual area in absolute mode. The digitizer tablet was horizontally placed in front of the user, where normally a keyboard would be.

The experiment was done in full-screen mode, with a black background color, on a Dell UltraSharp 1800FP 18-inch flat panel LCD Monitor running at a native resolution of 1280 by

⁶ The digitizer tablet used in this experiment responds to a range of forces between 0 and 300 grams force (gf).

PRESSURE WIDGETS

1024 pixels. The experimental software ran on a 2GHz P4 PC with the Windows2000 operating system.

4.2.2 Task and Stimuli

We use a serial target acquisition and selection task. Pen pressure controls the movement of a small blue circle cursor along a vertical line. 1024 pressure values are mapped uniformly to a spatial distance of 256 pixels. A set of consecutive rectangles are drawn along the line's length. The size of the rectangles is experimentally manipulated. During each experimental trial, one of the targets is highlighted in green, and the user's task is to apply the appropriate amount of pressure to move the blue circle cursor into that target. When the cursor enters the target, the target color changes to red.

We use two different visual feedback conditions (Figure 4-1): *Full Visual (FV)* and *Partial Visual (PV)*. The *FV* condition shows the target in context with the other adjacent ones and provides continuous feedback in the form of the cursor's position along a vertical line. In the *PV* condition, only the target is visible, and the cursor is only shown at the start of the trial. Once movement begins, the cursor is hidden, and the user has to rely on proprioceptive cues and memory to determine accurately the amount of pressure to apply in order to get the hidden cursor into the target. This simulates the condition where expert users may be able to use pressure for quick selection in an eyes-free manner, similar to behavior exhibited by expert users of Marking Menus (Kurtenbach & Buxton, 1993). However, this condition is not completely free of visual feedback: as in the *FV* condition, the target color changes from green to red when the hidden cursor is inside the target. In other words, feedback is provided at the final stage of the task. A similar approach has been used successfully in previous experiments studying the limits of kinesthetic cues in interface tasks (Balakrishnan & Hinckley, 1999).

PRESSURE WIDGETS

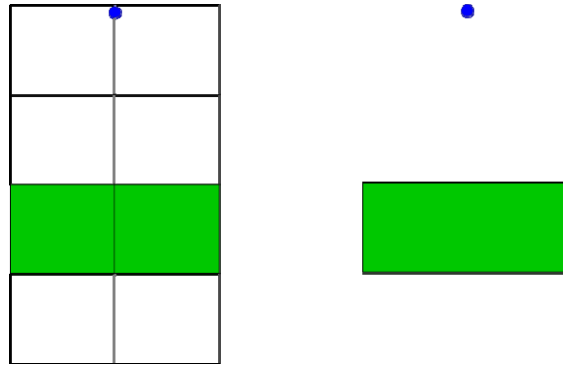


Figure 4-1: Visual feedback conditions. (Left) FV: green target is shown in context with other possible targets; the blue cursor is always visible. (Right) PV: green target is shown alone, the blue cursor disappears once pressure is applied.

Once the cursor is in the target, there has to be a mechanism for the user to confirm the selection. In standard GUI interfaces, this is typically done by clicking the mouse button. An analogous mechanism for the pen would be clicking the barrel button. In practice, however, the ergonomics of the pen makes this action less than ideal since users often rotate the pen and the button may not always be in a position to facilitate clicking. Further, clicking the barrel button can cause inadvertent movement in the x-y direction, which is problematic if we are to design interaction widgets where pressure is used in conjunction with x-y pen movement. In contrast, mouse buttons are orthogonal to x-y movement, reducing the possibility of inadvertent movement during button clicks.

We tested four selection techniques: *Click*: pressing the pen's barrel button; *Dwell*: maintaining the cursor within the target for a prescribed amount of time (in our experiment, a one-second delay was used); *Quick Release*: quickly lifting the pen from the tablet's surface; and *Stroke*: quickly making a spatial movement to the right. Each method has a particular signature in terms of pressure, spatial position, and button state, as illustrated in Figure 4-2.

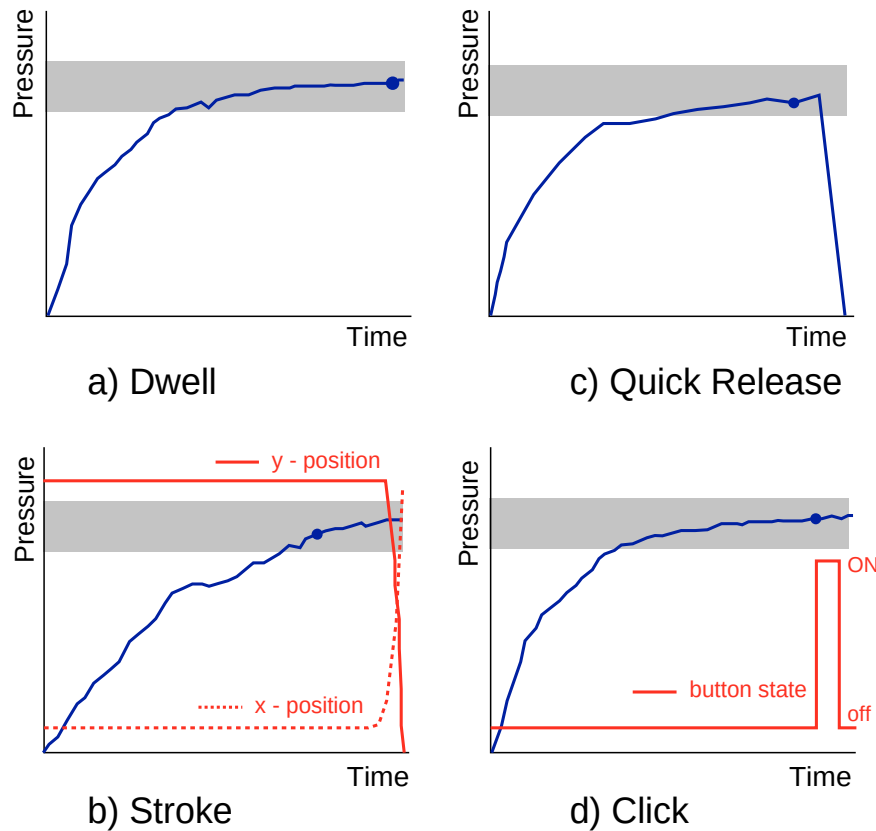


Figure 4-2: Signatures for the selection methods: Dwell, Stroke (in this case a tick mark with some vertical displacement), Quick Release and Click. The grey band shows target pressure range. The dot on each curve marks where the selected pressure value is taken.

4.3 Participants

Seven female and five male volunteers, 18-34 years old, participated in the experiment. All had normal or corrected-to-normal vision, were right-handed. Only three participants reported having some previous experience pressure sensitive devices such as the pen used in the experiment

4.4 Procedure and Design

We used a within-subjects full factorial design with repeated measures. The independent variables were selection method (Click, Dwell, Quick Release, and Stroke), visual feedback condition (*FV*, *PV*), the distance from the starting point to the target ($D = 205, 410, 615, 820$) and

PRESSURE WIDGETS

the target's width ($W = 85, 102, 128, 170, 256$). Distance and width are expressed in pressure units.

Recall that we used a fixed mapping of pressure values to cursor movement (1024 pressure values were linearly mapped to 256 pixels of cursor movement). As such, changing W changes the number of divisions of the 256 pixel potential target space. We used this approach rather than variable pressure to spatial mappings in order to resemble the designs we anticipate for pressure widgets where the overall widget size will likely remain constant with appropriate subdivisions into selectable targets, much like Marking Menus retains a uniform size regardless of number of menu items (Kurtenbach & Buxton, 1993). Furthermore, keeping the pressure to spatial movement mapping constant will likely facilitate user's ability to develop haptic memory of various pressure levels.

In order to keep the experiment balanced, special care was taken when choosing the W and D values, such that targets were appropriately distributed throughout the potential target space (Figure 3). This choice, however, resulted in D not always defining the distance from the start to the centre of the targets, but instead to *some* location in the targets. Because we are also interested to see if this pressure-controlled target acquisition task obeys Fitts' law (Fitts, 1954), we compute the index of difficulty (ID) for each condition using the amplitude A (i.e., the distance to the centre of the relevant target) rather than D (Figure 4-3).

PRESSURE WIDGETS

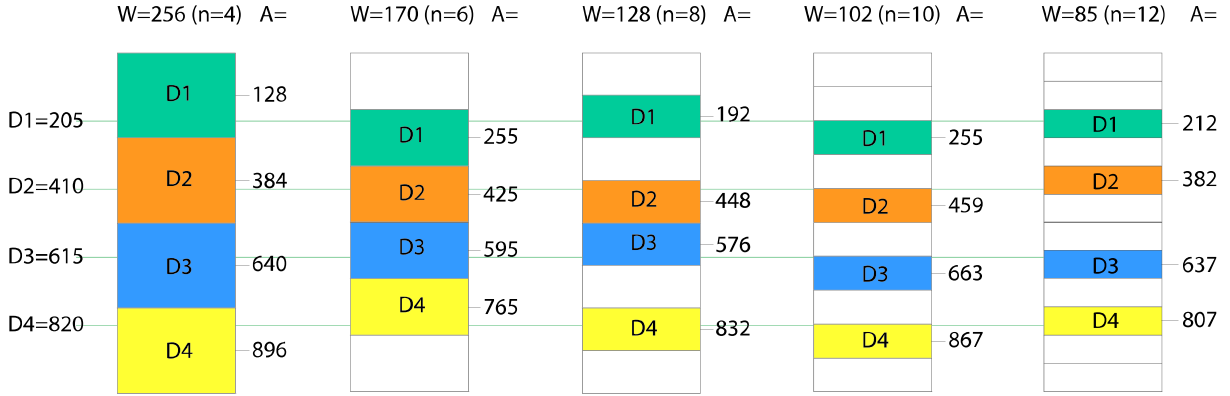


Figure 4-3: Subdivisions of pressure space used for the study. *W* dictates the number of pressure levels, *n*; *D* defines a target for a given *W*; *A* is the *amplitude* used to compute Fitts' *ID*. Distances D1-D4 sample the pressure space at four different points, which define four different targets across the different *W* levels.

Participants were randomly assigned to four groups of three participants each. In each group, participants were exposed to all four selection methods, whose order of appearance was balanced using a Latin square. For each selection method, we asked participants to complete two sessions of five blocks each. In the first session, the FV feedback was used and in the second session, the PV feedback was used. We used this presentation order to assess if users were capable of performing well under PV feedback, after they were exposed to the FV feedback level. Each block consisted of trials for all 20 D-W conditions, repeated three times, which resulted in 60 trials per block. Presentation of trials within a block was randomized. In summary, the experiment consisted of:

12 participants ×
 4 selection methods ×
 2 visual feedback conditions ×
 5 blocks ×
 20 D-W conditions ×
 3 repetitions = 28800 target selection trials.

Prior to performing trials for each selection method, participants were given a short warm-up set of trials to familiarize themselves with the selection technique. Participants were instructed to perform the task as quickly and accurately as possible. Participants could take

breaks between trials, and breaks were enforced between changes of visual feedback condition. The experiment lasted approximately 2 hours for each participant. A short questionnaire was administered at the end of the experiment to gather subjective opinions (Appendix A).

For each trial, we collected all the pen data events (position, pressure, and time). This allowed us to measure the time it took to perform a task, the result of the task (i.e. success or failure), as well as any extra information such as the number of times the cursor enters and leaves a target before the participant selects it, or changes in the pen’s spatial position. An audible beep provided error feedback if a selection was made outside the target. Timing began the moment the pen came into contact with the tablet’s surface (i.e. the tablet reported a pressure > 0) and ended when the appropriate selection technique was executed.

4.5 Performance Measures

The dependent variables were *movement time* MT – defined as the time from when the pen came into contract with the tablet’s surface until the appropriate selection technique was executed; *error rate* ER – defined as the percentage of trials for a particular condition that resulted in erroneous selections; and *number of crossings* NC – defined as the number of times the cursor enters or leaves a target for a particular trial, minus 1 (e.g., $NC = 2$ for a task where the user overshoots and reacquires the target). These measures complement each other: while MT and ER give us an indication of the overall success rate, NC tells us about the degree of pressure control that participants exerted.

4.6 Results

We removed outliers from the data set. A trial was considered an outlier if the time it took to complete the task was beyond two standard deviations from the mean task completion time. We discarded 1326 outliers, which represent 4.6% of the data collected. These 1326 outliers

PRESSURE WIDGETS

were distributed evenly across the full visual (2.34%) and partial visual (2.26%) conditions. Table 4-1 illustrates the distribution of outliers across the main experimental conditions. While in an ideal case, we would like the outliers to be uniformly distributed across conditions, our data does not reveal disproportionate unbalances that might bias our experimental results.

Table 4-1: Distribution of outlier samples

	Dwell	Stroke	Q. Release	Click	Total
Full Visual	0.6%	0.4%	0.8%	0.6%	2.3%
Partial Visual	0.5%	0.5%	0.7%	0.5%	2.3%
Total	1.1%	0.8%	1.5%	1.1%	4.6%

We performed a 4 (selection) $\times$ 5 (W) $\times$ 4 (A) $\times$ 5 (blocks) repeated measures analysis of variance (RM-ANOVA) on *MT*, *ER* and *NC* for each of the full visual (FV) and partial visual (PV) levels of the visual feedback condition. Unless stated otherwise, the values and differences we do not report are not statistically significant.

4.6.1 Selection Techniques

Our analysis showed a significant main effect for selection technique on *MT* for the *FV* ($F_{3,18}=35.52$, $p<0.01$) and *PV* ($F_{1,7}=79.46$, $p<0.01$)⁷ levels. For the *FV* level, pairwise means comparisons showed significant differences in *MT* between *Dwell* & *Q.Release* ($p<0.01$), *Dwell* & *Click* ($p<0.01$), and *Stroke* & *Q.Release* ($p<0.02$). For the *PV* level, pairwise means comparisons showed significant difference in *MT* between all pairs of selection techniques ($p<0.01$), with the exception of *Q.Release* and *Click* ($p=1$). Overall, the fastest selection tech-

⁷ Mauchly's test indicates that the assumption of sphericity is not met; we corrected the degrees of freedom using lower bound estimates.

PRESSURE WIDGETS

nique was *Quick Release*, followed in order by *Click*, *Stroke*, and *Dwell*. Figure 4-5 and Figure 4-8 illustrate these effects.

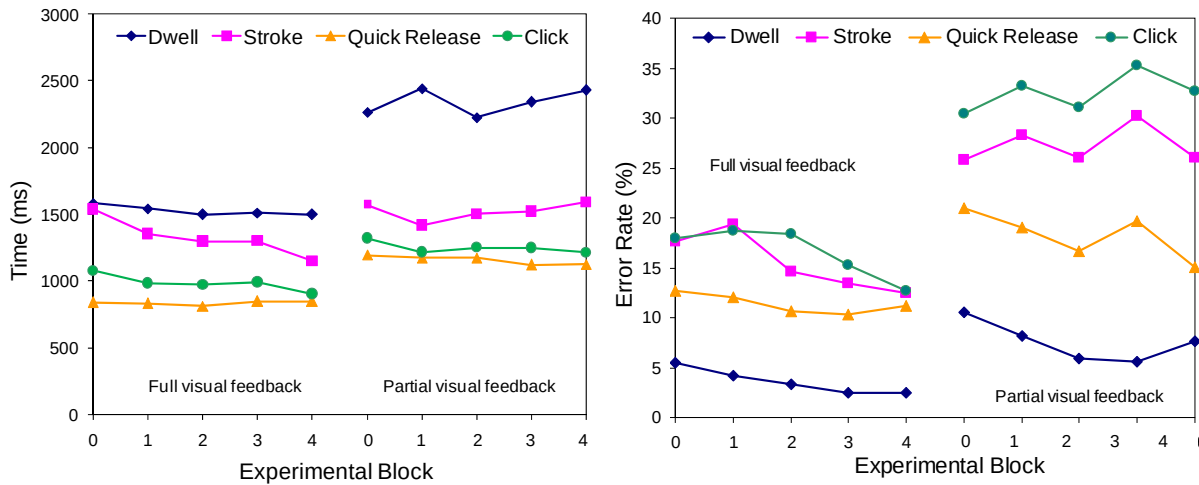


Figure 4-4: Effect of selection on Mean Time (left), and Error Rate (right).

Error rate *ER* was also significantly different across selection methods for both *FV* ($F_{3,18}=35.52$, $p<0.01$) and *PV* ($F_{3,21}=10.85$, $p<0.01$) levels. Pairwise means comparisons showed no significant differences between selection techniques ($p>0.1$). At the *PV* level, pairwise comparisons revealed significant differences only between *Click* & *Dwell* ($p<0.05$) and *Click* & *Q.Release* ($p<0.05$). Overall *Dwell* had the lowest error rate followed in order by *Q.Release*, *Stroke*, and *Click*. Figure 4-4 illustrates these effects.

As a measure of pressure control, the number of target crossings *NC* was also significantly different across selection methods for both *FV* ($F_{1,6}=22.75$, $p<0.01$)⁸ and *PV* ($F_{3,21}=31.73$, $p<0.01$) levels. Pairwise means comparisons at the *FV* level showed significant difference between *Stroke* and *Click* only ($p<0.05$). At the *PV* level, pairwise comparisons reveal significant differences between all pairs with the exception of *Dwell* and *Stroke* ($p=1$) and *Q.Release* and *Click* ($p=1$). People attained different levels of control depending on visual

⁸ Mauchly's test indicates that the assumption of sphericity is not met; therefore we corrected the degrees of freedom using lower bound estimates.

PRESSURE WIDGETS

feedback presented. For the *FV* condition, *Stroke* was the most difficult technique to control, followed by *Q.Release*, *Click*, and *Dwell*. For the *PV* condition, *Dwell* was the most difficulty, followed by *Stroke*, *Click*, and *Quick Release*. The overall higher *NC* rate for the *PV* condition is consistent with our observations that without visual feedback, users had to resort to “fishing around” in order to acquire the desired target. Figure 4-5 illustrates these effects.

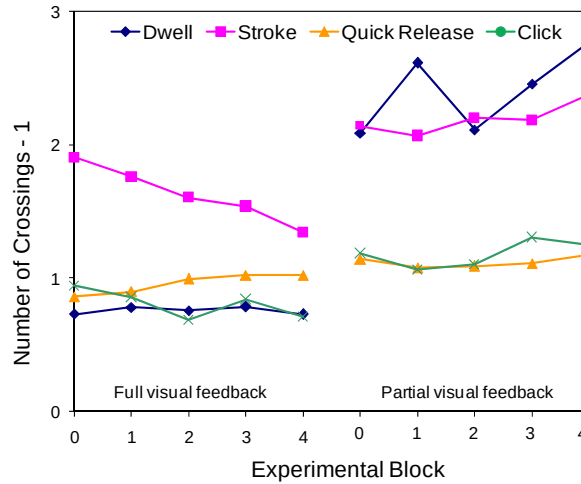


Figure 4-5: Effect of selection on *Number of Crossings* throughout experimental blocks. The left and right half of each graph shows data for the *Full Visual* and *Partial Visual* condition, respectively.

For the *FV* condition, we observed significant differences across blocks for *MT* ($F_{4,24} = 8.69, p < 0.01$). All selection methods consistently improved over time except for *Quick Release*. Differences in *ER* were not significantly different ($F_{4,24} = 2.29, p < 0.09$). A similar lack of significant differences was observed across blocks for *NC* ($F_{1,6} = 2.19, p = 0.19$)⁹. Still, we observed a marked decrease on *NC* for *Stroke*.

For the *PV* feedback condition, *MT* slightly improved with practice for the *Quick Release* and *Click* techniques, however they did not approach the times seen in the *FV* condition. Time performance for *Dwell* and *Stroke* was erratic, and we could not clearly identify a trend. Some improvement in error rate was seen for the *Dwell* and *Quick Release* techniques, but not for

⁹ Mauchly's test indicates that the assumption of sphericity is not met; therefore we corrected the degrees of freedom using lower bound estimates.

PRESSURE WIDGETS

Click or *Stroke*. Erratic *NC* values were observed for *Dwell*, while the three techniques had high *NC* values that were fairly constant throughout.

A significant selection $\times$ block interaction for *MT* ($F_{12,72}=3.07, p<0.01$) and *NC* ($F_{12,72}=3.34, p<0.01$) in the *FV* condition suggest that participants progressed at varying rates for the different selection techniques.

In the post experiment questionnaire, participants were asked to rate on a 7-point “agree – strongly disagree” Likert scale if they believed they made many mistakes. The average result was 5.1, falling in the “somewhat agree” slot. Participants also ranked the selection techniques according to how easy they were to use. *Quick Release* ranked highest, followed by *Dwell*, *Click*, and *Stroke*. This resembles the ranking for error rate, which may suggest that participants associate successful selections with ease of use.

4.6.2 Discernable Number of Pressure Levels

One of the main purposes of this study was to determine how many discrete levels of pressure, *nLevels*, users could discriminate between at a decent level of performance. For each selection method (except *Stroke*) error rates were not significantly different when *nLevels* ≤ 6 for the *FV* condition (Figure 4-6). In addition, *NC* values for all techniques reach a plateau (i.e. for a particular selection technique, values of *NC* do not differ significantly) at *nLevels* ≥ 8 , except for *Stroke* which continues to increase (Figure 4-7).

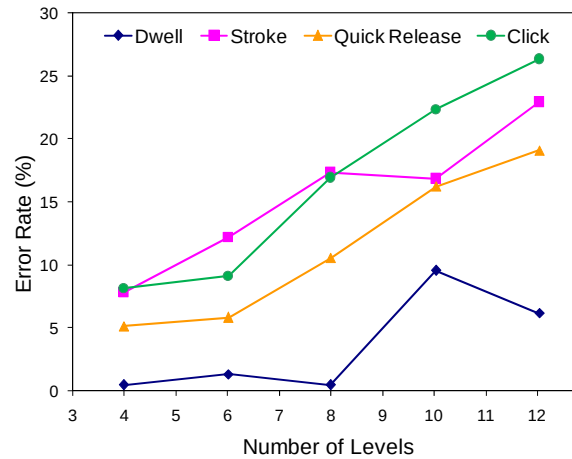


Figure 4-6: Effect of the number of pressure levels ($nLevels$) on Error Rate for the Full Visual condition.

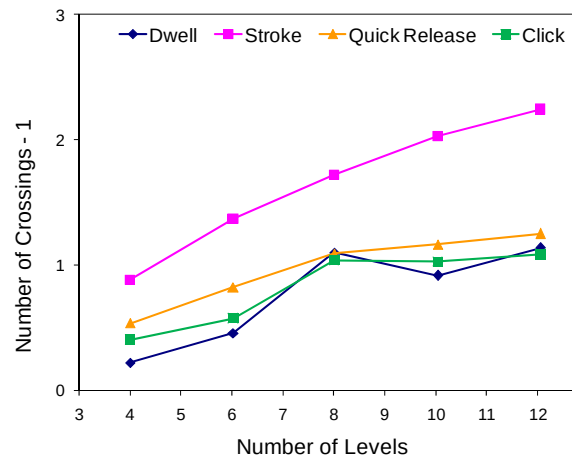


Figure 4-7: Effect of the number of (pressure) levels ($nLevels$) on Number of Crossings for the Full Visual condition.

Overall, best control is seen for $nLevels \leq 6$, where the average $NC < 1.8$ (i.e., the actual number of overshoots is less than 1) for all but the *Stroke* technique. These results point at a “sweet spot” around 6 levels of pressure that represent how many pressure levels a person can at least reliably navigate through if provided with adequate visual feedback using one of the four selection techniques we consider. For a number of levels below this “sweet spot”, user can target discrete pressure zones, quickly, accurately and with good control.

PRESSURE WIDGETS

For the *PV* condition, both *ER* and *NC* have increasingly poor levels for *nLevels* > 4. At *nLevels* = 4, *NC* values remain below 1.78 for all selection techniques, and *ER* values remain close to 5%¹⁰ for *Dwell* and *Click*; and 15% for *Quick Release* and *Stroke*.

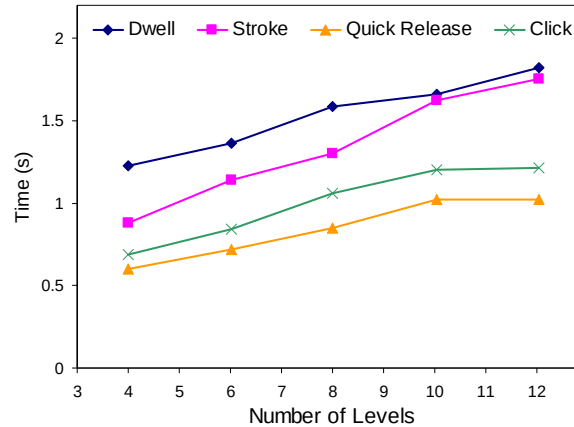


Figure 4-8: Effect of the number of (pressure) levels (*nLevels*) on Mean Time for the Full Visual condition.

As our data shows, the point where user performance degrades beyond the point of usability depends on the particular selection technique and the type of visual feedback available.

4.6.3 Effect of Visual Feedback

We included a partial visual feedback (*PV*) condition in our experiment to simulate the situation where expert users would perform pressure-based selection without looking at the visual feedback, relying instead on their haptic memory of the amount of pressure to apply. In particular, all participants performed the experiment with full visual feedback (*FV*) first, in order to gain expertise with the techniques, before attempting expert behavior. We performed a 4 (selection technique) $\times$ 2 (visual feedback) RM-ANOVA on *MT*, *ER* and *NC* and our results indicate that performance in the *PV* conditions are significantly lower than in the *FV* condi-

¹⁰ The choice of an acceptable error rate is a matter of choice that takes into consideration the task at hand. In other words, the cost of error recovery of an interaction depends on the specific action being performed, e.g., an error in the context of a nuclear reactor facility is more severe than one occurring during selecting a font's weight in a word processor. Fitts assumed in his seminal paper (Fitts, 1954) an error rate of 4% (equated to the level of reasonable noise when movement amplitudes are interpreted with a signal's amplitude) for pointing tasks, a threshold that many (ourselves included) chose to adhere to.

tions for MT ($F_{1,11}=110.17$ $p<0.01$), ER ($F_{1,11}=14.62$, $p<0.01$), and NC ($F_{1,11}=12.73$, $p<0.01$).

Given that our experiment lasted approximately 1 hour per participant for the FV condition, it is likely that this was not sufficient time to develop expert behaviour. However, we cannot rule out the possibility that there will always be a fundamental difference between FV and PV performance, regardless of the amount of practice.

4.6.4 Conformity with Fitts' Law

Most target-acquisition tasks tend to follow Fitts' law, where MT is modeled by the following relationship:

$$MT = a + b \underbrace{\log_2 \left(\frac{D}{W} + 1 \right)}_{ID}$$

The logarithmic term is called the *index of difficulty (ID)* for the target acquisition task. Linear regression of our MT data by ID for each selection technique (Figure 4-9) indicated a good fit with Fitts' law for *Quick Release* ($r^2=0.9$) and *Dwell* ($r^2=0.84$), and poorer fits for *Click* ($r^2=0.74$) and *Stroke* ($r^2=0.44$). For the *Stroke* technique, this is perhaps unsurprising since it involves two distinctly separate actions: pressure followed by a spatial stroke gesture. Our data logs did not identify the moment separating those two stages, thus the time data for that selection technique contains a significant amount of noise. The *Click* and *Stroke* technique involved actions that likely affected a user's ability to maintain a particular pressure level, resulting in the high error rates observed and thus more variance in the regression.

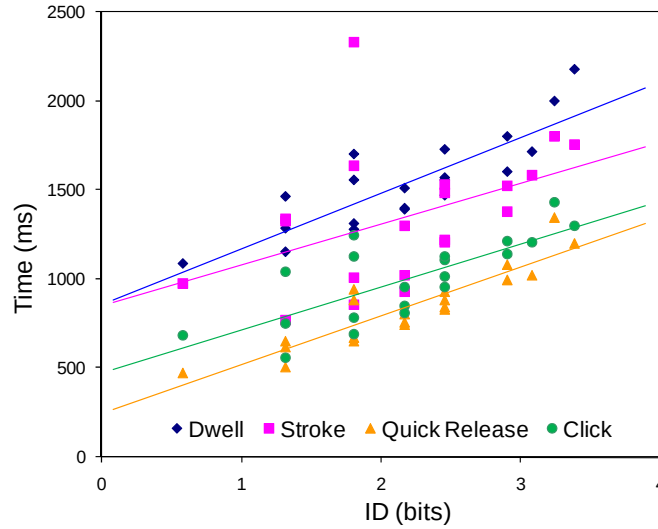


Figure 4-9: Linear regression of Index of Difficulty (ID) versus *Mean Time* data by selection technique.

4.6.5 Control at Different Pressure Levels

Our observations during the experiment indicated that participants' abilities to control pressure varied according to the amount of pressure required. Participants also reported that "the pen was too sensitive" when they tried to acquire a target at a low-pressure value. Analysis of variance confirms these observations and shows that *NC* values are significantly different across amplitudes for both the *FV* ($F_{1,6} = 34.36, p < 0.01$)¹¹ and *PV* ($F_{3,21} = 107.73, p < 0.01$) conditions. There was also a significant selection method x amplitude interaction for the *FV* ($F_{9,54} = 13.19, p < 0.01$) and the *PV* ($F_{9,63} = 10.42, p < 0.01$) levels of the visual feedback condition. Figure 4-10 illustrates these effects.

¹¹ Mauchly's test indicates that the assumption of sphericity is not met; therefore, we corrected the degrees of freedom using lower bound estimates.

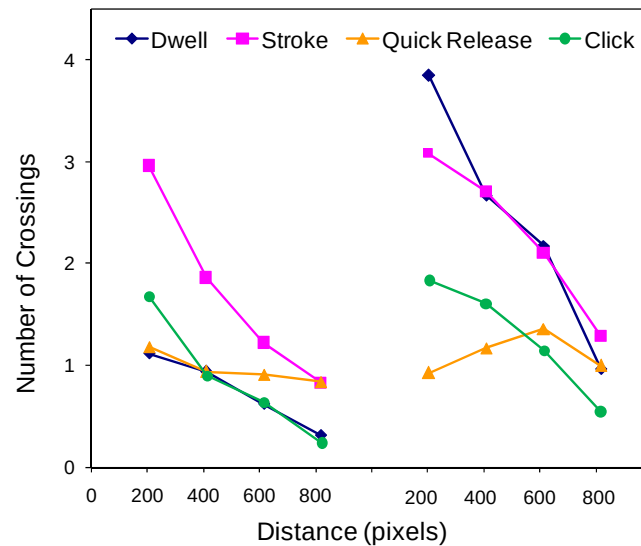


Figure 4-10: Effect of distance (amplitude) on pressure level on *Number of Crossings*: the left half of the graph shows the *Full Visual* condition and the right shows the *Partial Visual* condition.

4.7 Summary

Our results have shown that the different selection techniques have significant effects on the usability of pen pressure for discrete pressure-space navigation tasks. It is important to note that all the techniques have a common first phase: applying the right amount of pressure to move the cursor into the target. It is the second, selection, phase that differs between techniques. Some of the techniques (e.g., *Dwell*) have a smoother transition relationship between the two phases, while others required a distinctly separate action that could have interfered with performance.

The *Dwell* technique was the most accurate and allowed for the highest degree of pressure control (low *NC*), at least in the *FV* condition. This is perhaps unsurprising since the second phase in this technique involves simply waiting for the appropriate time delay to pass without any other movements required. However, this incurs a built in 1-second penalty, resulting in a tradeoff between accuracy, control and time.

PRESSURE WIDGETS

Participants consistently rated *Stroke* as being quite difficult. Completion time using *Stroke* depended greatly on people's skill in performing the stroke gesture. Participants learned this gesture at different rates, and performed the selection at different speeds. While some participants performed a quick flick motion, others were very careful and did a slow and controlled motion. Our implementation recognized any significant movement that went to the right as being a valid gesture, but some participants commented that it was more natural for them to perform the gesture in an upward diagonal motion instead of a horizontal one. Difficulties in maintaining a stable pressure value while moving the pen also contributed to the poor performance of this technique. Our algorithm attempts to compensate for this by estimating the point at which the stroke gesture begins and ignoring pressure fluctuations thereafter. However, this estimation process is not always successful.

Quick Release was subjectively the highest rated, and quantitatively the quickest technique. It was also the fastest in the absence of full visual feedback. This is because the second phase of this technique is very fast, and thus does not prolong the need to maintain a particular pressure level after the first phase. Overall, *Quick Release* appears as the most attractive selection technique compared to the ones studied because it allows for selections with the low error, good control (i.e., low *NC*) and fastest execution.

As might be expected from our earlier discussions about the ergonomics of the pen, we found that the button presses in the *Click* technique interfered significantly with pressure control. Unless the pen's design can be changed significantly, our results indicate that this is not a good technique for pressure based target selection.

4.8 Implications for Design

The results of our experiment suggest several guidelines for the design of pressure sensitive widgets:

- *Have the “right” number of pressure levels.* Our results show that one can partition the pressure range into at least six levels, which users can navigate through successfully. Our choice of six as a minimum number is a conservative one and it stems from the integration of our experimental results in terms of interaction accuracy, speed and control. The interactions that used Dwell as a selection mechanism were the most accurate, yet this accuracy came at the price of being the slowest of all interactions. These results made us look into the second most accurate interactions, i.e., the ones performed with the quick release selection mechanism. For this type of interaction, performance starts to deteriorate after tasks where the pressure space has more than 6¹² regions or zones (Figure 4-6,7). The digitizer device that we used in our experiments, a Wacom Intuos 9x12 digitizer tablet, discriminates up to 1024 levels of pressure. Six levels of pressure is several orders of magnitude lower than 1024, so we argue that the sampling of the pressure space provided by the device was adequate for our study. Because of this, we also argue that our results could generalize to other digitizers working within similar force ranges¹³ than the Wacom device.
- *Provide real-time, continuous feedback.* Even though pressure activated target acquisition tasks are achievable without continuous feedback (*PV* condition), pressure control was consistently poor, if not erratic, under this condition for almost all selection methods.

¹² It is possible that the number of pressure levels a person can navigate accurately through increases above this threshold and that error rates decrease if an appropriate transfer function is used. Likewise, experiments using psychophysical methods may yield different results and remain a valid, yet not known, experiment to perform.

¹³ We found the force range for this particular device to be between 0 and 250gf.

PRESSURE WIDGETS

While this could improve with lots of practice, it is clear that good feedback is required for pressure sensitive widgets.

- *Minimize the impact that movement has on the selection phases.* Movement and selection with an isotonic pointing device (e.g. mouse) occur on uncorrelated (orthogonal) input channels. With an isometric pen, however, appropriate techniques need to be utilized to minimize inherent interferences between movement and the application of pressure. Our *Dwell* and *Quick Release* techniques are good candidates in this regard.
- *Minimize the impact spatial x-y movement has on pressure control.* A poor visual design may inadvertently make users move the pen while trying to perform a pressure control task. We observed this phenomenon with our experimental setup. Even though participants were instructed that only pressure had an effect on the target acquisition task, many moved the pen spatially in the vertical direction, corresponding to the movement direction of the blue cursor (Figure 4-1). It was interesting to observe that while some users did this inadvertently, others used this motion as an *explicit* way to control pen pressure. We will elaborate later on widget designs that aim to minimize this source of interference.
- *Choose a good transfer function.* Participants consistently demonstrated less pressure control for low levels of pressure, and described the widget as “very sensitive” at these levels. The simple linear transfer function used in our experiment could be improved to take into account this variation of control at different pressure levels.

4.9 Factors for the Design of Pressure Widgets

Pressure widgets can be *continuous*, if they control a continuous value, e.g., the speed of a video, or the opacity of an object; or *discrete*, if they are used to select an element or value

PRESSURE WIDGETS

contained in a finite, small set, e.g. a color from a palette, or the typeface of a font. In this chapter, we have studied *discrete* pressure widgets, and for the remainder of this chapter (unless stated otherwise) will simply refer them as pressure widgets. We think it is useful to describe the different factors that characterize pressure widgets in general and then discuss how these factors can be used to guide us in the visual design of specific pressure widgets.

4.9.1 Interference

In designing pressure widgets, we already identify the need to minimize the impact between x-y movement and pressure control. We call this tension between input channels *interference*. Interference is an undesirable feature, since we observed that it could be difficult to both simultaneously move a pen and effectively control the pressure applied with it. If a person's objective is to control pressure, a sudden spatial movement may be disruptive. Conversely, a carefully planned motion may be disrupted by a sudden, even intentional, change in pressure. These behaviors can be traced back to the open-loop nature of pointing or handwriting tasks, as reported by Woodworth and cited by Elliott et al. (Elliott, Helsen, & Chua, 2001). In the context of pressure widget design, three factors could potentially affect the amount of interference:

- *The widget's visual appearance and feedback:* An improper widget visual design may lead to erratic pressure control, e.g. a person may not feel compelled to vary the pressure applied with the pen, or a person may feel compelled to move the pen, both of which cause interference.
- *The widget's selection method:* A widget must provide a way for the user to select the desired target. For example, the selection method may rely on the x-y variation of the same pen with which a user applies pressure for navigation. Such a selection method may dis-

rupt the navigation task, causing the pressure applied to vary inadvertently before or during selection. Also, in this case, it is difficult for a system to identify the moment at which the navigation phase ends and the selection phase begins; i.e., the moment at which a person makes the decision to select, but prior to that decision being translated into a change in the pen's parameters. For this particular study, we explicitly remove the selection phase from the tasks, in order to focus our study on how a pressure widget's visual feedback affects the navigation phase.

- *Tracking* is factor that we will take into consideration in the design of pressure widgets. This factor can assume two values, active and inactive, and is of particular interest because of its potential impact on interference. A widget with tracking active will tightly follow the pen's x-y position. An example of such behavior can be found in Tracking Menus (Fitzmaurice, Khan, Pieké, Buxton, & Kurtenbach, 2003). Conversely, a widget with tracking inactive will maintain its location independent of the pen's x-y position. This behavior can be found in context menus that are usually invoked with a right-click command in many applications.

4.9.2 Visuals Elements and Behavior

Users' interactions with pressure widgets can be divided into two stages: *navigation*, where users apply the right amount of pressure so that the cursor indicates the desired target; and *selection*, where users effectively confirm picking the target/value identified in the navigation stage, and thus complete the interaction. During these two stages of interaction, we differentiate between three distinctive visual elements that can be present in pressure widgets: *cursor(s)*, *target(s)*, and *glyph(s)*. We informally define *cursor* as the visual feature that indicates what item will be chosen if a selection occurs. We will refer to *targets* as the visual represen-

PRESSURE WIDGETS

tation of the set of items available for selection. Finally, a glyph is symbol or abstract representation of a particular quantity or information, e.g., a number indicating the magnitude of a particular entity. Figure 4-11 illustrates an example of these visual elements in the context of a traditional linear menu.

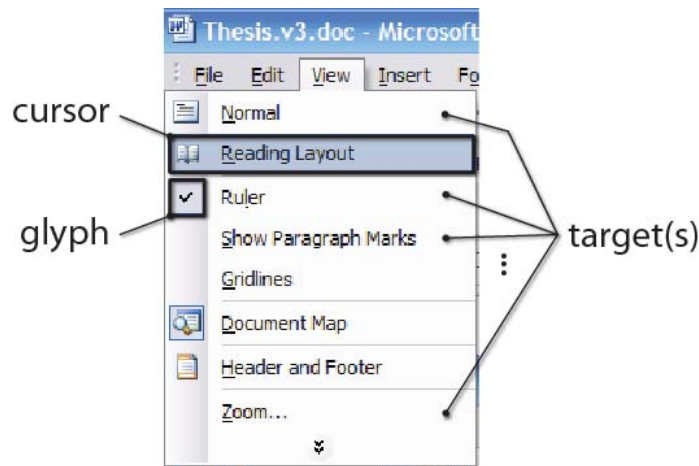


Figure 4-11: Example of visual elements on a traditional linear menu. The available selectable *targets* are laid out in a linear fashion. A rectangular *cursor* moves up or down and highlights the item to be selected. A checkmark is a type of *glyph* that changes its appearance depending on the state of an internal variable.

Our experiment shows that users perform better if they are provided with a visual indication of the amount of pressure they are applying. In order to do this we connect the pressure value reported by the pen to control one or more properties of the widget's visual elements – e.g., *cursor*, *targets* or *glyphs*. We call this relationship *coupling*. When the pressure affects the widget's targets we call it *target coupling*, and when it affects the widget's cursor we call it *cursor coupling*. When pressure variations were tied to glyph variations like in Mizobuchi's design (Mizobuchi et al., 2005) we are under the presence of *glyph coupling*.

For either of these couplings, we consider three particular attributes of a widget's visual element that can be affected by changes in pressure: *position* if the variations in pressure produces changes to an element's x-y coordinates; *scale* if the variations in pressure produce

changes in an element's size or scale¹⁴; *angle* if the variations in pressure produces changes in an element's angle or orientation; or *self* if variations in pressure produce changes in an element's appearance beyond what a deformation can achieve – e.g., numbers changing. We will refer to this effect that pressure has on a (visual) element as *behavior*. For example, using the above terminology, we can say for a particular widget that its targets have a position behavior.

4.10 A Study on Visual Features and Interference

We performed a controlled experiment that studied users' performances with a number of discrete pressure widget designs corresponding to three different *behaviours*: the *Flag* (cursor with position behaviour), the *Rotating Pie* (targets with angle behaviour) and the *Bullseye* (cursor with scale behaviour) (Figure 4-12a-c).

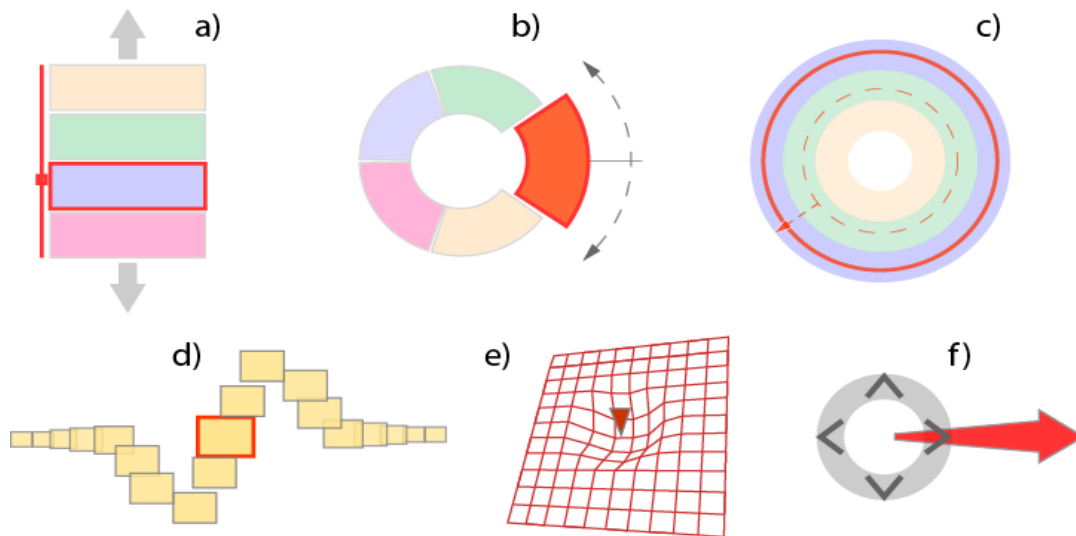


Figure 4-12: Concept designs for different pressure widgets. (a) Flag (discrete). (b) Rotating Expanding Pie (discrete). (c) Bullseye (discrete). (d) Twist-Lens Slider (continuous). (e) Pressure Grid (continuous). (f) Pressure Marking Menu (discrete).

When possible for each widget, we also considered two pressure *couplings* (i.e. cursor and targets) and a *tracking* condition that dictates when the whole widget follows the pen's x-y position and when it does not. Our study required users to perform a serial target acquisition

¹⁴Changes in scale could be non-linear in nature, such as a fish-eye or hyperbolic deformation.

PRESSURE WIDGETS

and selection task. In this task, a pressure-sensitive pen was used to control the behaviour of five different types of pressure widgets: *Flag*, *Moving Flag* (targets with position behaviour), *Bullseye*, *Pie* (cursor with angle behaviour) and *Rotating Pie* (Figure 4-13). The experiment also investigated the effects of *tracking* as applied to each of these widget designs. In each experimental trial, participants confirmed the selection of the target by pressing a key with their non-dominant hand. This choice of selection mechanism isolated any interference that the selection event may introduce if it were originating from the pen itself.

PRESSURE WIDGETS

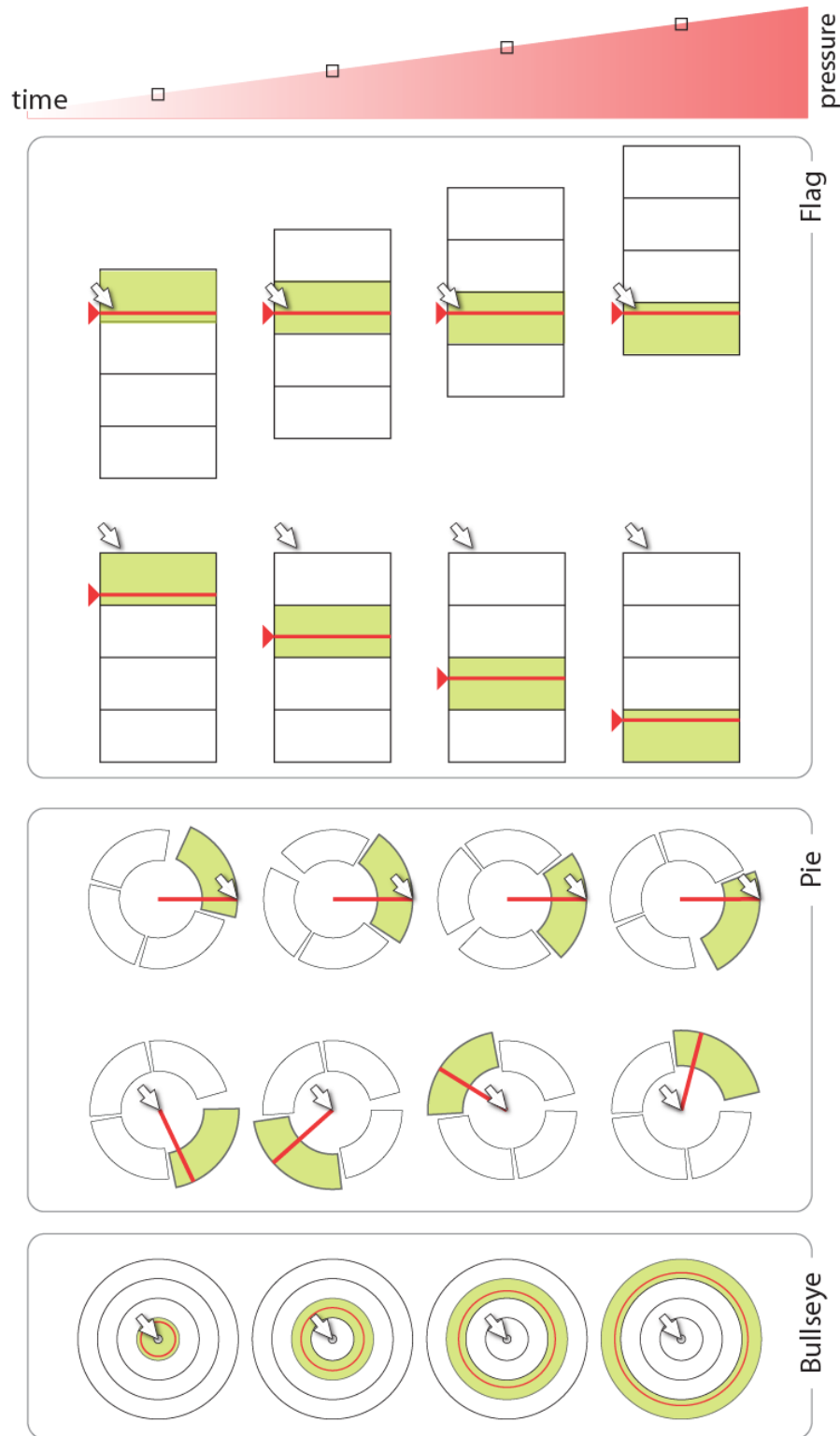


Figure 4-13: The diagram illustrates the behavior of different discrete pressure widget designs as pressure applied by a user increases. (First row) Moving Flag: targets coupled to position. (Second row) Flag: cursor coupled to position. (Third row) Rotating Pie: targets coupled to angle. (Fourth row) Pie: cursor coupled to angle. (Fifth row) BullsEye: cursor coupled to scale. Notice the large gap between the first and last target. The white arrow indicates the position of the pen. A video displaying these designs can be found at <http://www.youtube.com/watch?v=n3Ybz8KiB68>

4.10.1 Apparatus

We used a Wacom Cintiq 18-SX interactive LCD graphics display tablet with a wireless stylus that has a pressure-sensitive isometric tip that reported values between 0 and 1024¹⁵. The experiment software ran on a 2GHz P4 PC with the Windows XP Professional operating system. The tablet was laid out flat over a desk in front of users as if it were a slate resting over a table.

4.10.2 Participants

Seven female and eight male volunteers, 18-34 years old, participated in the experiment. All were right-handed, had normal or corrected-to-normal vision and had little to no prior experience using pressure-sensitive devices such as the stylus used in the experiment.

4.10.3 Procedure and Design

Our experimental design and stimulus was similar to the one described in section 4.2. The differences between studies were: b) the addition of a *widget* independent variable (*F*, *MF*, *B*, *P* and *RP*), b) all widgets presented full visual feedback, c) the addition of a tracking *tracking* independent variable (*t_on*, *t_off*), and d) the use of a reduced number of pressure zones, i.e., *breadth* (4, 6, 8).

Presentation of the widget condition was balanced across users using using a Latin square. For each widget condition, we asked participants to complete two sessions of eight blocks each. Tracking was active (*t_on*) during the first session and inactive (*t_off*) during the second one. Each block consisted of three selection trials of for all three *breadth* conditions, repeated four times for a total of 36 trials per block. Presentation of trials within a block was randomized. In summary, the experiment consisted of:

¹⁵ The digitizer tablet used in this experiment responds to a range of forces between 0 and 300gf.

PRESSURE WIDGETS

15 participants ×
5 widget conditions ×
2 tracking conditions ×
8 blocks ×
3 selection tasks (Amplitude) ×
3 breadth conditions (Width) ×
4 repetitions = 43,200 target selection trials.

Prior to performing trials for each widget condition, we administered participants with a short warm-up set of trials to familiarize them with the widget. We then instruct participants to perform the upcoming tasks as quickly and accurately as possible. Participants were allowed to take breaks between trials, and were instructed to rest between different widgets. The experiment lasted an average of 2 hours per participant.

4.10.4 Performance Measures

The dependent variables were *navigation time* NT (i.e., the time from when the stylus came into contact with the tablet's surface until the user selected a target); *distance traveled* by the stylus D; *number of crossings* NC (defined as the number of times the cursor enters or leaves a target for a particular trial); and *error rate* ER (defined as the percentage of trials for a particular condition that resulted in an erroneous selection).

4.10.5 Results

A trial was considered an outlier if the time it took to complete the task was beyond 2 standard deviations from the mean NT. 1630 outliers were discarded, representing 3.7% of the data collected. Table 4-2 illustrates the distribution of outliers across the main experimental conditions. While in an ideal case, we would like the outliers to be uniformly distributed

PRESSURE WIDGETS

across conditions; our data does not reveal disproportionate unbalances that might bias our experimental results.

Table 4-2: Distribution of outlier samples

	Moving Flag	Flag	Bullseye	Rotating Pie	Pie	Total
Tracking Disabled	0.5%	0.5%	0.3%	0.6%	0.4%	2.2%
Tracking Enabled	0.4%	0.3%	0.2%	0.4%	0.3%	1.6%
Total	0.9%	0.7%	0.5%	1.0%	0.7%	3.7%

We performed a 2 (tracking state) $\times$ 5 (widget) repeated measures analysis of variance (RM-ANOVA) on *NT*, *D* and *NC*. Unless stated otherwise, the values and differences we do not report are not statistically significant.

4.10.5.1 Distance

Because in the case of the pressure widgets that we study users do not need to move the pen in order to perform the assigned task, we consider distance (*D*) as a fair estimator of the interference produced by a visual design. There was a significant effect of widget on distance ($F_{4,56}=4.05$, $p<0.01$) with the Moving Flag eliciting the most movement and Flag least pen displacement. This particular result indicates that coupling a widget's cursor with pressure can decrease interference. Figure 4-14 illustrates this.

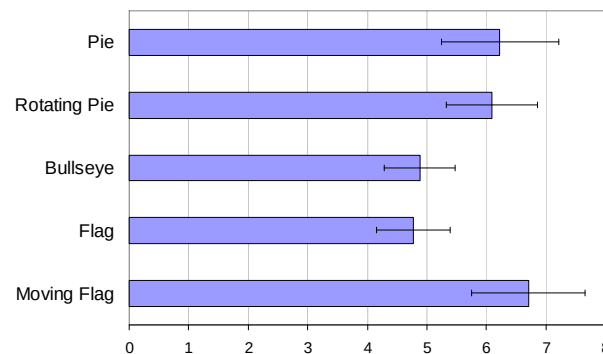


Figure 4-14: Average distance (in pixels, +/- SE) traveled by the pen's tip during a selection task per widget design.

PRESSURE WIDGETS

4.10.5.2 Error

There was a significant effect of widget on Error ¹⁶ ($F_{4,14}=3.44$, $p<0.02$) with the Flag design causing the highest error rate (~5.6%) and Pie being the design with lowest error rates (~3%). One hypothesis for this behavior is that unlike the Moving Flag, the Pie design remains within the locus of the user's attention, i.e., there is an element of spatial stability, which might benefit the overall interaction.

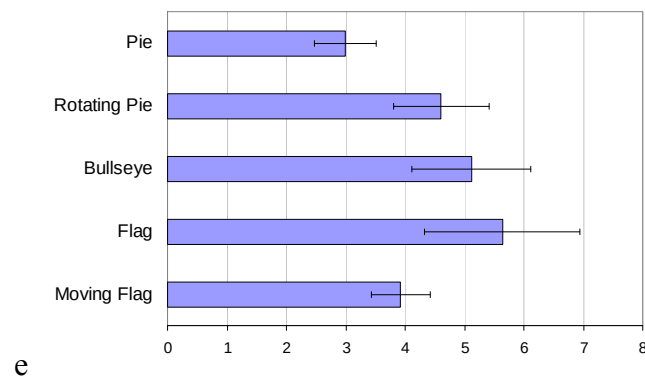


Figure 4-15: Average % error rate (+/- SE) per widget design.

4.10.5.3 Number of Crossings

There was a significant effect of tracking on Error ($F_{1,14}=17.72$, $p<0.01$) yet this difference's magnitude is small and likely to have no weight in deciding to use the tracking feature in a particular design. Table 4-3 shows these values.

Table 4-3: Average number of crossings (NC) per tracking level. A value of 3 indicates a user overshoots a target once.

	Avg. NC	SE
No Tracking	1.71	0.09
Tracking	1.65	0.08

¹⁶ Mauchly's test indicates that the assumption of sphericity is not met; we corrected the degrees of freedom using lower bound estimates.

PRESSURE WIDGETS

4.10.5.4 Navigation Time

There was a significant effect of tracking on Error ($F_{1,14}=44.25$, $p<0.01$) yet this difference's magnitude is small ($\sim 100\text{ms}$) and likely to have no weight in deciding to use the tracking feature in a particular design. We observed no significant effect of widget on navigation time (Figure 4-16), still we can observe how interactions done with the Bullseye were the fastest of the group but by amounts in the order of milliseconds.

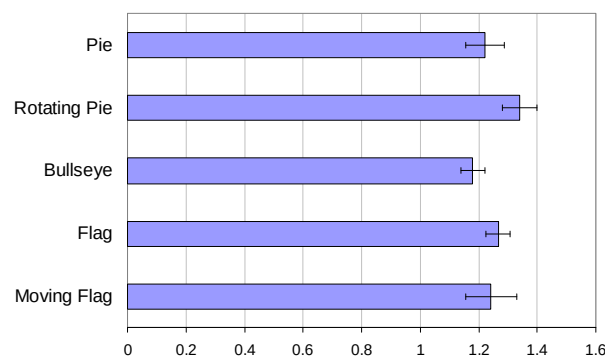


Figure 4-16: Average Navigation time (seconds +/- SE) per widget.

4.10.5.5 Subjective Impressions

At the end of the study, we gathered subjective information from participants through the post-experiment questionnaire. All users quickly learned how to manipulate the widgets. Participants were asked on a 7-point “agree – strongly disagree” Likert scale if they felt they knew how to use a particular widget after a few trials. Answers averaged above 5 (agree – strongly agree) for all widgets except the *Rotating Pie*. This last widget scored at 4.7 (somewhat agree – agree).

Diversity in participants' personal preferences surfaced when we asked them which widget they prefer to use, or which one they find visually attractive. For these questions, participants rated each widget numerically from 1 to 5 (with 1 being “best”). Participants rated the *Moving*

PRESSURE WIDGETS

Flag highest (2.2) on preference and *Rotating Pie* last (3.4). This assessment was reversed for visual attractiveness, where *Rotating Pie* was ranked highest (2.2) and *Moving Flag* was ranked second lowest (3.2), only better than the *Flag* (3.7).

Users almost unanimously commented as to how quickly they were able to select a target when it was the last one on the pressure intervals. Similar to the case of target selection in x-y space with a target of infinite width, users performed a fast, ballistic increase in pressure that reached the last interval without the risk of overshooting. In this regard, the recommendation made by Walker et al. (Walker & Smelcer, 1990) remains true: borders are effective in decreasing selection time. However, unlike spatial movement, pressure space has only one “effective” upper limit, or border. Still, this observation can help us choose what actions to map to different pressure intervals, e.g. the last interval of a discrete pressure widget could be assigned to a frequently used operation.

4.10.6 Discussion

Our results from this study indicate that variations in visual design can have a significant impact on a pressure widget's usage speed, accuracy, and the interference between the pressure and the spatial x-y movement components of the pen. The following two points summarize our experimental results:

- Designs in which the widget's cursor was affected by pressure resulted in less interference than those in which it was not. Designs where pressure affected the angle of a widget's cursor or targets (i.e., *Pie* and *Rotating Pie*) also served to decrease interference, though to a somewhat lesser degree.

PRESSURE WIDGETS

- The *Moving Flag* and *Pie* designs exhibited comparably low error rates (below 5%), and were the widgets with the fast selection speeds. These widgets were also ranked the top two “easiest to use” designs, in the users’ subjective evaluations.

In light of these recommendations and our studies, the use of the *Pie* design emerges as the overall top performer design in terms of low interference, fair speed, good accuracy and ease of use. Our research also reveals a measurable upper bound of 12 pixels on average for the interference effect. This observation can be of assistance for designers of pressure widgets. For example, one can identify x-y movements greater than a prescribed upper bound (for a particular input device) as a voluntary user action and not as the result of an interference effect. Readers interested in a detailed account of this experiment can find it in its correspondent technical report (Ramos & Balakrishnan, 2004).

4.11 Summary

In order to design effective, useful and meaningful interactions that leverage the pressure a person applies through a stylus it is crucial for us to know people’s limits when controlling that pressure. In this chapter, we have presented the results of a controlled experiment that investigated human ability to use pen pressure to perform discrete target acquisition tasks, with different selection techniques.

Our results indicate that the *Quick Release* selection technique was preferable overall, and that dividing pressure space into six levels or less is optimal. An interesting observation regarding this result is that even though it stems from a particular experimental hardware setup, other researchers seem have observed similar performance limits on other devices. Mizobuchi, et al. recommend using no more than seven pressure regions on their hand-held experimental device (Mizobuchi et al., 2005). Cechanowicz et al. investigate people’s performance

PRESSURE WIDGETS

when interacting with a pressure-sensitive mouse (Cechanowicz, Irani, & Subramanian, 2007) and find that performance degrades rapidly when the number of pressure level presented to users is greater than six. This similarity of outcomes across different hardware configurations is remarkable and suggests that such limit in performance might in fact dependent of people's force-sensing capabilities.

We also found that appropriate visual feedback is critical: users were not able to perform effectively eyes-free pressure selection with only an hour of practice. In particular, we observed that users' performances between full and partial visual feedback conditions were drastically different, suggesting that achieving expert behavior for the presented task is difficult, if entirely achievable. The choice of appropriate visual feedback has a direct impact on the widget's usability: results for a follow-up study (Ramos & Balakrishnan, 2004) indicate that a widget's design can affect significantly on metrics such as interference, speed, and error rate.

While the results from this chapter are important, they apply to a particular aspect of pressure-based interaction techniques: discrete pressure manipulation. Other types of interactions might benefit from the use of pressure as a continuous entity and in the next chapter we investigate in more detail what form such interactions might take.

5 Zooming and Sliding for High-Precision Parameter Manipulation

Chapter 4 mainly considers the use of pressure for the selection of discrete targets. However, there are opportunities to use pressure as the means of controlling a continuous parameter in a variety of tasks. The manipulation of a parameter is a fundamental task in most graphical user interfaces and we see that the use of pressure in such tasks can be beneficial, especially with high-precision parameter manipulations.

Although high-precision parameter manipulation can be accurately achieved by simply entering an exact numeric value with an appropriate text input technique, from the user's point of view this exact method is not always the most appropriate or preferred. Interactions such as identifying and then picking a single pixel from a high resolution image, seeking a particular frame in a long video stream, or adjusting a continuous image color parameter are examples of parameter manipulation tasks where more interactive direct manipulation techniques can be preferable since the user may not be certain *a priori* as to what value to enter. Furthermore, the immediate feedback that an interactive widget or technique can provide while the user adjusts the parameter is immensely valuable as it affords a more continuous style of interaction rather than the discrete style that results when specific values are entered explicitly.

The challenge in designing interactive techniques for continuous high-precision parameter manipulation is that the manipulation granularity desired by the user when adjusting parame-

ters may differ from one parameter to another, or even within the same parameter in different usage scenarios. Thus, interaction techniques for high-precision parameter manipulation should support adjustment of the granularity within which the manipulation occurs, allowing users to make coarse granularity manipulations for initial adjustments followed by finer granularity manipulations for the final precise parameter specification.

In this chapter we propose and study a mechanism for pressure-sensitive input devices, called Zoom Sliding, or Zliding for short, in which users use the pressure modality to explicitly zoom or adjust the granularity of the parameter space, while sliding or dragging the input device to perform high-precision parameter manipulation within that zoomed parameter space¹⁷.

5.1 Related Work

Common strategies to facilitate spatial parameter selection tasks include reducing its Fitts' Index of Difficulty (ID) (Fitts, 1954) by making targets larger, or by bringing them closer to the user's pointer. For example, McGuffin et al. (McGuffin & Balakrishnan, 2002) show how increasing a target's size even at the final stages of a pointing task can be beneficial. Drag-and-Pop (Baudisch et al., 2003) reduces the distance that a user has to travel by bringing objects closer, using the pointer's trajectory (Figure 5-1).

¹⁷ Much of the material presented in this chapter was previously published in the Proceedings of the UIST '05 Symposium on User Interface Software and Technology (Ramos & Balakrishnan, 2005).

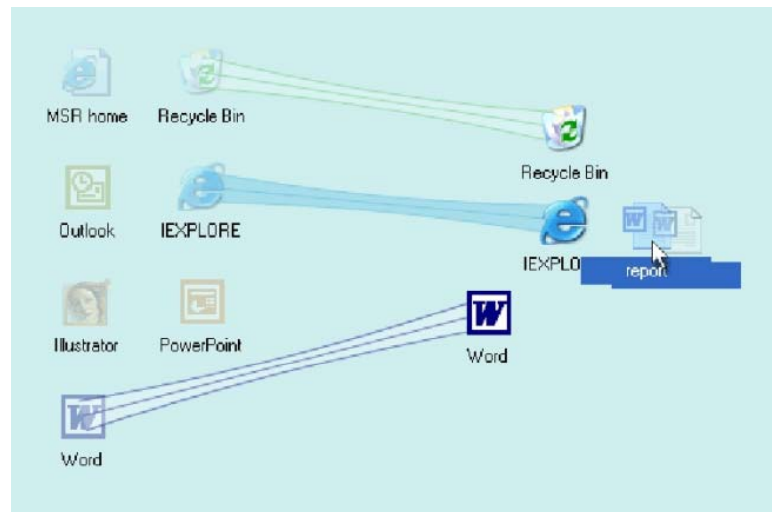


Figure 5-1: Drag and Pop interaction technique. As the user drags an object over the workspace, potential targets are brought closer to it.

Another strategy to increase the precision of the user's interaction is to adjust the input device's control-display (CD) ratio. Semantic Pointing (Blanch, Guiard, & Beaudouin-Lafon, 2004) improves the selection of objects by assigning them different CD ratios according to their importance—i.e., one need to transverse a larger distance with a mouse in order to escape the area of an important UI element, such as the “Save” button on Figure 5-2. However, both these elements are fixed, which could be problematic if the user's assessment of what is important changes. All these approaches are aspects of the same solution: changing the scale of a target or the space that contains it. In some instances, zooming occurs in the visual domain, and in others, the motor domain.



Figure 5-2: Example of a dialog window where its buttons are weighted according to their semantics. (Top) how the window looks visually. (Bottom) how the window looks in motor space –i.e., what it feels like when interacting with it.

There has been a consistent effort to develop controls and interactions tailored for precise parameter selection and manipulation tasks. The Alphaslides (Ahlberg & Shneiderman, 1994) is a compact selector that allows users to quickly pick a single item from a list of thousands, essentially by providing two or three sub-sliders with different levels of granularity (Figure 5-3 left). The FineSlider (Toshiyuki, Kouichi, & George R. Borden, 1995) extends the Alphaslides's idea and lets users adjust the rate at which the slider's selection changes, by using a rubber-band metaphor (Figure 5-3 right). The PVS slider control that we described in chapter 3 also uses a rubber-band metaphor to adjust the granularity with which users slide through a video stream. The issue of precise manipulation also applies to scenarios where the input mechanism can be imprecise by nature.

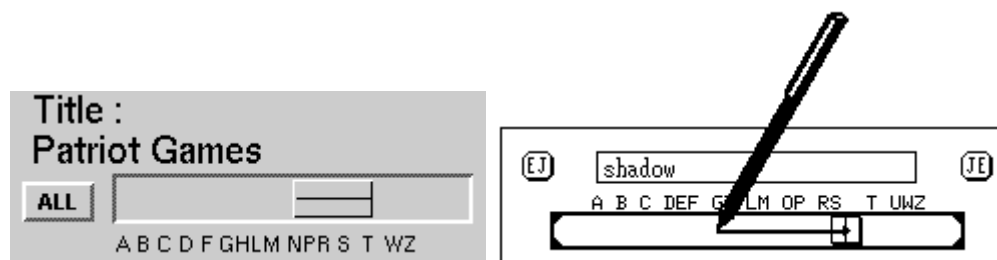


Figure 5-3: (left) The Alphaslides with two levels of granularity (i.e., the upper and lower parts of the control's handle explores the dataset at different granularities). (Right) The FineSlider widget: the pointer is connected to the slider's handle by means of a virtual "elastic band" that dictated the rate at which the parameter space is browsed.

FLUID ZOOMING & SLIDING...

Potter et al. (Potter, Weldon, & Shneiderman, 1988) investigate how to increase the accuracy of a bare finger on a touch screen and show how their “take-off” approach outperforms traditional touching techniques. With “take-off,” a target is defined not by the position a finger lands on, but by the position it is lifted at. This lets users adjust a cursor’s position while their finger stays in contact with the touch screen (Figure 5-4).

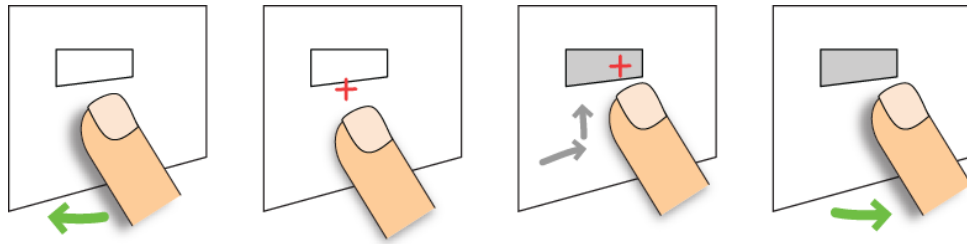


Figure 5-4: Diagram of the take-off technique. From left to right: when a user's fingerpad touches the surface a crosshair cursor appears slightly above it (red cross). While in contact with the surface, the user can adjust his finger's position until the desired target is reached (gray rectangle). At that point the user lifts his finger and finalizes the selection.

However, this approach does not allow for the gain of the finger’s movement to be changed. The *precision handle* (Albinsson & Zhai, 2003) lets a user’s bare finger manipulate a graphic handle (Figure 5-5) around a pivot in order to change the interaction’s granularity and achieve pixel-level accuracy.

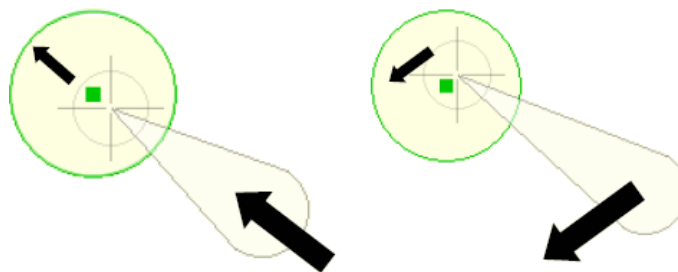


Figure 5-5: Precision handle. The cross hair follows (small arrow) the direction of the finger (big arrow) Guiard et al. recognize that high precision selection tasks can be thought of as multi-scale navigation tasks (Guiard, Beaudouin-Lafon, & Mottet, 1999). In addition, there is a significant body of work establishing a comprehensive theoretical framework for multi-scale tasks. Furnas’ space-scale diagrams (Furnas & Bederson, 1995) give us the means to understand and

FLUID ZOOMING & SLIDING...

analyze multi-scale interactions and interfaces. Guiard et al. have also shown that multi-scale pointing still obeys Fitts' Law (Guiard, Beaudouin-Lafon, Bastin, Pasveer, & Zhai, 2004). Building on this evidence, we argue that facilitating space scaling and manipulation operations should help users with high precision tasks. Some of the literature suggests that panning (i.e., equivalent to parameter manipulation) and zooming is an integrated task (Jacob, Sibert, McFarlane, & Mullen, 1994), and recommends it be driven by an integrated device.

Igarashi and Hinckley (Igarashi & Hinckley, 2000) introduce speed-dependent automatic zooming, a technique that facilitates navigation tasks over large spaces, using a 2-DoF integrated device (Figure 5-6). This technique keeps the visual flow of the navigation constant, while scrolling at different speeds, thus improving users' performances over traditional scroll and pan-and-zoom methods. However, other results indicate that there may be benefits in separating pan from zoom. It has been shown how bi-manual interaction techniques can be faster (Buxton & Myers, 1986) than interactions driven by one hand, and can permit parallelism in multi-scale tasks (Bourgeois, Guiard, & Beaudouin-Lafon, 2001; Hinckley, Czerwinski, & Sinclair, 1998; Zhai & Smith, 1999).

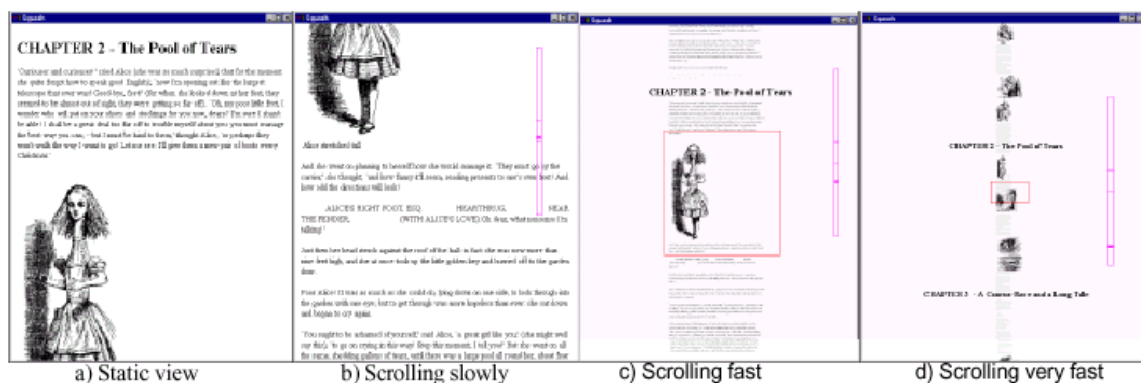


Figure 5-6: Igarashi and Hinckley's implementation of speed dependent zooming. The zoom factor is altered to elicit the perception of a constant visual flow.

A common theme present in all the above uni-manual designs and techniques is that both scale and parameter values are specified as a function of the cursor's x-y position. Further-

more, in many of these techniques, scale adjustments are determined by the system without giving users much say as to what scale values to apply and when. With bimanual techniques, the non-dominant hand controls the scale, while the dominant hand performs the parameter manipulation within that scale. In contrast, our current work focuses on how users can control scale via a pressure transducer while simultaneously manipulating a parameter within that scale space using a spatial x-y cursor.

5.2 Motivation and Goals

We are driven by the steady technological progress in pen-enabled and touch-sensitive platforms, where high precision manipulation tasks are made even more challenging by very small or very large physical form factors and interfaces. The fixed granularity of standard GUI widgets like sliders may work reasonably well in a desktop computing environment, but may not scale to tiny PDAs or very large wall sized displays. On a TabletPC or PDA with small screen and input space, for example, the fixed relatively coarse granularity of some GUI widgets can hinder user's ability to make high precision adjustments. Example scenarios where users may need to adjust the granularity of the parameter space in order to make precise parameter adjustments include:

- *Graphic applications*: Users may need to select quickly a precise pixel from a large bit-map that cannot be displayed at pixel-visible resolution on a small screen. Also, users may need to adjust precisely a value controlling a visual feature, such as the blending across several images.
- *Browsing on ZUIs*: Users may need to navigate through a map both at a very large and at a very fine scale, i.e., coarse and fine granularities.
- *Acquisition of small controls in the GUI*: Elements in an interface can present very small selection footprints, requiring a change in CD ratio to facilitate selection.

- *Analog-like controls*: These controls offer a granularity that depends on their physical size and the input device's CD ratio. Users interacting with such controls may need to do fine tuning in order to attain a precise value, such as a frequency in a radio tuner.

In designing an interaction mechanism that facilitates high precision parameter manipulation, we have the following goals:

- *Integrated scale (granularity) and parameter manipulation*. The interaction should support zooming of the parameter's scale space and concurrent high precision adjustment of the parameter within that space.
- *Infinite parameter scale adjustment*. It should be possible to adjust fluidly a parameter to an infinitely small or large value. We argue that having the ability to attain virtually infinite precision or gain is a rarely explored objective is worthy of attention.
- *Familiar interactions*. The new interaction should feel familiar, leveraging the typical user's vast experience with standard GUI widgets and interaction techniques.

5.3 The Zlider

The Zlider widget (Figure 5-7) consists of a rectangular working area that the user can scrub in order to adjust a parameter $v \in [low, high]$, where *low* and *high* are arbitrary limit values. There is no particular handle the user needs to grab to use the zlider widget. To operate the Zlider the user taps and drags its pointer across the working area until the desired value is reached or effect is achieved. At all times a red needle indicates the position of the value being adjusted relative to the possible minimum and maximum values at the extremes of the widget.

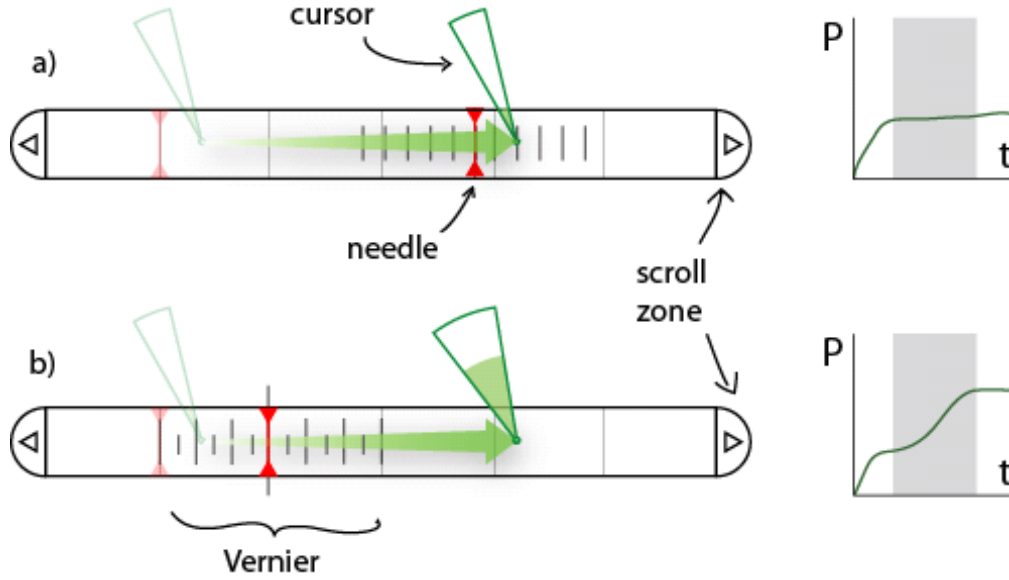


Figure 5-7: Zliding on the Zlider widget. a) A user manipulates a parameter at coarse granularity by sliding through the control while applying low pressure with the pressure transducer. b) The same x-y sliding action while pressing harder increases the granularity of the parameter space, allowing for more precise parameter manipulation when desired. The graphs on the right plot pressure over time, with the interval when the sliding occurs highlighted. A video displaying this design can be found at <http://www.youtube.com/watch?v=EcE3XBytN-U>

We say that at time i , the Zlider has a scale¹⁸ s_i that dictates how the parameter v

changes: $v_i = v_{i-1} + \frac{\Delta \times range}{scale \times length}$; where $v_0 = low$, Δ is the distance between the tapping point

and the pointer's current dragging location, $range = high - low$, and $length$ is the working area's length. The Zlider also displays a Vernier as suggested by Ayatsuka et al. (Ayatsuka, Rekimoto, & Matsuoka, 1998); however, the Vernier in our Zlider adapts its grid spacing depending on the widget's current scale factor.

5.3.1 Pressure Cursor

Though not integral to the Zlider design, we use a pressure cursor (Figure 5-8) across our implementations, instead of the default cursor found in most GUIs. Our pressure cursor provides users with a real-time indicator of the pressure they are applying with the input transducer.

The pressure cursor has a wedge-like shape that changes its aperture with the amount of pres-

¹⁸ While the Zlider's scale depends on the amount of pressure a user applies with a pen's tip, it is possible to map this parameter to other types of transducers such as discrete keys or (scroll) wheels.

sure applied. The wedge's area fills as the pressure increases, until completely filled when the pressure reaches the maximum level the device can sense. The cursor's hot spot, i.e., the point that is aligned with the pen's tip, is located at the wedge's vertex.

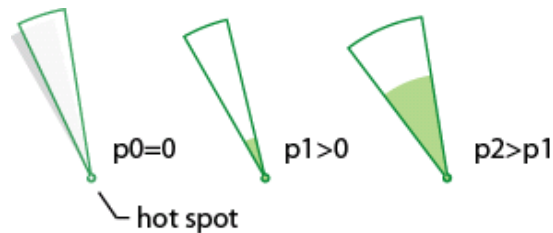


Figure 5-8: Pressure cursor. The wedge increases in size and fills up as pressure increases. ($p_2 > p_1 > p_0$). A video illustrating this design can be found at <http://www.youtube.com/watch?v=EcE3XBvtN-U> - timecode 00:30.

5.3.2 Integrated Zoom & Slide Control

Our default interaction design uses a pressure-sensitive stylus as an integrated input device for zooming of the parameter's scale space and sliding (i.e., manipulation) of the parameter's value within that scale space. The scale factor of the Zlider is adjusted by changes in pressure at the stylus' tip, and the stylus' x-y position enables sliding of the parameter's value. We use an exponential function of the form $scale = base^{f(p)}$ to calculate the scale factor, where $f(p)$ is a function of the stylus' reported pressure at a particular time.

Chapter 4 highlights the difficulty users can experience in maintaining a constant level of pressure while dragging a stylus. We therefore utilize a combination of both signal processing and interactive techniques to minimize unwanted changes in the control's scale. Raw pressure data first passes through a low-pass filter. Then, it passes through a hysteresis process that stabilizes the signal further. Finally, a parabolic-sigmoid transfer function (Figure 5-9) is used to account for users' performance when they apply force through an isometric input device like the stylus' tip pressure sensor.

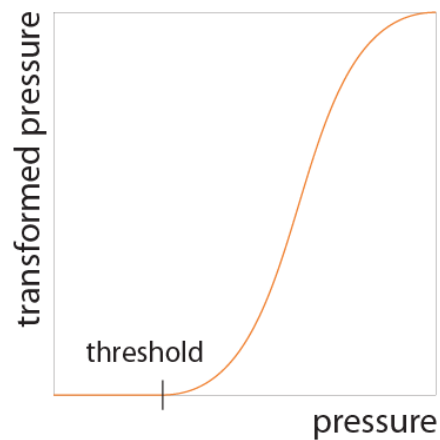


Figure 5-9: Sketch of a parabolic-sigmoid transfer function used to preprocess the pressure signal.

This transfer function has been used in similar scenarios (Barrett, Selker, Rutledge, & Olyha, 1995) and is consistent with the effect we want to achieve. This effect is comprised of: 1) an initial “dead zone”, 2) a slow response region at low pressure levels (where users can vary pressure significantly without noticing), 3) a linear behavior region in the mid pressure ranges (where users have good control of pressure), and 4) a slow response region at high levels of pressure (where the user’s applied force can produce tremors, causing sudden pressure variations that are magnified by the exponential scale function).

Figure 5-10 shows an example of how the different pressure-stabilizing stages affect the resulting scale factor.

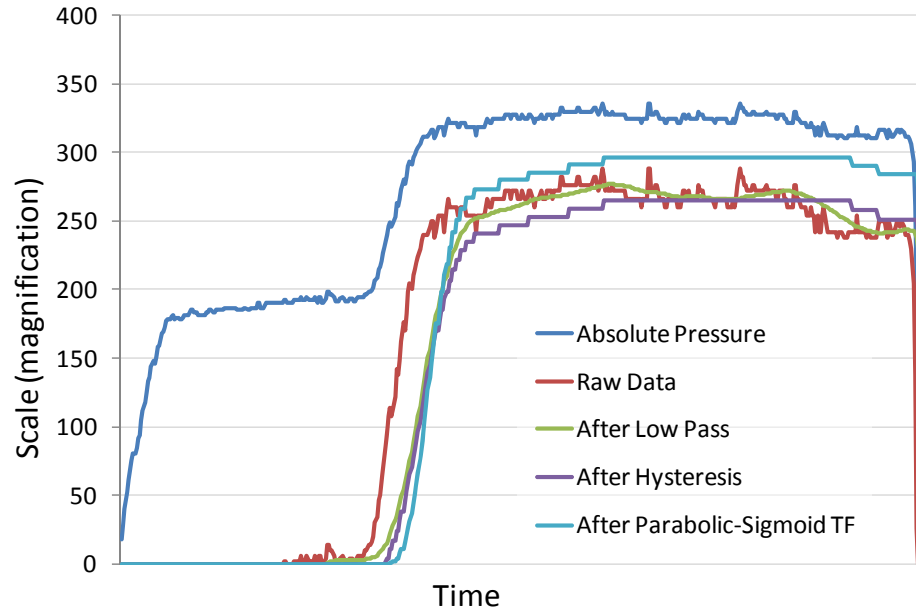


Figure 5-10: Effect of the signal's stabilization and filtering techniques. The absolute pressure line represents the transducer's raw signal.

5.3.3 Clutching the Zoom Level

The Zlider design has a clutching mechanism that enables users to completely stabilize pressure and hence lock the zoom level while sliding. Users clutch by sliding the cursor away from the Zlider's working rectangle (Figure 5-11.2). While clutched, users can still slide outside the working rectangle (Figure 5-11.3) but the widget maintains its scale at the last reported value regardless of pressure variations. Re-entering the working rectangle declutches (Figure 5-11.4).

While the widget's scale factor is the same at the point in time when users clutch and declutch, it is possible that the pressure they applied at these moments is not. By design, we use this situation to let users increase the Zlider's scale factor arbitrarily in a relative manner. In other words, by clutching, users can not only stabilize scale variations, but also achieve as much precision as needed. To go beyond the scale value attainable when the pressure at the stylus' tip reaches its maximum value, users can in one continuous gesture:

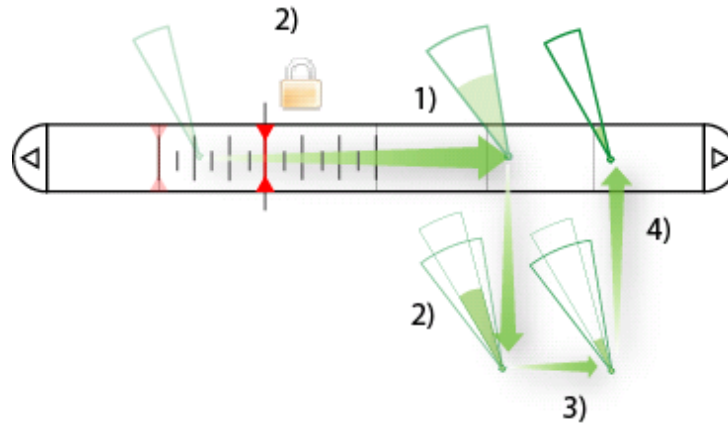


Figure 5-11: Clutching the zoom level: (1) the user slides the pen over the zlider's working area, (2) then locks the pressure by dragging the pen outside the working area. (3) While the pressure is locked, pressure variations do not change the scale if the parameter adjustment. (4) The user unlocks the pressure by re-entering the control's working area. A video illustrating this design can be found at <http://www.youtube.com/watch?v=EcE3XBytN-U> – timecode 01:09.

- Increase pressure and hence scale factor (Figure 5-11.1);
- Clutch (activate pressure lock, (Figure 5-11.2);
- Decrease applied pressure (Figure 5-11.3);
- Declutch (deactivate pressure lock, (Figure 5-11.4)); and
- Increase pressure and hence scale beyond the value at step (a).

This process can be repeated in order to attain higher precision levels if the user so desires.

Conversely, an inverse series of steps allows users to decrease the scale factor from a high to a low level.

There are three ways in which the user is notified that they are clutching or declutching: a) a very brief auditory feedback, b) an icon that follows the Zlider's needle (Figure 5-11.2), and c) a change in the physical appearance of the pressure cursor (Figure 5-11.2-3). Pilot studies revealed that while visual feedback is important, auditory feedback was beneficial to users who were not visually focusing on the Zlider control.

5.3.4 The Selection Mechanism

The Zlider design uses the release of the stylus from the interaction surface as an indication of selection. This is consistent with the behavior of regular slider controls, and Chapter 4 also

FLUID ZOOMING & SLIDING...

supports lifting the stylus as a selection technique for pressure-aware widgets. However, some issues remain that deserve our attention.

First, we need to determine the Zlider's behavior when the stylus is lifted from the interaction surface (i.e., the applied pressure becomes zero). Even though one possible design decision is to make the scale=1, this is not always desirable. Pilot studies revealed that users might lift the stylus because they wanted to re-invoke the Zlider from a different point when they found themselves sliding very close to either extreme of the working rectangle. Users indicated that resetting the scale to 1 was annoying, since it forced them to reacquire the scale value. The same situation was found when users missed the target parameter value by a small amount. In this case, they explicitly voiced the need to perform, as one user called it, "quick micro-adjustments".

Based on this feedback, we modified the Zlider's behavior so that it maintains its last reported scale value as long as the stylus is within sensing proximity and the widget's working area – i.e., hover. This sensing or tracking capability can be found in most modern digitizing tablets as well as in other display technologies (such as the SmartBoard), and has started to be used as a design element in a number of novel user-interface widgets (Bezerianos & Balakrishnan, 2005; Fitzmaurice et al., 2003). While it is possible to use time-based techniques to simulate to some degree this behavior in devices that lack proximity sensing, we decide to concentrate our attention into the use of pressure and tracking modalities available on our experimental devices. Figure 5-12 shows the state-transition diagram of the Zlider's behavior.

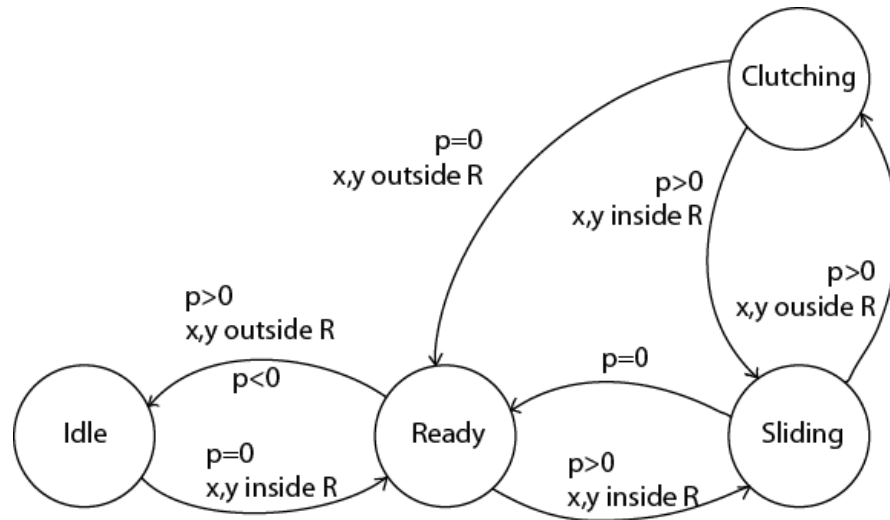


Figure 5-12: Zlider's state-transition diagram. *R* is the working rectangle; *x,y* the cursor's position. *p* < 0: stylus out of range; *p* = 0: stylus is being tracked; *p* > 0: stylus is touching the tablet. Zlider's scale is reset to 1 at the idle state.

The second issue we need to consider is estimating what the Zlider's last reported scale (pressure) value should be at the time the stylus is lifted from the interaction surface. We need to identify as accurately as we can the exact moment when users start their lifting action. This is important as we do not want the Zlider to change accidentally its scale factor.

In our case, looking back a fixed number of samples, as was proposed by Buxton (Buxton et al., 1985), is not sufficient because the number of samples we need to trace back depends on how fast users lift the stylus. Observations in our pilot studies also revealed that users generally pause for a few milliseconds before lifting the stylus, thus defining a very small pressure valley. Furthermore, pressure values from that point onwards follow a monotonically decreasing trend. With the above information we estimate the scale factor at the time the user starts to lift the stylus. Our algorithm looks backwards in the device's buffer until the small valley is found, or the curve stops its decreasing trend. Since the Zlider control's scale responds in real-time to variations, it is possible that there is a mismatch between the estimated scale value and the scale at the point the stylus is lifted. Sudden changes in the Zlider's scale factor

FLUID ZOOMING & SLIDING...

would result in an undesired disorienting effect on the user. In order to mitigate this effect, the Zlider's scale smoothly changes to the estimated last reported scale value by presenting an animation that linearly interpolates the two different scale values. The same type of smooth transitions is consistently used in the Zlider widget when changes, otherwise too abrupt, need to occur.

These two design features provide functionality similar to clutching, wherein users have another way to achieve an arbitrarily high level of precision. In this case, to increase the scale once no more pressure can be applied, a user can:

- a) Increase pressure and hence scale;
- b) Quickly lift the stylus from the interaction surface, staying within tracking distance;
- c) Touch the working area again; and
- d) Increase pressure until the desired magnification is achieved.

We call this interaction mechanism *air clutching* and unlike clutching, this interaction does not allow users to un-zoom in a controlled fashion. Nonetheless, we observed that both the locking and clutching mechanisms served different users' interaction styles when adjusting the zoom. Figure 5-13 illustrates the air clutching mechanism.

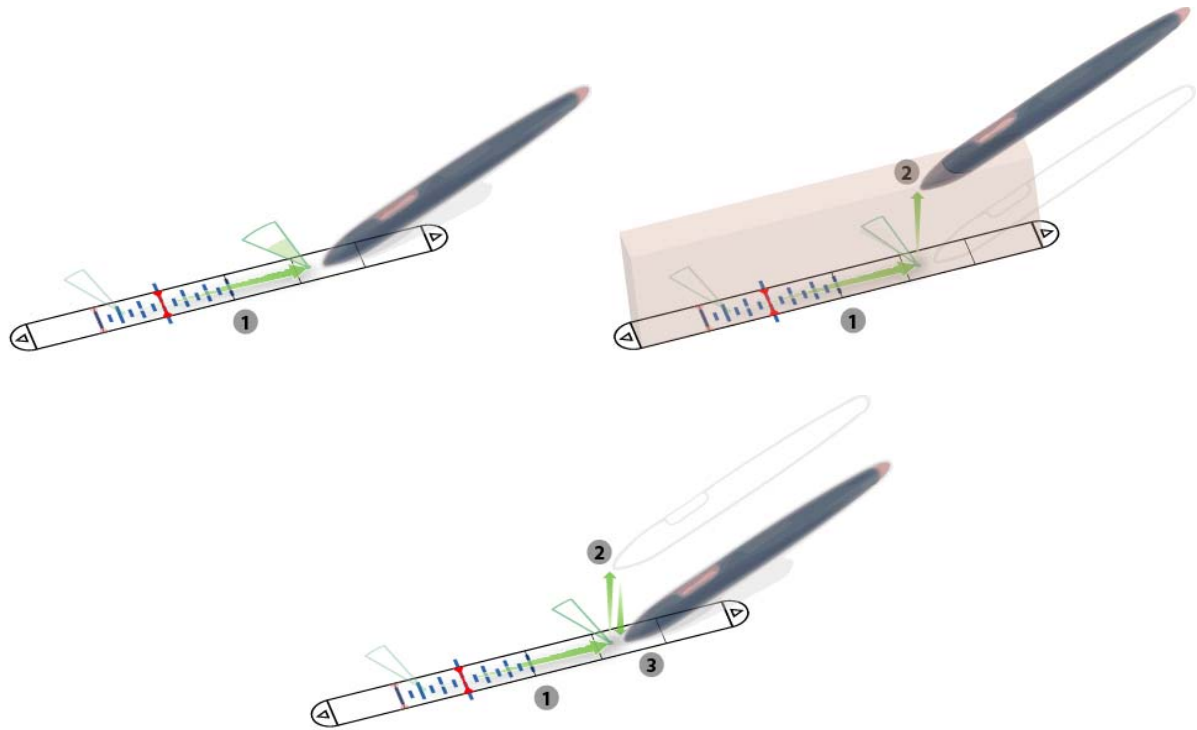


Figure 5-13: Air Clutching in action. (Top-left) During a zlidering operation the scale increases to a value of S_1 as the applied pressure increases to P_1 (1). (Top-right) The pen is rapidly lifted and remains within tracking distance causing the zlider's scale to remain at value S_1 (2). (Bottom) The pen engages the Zlider control with pressure $P_0 \ll P_1$, yet the scale is S_1 (3). From this point forward, one can repeat air-clutching operations to increase the control's scale to an arbitrarily high value. A video illustrating this design can be found at <http://www.youtube.com/watch?v=EcE3XBytN-U> – timecode 02:02.

5.3.5 Scrolling

The Zlider is controlled by relative displacements in its working area. However, pilot studies showed that some users wished the familiarity of continuous scrolling found in ordinary scroll and slide controls. Our design easily incorporates continuous scroll zones at the extremes of its working area (Figure 5-7). If, while sliding, the cursor reaches a scroll zone the Zlider enters a scrolling mode. Sliding has no effect in this mode and the parameter it controls changes at a constant rate proportional to the current scale. By adjusting pressure, scale can be changed while in scroll mode thus affecting the scrolling speed.

5.4 Alternatives for Decoupled Zoom Control

The Zlider was designed to be operated by an integrated pressure and position sensing input device, such as a pressure-enabled stylus. However, our design can easily support other ways to adjust the Zlider's scale factor. In particular, we can use input originating from the user's non-dominant hand. Decoupling the scale control from the dominant hand has the potential to eliminate undesired interference between zooming and panning that may occur while using the stylus as the only input device. At the same time, this decoupled way of controlling scale has the potential to still allow users to perform zooming and sliding concurrently (Bourgeois et al., 2001). In this section, we explore two instances of decoupled design strategies for adjusting the Zlider's scale: a force-sensing button, and two discrete keys.

5.4.1 Force Button

A force button is an isometric input device that can have a minimal footprint. This makes it an attractive design choice that can be incorporated in many form factors such as hand-held devices, tablets, and even in traditional input devices such as mice or keyboards. In addition, previous research (Harrison, Fishkin, Gujar, Mochon, & Want, 1998) shows the potential advantages of embedding force sensors on hand-held devices.

For our exploration of this style of input we used a phidget (Greenberg & Fitchett, 2001) force sensor Figure 5-15. The signal reported by this sensor is very similar to the one given by the stylus' tip and we use it in the same way. Because of this similarity, many of the issues regarding signal stabilization that we discussed in the previous sections apply to this input device. However, since the force button is decoupled from the stylus, it is easier to determine what the scale factor is at the time users lift the stylus. Nonetheless, we found that both the signal stabilization techniques and clutching mode already discussed were effective at miti-

gating signal instabilities while users slide. Clutching and tracking can be used with this input mode to achieve arbitrarily high precision levels.

5.4.2 Discrete Keys

The second decoupled method of controlling the Zlider's scale uses two discrete keys: one for increasing the scale and another for decreasing it. This input mechanism is easy to implement in a variety of form factors and sizes and it can be seen as the lowest common denominator method for changing the scale in many scenarios.

We implement this input mode using the Shift and Ctrl keys found in most computer keyboards. Users tap on Shift and Ctrl in order to respectively increase or decrease the scale factor by a constant increment. Also, users can tap and hold on either key in order to zoom or unzoom at a continuous rate. The signal from this input is stable, making it easy for users to slide at constant scales. Consequently, we neither need to filter the input, nor use the parabolic-sigmoid transfer function. Also, finding the scale value when the stylus is lifted becomes trivial.

Though clutching and tracking are still available, users can use the keys alone to reach arbitrarily high precision levels. However, this discrete input has a drawback in the amount of time a user requires to reach a determined scale factor. This time depends both on the mechanical properties of a key that needs to be pressed and released, and the rate at which scale is adjusted when a key is held. This rate needs to be carefully considered. A rate that is too fast will make the interaction quicker, yet difficult to control (i.e. users will overshoot the desired scale). Conversely, a slow rate will make the interaction more controllable, yet unacceptably sluggish. Equally important is the choice of the step the scale should change for each

key press. For our experiments, we updated the scale every 30ms after a key was held for 400ms.

5.5 Experiment

Our experiment investigates how three different scale-adjusting strategies: Stylus, Force Button, and Keys affect people's interactions and performance in a high precision selection task that uses the Zlider.

We are particularly interested in investigating how these strategies compare to one another. In particular, the simplicity of the Keys and Force Button techniques could outperform the Stylus technique where the combination of linear x-y movement and pressure control with the stylus tip might interfere with one another. On the other hand, the integrated nature of the Stylus technique has the advantage in that users will likely conceive of the zoom and sliding task as a conceptual whole, rather than two separate subtasks as with the Keys and Force Button techniques where zooming and sliding control are separated across the two hands.

Because each technique has its own idiosyncrasies, the experiment's results can help both designers and users to choose the best solution for a given situation. In addition, the experiment will provide us with valuable user feedback regarding the Zlider control and the overall experience of Zliding.

5.5.1 Apparatus

We used a Wacom Cintiq 18SX interactive LCD graphics display tablet with a wireless stylus that has a pressure-sensitive isometric tip (Figure 5-14). The stylus reports 1024 levels of pressure, and has a binary button on its barrel¹⁹. The stylus does not provide any distinguish-

¹⁹ The digitizer tablet used in this study responded to forces between zero and 300gf.

FLUID ZOOMING & SLIDING...

able haptic feedback in relation to the pressure applied. The tablet's active area was mapped onto the display's visual area in an absolute one-to-one manner. The tablet was laid out flat over a desk in front of users as if it were a slate resting over a table.



Figure 5-14: Wacom CintiQ interactive display with its accompanying pressure sensitive pen (enhanced detail on the right).

To implement the Force Button condition we use a phidget (Greenberg & Fitchett, 2001) interface board that read data from a force sensor. Users applied force on the sensor through a thin layer of hard rubber protecting them from its uncomfortable original profile. Although this force sensor reports up to 1000 levels of force, we only use the first 2/3 of them, as in our pilot studies users showed discomfort when reaching values above 2/3 of the way. The Keys condition was implemented using the Shift and Ctrl key on a regular PC keyboard. The experimental software ran on a 1.4GHz P4 PC with Windows XP Professional.

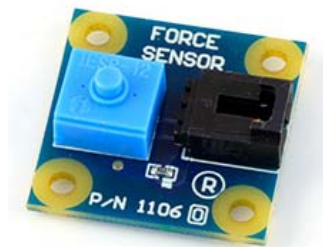


Figure 5-15: Phidgets pressure transducer used in our experiment.

5.5.2 Participants

Four female and eight male volunteers, 18-44 years old, participated in the experiment. Ten were right-handed. All the participants had normal or corrected-to-normal vision and less than half (5 out of 12) of them reported having some prior experience with tablets like the one used in the experiment. Participants were recruited from our university population through e-mail lists. No compensation was provided.

5.5.3 Task and Stimuli

A serial target acquisition and selection task was used. The stylus was used to control the sliding behavior of a Zlider widget with its scrolling zones disabled and its clutching and hover mechanisms enabled. The experimental trials simulate a pan-and-zoom task on a reduced interaction footprint, like the ones found in hand-held computers or dialog windows. In each trial the user controls the Zlider in order to locate and select a target in a workspace area 1500 pixels long, shown through a viewport 256 pixels long (Figure 5-16).

The target to be selected is represented as a green rectangle and can have three possible widths: $1/10$, $1/1,000$ and $1/100,000$ of the workspace's length. In turn, the target can be located at a near, mid or far distance from the top of the workspace. Distance is chosen according to the target's width so that $distance = n \times width$, where n is an integer, $width$ is the target's width, and $distance$ belongs to either the intervals $[150, 450)$, $[600, 900)$ or $[1050, 1350)$. Besides the target, the workspace contains a horizontal grid that increases in density in the vicinity of the target, helping users locate it. As the user scrubs across the Zlider the workspace scrolls accordingly under the viewport in the same way a document scrolls in a text editor.

During the trials, users can adjust the interaction scale through one of three methods: *Stylus*, *Force Button* and *Keys*. Changes in the scale are reflected by magnification changes on the working area and on the pen's C:D ratio. This scaling operation makes accessible targets that otherwise would be too small to select. Users are instructed to scroll through until the target is inside the viewport, visible, and covering the selection line (Figure 5-16). When this happens the target changes its color from green to red, and users finish the selection by lifting the stylus from the interaction surface. The workspace has a textured background, which helps users to be aware of the current scale factor they are at, thus alleviating desert fog (Jul & Furnas, 1998) effects in the scale space.

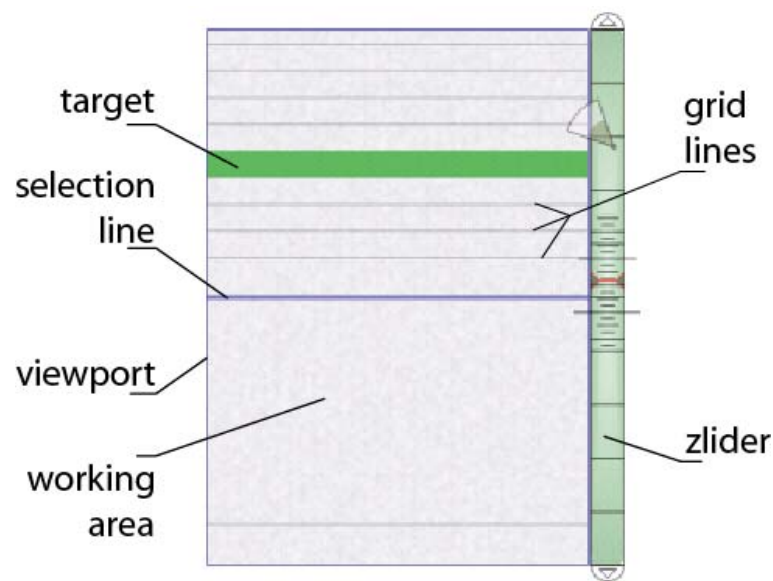


Figure 5-16: Elements in the experimental setup.

5.5.4 Procedure and Design

A within-participants full factorial design with repeated measures was used. The independent variables were *Technique* (*Stylus*, *Force Button* and *Keys*), *Width* (*large*, *small*, *smallest*), and *Distance* (*near*, *mid*, *far*). The order in which techniques were presented to users was included as a between-subjects factor. The dependent variables were *Selection Time* – defined as the time from the moment the stylus touches the tablet's surface until the moment the user selects

the target; and *Crossings* – defined as the number of times the selection line enters and leaves the target per trial (e.g., this value is equal to 1 when a participant does not overshoot the target).

Crossings gives us information about the degree of control shown by participants during a trial, as well as hints about their strategy during trials. For each experimental trial, we collected all the stylus, force button, and key data events. Also, since each trial can only be completed successfully, we end with a set of error-free selections. Participants were randomly assigned to six groups of two participants each. In each group, participants were exposed to all three *Technique* conditions, whose order of appearance was counterbalanced across groups to minimize ordering effects. For each *Technique*, participants were asked to complete four blocks each. Each block consisted of nine selection trials (3 *Distances* x 3 *Widths*), repeated 5 times. Presentation of trials within a block was randomized. In summary, the experiment consisted of:

- 12 participants ×
- 3 technique conditions ×
- 4 blocks ×
- 3 width conditions ×
- 3 distance conditions ×
- 5 repetitions = 6480 target selection trials.

Prior to performing the trials for each *Technique*, the experimenter explained to the participant how the Zlider worked with a particular technique. Then participants did a warm-up block of 45 trials to practice with the corresponding technique. Participants were instructed to perform the upcoming tasks as quickly and accurately as possible. While participants could take breaks between blocks, we enforced a 5 minutes break between techniques. A short ques-

tionnaire was administered at the end of the experiment to gather the participants' opinions (Appendix B).

5.6 Results

The experiment took an average of 1.25 hours per participant. A trial was considered an outlier when *Time* was beyond two standard deviations from the mean per participant. A total of 245 outliers (~3.8%) were removed from our analysis.

Table 5-1: Distribution of outlier data points across main experimental conditions.

	Large	Small	Smallest	Total
ForceButton	0.0%	0.0%	1.2%	1.2%
Keys	0.0%	0.1%	0.8%	0.9%
StylusPressure	0.0%	0.1%	1.5%	1.6%
Total	0.0%	0.3%	3.5%	3.8%

There were no main effects or interactions for the *Order* condition on either *Selection Time* or *Crossings*. Unless stated otherwise, the values and differences we do not report are not statistically significant.

5.6.1 Selection Time

As might be expected from Fitts' law, analysis of variance revealed a significant main effect on *Selection Time* for *Width* ($F_{2,14} = 392.8$, $p < 0.01$), and *Distance* ($F_{2,14} = 13.23$, $p < 0.01$). However, there were no significant main effect on *Selection Time* for *Technique* ($F_{2,14} = 0.31$, $p = 0.73$), or *Technique*Width* ($F_{4,28} = 1.53$, $p > 0.2$), and *Technique*Distance* ($F_{4,28} < 1$, $p = 0.6$) interactions. Also, post-hoc pairwise comparisons did not show any main effects between *Technique* for all levels of the *Width* condition. This is an interesting finding because we did not expect users to perform statistically similarly with such distinct techniques. Figure 5-17 illustrates these results.

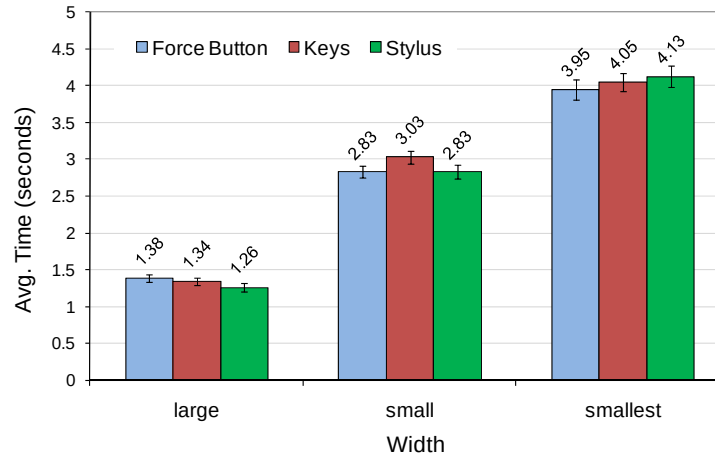


Figure 5-17: Average selection time per technique and width.

An analysis of *Selection Time* across experimental blocks (Figure 5-18) shows participants improving marginally as the experiment progressed for both the *Force Button* and *Stylus* conditions. For the smallest condition, participants' performance degraded and then recovered when using *Keys*, suggesting it may have taken longer for users using this technique to find a good strategy to complete a trial. Variations on the last experimental block suggest that fatigue effects may be present.

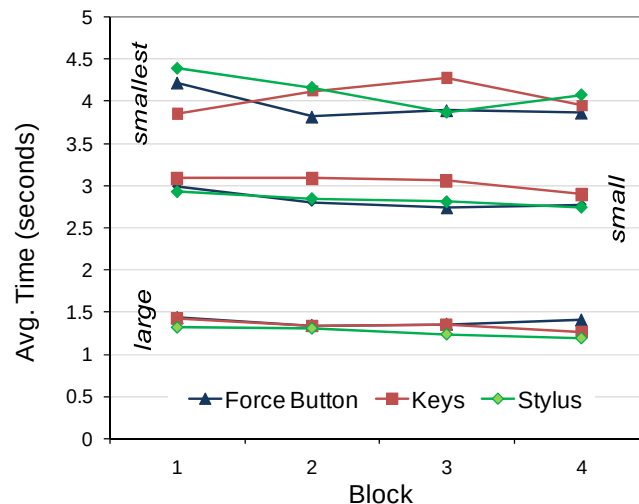


Figure 5-18: Average selection time per block and scale-adjusting technique.

5.6.2 Crossings

We found through empirical observations that participants crossed a target more than once for a number of reasons:

- They were sliding too fast and the target passed under the selection line without them noticing;
- They tried to acquire the target when it was visible, yet unreachable because the CD ratio was not high enough;
- Fluctuations in their control of the scale caused the target to move; and
- A combination of all of the above.

Our analysis shows a significant main effect for *Width* ($F_{2,4} = 357$, $p < 0.01$) on *Crossings*. Pairwise comparisons indicate that *Crossings* at *mid* distances were significantly higher than *Crossings* at *near* distances ($p < 0.03$). There was no main effect for *Distance* ($F_{2,4} = 5.46$, $p = 0.07$). However, we observe for the large condition that *Crossings* increase as distance decreases. Post-hoc comparisons indicate that *near* targets are crossed more often than *mid* ones ($p < 0.03$). Figure 5-19 illustrates these effects.

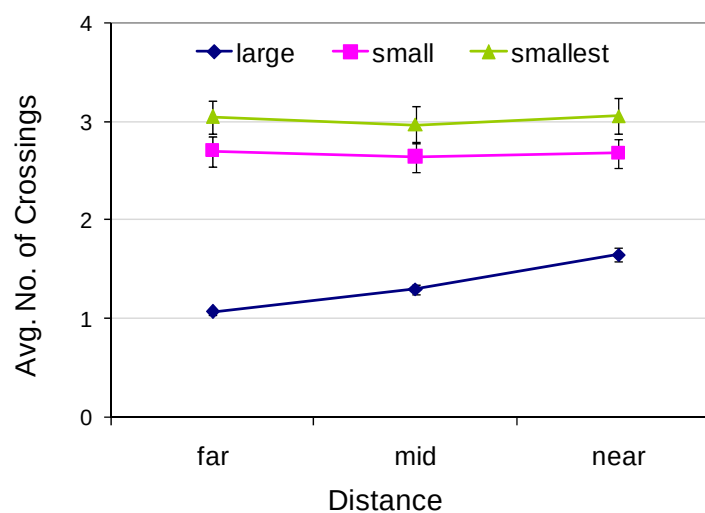


Figure 5-19: Average crossings per distance*width.

There was no main effect for *Technique* ($F_{2,4} = 2.24$, $p = 0.22$) on *Crossings*. However, analysis of variance shows a *Technique*Width* interaction ($F_{4,8} = 4.49$, $p < 0.04$) a closer inspection of the means shows *Stylus* resulting in fewer crossings than the other techniques for the *small* and *smallest* conditions. Figure 5-20 illustrates these effects.

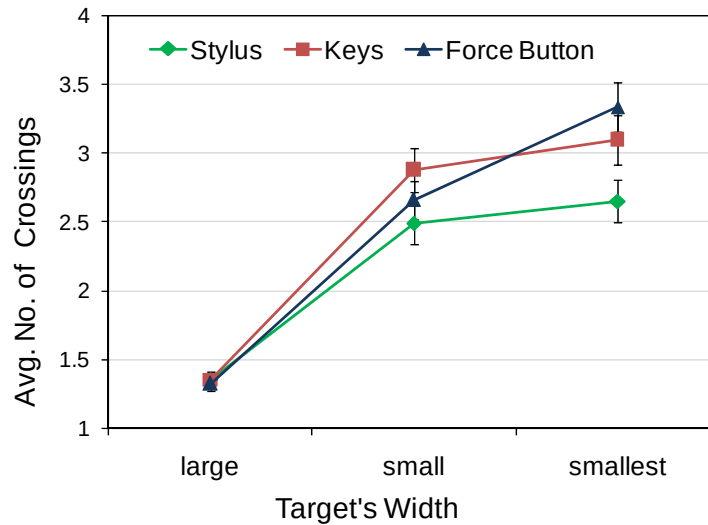


Figure 5-20: Average crossings per target's width and scale-adjusting technique. Note than 3 crossings correspond to overshooting a (selected) target once.

Figure 5-21 illustrates the number of crossings as the experiment progressed for each of the techniques. While participants do not seem to do better or worse with the *Stylus*, there is some improvement in participants' control for the *Force Button* with practice.

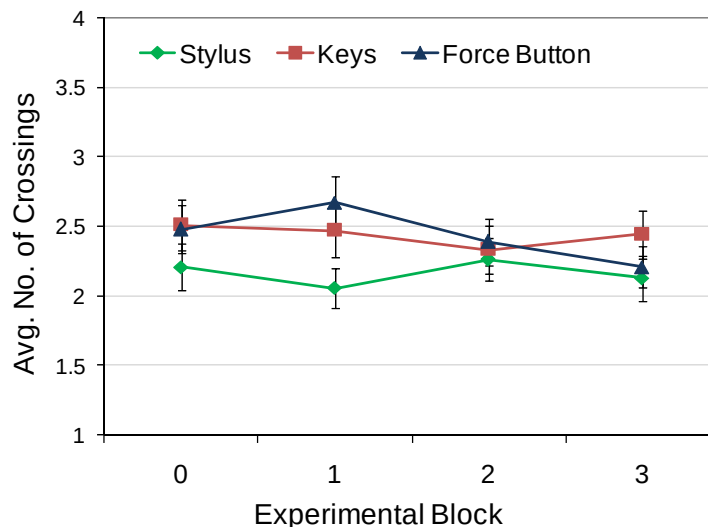


Figure 5-21: Average number of crossings per experimental block and scale-adjusting technique.

5.6.3 Qualitative Results: User Preferences

At the end of the experiment, we asked participants to rank each of the techniques presented to them (Appendix B). Their responses revealed mixed opinions. While some participants preferred and showed more skill with the *Stylus*, others preferred the *Force Button* or the *Keys*. This was interesting, as the experimenter heard how much a participant “loved” a technique not long after another participant expressed her dislike for it.

Six participants ranked *Keys* as their first preferable method to use and ten people as their 1st to 2nd choice. Users gave several reasons for their ranking, e.g., “it was easy to learn”, “it was simple to use”, “it was predictable”, “I could keep the scale stable”. However, more than half the people in that group expressed that for many scenarios they would probably like to use the stylus alone, because it requires “only one hand” and “does not need a keyboard”. It was not surprising to hear from some participants that “using the keys was slow”, although the quantitative data does not support this claim.

Only two participants ranked *Stylus* as their first choice and eight as their 1st to 2nd choice. Even though “it took longer to get used to” people expressed that once they “got it” the selecting task had a “cool fluid feeling” to it. While people in this group commented that it felt “quite natural” to zoom, they also expressed that it was challenging not to affect the pressure they are applying when sliding over long distances. This was a nuisance, if users did not want to alter the *Zlider*’s scale while browsing for the target. One participant expressed that using the stylus alone was “incredibly fast when the target area was on sight”.

Four participants ranked the *Force Button* as their first choice, and six did so as their 1st to 2nd one. There was a mixed set of responses for this condition. While some people exhibited very good control, others did not. As was observed with the *Keys* condition, people in this

group liked the fact that zooming and sliding were decoupled. However, people who did not like this condition complained about difficulties coordinating both zooming and sliding with separate hands. For example, we observed how users inadvertently accompanied the selection lifting action with a quick release of the *Force Button* as well.

5.7 Discussion

We had the opportunity to both assess a widget of our design, and to observe people using it to perform successfully selection and micro-adjustment tasks in sizes that ranged from the large to the almost infinitesimal. The fact that we found no significant differences in terms of average Selection Time for the three scale adjusting Technique conditions we studied is both unexpected and remarkable. This result shows that the Zlider's design can be used in different scenarios and hardware configurations without any performance degradation.

Nevertheless, our results prompt us to consider metrics other than Selection Time in order to identify if a Technique is preferable. Our analysis of the number of *Crossings per Technique* favors the *Stylus* condition, which results in fewer crossings for small targets. This conclusion is reinforced by the participants' qualitative feedback, which not only helps us identify what works well with the *Stylus*, but also what can be improved. A critical area of Zliding requiring improvement is supporting users' ability to zoom only when they want to. Our Zlider design supports this feature with its clutching mechanism and by making use of the tracking capability of its input device. However, it may be that Zliding needs to occur without an explicit physical area, or widget that can act as a clutching delimiter. Examples of these cases are a widget with no area, such as a crossing widget (Apitz & Guimbretière, 2004), or panning and zooming on a 2D map. It is then necessary to think of alternate strategies to achieve this design goal.

Since most of the undesired scale changes were observed while the user was dragging the stylus, one solution is to alter the rate at which the interaction's scale is allowed to change, based on the speed at which the stylus moves. This solution can include extreme cases such as disabling scale changes when the pointer moves above a certain threshold speed, or allowing scale changes only when the stylus is not moving in x-y space. Our current implementation is but a particular case of this general strategy. In addition, this solution provides users with an interaction style that models tasks that are purely serial (e.g., pan then zoom), purely coordinated (e.g., pan while zoom), or in-between.

5.7.1 Other Designs: The Zliding Wheel

The Zliding Wheel (Figure 5-22) operates by the same principle as a knob control with the exception that one can control the granularity of the wheel's increments. This control provides functionality similar to the Zlider's, but with a potentially smaller footprint and no boundaries on the parameter it controls. With the Zliding wheel, in addition to using the curvature of the arc being drawn to regulate the granularity, users can also adjust it through a degree of freedom other than a cursor's position, such as the pressure applied with a stylus input device. We consider two main variations of the Zliding wheel: a fixed version (Figure 5-22-a), and a floating one (Figure 5-22-b).

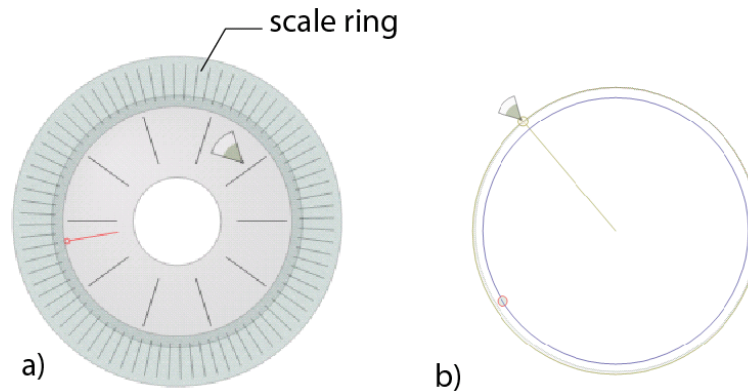


Figure 5-22: Zliding Wheels. (a) Fixed wheel remains in the same location on the display. (b) Floating wheel follows the cursor's position as the user performs circular motions on the interactive surface. A video illustrating this design can be found at <http://www.youtube.com/watch?v=EcE3XBytN-U> – timecode 03:33.

The fixed version (Figure 5-22-a) consists of a circular disk that users can rotate by scrubbing on its surface. A red needle inside the disk indicates the absolute rotation the disk is subjected to. The fixed Zliding wheel is very similar in its behavior to the Zlider widget: users can modify the wheel's granularity while they are rotating it, as well as access a clutching zone when they drag the pointer outside the disk's area. A scale ring is displayed above the wheel when its scale factor > 1 , and provides a “gear-like feedback”, which helps users understand the differences between the wheel's and the pointer's absolute motion.

The floating or “drifter” wheel (Figure 5-22-b) follows a pointer's circular motion, which is not necessarily centered on a fixed point, and works under the same principles of the control described by Moscovich (Moscovich & Hughes, 2004). This drifting is usually the result of the user not focusing visually on the control but instead paying attention to the changes the control causes. Because of this, the floating wheel has a lightweight visual design that consists of two concentric rings: an internal one that provides the absolute rotation the wheel is subjected to; and an external one that keeps track of the pointer's current motion. The floating wheel provides minimal feedback about its granularity by altering the thickness of its outer ring. As it stands, this design cannot incorporate a clutching zone, and because of this, it

would be appropriate to use the pointer's speed to determine when scale adjustments should be permitted.

It is worth mentioning that unlike other similar wheel controls, the Zliding wheel has the advantage of providing a way to adjust its granularity even in scenarios where it is not possible to move beyond the control's boundaries (e.g., a notebook's touchpad, or an iPod's scroll wheel).

5.7.2 Other Technologies, Other Directions

There are other technologies that have the potential to be used to zoom and slide. For example, capacitive touchpads are becoming widespread as a means of input for notebook computers, portable music players, and handheld devices. Even though capacitive touchpads have been traditionally used to sense a finger's position, some are capable of also sensing the amount of pressure that is applied to them. In other types of touch-sensitive surfaces, it is possible to estimate the contact area of a touching finger, thus estimating the applied pressure.

It is possible to leverage devices with decoupled continuous degrees of freedom such as Microsoft's Digital Media Pro keyboard or Wacom's Cintiq 21UX tablet, to facilitate the acquisition of very small targets in current GUIs. For example, in a fashion similar to the map application presented in (Hinckley et al., 1998), a user can apply scaling operations on a window's manager using a pointer's current location as a center of magnification. In this way, users can browse the GUI until a desired scale or CD ratio is reached. In a similar fashion, the "take-off" technique (Potter et al., 1988) can be greatly enhanced by Zliding. We imagine such interactions becoming commonplace in new environments that use resolution-independent graphics as their rendering engine and ZUIs.

Future work includes additional interactions and widget designs that can take advantage of Zliding to facilitate high precision manipulation tasks. In particular, we would like to see how crossing widgets (Accot & Zhai, 2002; Apitz & Guimbretière, 2004) can incorporate Zliding techniques. Finally, we are interested in studying, in the context of Zliding tasks, the degree of coordination people exhibit when using both coupled and decoupled input strategies for a variety of form factors and devices. Examples of these include capacitive touch pads, interactive surfaces like the Smartboard ("Smart Technologies Inc."), the Smartskin (Rekimoto, 2002) or the Diamondtouch (Dietz & Leigh, 2001), and devices with decoupled continuous degrees of freedom such as the Digital Media Pro keyboard (Mizobuchi & Yasumura, 2004) or Wacom's Cintiq 21UX tablet.

There is also potential for the Zlider control to have good performance in scenarios where screen real state is limited. Scrolling through large documents on a portable device such as a PDA or mobile phone is an example of one of these scenarios. In order to verify or challenge our claims, one needs to collect usability data comparing the Zlider control against other recent parameter selection approaches such as the OrthoZoom scroller (Appert & Fekete, 2006) or a virtual scroll ring (Moscovich & Hughes, 2004) under different form-factor conditions.

5.8 Summary

This chapter explores the use of pressure as a continuous parameter and it offers a novel interaction technique for arbitrarily high-precision parameter manipulation: *Zliding*. *Zliding* is an interaction technique in which one drags the input device to perform a parameter manipulation while at the same time one uses an input channel other than a pointer's x-y position to adjust the granularity, or scale, of the parameter space. We instantiate and study *Zliding* through a novel selector widget called *Zlider*, which users scrub in order to adjust a parameter

at a granularity that depends on the pressure applied through the pen. While the use of pressure sensed by a pen's tip allows for a zlliding experience on an integrated input device, it was of interest to find out that different input device configurations are comparable in terms of the time it takes to perform a task. Still, there are advantages in the use of the pressure-enabled pen such as better control over the interaction and its availability in scenarios where a keyboard is not available or accessible. A user commented on how performing a selection using only one hand was appealing to him and “felt right”.

Our experimental results not only show that zlliding enables people to perform high-precision parameter manipulations, but also suggest that one could the apply the zlliding concept beyond pen-based systems and to a diverse set of devices and form factors capable of direct, touch-based input. Our observations also provided anecdotal evidence of some concurrency taking place while users were zooming and sliding, but we did not quantitatively measure it. The following chapter presents and elaborates a technique defined by the concurrent variation in pressure applied while dragging a pen.

6 Pressure Marks

In pen-based interfaces, the choice of a command generally follows the selection of a group of objects. Despite a variety of instantiations, this *selection-action* (Hinckley et al., 2005) interaction pattern is typically serial in nature, i.e., the scope and command specifications occur in sequence, one after the other. This sequential nature not only makes the time necessary to perform a selection-action interaction at least equal to the sum of the time it takes to execute its parts, but it can also impose a sequential structure on potentially concurrent or integrated tasks (Jacob et al., 1994). Moreover, the nature of the delimiter between the selection and action components of the interaction can further increase the overall execution time. This chapter introduces and investigates *pressure marks* (Figure 6-1), pen strokes where the variations in pressure a user applies while drawing, or *pressure signature*, has meaning. These marks can potentially improve selection-action interactions by allowing the selection and action to be specified *concurrently*.

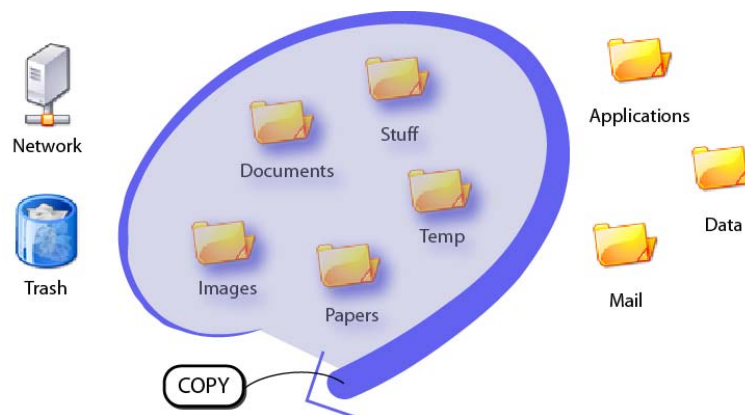


Figure 6-1: A pressure mark is used to select and copy (e.g., to a clipboard) a group of items in a single stroke. The selection is indicated by the enclosure of the stroke, while the command is specified by the pressure signature (thin-THICK) over the stroke. The selection and action components occur concurrently, with no delimiters between them.

PRESSURE MARKS

In addition to this characteristics, pressure marks have the potential benefit of providing pen gestures that are orientation independent, i.e., a command corresponds to a particular pressure signature (signature for short), rather than to a stroke direction or orientation. Thus, pressure marks can be useful in scenarios where the user's orientation relative to the display varies. Examples of such scenarios include cases where an artist rotates the underlying drawing canvas, or when one or more users interact with the same surface from different directions as commonly occurs in co-located collaborative tabletop environments.

Finally, pressure marks can be used to enhance traditional marking techniques, such as pie or marking menus (Kurtenbach & Buxton, 1993), which are most effective when the relative orientation of the screen and the user remains constant. For example, if one is to use a stroke's direction as part of a command, as with marking menus, pressure marks can potentially increase the number of available commands at a user's disposal.

Despite their potential, pressure marks present interesting design challenges. For example, in order for novice users to browse through a given set of available pressure marks, one has to produce visual designs that prompt them to interact through both pressure and x-y spatial movements. While browsing is straightforward when the interaction only uses x-y space, this is not the case when both x-y and pressure spaces are used concurrently. For example, items in a pie menu are laid out in different x-y positions making it easy for the user to glance at them. However, different pressure marks can have the same x-y spatial movement, making a visual representation that permits browsing non-trivial since overlapping icons may be confusing.

Providing an effective browsing experience for pressure marks is key both for assisting novice users in becoming experts and for allowing the exploration and discovery of otherwise

hidden commands. However, it is not obvious what the basic set of pressure marks should be, and how well users will perform when using such a set. Complex marks can potentially expand a system's interaction vocabulary, but they can also impose a steep learning curve and lead to high error rates. This chapter not only addresses many of these challenges, but also provides measurable evidence of the benefits of pressure marks benefits as a selection-action technique as well as offers a number of interaction designs that leverage the concept of pressure marks²⁰.

6.1 Previous Work

Pen-based systems generally offer users a *marking* and an *inking* mode. For the purposes of this chapter, we assume that non-dominant hand mode indication is present as researched by Li et al. (Li et al., 2005) and have chosen to concentrate on pen interactions while in marking mode. While selection-action patterns are traditionally sequential, there have been efforts to improve on this experience. Guimbretière et al.'s FlowMenus (Guimbretière, Martin, & Winograd, 2005) fluidly connect command selection and direct manipulation. FlowMenus consist of eight octants arranged around a central rest area. By entering and/or leaving the rest area and moving through the octants, a user can, in one continuous motion, navigate through the menu hierarchy, adjust a parameter or manipulate an object on a display. Figure 6-2 illustrates a flowmenu in action.

²⁰ Much of the material presented in this chapter was previously published in the Proceedings of the CHI '07 Conference on Human Factors in Computing Systems (Ramos & Balakrishnan, 2007).

PRESSURE MARKS

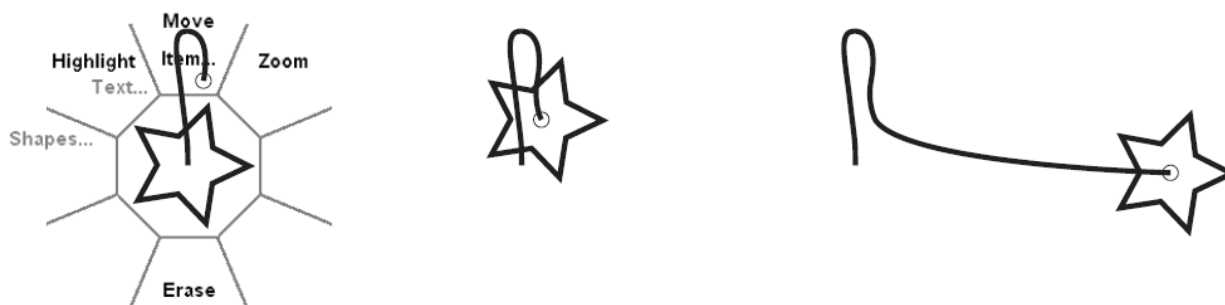


Figure 6-2: Flowmenu can integrate selection and direct manipulation in one continuous interaction phrase. From left to right: a user activates the menu, selects the move operation and proceed to manipulate the corresponding object with one connected stroke.

Similarly, Hinckley et al.'s pigtail delimiters (Hinckley et al., 2005) allow selection-action patterns to be performed in one continuous stroke. A user explicitly creates a pigtail by intersecting his/her own stroke and then uses the stroke's direction to specify an action or manipulate an object. Pigtails (Figure 6-3) provide a way to integrate an explicit command invocation in a continuous stroke following the selection specification. This is unlike previous selection-action schemes where users signal a command selector using buttons or timeouts.



Figure 6-3: This image illustrates how a pigtail delimiter not only marks the invocation of a contextual marking menu but also permits a command specification in one continuous stroke.

Saund and Lank (Saund & Lank, 2003) present a technique that guesses the user's intent by using the stylus' trajectory and context. While in some cases this protocol does not need an explicit command, the system presents a selector widget if the stroke drawn is ambiguous. These techniques remain sequential – i.e., a delimiter separates selection and action.

Previous research suggests that interactions where parallelism occurs can outperform sequential tasks. For example, researchers have shown how bi-manual interaction techniques permit

PRESSURE MARKS

parallelism (Buxton & Myers, 1986) as well as outperforming (Latulipe, Kaplan, & Clarke, 2005) one-handed ones. Baudisch et al.'s marquee menus (Baudisch, Xie, Wang, & Ma, 2004) are a technique where the selection-action pattern occurs concurrently. The marquee menu's selection is specified by the rectangular area defined by a straight stroke and its action is determined by drawing the stroke in one of four directions (Figure 6-4). While providing a compact interaction phrase, this type of menu is sensitive to both a mark's point of origin and direction. Although promising, the authors did not elaborate on if and how this technique scales for non-straight strokes with arbitrary orientations, or for larger command sets.

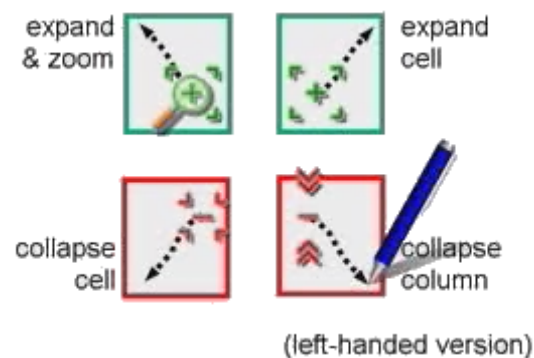


Figure 6-4: Marquee menu's four types of commands. Each command's stroke follows a fixed direction. The control of pressure has the potential to be used concurrently with spatial movement. Srinivasan and Chen (Srinivasan & Chen, 1993) presented evidence that when provided appropriate visual feedback, users can control variations of pressure over time. In their study, participants applied force to a pressure sensor using the pad of their index finger. Their results however, do not include situations where the applied force changes as the user's finger slides over a rigid surface. Our own work with Zliding presented in the previous chapter also explore integrated panning and zooming by concurrently controlling input pressure while sliding in x-y space. While we provide insight on the issue of concurrent input modalities, zliding

PRESSURE MARKS

does not integrate spatial movement and pressure input for concurrent selection-action operations.

While the use of marks for both selection and action patterns in the GUI is not new, there has not been significant examination of marks capable of concurrent selection-action. Similarly, while there have been efforts supporting the use of an interface from different orientations, they often assume a system that senses a user's position, or has preordained rules of engagement – i.e., regions of the display are meant to be approached from a particular direction (Fitzmaurice, Balakrishnan, & Kurtenbach, 1999; Shen, Vernier, Forlines, & Ringel, 2004; Wu & Balakrishnan, 2003). Kara and Stahovich (Kara & Stahovich, 2004) use polar coordinates to recognize gestures independently of their orientation in the context of their Simus-ketch system. However, some gestures such as the marks in hierarchical marking menus are ambiguous when seen from different orientations. The exploration of interactions that do not depend on how the user approaches the interface deserves further exploration. This exploration has the potential to benefit the usability of both portable small displays such as those on PDA's and larger form factors such as tabletop displays.

6.2 Pressure Marking

Through history, people have used hand-made marks both as a channel of artistic expression and as a way to encode information. In particular, the written word is probably the predominant examples of this type of expressivity. These marks can be made by someone's bare hands and with instruments such as chisels, quills, brushes and pens. They are a testament to the fine pressure control achievable by the human hand. However, people's skills with a stylus can vary dramatically from one person to the next. To be successful it is thus important to design a simple and significant set of signatures that people can perform effectively.

PRESSURE MARKS

Writing or drawing a predetermined stroke, such as a pressure mark, can be described as an open-loop task, as it was early reported by Woodworth and reviewed by Elliot et al. (Elliott et al., 2001). In open loop tasks, people develop a motor program, or plan, and later execute it without using sensory feedback. Open-loop tasks can be very fast once people start them, so fast that their sensory system cannot keep up with all the events that occur during it. Because of this perceptual limitation, once the task execution has started it is very difficult to modify the task's goal. However, a person's sensory feedback can affect the task at the planning stage or during execution if the speed at which the task is reduced to below a certain threshold. Building on this information and acknowledging that people's skills vary, we use a set of design guidelines for pressure marks:

- *Small and Simple*: we will aim for a small number of pressure signatures. In addition, signatures should be simple enough so an average user can do them.
- *Continuous Feedback*: appropriate visual feedback should enable users to be aware of what they are doing and the effect pressure has on the interaction.
- *Browsable*: a mechanism for browsing through the available pressure marks should be available.

We initially consider two basic signatures when a mark is drawn: a) pressure remains constant within a certain margin of variance, and b) pressure changes. We can then increase the number of signatures by a factor of two by considering constant pressure at low or high levels, and pressure changing in a monotonically increasing or decreasing way. This process produces a set of four basic signatures (Figure 6-5 left): low-low (LL), low-high (LH), high-high (HH) and high-low (HL).

PRESSURE MARKS

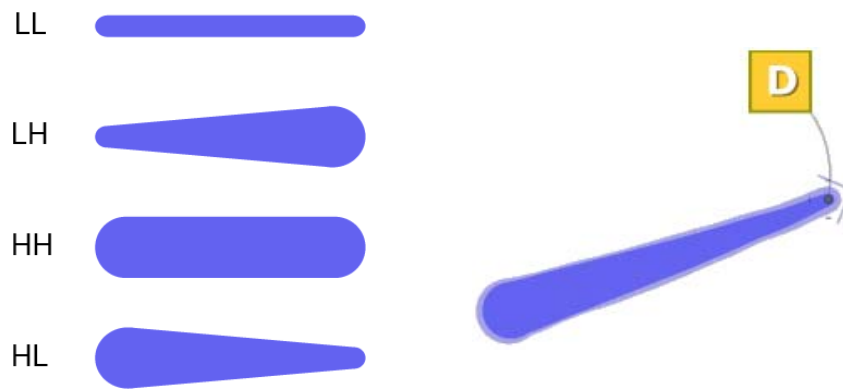


Figure 6-5: (left) Profiles of the four proposed pressure signatures. (Right) User-made stroke with recognized profile overlaid on top. The icon “D” represents some action associated with the HL profile. A video illustrating these marks can be found at <http://www.youtube.com/watch?v=qR2mKwkATpk> – timcode 0:33.

While it is possible for users to draw pressure marks with signatures that do not match exactly either of these four classes, we will later describe a way in which we can reduce any pressure mark into one of these four classes.

In order to give meaningful information to users as they create a mark, we use different types of feedback. First, we use a pressure cursor similar to the one we present in the previous chapter in order to provide continuous information about the pressure being applied through the stylus. Second, we draw a stroke whose width at any point is proportional to the amount of pressure applied at that point. This last feedback is analogous to strokes produces by brushes in Japanese sumi-e paintings or Chinese calligraphy, which are a familiar feedback for a great number of people. For example, the Virtual Heritage system (Song, Elias, Müller-Wittig, & Chan, 2005) uses a Chinese brush and the marks it makes on a virtual slate as an input device to enable users to navigate through a virtual reality (VR) scenario.

Finally, when the user lifts the stylus from the display’s surface (i.e., the mark is finished) a stylized semi-transparent representation of the pressure mark is overlaid on it, confirming the interaction (Figure 6-5 right). We call this stylized visual representation of a pressure mark its *profile*. We use a drawing pause-timeout delimiter to enter a *browsing mode* where novice

PRESSURE MARKS

users can browse through the available set of pressure marks. Many selection-action scenarios use delimiters in this way and in our case, it reverts pressure marks into a standard sequential pattern.

6.2.1 Browsing Through Pressure Marks

During the early stages of our research, we explored different designs for this browsing mode. We first considered a design where available profiles of the pressure marks appear at different locations in x-y space. While this design has the advantage of showing at-a-glance all the available options, pilot studies showed that it also elicits an unwanted response from users in that users often drew marks towards the different profiles as if they were targets in a spatial menu. People exposed to this browsing mode were confused, as they were not sure what it meant and they often treated it as a spatial menu. In the end, this type of design did not facilitate the exploration of available pressure marks. In light of the unwanted response we got from users to these “spatial browsing” designs, we developed an alternate mechanism for browsing pressure marks.

The browsing mechanism we propose exploits the property that each of the four proposed pressure signatures can be connected like domino pieces – e.g., a user can “draw a stroke” that passes through all four signatures: ... LL → LH → HH → HL → LL ... These “connections” allows us to suggest follow-up marks when the browsing mode is active (Figure 6-6).

In particular, our implementation draws profiles of signatures that the user can execute next and that differ from the mark currently in progress. These suggested profiles follow the last known direction of the mark in progress. In addition to drawing the profiles of potentially different marks, we draw a dashed line that indicates an available “cancel” crossing gesture. This cancel gesture is simply a self-intersecting pigtail at the end of the mark.

PRESSURE MARKS

This browsing mechanism follows a quasi-sequential way of exploring the set of available pressure signatures. While spatial browsing can elicit unwanted responses from users, sequential browsing can be tedious and time consuming. We find that our proposed type of quasi-sequential access is a good compromise between the random access that full spatial browsing provides, and purely sequential exploration. This way of revealing otherwise hidden gestures pushes Scriboli's stroke extension (Hinckley et al., 2005) further and addresses the issue of revealing more than one available gesture. In contrast to the original "spatial layout" approaches we considered, informal user feedback on our quasi-sequential browsing mode has been very positive. The question of how useful this browsing mechanism is for first-time users merits exploration, and is an exciting topic for future research.

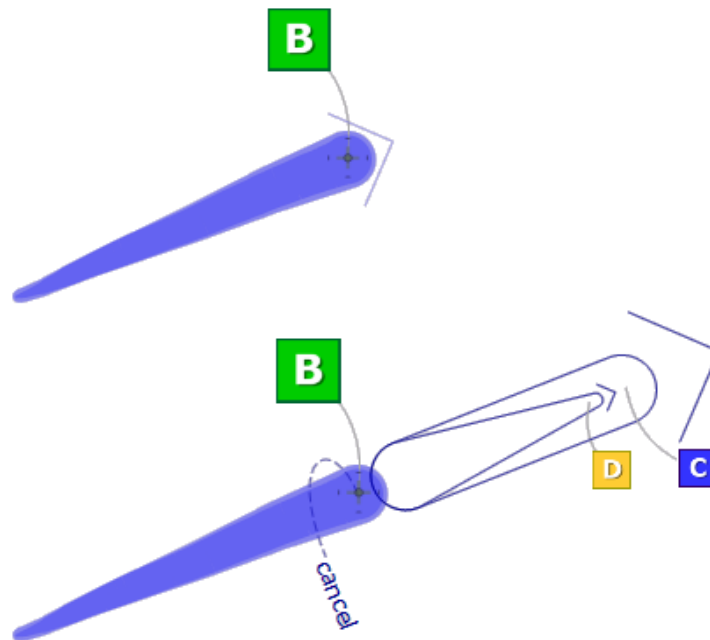


Figure 6-6: Browsing through pressure marks. A pause-timeout delimiter while marking (top) triggers a browsing mode. We suggest available pressure profiles from that point (bottom). The boxed letters represent actions associated with the profile they are connected to. The dashed line (bottom) indicates that pig-tailing at that point cancels the mark. A video illustrating this design can be found at <http://www.youtube.com/watch?v=qR2mKwkATpk> - timecode 00:47

6.2.2 Pressure Marks' Anatomy: Reduction and Parsing

The four basic types of pressure marks we propose do not capture all possible signatures that a user can make. What do we do if a mark's signature does not fit our prescribed types? There are three options:

- a) Treat the mark as null, as if nothing happened;
- b) Treat the mark as ambiguous and solicit additional feedback to resolve the ambiguity; and
- c) Reduce the mark into one of the four types.

After conducting preliminary heuristic evaluations at the early stages of our research, we decided in favor of the third option because it allows us to provide a response that better matches users' expectations. For example, users who made a mark wherein pressure first increases and later decreases often expected the system to recognize a HL signature.

Analysis of a pressure mark consists of both detecting the selection and the action that it encodes. The selection part of a mark, which determines which UI object(s) are selected, depends on the mark's shape – e.g., straight line, lasso, etc. As this is a well-studied topic (Hinckley et al., 2005; Hinckley, Guimbretiere, Agrawala, Apitz, & Chen, 2006; Mizobuchi & Yasumura, 2004), we will concentrate instead on how signatures are parsed and reduced. We performed parsing over the curve defined by the distance traveled by the stylus and the pressure applied through it. Figure 6-7 shows an example of such a curve.

PRESSURE MARKS

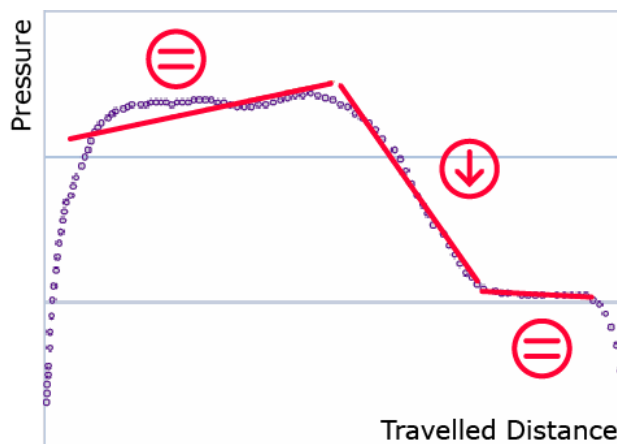


Figure 6-7: Pressure vs. traveled distance. The red segments show fitted lines on the pressure curve. In this case, the curve is parsed as a HL (↓) mark.

We first discard several data points at the ends of the curve to account for noise from the stylus engaging with and disengaging from the digitizing surface. We later analyze the resulting curve and simplify it using a piecewise linear approximation. This scheme tries to fit a straight line on the curve, and then, if the error is above a certain threshold, it divides the curve at the point of maximum error. Later, the algorithm recursively finds a linear fit to those pieces. The analysis stops when it meets a convergence threshold.

For a given signature, this analysis produces a set of n straight lines; each with characteristic features such as length, % length, slope, and pressure change. We use these features to label each line as being constant (=), ascending (↑) or descending (↓). The labeling takes into consideration how people perceive variations in pressure at different levels, a phenomenon that can be described by the Weber-Fechner Law (Boothe, 2001; Gescheider, 1997; Stevens, 1975). Finally, we classify a signature according to its last observed ↑ or ↓ label as LH or HL, respectively. If we observe no ↑ or ↓ labels, we use a simple threshold to classify the signature as LL or HH. Listing 6-1 outlines the algorithm used to parse a pressure signal.

Our anecdotal observations reveal that our reduction-parsing technique performance is affected under the presence of extreme variability among different users. Still its straightfor-

PRESSURE MARKS

ward implementation and overall robustness make it an attractive parsing technique. For improved performance one can imagine using other parsing schemes that utilize a training set of gestures, such as Rubine's (Rubine, 1991). Also, it might be interesting in the future to explore using the speed or tilt variations of the pen for command specification, however these properties may be less suitable than pressure since speed is highly variable across users and tilt is orientation dependent.

```

//Function Tokenize
//Input
//    signal:
//    vector containing
//    x, y, pressure and time values of a mark
//    (global signal comes trimmed at the extremes
//    to remove sharp artifacts - i.e., first touch, lifting)
//
//Output
//    token:
//    string describing profile of pressure signal

Tokenize(signal) {

    line = fit straight line on signal

    // r_threshold ~ 0.95
    if (line.r_value > r_threshold)

        // find the point i that maximizes the
        // difference between line and signal
        // (as an estimate of inflexion point)

        i = maxj(Abs(line.point[j] - signal[j ]));

        // divide signal in two pieces (recursion)
        left = Tokenize(signal[0,i-1]);
        right = Tokenize(signal[i, signal.length]);
        token = left + right;
    } else {
        if (line.fit < r_threshold)

            // JND ~ 10%
            if (line.change >= JND) {

                if (line.slope > 0)
                    token = "↑";
                else
                    token = "↓";

            } else {

                // line.change < jnd
                token = "=";
            }
    }
    return token;
}

```

Listing 6-1: Pseudo-code describing the pressure marks' parsing algorithm. The *Tokenize* function takes an array of signal points and returns a string describing the signal's features. For example, for the signal illustrated in Figure 6-7 *Tokenize* would return the string "↑↓=".

6.3 User Study #1

It is important to determine if users can successfully execute pressure marks using the proposed set of signatures. We also wish to assess how well our reduction-parsing technique performs. Favorable answers to these issues will provide evidence in support of pursuing interaction designs that leverage the use of pressure marks. In order to answer these questions, as well as to gain usability information about pressure marks in general, we performed an exploratory quantitative study, which we describe in this section.

6.3.1 Apparatus

We used a Toshiba Portégé M200 TabletPC running Windows XP Tablet Edition, with a 1400 by 1050 pixel display at ~140 dpi. Participants used the TabletPC in slate mode and interacted with it using a wireless stylus that has a pressure-sensitive tip²¹ that provides 1024 levels of pressure. The tablet was set flat on a desk, but we allowed participants to adjust the tablet's position on the desk, according to their preference.

6.3.2 Participants

Sixteen people (eleven male, five female), 18-44 years old, recruited from our university population through e-mail lists, participated in the study. One person was left handed. All participants had normal or corrected-to-normal vision and had some familiarity with the TabletPC. 93% of the participants had used a TabletPC or similar device (such as a PDA). Also, 53% considered themselves as having some drawing experience. Participants were recruited from the University of Toronto population using e-mail and flyer advertisings. We provided no compensation.

²¹ This digitizer tablet responded to forces between zero and 300gf.

6.3.3 Task and Stimuli

We studied two styles of marks: straight lines and lassos. For each mark's style, we also considered three possible lengths and various different orientations. For the straight line case, we asked participants to draw lines of different *length*: *small*, *medium* and *large* in four possible *orientations*: *east (E)*, *west (W)*, *south (S)* and *north (N)*. For the lasso case, we also asked participants to draw lassos of different *length*: *small*, *medium* and *large*, but in eight possible *orientations*: *clockwise north (CN)*, *south (CS)*, *east (CE)*, *west (CW)* and *counterclockwise north (CCN)*, *south (CCS)*, *east (CCE)*, and *west (CCW)*.

For each experimental trial, we presented as stimulus a stylized representation of the pressure mark users should make and dashed lines that users should cross in order to complete marks of a particular length. Figure 6-8 illustrates the stimulus for straight lines and lassos. In the case of lassos, we also showed a gray circle that indicated the particular object that users had to lasso. Start and stop icons, showing where a mark should start and end respectively, reinforced the task's orientation.

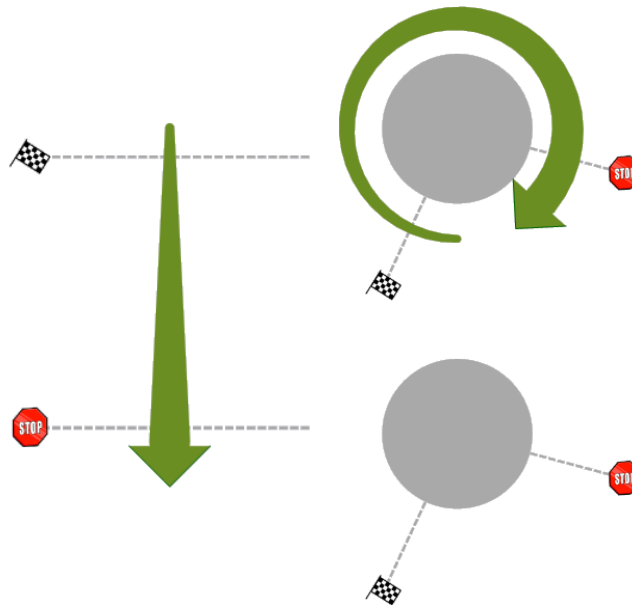


Figure 6-8: An example of experimental stimuli for the straight line (left) and lasso (right) cases.

PRESSURE MARKS

Each trial ran the same way for both the straight line and lasso cases. After the stimulus was displayed, users were required to draw a pressure mark with a profile and trajectory similar to the one displayed. After the mark was completed, we tested for two conditions: a) that the mark's trajectory is similar to the stimulus' – i.e., dashed lines are crossed in the right order; and b) that the mark drawn by the user is parsed as one with a similar signature as the stimulus. While failing either test causes an error sound to be played and the trial to be repeated; only failing b) was counted as an error since our primary goal was to determine user ability to generate the given pressure signatures and not their ability to draw straight lines or lassos per se. The mark's browsing mode was disabled for this study.

6.3.4 Procedure and Design

For straight marks, we used a 3 length (*small* – 2cm, *medium* – 4cm, *large* – 8cm) $\times$ 4 orientation (*N*, *S*, *E*, *W*) $\times$ 4 signature (*LL*, *LH*, *HH*, *HL*) within-subjects design. For lasso marks, we used a 3 diameter (*small* – 2.5cm, *medium* – 4cm, *large* – 6.5cm) $\times$ 8 orientation (*CN*, *CS*, *CE*, *CW*, *CCN*, *CCS*, *CCE*, *CCW*) $\times$ 4 signature (*LL*, *LH*, *HH*, *HL*) within-subjects design. For both types of marks, the dependent variables were *trial time* and *errors*. We computed trial time as the time elapsed between the moment a participant touches the tablet's surface with the stylus after a trial's stimulus was presented and the trial's successful completion. A trial was erroneous if the system could not match the user's input with the stimulus. Since one could only advance to the next trial after being successful in the preceding, participants were motivated to perform well. For straight lines, participants completed three blocks of trials. Each block consisted of 48 marking tasks repeated twice. Presentation of trials within a block was randomized. For lasso marks, participants completed two blocks of trials. Each block consisted of 96 marking tasks repeated twice. Again, presentation of trials within a block was

PRESSURE MARKS

randomized. The difference in the number of blocks between the lines and the lasso cases is due to our intention to keep the total number of trials per case roughly similar. All users did the straight line case first, followed by lassos so as to present cases in an increasing level of complexity. In summary, the experiment consisted of:

$$\begin{aligned} &16 \text{ participants} \times (\\ &\quad (3 \text{ blocks} \times 3 \text{ lengths} \times 4 \text{ orientations} \times 4 \text{ straight signature marks} \times 2 \text{ repetitions}) + \\ &\quad (2 \text{ blocks} \times 3 \text{ lengths} \times 8 \text{ orientations} \times 4 \text{ lasso signature marks} \times 2 \text{ repetitions}) \\ &\quad) = 10752 \text{ trials.} \end{aligned}$$

Prior to the study, the experimenter explained the task to the participants. Before each type of mark was presented, participants practiced with two warm-up blocks of 48 trials. The experimenter then told participants to do the trials as quickly and accurately as possible. Participants completed a questionnaire at the end of the experiment (Appendix C).

6.3.5 Results

This study averaged 1 hour per participant. For the straight lines case, we conducted a 3 (block) $\times$ 3 (length) $\times$ 4 (signature) repeated measures analysis of variance (RM-ANOVA) on the logarithmically transformed trial time and on the errors. For the lasso case, we conducted a 2 (block) $\times$ 3 (length) $\times$ 4 (signature) RM-ANOVA, also on the logarithmically transformed trial times and on the errors. The logarithm transform corrects for the skewing present in human response data, and removes the influence of outliers. Unless stated otherwise, the values and differences we do not report are not statistically significant.

6.3.5.1 Errors

In this study, we are most interested in error rates across blocks of trials because they will give us information as to learning effects, as well as an indication of the performance of our

PRESSURE MARKS

reduction-parsing algorithm. Figure 6-9 illustrates how, for both straight lines and lassos, error rates decrease as the study progresses until it reaches levels of about 4%.

There was no main effect for blocks in the non-warmup experimental trials, both for straight lines ($F_{2,30}=2.00$, $p=0.15$) and for lassos ($F_{1,15}=3.99$, $p=0.06$). However, when we examine this data in the context of the warm-up data, we observe a marked improvement in the users' accuracy as they progress through the entire experiment.

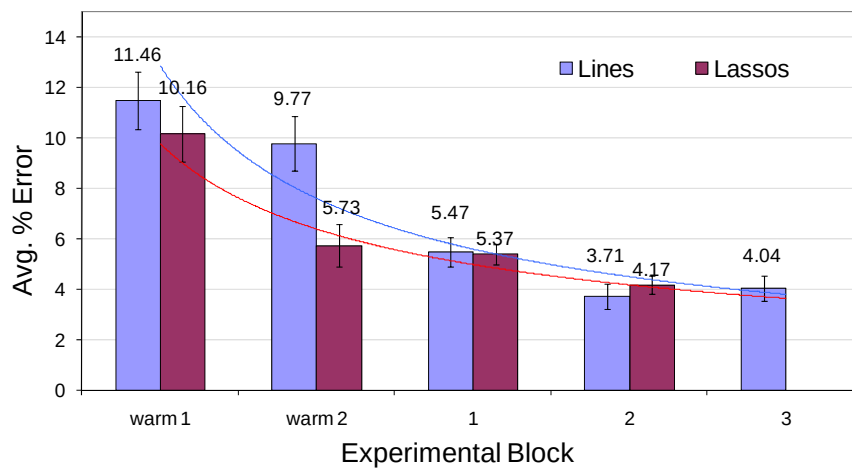


Figure 6-9: Average errors per block for straight lines and lassos. Power regression lines suggest the presence of learning taking place.

We found a main effect for signature in the case of straight lines ($F_{3,45}=3.39$, $p=0.02$) and lassos ($F_{3,45}=8.82$, $p<0.01$). Participants were most accurate when performing a LH line or lasso. Unlike the case of straight lines where accuracy was similar for LL, HH and HL, with lassos people made more mistakes when trying to maintain constant pressure, especially the HH mark. Figure 6-10 illustrates these results. This is consistent with the observations we do in both chapters 4 and 5 about users maintaining a constant level of pressure.

PRESSURE MARKS

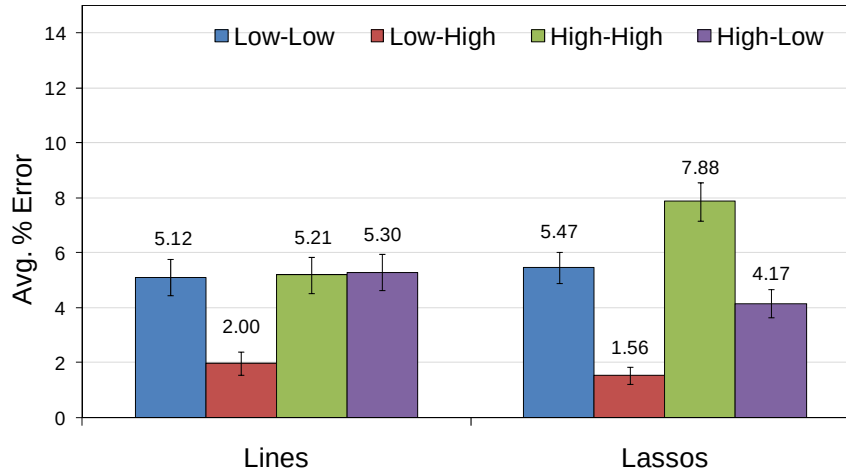


Figure 6-10: Average errors per pressure signature for lines and lassos.

6.3.5.2 Trial Times

For straight lines, we found a main effect for blocks ($F_{2,30}=20.9$, $p<0.01$), where users performed faster as the trials progressed. For lassos, there were no main effects for blocks ($F_{1,15}=0.23$, $p=0.63$). This was probably due to users' prior exposure to the straight lines case. However, as Figure 6-11 illustrates, we still observe a small speed improvement. We also found a main effect for signature in the case of straight lines ($F_{3,45}=31.02$, $p<0.01$) and for lassos ($F_{3,45}=16.78$, $p<0.01$).

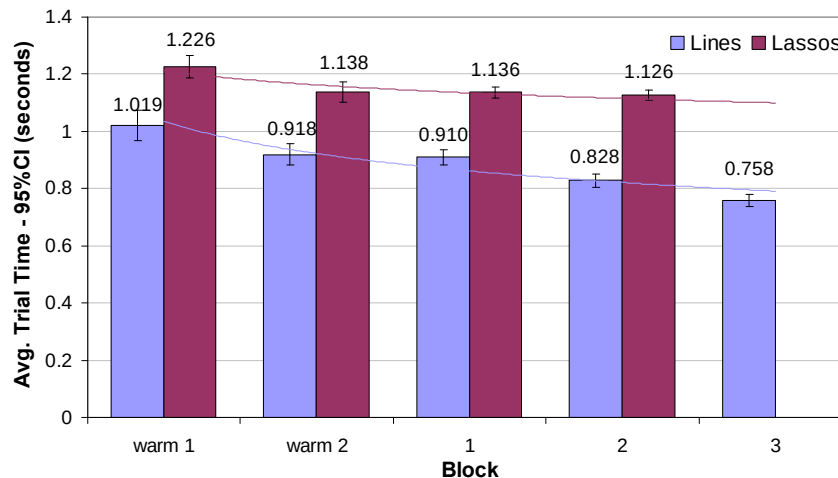


Figure 6-11: Average Trial Time per block for lines and lassos. Power regression lines are shown.

PRESSURE MARKS

Bonferroni-corrected pairwise comparisons reveal significant differences between all signatures ($p < 0.01$), except between LH and HH ($p = 1.0$) for straight lines. A similar trend was seen for lassos where all pairs of signatures were significantly different ($p < 0.05$) except for LH and HH ($p = 0.74$). Figure 6-12 illustrates this.

We can explain these differences by reflecting on the nature of the pressure profiles of the four signatures. LL was the fastest. It requires users to keep pressure relatively constant once the tip of the stylus touches the screen. Users commented on how easy it was to perform LH marks. We observed how they did them in an almost ballistic way. Once the stylus touches the screen, users start increasing pressure as they drag the stylus on the screen's surface. It took users almost the same time to do HH marks as to make LH marks. For the HH case, users reached a high level of pressure in a ballistic way, before dragging the stylus. Our observations showed that when drawing lassos keeping pressure constant was more challenging than it was when drawing straight lines. While users took the longest to do HL, they did achieve good levels of accuracy. This is consistent with general user feedback where users describe the HL mark as the most difficult to perform of the four pressure signatures.

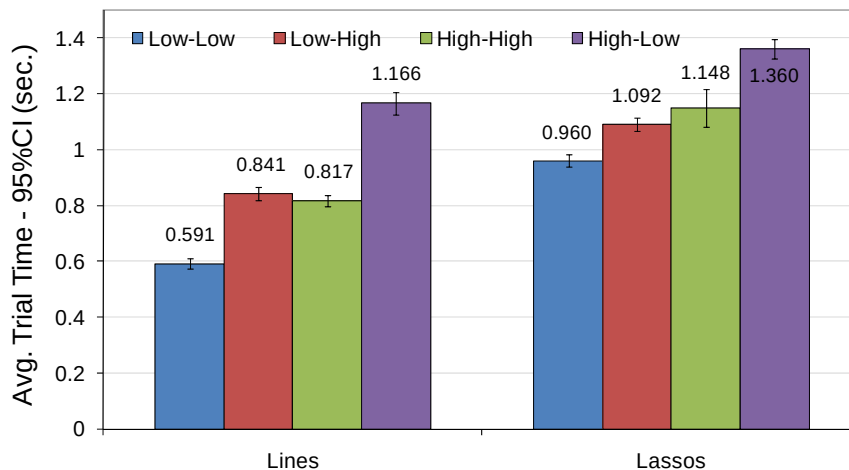


Figure 6-12: Average Trial Time per pressure signature for lines and lassos.

PRESSURE MARKS

Qualitative results are consistent with our experimental observations. Users rated in a scale from 1 (strongly disagree) to 7 (strongly agree) on the ease of use of the different marks. On average, LH (6.3) was rated between agree and strongly agree; HH (5.4) and LL (5.2) between somewhat agree and agree; and HL (4.6) rated least easy, between neither agree/disagree and somewhat agree.

As we might expect from Fitts' law, there was a main effect for length for lines ($F_{2,30}=191.73$, $p<0.01$), as well as for lassos ($F_{2,30}=212.09$, $p<0.01$).

6.3.6 Summary

The results from this study provide encouraging evidence supporting pressure marks as a viable interaction technique – i.e., people both can learn how to use them and are able to perform them accurately. Moreover, the initial set of pressure signatures we propose seems adequate, albeit with improvements required to our reduction-parsing algorithm. In particular, our heuristics seem to be sensitive to pressure variations while doing a LL or HH mark.

6.4 User Study #2

A concurrent selection-action technique like pressure marks has the potential to produce faster interactions than sequential techniques. However, it is not clear whether the increased complexity inherent in concurrently controlling both pressure and spatial x-y positioning would negate the benefits of concurrency. Therefore, we wanted to gather data as to the performance of pressure marks in comparison to a fluid serial selection-action technique. Accordingly, we ran a study that delves further into the use of pressure marks and contrasts its performance with lassoing + pigtail2 (LP2) (Hinckley et al., 2005), one of the latest state-of-the-art pen-based serial selection-action technique available to date.

6.4.1 Apparatus and Participants

For this contrast study, we used the same apparatus as in the first study. Fourteen people (nine male, five female), 18-44 years old, recruited from our university population through e-mail lists, participated in the study. All participants had normal or corrected-to-normal vision. None of these fourteen people participated in the first study. No compensation was provided.

6.4.2 Task and Stimuli

For this study we used an experimental task similar to the one used by Hinckley et al. (Hinckley et al., 2005). Users were asked to lasso (i.e., select) elements in a selection region (either a *single* square or a full row/column, i.e. *multiple* squares) and apply the correct action to the selected elements using either a pigtail menu or a concurrent pressure mark. The selection region consisted of 9 squares arranged in a 3 x 3 grid. The squares' size and spacing in our study were chosen to match the experimental setup in Hinckley et al. (Hinckley et al., 2005). Figure 6-13 illustrates an example of the experimental task and stimuli.

For each trial, we highlighted the squares to be selected in bright green and indicated the action to be taken by displaying the word “North”, “South”, “East” or “West” for LP2 and “thin-thin”, “thin-THICK”, “THICK-THICK” or “THICK-thin” for pressure marks. We chose this type of text stimuli instead of a graphic one, because we did not want to impose on users any prescribed way to lasso the targets. Also, we showed above the text stimuli the icon that corresponded to the action users had to apply. In our study, the icons were the letters “A”, “B”, “C” and “D” framed inside a colored box (Figure 6-13). After the stimulus was displayed, users were required to lasso the green squares and to indicate the requested action.

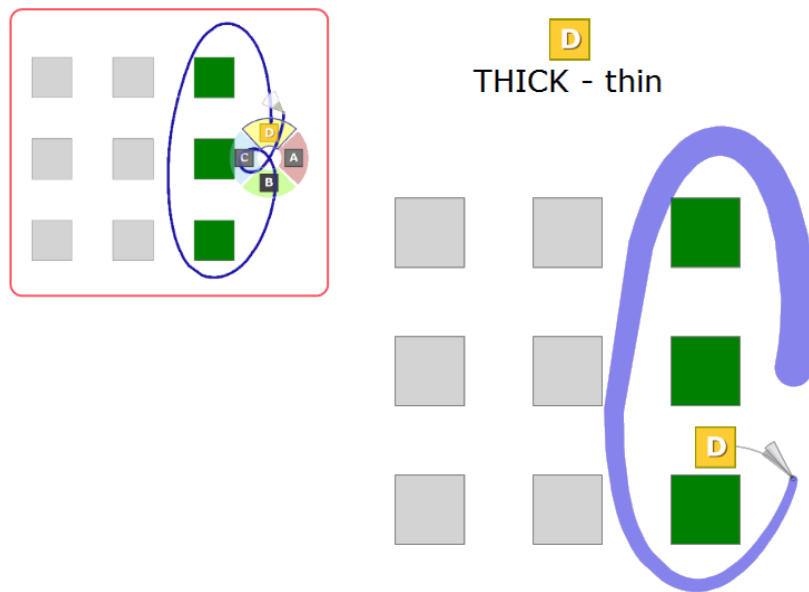


Figure 6-13: (right) Example of an experimental trial for the pressure marks condition. (Left) Scale-down version of similar trial for the Lasso+Pigtail2 condition –not shown in the study. A video displaying an example of what the trials look like can be found at <http://www.youtube.com/watch?v=qR2mKwkATpk> – 02:07.

After users completed the selection-action pattern, we tested for two conditions: a) that the lasso included all of the green squares and no distractors; and b) that the action performed matched the one presented as stimulus. While failing either test caused an error sound to be played and the trial to be repeated; only failing condition b) was counted as an error since our primary goal was to study user ability to lasso some number of targets and specify a command, rather than their ability to lasso perfectly a given number of targets per se. A target was inside a lasso if the target's center was inside it. The mark's browsing mode was disabled for this study.

6.4.3 Procedure and Design

We used a 2 technique (*pressure mark*, *lasso+pigtail2*) $\times$ 2 selection type (*single*, *multiple*) $\times$ 6 selection (targets) $\times$ 4 action (*N*, *S*, *E*, *W* for LP2 and *LL*, *LH*, *HH*, *HL* for pressure marks)

PRESSURE MARKS

within-subjects design. For *multiple* selection tasks, the selection was always 3 contiguous squares randomly selected as a row or column (Hinckley et al., 2005).

The dependent variables were *trial time* and *error*. Trial time was the time between the moment the stylus touched the tablet's surface after a trial's stimulus was presented and the trial's successful completion. A trial was erroneous if the user lassoed the targets, but performed an incorrect action. Since one could only advance to the next trial after completing the preceding one, participants were motivated to perform well. We divided participants in two groups, according to the order in which techniques were presented to them (pressure marks first or LP2 first). This order was included as a between-subjects factor. For each technique, we asked participants to complete three blocks of trials. Each block consisted of 48 selection-action tasks repeated twice. Presentation of trials within a block was randomized. In summary, the study consisted of:

14 participants ×
2 techniques ×
3 blocks × 2 selection types × 4 lasso signature marks ×
6 tasks ×
2 repetitions = 8064 trials.

Prior to the first use of a technique, we explained to participants the nature of the task. Participants practiced with two warm-up blocks of 48 trials. We also instructed participants to be as quick and accurate as possible.

6.4.4 Results

This study averaged 1 hour per participant. We conducted a 2 (technique) × 2 (block) × 2 (selection type) RM-ANOVA on the logarithmically transformed trial times and on the errors. The logarithm transform corrects for the skewing often present in human response data, and

PRESSURE MARKS

removes the influence of outliers. The presentation order of the techniques had no effects on the trial times or the errors. Unless stated otherwise, the values and differences we do not report are not statistically significant.

6.4.4.1 Trial Times

There was a main effect for technique ($F_{1,11}=18.22$, $p<0.01$), with pressure marks being an average of 320 milliseconds (27%) faster than LP2. As expected, selection type ($F_{1,11}=396.38$, $p<0.01$) had a significant effect – i.e., it takes longer to lasso a larger target. Post-hoc pairwise comparisons show significant differences between technique for both single ($p<0.01$) and multiple ($p<0.01$) selections. Pressure marks were consistently faster, as Figure 6-14 illustrates.

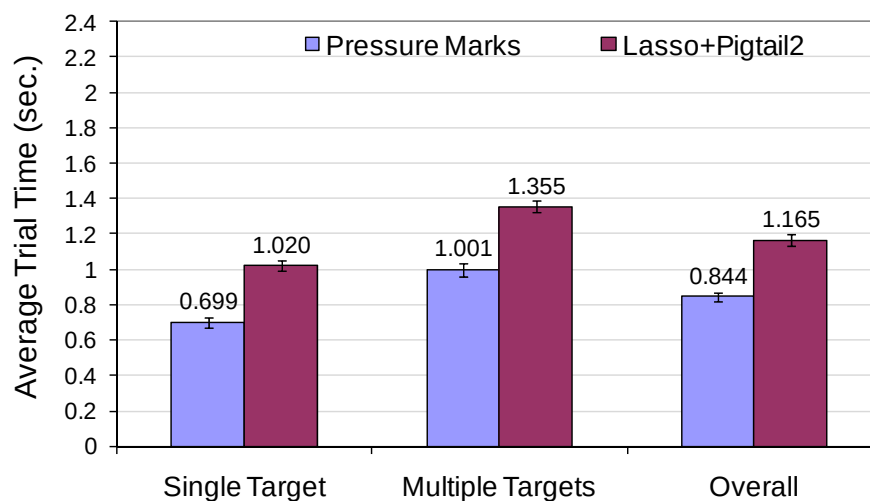


Figure 6-14: Average trial time (+/- SE) per technique and selection type.

There was a main effect for block ($F_{2,22}=13.64$, $p<0.01$), and a marginal technique*block interaction ($F_{2,22}=3.33$, $p=0.05$). Average trial times improved for both techniques as the study progressed. However, trial times decreased more drastically for the LP2 condition. Bonferroni-corrected post-hoc comparisons show no significant differences between the last two experimental blocks for either LP2 or pressure marks; and also reveal that the difference be-

PRESSURE MARKS

tween techniques at the last block is still significant ($p < 0.01$). Figure 6-15 illustrates these results.

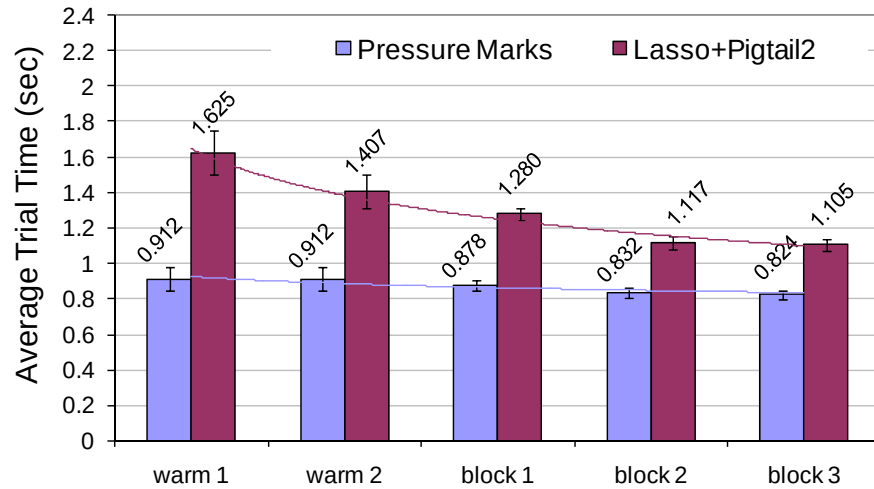


Figure 6-15: Average trial time per block. Power regression lines are shown.

6.4.4.2 Errors

We found no significant effects for technique ($F_{1,11}=0.29$, $p=0.59$) on errors (Figure 6-16). While participants made slightly fewer errors with LP2 when selecting multiple targets, Bonferroni-corrected post-hoc comparisons show that this difference was not significant ($p=0.22$).

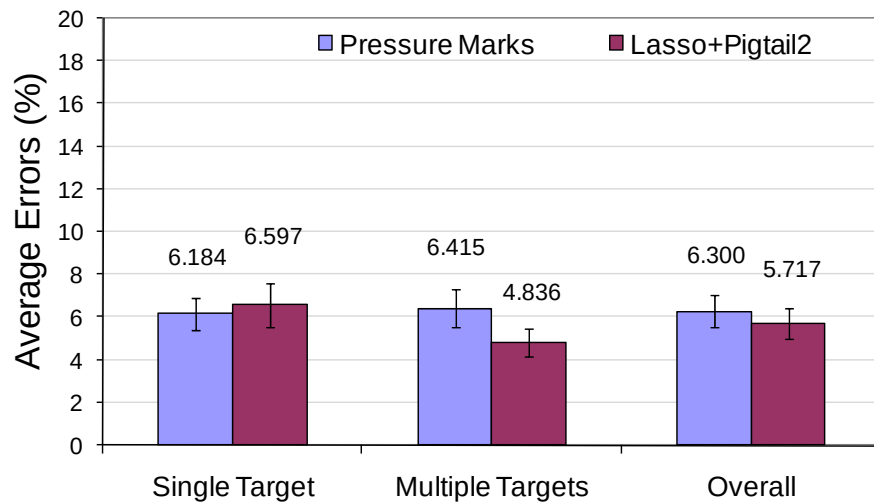


Figure 6-16: Average error per technique and selection type.

PRESSURE MARKS

While we saw some improvement in the users' accuracy as the study progressed, we did not find effects for blocks ($F_{2,22}=0.07$, $p=0.93$) or technique*block ($F_{2,22}=0.14$, $p=0.86$). Figure 6-17 illustrates these results.

6.4.4.3 Subjective Observations

Many participants reported how they developed different strategies for performing pigtails efficiently, depending on the (action) stimulus. These strategies usually involved starting the lasso from a particular point relative to the target(s) and doing a clockwise or counterclockwise motion. We also observed that participants had a preferred starting position and orientation that stayed almost unchanged when using pressure marks. Participants liked pressure marks because they helped them not having to think about the direction and orientation of a lasso.

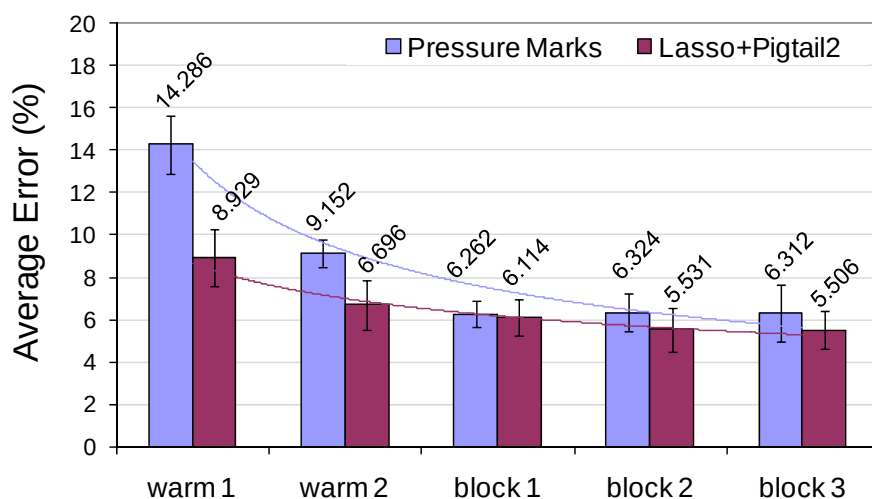


Figure 6-17: Average error per block per technique. Power regression lines are shown.

6.5 On The Use of Pressure Marks

Our studies provide us with evidence in support of pressure marks as a viable interaction technique whose ability to specify both selection and action concurrently outperforms existing techniques that require these operations be performed in a sequentially. This evidence encour-

PRESSURE MARKS

ages us to explore different designs where we can leverage the use and properties of pressure marks.

6.5.1 Pressure Marking Menus

We originally suggested the idea of pressure marking menus (Ramos et al., 2004), but did not elaborate upon it. This idea considers both the direction and the signature of a mark to increase the number of items available at any particular level or depth of a marking menu. With our proposed set of signatures, a menu's breadth can increase by a factor of four. While a single-level menu has a straightforward design, the visual design for novice users becomes challenging for menu depths greater than one. Whereas expert users will move through the marking menu's levels by changing a mark's inflexion, novice users need a visual design to help them browse through the menu's options at the current and sometimes at the next levels.

Figure 6-18 shows a design that aims to address this issue.

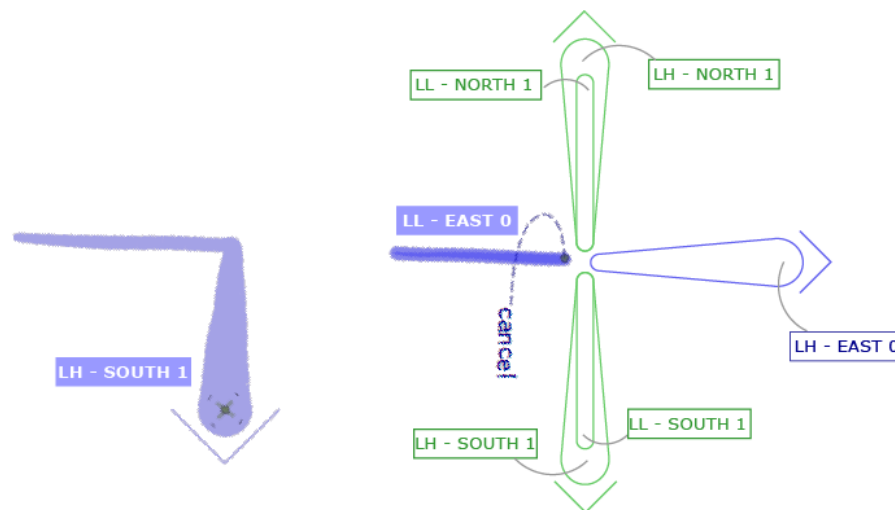


Figure 6-18: Pressure marking menu design. (Left) Expert mark. (Right) Feedback shown in the novice / browsing mode. The labels indicate the mark/level.

If we detect a pause-timeout delimiter while a stroke is drawn, we enter a browsing mode that shows available options for the menu's next level that are not collinear with the current mark.

PRESSURE MARKS

We use the direction of the current mark for browsing within the current level. Finally, users can pigtail on the mark to go back to the previous menu level.

6.5.2 Simple Pressure Marks

We can leverage pressure marks to expand other marking schemes such as Zhao and Balakrishnan’s simple marks technique (Zhao & Balakrishnan, 2004). As with compound marks, our proposed signatures can increase simple mark’s breadth by a factor of four. However, we think that it is interesting to explore a variation of the simple marking scheme, one that is orientation invariant. We call this design *simple pressure marks*. While traditional marking schemes rely on the presence of a “north” direction, simple pressure marks do not. An orientation-invariant marking scheme can be advantageous in situations when users engage an interactive surface from an arbitrary orientation – e.g., when an artist draws on a sheet of e-paper, or at collaborative tabletop environments.

With simple pressure marks, users specify a command by concatenating pressure signatures made in any direction (Figure 6-19). In this fashion one can define a menu structure with four choices per level. We argue that this type of “arbitrary flicking” makes connecting marks easy and independent from a user’s handedness, screen layout or orientation. For example, preliminary user observations revealed that users tend to develop a zigzag flicking pattern, which varies in orientation depending on the user.

PRESSURE MARKS

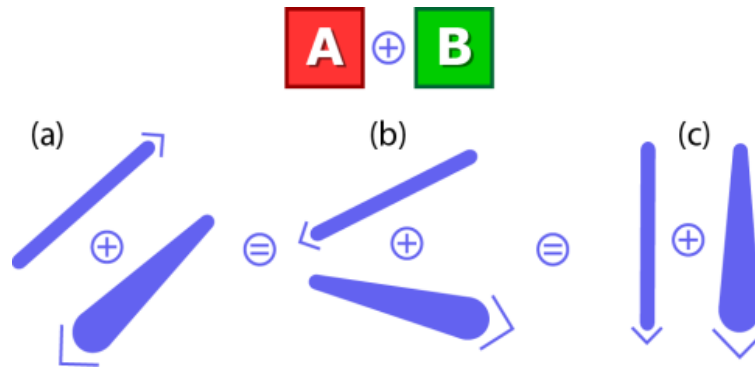


Figure 6-19: Different simple pressure mark flicks. (a) zig-zag pattern. (b, c) random directions. All correspond to the same command sequence A+B. A video illustrating this interaction can be found at <http://www.youtube.com/watch?v=qR2mKwkATpk> – timecode 01:48.

We consider two simple pressure marks to be connected if a user draws them within a certain time window. Whenever a user draws a mark, if it is not a leaf of the menu tree, we display a “ripple” originating at the mark’s end. This animated ripple lasts for as long as the connection time window and aims to make users aware of the opportunity to concatenate marks. If the user starts drawing a mark while the ripple is visible, the new mark connects to the previous one. We provide feedback for this concatenation by displaying the sequence of icons/labels (i.e. menu options) selected up to that moment. Figure 6-20 illustrates this.

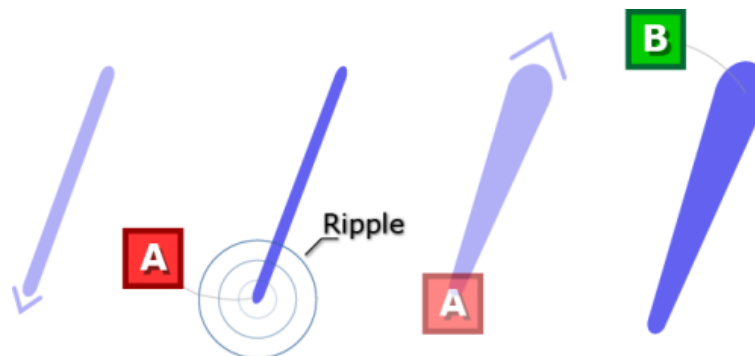


Figure 6-20: Ripple feedback. From left to right: a LL mark is made that triggers an expanding circular ripple. A LH mark is made while the ripple was active resulting on a LL+LH compound mark. A video illustrating this interaction can be found at <http://www.youtube.com/watch?v=qR2mKwkATpk> – timecode 01:48.

Whereas we envision expert users performing simple pressure marks straightforwardly, we argue that novice users can take advantage of the visualization and browsing mode discussed previously in the pressure marks section.

6.5.3 Pressure Tails

Pressure marks can also leverage pigtail delimiters to produce a technique called *pressure tails*, which allows for a selection-action-manipulation phrase where the pressure signature comes into play only when the pigtail gesture is performed. The advantage of integrating pigtails and pressure marks is that users do not need to be concerned as to the direction they are pigtailing, since the action is specified by the pressure signature. Instead, a pigtail delimiter marks the beginning of a parameter manipulation, such as the position of an object or its scaling factor. The self-crossing gesture defines a crossing interaction, wherein the pressure before and after the cross can be used to produce simple heuristics for the mark's parsing. Figure 6-21 illustrates an example of pressure tails.

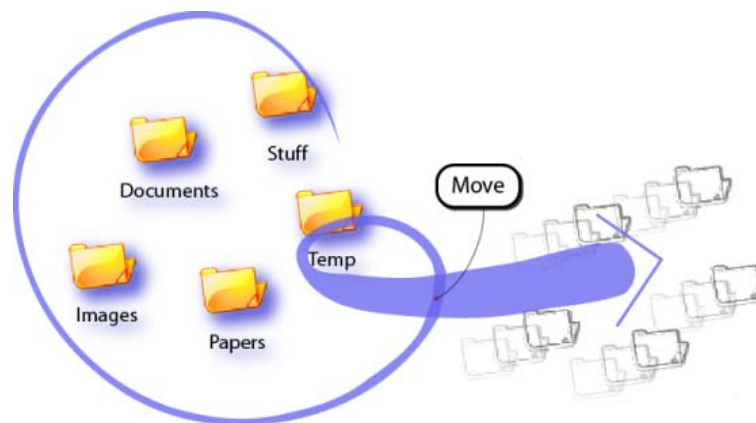


Figure 6-21: Pressure tails example. A LH crossing signature lets users move a group of objects.

6.5.4 Pressure Fanning

There are situations when users need to inspect information inside a container such as a folder. Similarly, information or GUI elements can be structured in piles. Agarawala and Balakrishnan (Agarawala & Balakrishnan, 2006) explore fanning as an interaction technique for revealing the content of piles in the interface. Pressure marks offer the means to provide additional semantics to such a fanning gesture – e.g., depending on a stroke's pressure signature, one can fan out the contents of a pile sorted in ascending or descending order, unsorted

PRESSURE MARKS

or as a means to break the pile (Figure 6-22). We have implemented this technique and initial feedback is encouraging.



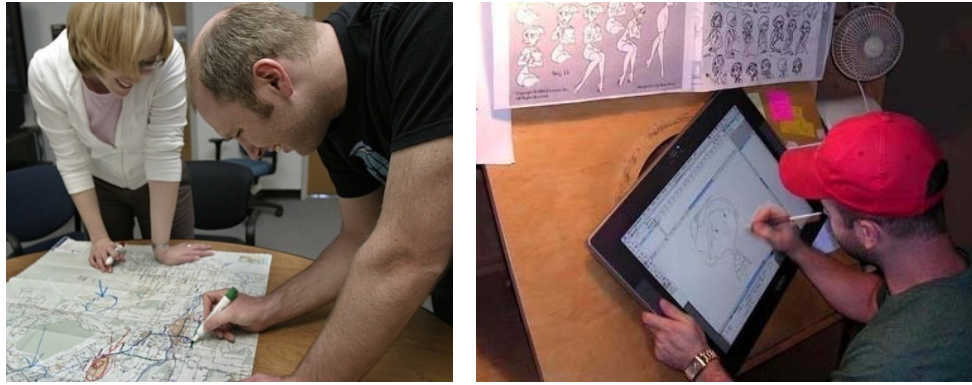
Figure 6-22: Example of pressure fanning. (a) LL mark fans the elements of the pile in their normal order. (b) HL mark fans the elements in ascending order. (c) LH mark fans the pile's elements in descending order. A video illustrating this interaction can be found at <http://www.youtube.com/watch?v=qR2mKwkATpk> – timecode 01:57.

6.6 Summary

Pressure marks are a novel way to use the pressure variations within a pen stroke in the user interface. In contrast with most interaction techniques used today, pressure marks can encode selection-action patterns in a concurrent, parallel interaction. The results we present in this chapter not only show that pressure marks are a viable interaction technique, but also reveal that their use can result in a significant reduction in the time it takes to perform selection-action patterns. In addition to these positive results, pressure marks have potential as orientation-independent marks, thus enhancing existing marking techniques and boosting new ones. We propose several designs that explore this possibility. Still, the designs we propose remain to be evaluated with a future user study.

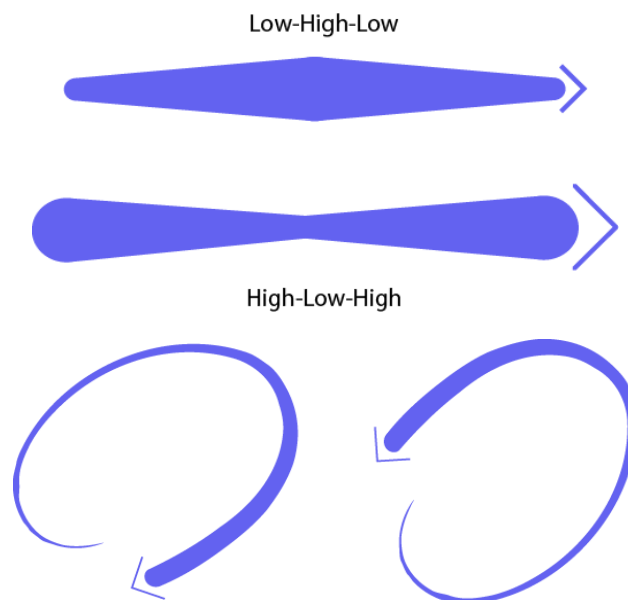
Several paths for future research stem from the concept of Pressure Marks. In particular, the concept of orientation independent marks can have a significant impact on single-user applications as well as multi-user tabletop settings. For example, artists frequently vary the orientation of their artwork during a drawing or design session; also, groups of collocated people can engage a surface from different orientations.

PRESSURE MARKS



6-23: (Left) A map annotation in a multi-user scenario where users can engage a surface from different orientations – Photo courtesy of Edward Tse. (Right) The ability to rotate the working surface is important for an artist. The image shows an electronic version of cartoon animator’s desk. – Photo credits from Steven Geer and Ronnie Williford.

Another path for future research includes investigating extensions to our proposed signature set of pressure marks to include compound marks – i.e., marks where one changes pressure more than one time such as Low-High-Low, or High-Low-High (6-24 left). It is worthwhile investigating expanding a signature set of marks by considering the rotation at which a stroke is drawn – i.e., clockwise or counterclockwise. In addition to encoding an extra bit of information, rotations are also orientation independent and can expand a given signature set by a factor of two (6-24 right).



6-24: (Top) Example of compound pressure marks. (Bottom) Example of two Low-High marks drawn with different directions.

7 Conclusions and Future Research

Pens are instruments that can react to the expressiveness of the human hand. In turn, pen-based computers have the potential to provide rich interactions and functionality. Yet, pen-based computers remain little more, and far less successful than the more widely used keyboard- and mouse-driven computers. This unrealized potential is primarily caused by the almost direct application of the standard point-and-click keyboard/mouse-based interface to pen-based computers. To maximize the potential of pen-based computers, it is necessary to systematically design and implement interactions and interfaces that are sensitive to both the pen's idiosyncrasies and the way users handle such a unique input device.

This dissertation constitutes a significant contribution to these efforts. In particular, we have advanced our knowledge about the design space of interaction techniques that utilize the pressure a person applies through the tip of a pen. At the same time, we have expanded our understanding of how people interact with both pen-based devices and interaction techniques that take advantage of pressure-sensing pens.

Chapters 3, 4, 5 and 6 represent four pieces of research that stand on their own as significant contributions to the field of pen-based computing. At the same time, these research efforts constitute important milestones in a more general discussion, or narrative about the design space of pressure-aware, pen-based interactions. Starting with a general exploration of how pressure could impact the vocabulary of pen-based interactions (Chapter 3), we investigated pressure as a discrete space (Chapter 4), then as a continuous space (Chapter 5) and later as a

CONCLUSION & FUTURE WORK

way not only to produce compact interaction phrases that encode scope and command in a single stroke, but also to enhance existing interaction techniques (Chapter 6).

Many of the artefacts we have introduced in this dissertation take the form of novel widgets and interaction techniques, which stand as individual contributions to the field. At the same time, these artefacts serve their intended purpose as crucial instruments for our research, which enable us to make a number of contributions.

7.1 Design Issues & Guidelines

Through our research, experimental studies and observations we were able to identify significant design issues as well as to suggest design guidelines for the development of pressure-sensitive pen-based widgets and interactions. The following sections summarize these.

7.1.1 Interaction Duration

Uninterrupted, long interactions involving the accurate control of pressure are not optimal, as the adaptation of human mechanoreceptors desensitize after only a few seconds of applying a constant level of force (§ 2.1.4). Thus interaction designs that rely on maintaining constant force through time are not ideal, since feedback from a person's hand becomes hindered. On the other hand, interaction techniques where variations in pressure are common have a better chance of being perceived correctly by a person's somatic sensory system (§2.1.4). Our experimental observations provide evidence that maintaining constant pressure might be even more challenging as a person's arm moves: It was more difficult for participants to draw pressure marks at constant levels (e.g., LL or HH) than marks with varying pressure (§6.3.5).

7.1.2 Visual Feedback

Continuous, real-time visual feedback is crucial for the accurate control of pressure. Feedback in the GUI should come at least in two forms: a) an application-agnostic notification that responds with the current pressure being sensed by the input device, such as the pressure cursor (§5.3.1), and b) an application- or widget-specific notification that provides users with information pertaining the consequences of their actions, such as the Zlider’s dynamic Vernier (§5.3), or Pressure Marks’ width-changing trace (§6.2). Subsequent studies by Mizobuchi in the context of pen input for hand-held devices also confirm that visual feedback is crucial for accurate pressure interactions (Mizobuchi et al., 2005).

7.1.3 Pressure Taming: Noise

Pressure signal can be noisy and if left unchecked can significantly diminish an interaction technique. In the context of this dissertation we define noise as the unexpected fluctuations in the pressure values reported by pressure transducers. The origin of this noise is not unique and can be attributed to three main sources. First, the input device’s signal might carry a certain level of background noise. A second source of noise is the physical environment, e.g., small vibrations. A third source of noise comes from people’s muscles, as hardware equipment can be sensitive to motions unnoticed by people.

We used a number of ways to deal with these sources of noise throughout our different designs (§5.3.2, §6.2.2). We use a low-pass (or averaging) filter to mitigate any background noise coming from the hardware. Choosing an appropriate window size is important because a window that is too large will generate an unwanted delay in the signal’s response, and a window that is too short will not filter the signal enough. For our designs we always choose a window size that allows for a good compromise between responsiveness of the interaction

CONCLUSION & FUTURE WORK

and smoothness of the filtered signal. To compensate for environmental and muscular noise, we propose the use of hysteresis filters that govern when changes on the input signal should be interpreted as changes. Hysteresis filters are different depending on the application. For example, for a task where one navigates through discrete pressure regions or zones a hysteresis function, like the one shown in Figure 7-1, can help users achieve better pressure control and to decrease overshooting effects.

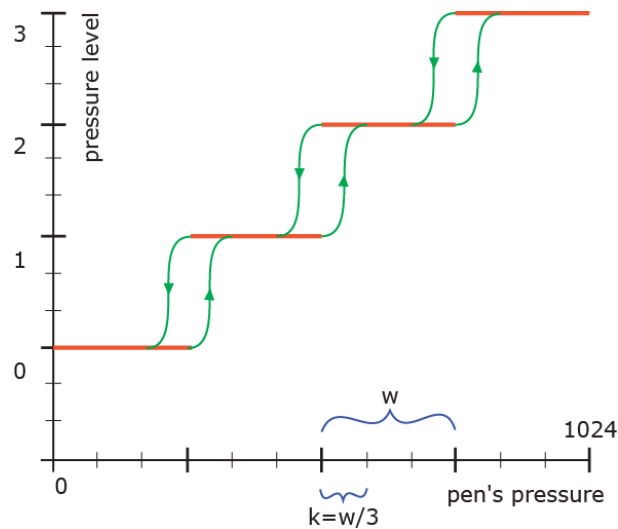


Figure 7-1: Example of a hysteresis function for navigating through four discrete pressure levels. To transition to an adjacent pressure level, users need to overshoot or undershoot by more than a fraction k of the pressure interval w . Here $k = 1/3$.

For tasks where one maps pressure to a continuous parameter, a hysteresis filter can correct for noise that the low-pass filter cannot detect and that is commonly caused by muscle tremor. In this case the hysteresis filter is similar to the one illustrated on Figure 7-1, but spanning all the possible levels of pressure that the input device can support, e.g., 1024 in our case. Depending on the desired result this filter can be tailored to provide different responses not only at different levels of pressure intensity, but also depending on the type of pressure variation, i.e., increasing or decreasing. The use of filtering techniques for the pressure signal is especially important for cases where pressure intensity is mapped to the control of a continuous parameter.

CONCLUSION & FUTURE WORK

While it is possible to envision more sophisticated filtering techniques in order to obtain a stable input signal from a pen's pressure-sensitive tip, we found our straightforward signal-filtering scheme worked very well our implementations of Zliding (§5.3) and Pressure Marks (§6.2).

7.1.4 Pressure Taming: Control

It is important to provide support for explicit pressure stabilization such as locking and clutching. Our observations tell us that while filtering the pressure signal does help to reduce sources of noise, providing users with pressure stabilization mechanisms such as locking and clutching is beneficial for two reasons: a) it further improves their control over pressure by stopping pressure fluctuations at mid/high intensity levels and b) it allows users to achieve arbitrarily high levels of pressure. Figure 7-2 illustrates the frequency of use of locking and clutching mechanisms during the studies of the Zliding technique (§5.5).

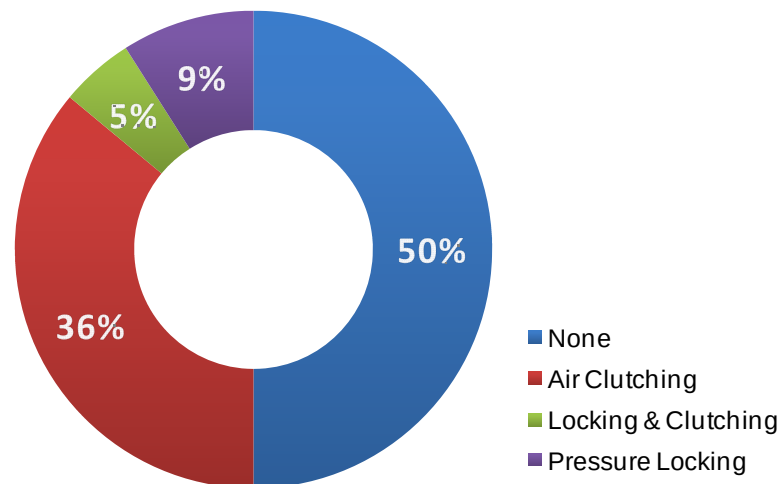


Figure 7-2: Frequency of use for pressure locking and air clutching mechanisms during the Zliding experiments (pen condition). Half the time one of these features was used, with air clutching being the most frequently used one.

7.1.5 Avoiding Task's Disruptions

One should avoid interaction techniques that force users to disrupt the task at hand. This is a design principle that applies to pen-based interactions in general, where there is potential for

CONCLUSION & FUTURE WORK

pressure-based interaction techniques. Examples of such a disruptive mechanisms are the round trips imposed by far away toolbars or the tap-and-hold delimiter, which we had the opportunity to observe in action during our heuristic evaluations of our LEAN system (§3.4). Unless such disruptions, breaks or pauses are a natural part of the user's workflow, one should avoid them. Examples of non-disruptive interaction techniques that fit within into a pen-driven interaction technique are pigtails delimiters (Hinckley et al., 2005), which can be incorporated in a more straightforward fashion into an interaction phrase, and trailing widgets (Forlines, Vogel, & Balakrishnan, 2006; Ramos, Cockburn, Balakrishnan, & Beaudouin-Lafon, 2007), which allow for an effective in-place mode switching mechanism.

Pressure can be used in interesting ways to facilitate mode switching or to disambiguate different pen modalities in a non-disruptive way. For example, pressure can reduce the need for disruptive mode-switching operations: we show how with *zsliding* (§5), one can combine zooming and sliding into an integrated interaction phrase that can eliminate the need to switch between separated pan and zoom modes. Also, we showed how using a pressure can be used to disambiguate between strokes and commands in the context of the LEAN system (§3.4). Li et al. show that using a pressure threshold is not significantly different in terms of speed than the fastest method in their study, a switch activated by the user's non-preferred hand (Li et al., 2005). The pressure-activated pointing lens technique (Ramos et al., 2007) also uses a pressure threshold to activate a lens that facilitate pen-based manipulation tasks. This pressure-activated lens results in faster activations than delay-activated or trailing pointing lenses.

Absolute pressure thresholds are straightforward both to conceptualize and to implement, though their use assumes that a single threshold applies to a variety of people. Since this is usually is not the case, sometimes errors can be observed for poorly chosen threshold values

CONCLUSION & FUTURE WORK

or across a population with a large perceptual variance. Neural science and psychophysics literature (Kandel et al., 1991; Stevens, 1975) tells us that humans perceive differences in force better than absolute values, and because of this insight it is possible to envision differential thresholds that will be less error prone than absolute ones. In their simplest form, differential thresholds are a collection of absolute threshold values that one chooses depending of the current level of pressure being applied. We used differential thresholds to transition between low and high states, while paring a pressure mark (§6.2.2) with very good results.

7.1.6 Performance Limits in Discrete Pressure Widgets

We showed how people with the aid of visual feedback can navigate through at least six different pressure zones or regions (§4.8). While navigating through more regions is possible, our studies indicate that once the number of regions increases above eight, the interactions become erroneous and difficult to control. We already mentioned (§4.11) how other researchers (Cechanowicz et al., 2007; Mizobuchi et al., 2005) found similar values across different hardware configurations and we remain curious as to what extent if these similar results dependent of people’s force-sensing capabilities.

Finally, an issue we did not investigate in depth in this dissertation is effect the use of appropriate transfer function would have on the numbers of regions a person can navigate through and on a user’s accuracy. The above issues remains areas for future research.

7.2 Charting the Design Space of Pressure Widgets

A way to summarize the factors involved in the design of pressure widgets and the specific widget designs that we have presented, is to use a *design matrix*. We originally proposed this matrix in our initial pressure widgets research (Ramos et al., 2004), with the goal of assisting in the design and creation of pressure widgets, much in the same way taxonomies can help to

CONCLUSION & FUTURE WORK

identify and describe the nature of existing and potential input devices (Buxton, 1983; Card, MacKinlay, & Robertson, 1991) or see-through tools (Bier, Stone, Fishkin, Buxton, & Baudel, 1994).

In the particular case of pressure widgets and the pressure expression channel, this *design matrix* has two dimensions (§4.9.2): *element* (e.g., *cursor*, *targets* or *glyphs*) and *behaviour* (e.g., *position*, *angle*, *scale*, *self*). The matrix, shown in Table 7-1, allowed us to initially inspect the design space of pressure widgets and prompt us to explore all possible combinations of cells so that one could imagine both the different visual designs and behaviors for pressure widgets.

Table 7-1: Design matrix for the design of pressure widgets. Combinations of cells describe the behavior for a particular pressure widget. For example, twist-lens' targets are affected both in terms of scale and position as pressure applied by a user changes.

Element ↓	Behavior →			
	<i>Position</i>	<i>Angle</i>	<i>Scale</i>	<i>Self</i>
Cursor	Flag (§4.9)	Pie (§4.9) Pressure Cursor (§5.3.1)	Bullseye (§4.9) Pressure Cursor (§5.3.1)	
Targets	Moving Flag (§4.9) Twist Lens (§ 3.7.2)	Rotating Pie (§4.9)	Twist Lens (§ 3.7.2)	
Glyphs				Mizobuchi's Pressure Menu

This design matrix helps us frame some existing widget designs; still it does not capture some of the more subtle design elements of pressure widgets designs that followed our initial research, such as the possibility of more than one visual element being present such as two cursors, or the participation of other expression channels in the interaction.

In the remainder of this section, we introduce a novel way of framing and expressing (pressure) widget designs in terms of atomic structures where expression channels are part of an atom's nucleus and visual elements are particles orbiting different levels of behavior. This

CONCLUSION & FUTURE WORK

chemistry-inspired representation not only helps us to grasp a widget's visual design properties but also allows us to compare different designs. Finally, we believe that this way of thinking about the design space of widgets allows for a compelling way of considering new designs that one might not originally imagined.

7.2.1 An Atom-Inspired Description of Pressure Widgets

Let us consider a simple pressure widget consisting of a number being displayed indicating the level of pressure being applied by a person. In this case, the dominant expression channel is *pressure (P)* and variations of this channel will change the *self* property of the widget's glyph (*G*) – i.e., numeric display. Figure 7-3 shows an atomic diagram representing this simple pressure widget.

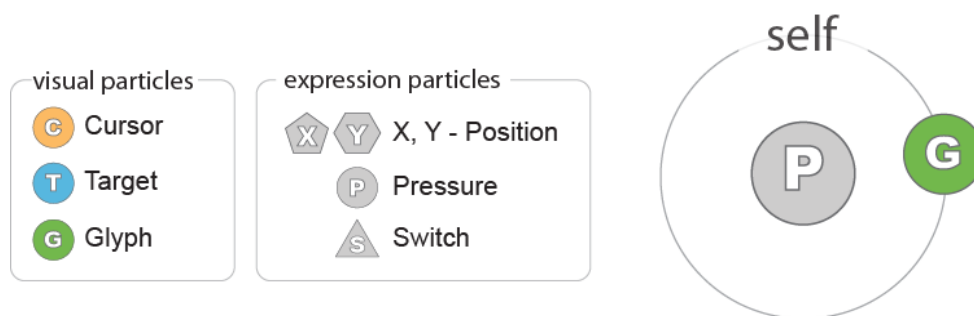


Figure 7-3: An atomic representation of a simple pressure widget. In this example, this diagram expresses the widget's design where a nucleus of P is orbited by a G particle at the *self* level. In other words, changing pressure causes the number(s) on the widget to change.

An atomic diagram has two main components: a nucleus and orbits. The nucleus is made of *expression* particles such as switch, x/y/z position, pressure, rotation, tilt, etc. Around the nucleus orbits different *element* particles such as cursors, targets and glyphs at different *behavior* orbit levels such as *self*, scale, angle and position. A good property of this representation is such that it scales up, permitting for the inclusion of unaccounted expression channels, behaviors and visual elements. Widgets that are more complex can warrant more complex atomic diagrams and sometimes the grouping of several atoms (Figure 7-4). One can explore

CONCLUSION & FUTURE WORK

the design space of pressure widgets by constructing arbitrary atoms by permuting different elements, particles and behaviors.

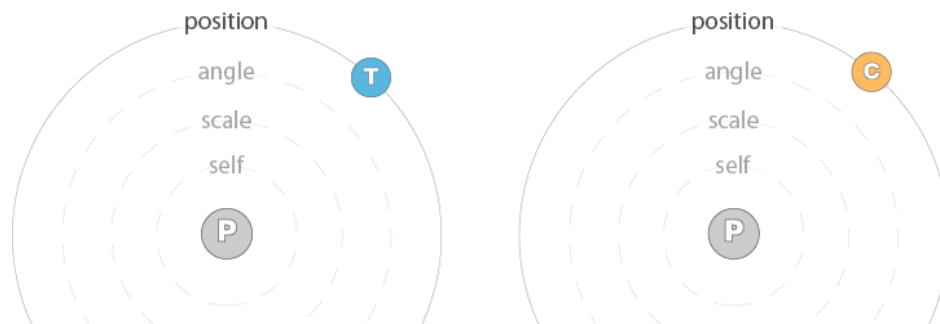


Figure 7-4: Atom diagrams for the *Flag* widget. (Left) *Moving Flag* design where changes in pressure result in changes on the targets' position. (Right) *Flag* design where changes in pressure result in changes on the cursor's position. Both diagrams also reveal similarities and differences between both widgets' designs.

The model we propose is not intended to model the behavior of every possible widgets or interaction technique. While there are other approaches that are better suited for modeling the behavior and predicting performance of interactive elements on the GUI such as CIS (Appert, Beaudouin-Lafon, & Mackay, 2004), our proposed atomic framework offers a tool to visually explain, compare and explore different widgets and interaction.

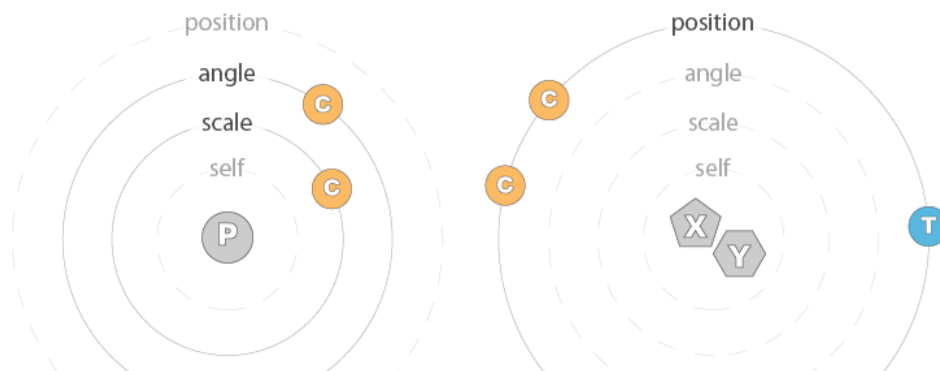


Figure 7-5: The atomic diagram of the Pressure Cursor, which uses two atoms. (Left) Variations of pressure result on changes on the cursor's aperture (angle) and fill (scale) – a pressure cursor has two elements that can be considered as independent cursors. (Right) Changes on the x/y expression channels move all the elements of the widget accordingly.

CONCLUSION & FUTURE WORK

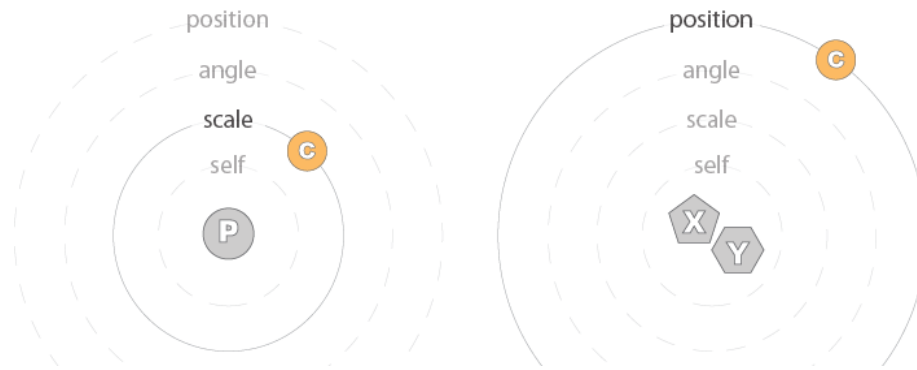


Figure 7-6: The atomic diagram for Pressure Marks, which uses two atoms. Variations of pressure result on changes on the cursor's diameter (scale). (Right) Changes on the x/y expression channels move all the elements of the widget accordingly. The diagram allows us to observe similarities and differences with the Pressure Cursor widget.

Finally, the scalability and modular quality of this framework has the potential to be applied beyond the design space of pressure widgets and constitutes a stimulating area for future work.

7.3 Future Directions

While effective in providing answers regarding the use of pressure in pen-based systems, our research also suggests a number of research directions in the areas including pressure-based interaction techniques, interaction and interface design, and pen-based systems.

7.3.1 Revisiting Digital Video Interaction

We believe that it is interesting to revisit our original designs for the LEAN system (§3.4) which were first introduced in 2003 (Ramos & Balakrishnan, 2003) and propose new possibilities and improvements that originate from the results and insights contained in this dissertation. This type of design exercise serves to strengthen our discussion about the use of pressure in pen-based interfaces and gives us an opportunity to apply the techniques and principles that we propose.

CONCLUSION & FUTURE WORK

7.3.1.1 Pressure Pulse as a Delimiter

The use of tap-and-hold as a delimiter was the object of criticism within the LEAN system (§3.9), i.e., user either thought that it was too long a wait, and that it got in the way of what they wanted to do. Our research in Pointing Lenses (Ramos et al., 2007) describes how a pressure threshold can be used effectively to transition to a different mode and hints at a method that we can use to allow users to quickly and unobtrusively activate different UI elements within LEAN.

Building upon our findings throughout this dissertation we propose the use of a *pressure pulse* as a gesture delimiter that replaces tap-and-hold. We define a *pulse* as a particular pressure variation over time, characterized by a fast change in the pressure applied through a pen's tip. The profile of this variation follows a rapid increase and decrease of pressure as illustrated by Figure 7-7.

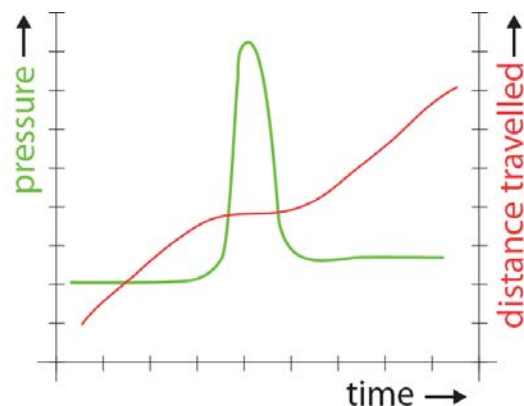


Figure 7-7: Profile of a pulse gesture. A pulse is a very quick variation on the pressure applied through the tip of a stylus while the pen remains stationary.

The use of a *pulse* has potential advantages over the use of a threshold. First, unlike a fixed threshold that might be challenging to find for a diverse user population, a pulse can be performed at different levels of base pressure, making it suitable across population of users with different sensitivities to pressure. Second, while visual feedback is necessary for the use of a fixed pressure threshold, a pulse has the potential to be performed eyes-free since people can

CONCLUSION & FUTURE WORK

sense differences in pressure much better than absolute pressure values. Third, a pulse is an quick, explicit gesture that can be phrased so as to have less false activations than a simple pressure threshold.

7.3.1.2 Position+Velocity Zlider

The PVSlider we described in section 3.7.1 requires an amount screen real-state that is not always available, e.g., portable multimedia devices. The Zlider presented in section 5.3 is an elegant solution that overcomes the space limitation of the PVSlider and constitutes an ideal candidate to replace it. The PVZlider has is a Zlider that is mapped to an interval of video frames of a video and provides all of the PVSlider's functionality in a compact format. Instead of directly using the Zlider's visual design from chapter 5, we replace the Zlider's abstract, adaptive Vernier for a texture composed of 1-dimensional slices of all the frames it maps (Figure 7-8). This visual representation of a video stream is similar to the one presented in the Video Streamer system (Elliott & Davenport, 1994) and has the benefit to reveal scene cuts and partial information about a particular scene.

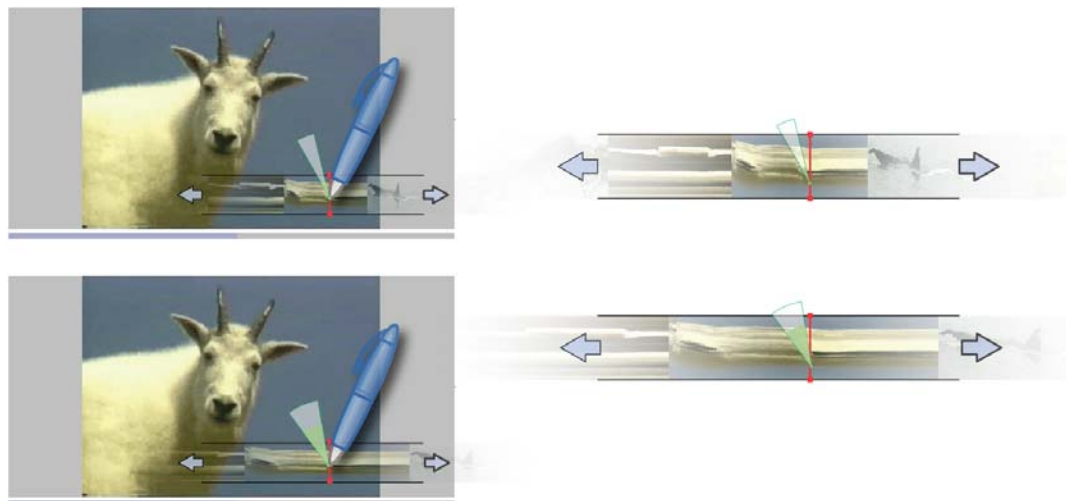


Figure 7-8: Mockup diagram of the PVZlider. (top-left) In place PVZlider shows frame slices as its background. (top-right) detail of the PVZlider's background. (bottom-left) a user increases the applied pressure thus changing the PVZlider's scale. This change is reflected on its texture as it stretches to match the new scale. (top-right) Detail of the PVZlider's texture at a nominal scale. (bottom-right) Detail of the PVZlider after pressure increased (i.e., more precision).

7.3.2 Learning Experience

One of the greatest challenges of any pen-based system that uses a gestural vocabulary is that of providing its users with mechanisms for learning such vocabulary. Using visual aids and imagery for this purpose is a traditional way of facing this challenge. Still, there are no silver-bullet strategies to straightforwardly express the dynamic, contextual and sometimes loosely defined shape of a particular gesture set, e.g., using static pictures such as the ones one might expect to find on a help document cannot capture a gesture's dynamic nature. This challenge in depiction only increases when one tries to express the role pressure interactions have on the interface, as variations in pressure do not have a direct mapping to a two-dimensional representation. There is an undeniable place for tools that can only fulfill their potential by an expert user. For example, the human ability of writing with a pen is often taken for granted and considered as something natural, whereas writing is a complex motor task that people train on for many years. Despite its steep learning curve, writing is far from being discarded as an input method or interaction paradigm. Likewise, we believe that some interactions that leverage the use of pressure might only achieve their full potential once users become dexterous with them. Nonetheless, we also believe that regardless of the learning curve a pressure-based interaction technique might possess; one should provide adequate means for learning how to use it.

We are aligned with Vogel's proposed design principle of *immediate usability* (Vogel & Balakrishnan, 2004) and believe that users training or learning a gestural interaction should learn as they do or explore their environment. Self-revealing help systems such as Vogel's (Vogel & Balakrishnan, 2004), Hinckley's (Hinckley et al., 2005) and pressure mark's browsing mechanism (§6.2.1) are steps in the right direction at producing a non-disruptive, interactive gesture learning experience. We foresee subtle animations and overlaid graphics playing

CONCLUSION & FUTURE WORK

a major role in the development of such learning solutions, which can ease the transition between an interaction's novice and expert modes of use.

7.3.3 Pressure as a Measure of Intention

This dissertation has shown the use of pressure to manipulate an explicit parameter, mode or discrete level. In addition to the above, it is also possible to use the pressure sensed by an input device to measure more subtle elements such as a user's intention. For example we describe how we use the pressure a person applies through a stylus to discern between a command mark and an ink stroke (§3.5), i.e., if a person presses hard onto a surface we assume that he or she *intends to leave a mark*, hence the drawn stroke is interpreted as ink. Similar strategies can be used in other contexts and systems, e.g., expanding targets only when one intends to drop a selection into a container, or providing an alternative to double-tap as an activation mechanism. In addition to the above, pressure channel information can enhance the accuracy of machine learning methods that try to predict users' intentions on the GUI. Other subtle uses of pressure include the instrumentation of input devices such as a mouse in order to reason about a user's stress or cognitive load (Ikehara & Crosby, 2005).

7.3.4 A Concert of Expression Channels

At the beginning of this dissertation, we discussed the need for pen-based interaction designs to consider and take advantage of the pen's idiosyncrasies and numerous expression channels, i.e., x/y movement, tilt, pressure, rotation, altitude (hover), etc. While this dissertation has explored the use of one of these channels, i.e., pressure, there is still significant work ahead for the research community at large. There are numerous ongoing efforts referencing our research. For example, the use of the hover state or the altitude of a pen have been explored through several efforts like Hover Widgets (Grossman et al., 2006), the Vacuum (Bezerianos

CONCLUSION & FUTURE WORK

& Balakrishnan, 2005) and Subramanian et al.'s above the surface, multi-level interaction system (Kattinakere, Grossman, & Subramanian, 2007; Subramanian, Aliakseyeu, & Lucero, 2006). Tilt has also been the subject of recent research efforts such as the Tilt Cursor, which has been shown to enhance stimulus-response compatibility in marking tasks (Tian, Ao, Wang, Setlur, & Dai, 2007). Other systems use many of the expressive capabilities of a pen in concert such as the pen-to-mime system (Oshita, 2005), which uses a pen as a physical proxy that defines the animation properties of a human figure or character. All these expression channels deserve not only individual investigations as to what they bring to an interaction, but they also need to be considered as input modalities that can, and probably should, operate in concert with each other.

7.3.5 Other Devices

The ideas and results from this dissertation have the potential of being applied to input devices other than pens. Direct input surfaces such as touchpads, Mitsubishi's DiamondTouch, or Microsoft's Surface computer are examples of such devices. While many of these touch surfaces do not sense pressure directly, they are still capable of sensing the area of contact of a finger against their surface, e.g., as done by the simulated pressure or *SimPress* (Benko, Wilson, & Baudisch, 2006) implemented on Microsoft's Surface computer. Simpler technologies like surfaces that only detect points of contact can simulate the perception of pressure by measuring/counting the number of fingers pressed against it.

7.3.6 Place in the Interface Ecology

One of the limitations of our work is that it does not investigate in depth how the use of pressure fits within a user-interface ecology. Desirable properties interaction phrases include that of being a) discoverable, b) understandable, c) learnable and d) consistent (through dif-

CONCLUSION & FUTURE WORK

ferent scenarios or applications). For example, clicking and dragging are ubiquitous elemental interactions that are universally associated across user interfaces with the act of selection and object movement, respectively. In contrast to these actions, the act of applying pressure does not have yet such a defined or even predictable role. In our exploration of the use of the pressure channel of expression we have found many possible uses for it, e.g., mode change, continuous parameter adjustment, discrete selector or command indicator. However, our research does not propose one use of pressure as the defining use across the GUI. It is not clear at this point which, how pressure should fit within the extensive vocabulary of interaction phrases in the GUI. This is a provocative issue that warrants future research and that can further shape the form and behaviour of interfaces to come.

7.4 Final Remarks

The research presented in this dissertation reveals that there is a promising and rich design space for pressure-sensitive, pen-based interaction techniques. Our exploration of this space is a step in the right direction which has already both influenced and inspired a significant number of scholarly papers and research efforts. These efforts not only explore different aspects of the pressure-based pen interaction techniques, but also consider non-traditional expression channels such as hover or tilt. Overall these efforts share the same goal of expanding existing interaction vocabularies and permitting users to use a broader set of expression channels to interact with their pen-based devices and systems.

8 Bibliography

- Abowd, G. D., Gauger, M., & Lachenmann, A. (2003). The Family Video Archive: An Annotation and Browsing Environment for Home Movies. *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, Berkeley, California, 1-8.
- Accot, J., & Zhai, S. (2002). More Than Dotting the I's - Foundations for Crossing-Based Interfaces. *Proceedings of the CHI '02 Conference on Human Factors in Computer Systems*, New York, NY, 73-80.
- Agarawala, A., & Balakrishnan, R. (2006). Keepin' It Real: Pushing the Desktop Metaphor with Physics, Piles and the Pen. *Proceedings of the CHI '06 Conference on Human Factors in Computing Systems*, Montreal, Quebec, Canada, 1283-1292.
- Ahlberg, C., & Shneiderman, B. (1994). The Alphaslider: A Compact and Rapid Selector. *Proceedings of the CHI '04 conference on Human Factors in Computing Systems*, Boston, Massachusetts, United States, 365-371.
- Albinsson, P.-A., & Zhai, S. (2003). High Precision Touch Screen Interaction. *Proceedings of the CHI '03 Conference on Human Factors in Computer Systems*, Ft. Lauderdale, Florida, USA, 105-112.
- Apitz, G., & Guimbretière, F. (2004). Crossy: A Crossing-Based Drawing Application. *Proceedings of the UIST '04 Symposium on User Interface Software and Technology*, Santa Fe, USA, 3-12.

BIBLIOGRAPHY

- Appert, C., Beaudouin-Lafon, M., & Mackay, W. (2004). Context Matters: Evaluating Interaction Techniques with the Cis Model. *Proceedings of the HCI'04 Conference on People and Computers XVIII*, 279-296.
- Appert, C., & Fekete, J.-D. (2006). Orthozoom Scroller: 1d Multi-Scale Navigation. *Proceedings of the CHI '06 Conference on Human Factors in Computing Systems*, Montreal, Quebec, Canada, 21-30.
- Ayatsuka, Y., Rekimoto, J., & Matsuoka, S. (1998). Popup Vernier: A Tool for Sub-Pixel-Pitch Dragging with Smooth Mode Transition. *Proceedings of the UIST '98 Symposium on User Interface Software and Technology*, San Francisco, California, United States, 39-48.
- Baecker, R., Rosenthal, A. J., Friedlander, N., Smith, E., & Cohen, A. (1996). A Multimedia System for Authoring Motion Pictures. *Proceedings of the Proceedings of the fourth ACM international conference on Multimedia*, Boston, Massachusetts, United States, 31-42.
- Baecker, R. M. (1969). *Interactive Computer-Mediated Animation*: Technical Report: TR-61, Massachusetts Institute of Technology.
- Baecker, R. M. (1974). Genesys - Interactive Computer-Mediated Animation. In Halas (Ed.), *Computer Animation* (pp. 97-115). New York: Hastings House.
- Balakrishnan, R., & Hinckley, K. (1999). The Role of Kinesthetic Reference Frames in Two-Handed Input Performance. *Proceedings of the UIST '99 Symposium on User Interface Software and Technology*, New York, NY, 171-178.
- Bao, S., & Silverstein, B. (2005). Estimation of Hand Force in Ergonomic Job Evaluations. *Ergonomics*, 48(3), 288-301.
- Barrett, R. C., Selker, E. J., Rutledge, J. D., & Olyha, R. S. (1995). Negative Inertia: A Dynamic Pointing Function. *Proceedings of the Conference companion on Human factors in computing systems.*, Denver, Colorado, United States, 316--317.

BIBLIOGRAPHY

- Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., et al. (2003). Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-Operated Systems. *Proceedings of the INTERACT '03 International Conference on Human-Computer Interaction*, Zurich, Switzerland, 57-64.
- Baudisch, P., Xie, X., Wang, C., & Ma, W.-Y. (2004). Collapse-to-Zoom: Viewing Web Pages on Small Screen Devices by Interactively Removing Irrelevant Content. *Proceedings of the UIST '04 Symposium on User Interface Software and Technology*, Santa Fe, NM, USA, 91-94.
- Bederson, B. B., & Hollan, J. D. (1995). Pad++: A Zoomable Graphical Interface System. *Proceedings of the CHI '95 Conference on Human Factors in Computing Systems*, Denver, Colorado, United States, 23-24.
- Benko, H., Wilson, A. D., & Baudisch, P. (2006). Precise Selection Techniques for Multi-Touch Screens. *Proceedings of the CHI '06 Conference on Human Factors in Computing Systems*, Montreal, Quebec, Canada, 1263-1272.
- Bezerianos, A., & Balakrishnan, R. (2005). The Vacuum: Facilitating the Manipulation of Distant Objects. *Proceedings of the CHI '05 Conference on Human Factors in Computing Systems*, Portland, Oregon, USA, 361-370.
- Bier, E., Stone, M., Pier, K., Buxton, W., & DeRose, T. (1993). Toolglass and Magic Lenses: The See-through Interface. *Proceedings of the SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, New York, NY, 73-80.
- Bier, E. A., Stone, M. C., Fishkin, K., Buxton, W., & Baudel, T. (1994). A Taxonomy of See-through Tools. *Proceedings of the CHI '94 Conference on Human Factors in Computing Systems*, Boston, Massachusetts, United States, 358-364.

BIBLIOGRAPHY

- Blanch, R., Guiard, Y., & Beaudouin-Lafon, M. (2004). Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaptation. *Proceedings of the CHI '04 Conference on Human Factors in Computing Systems*, Vienna, Austria, 519-526.
- Boothe, R. G. (2001). *Perception of the Visual Environment* (1 ed.): Springer.
- Bourgeois, F., Guiard, Y., & Beaudouin-Lafon, M. (2001). Pan-Zoom Coordination in Multi-Scale Pointing. *Proceedings of the CHI '01 extended abstracts on Human factors in computing systems*, Seattle, Washington, 157-158.
- Buxton, W. (1983). Lexical and Pragmatic Considerations of Input Structures. *Computer Graphics*, 17(1), 31-37.
- Buxton, W. (1986). Chunking and Phrasing and the Design of Human-Computer Dialogues. In Baecker, Grudin, Buxton & Greenberg (Eds.), *Readings in Human-Computer Interaction: Towards the Year 2000* (pp. 494-499). San Fransisco, CA: Morgan Kaufmann.
- Buxton, W. (2005). Resource Page on Early Hci Research by the Lincoln Lab Tx-2 Group. from <http://www.billbuxton.com/Lincoln.html>
- Buxton, W., Hill, R., & Rowley, P. (1985). Issues and Techniques in Touch Sensitive Tablet Input. *Proceedings of the SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, 215-224.
- Buxton, W., & Myers, B. (1986). A Study in Two-Handed Input. *Proceedings of the CHI '86 Conference on Human Factors in Computing Systems*, New York, NY, 321-326.
- Card, S. K., MacKinlay, J. D., & Robertson, G. G. (1991). A Morphological Analysis of the Design Space of Input Devices. *ACM Transactions on Information Systems*, 9, 99-122.
- Casares, J., Long, C., Myers, B., Stevens, S., & Corbett, A. (2002). Simplifying Video Editing with Silver. *Proceedings of the CHI '02 extended abstracts on Human factors in computing systems*, Minneapolis, Minnesota, USA, 672 - 673

BIBLIOGRAPHY

- Cechanowicz, J., Irani, P., & Subramanian, S. (2007). Augmenting the Mouse with Pressure Sensitive Input. *Proceedings of the CHI '07 Conference on Human Factors in Computing Systems*, San Jose, California, USA, 1385-1394.
- Diakopoulos, N., & Essa, I. (2006). Videotater: An Approach for Pen-Based Digital Video Segmentation and Tagging. *Proceedings of the UIST '06 Symposium on User Interface Software and Technology*, Montreux, Switzerland, 221-224.
- Dietz, P., & Leigh, D. (2001). Diamondtouch: A Multi-User Touch Technology. *Proceedings of the UIST '01 Symposium on User Interface Software and Technology*, New York, NY, 219-226.
- Elliott, D., Helsen, W. F., & Chua, R. (2001). A Century Later: Woodworth's, 1899. Two-Component Model of Goal-Directed Aiming. *Psych. Bulletin* (127), 342-357.
- Elliott, E., & Davenport, G. (1994). Video Streamer. *Proceedings of the CHI '04 Conference on Human Factors in Computing Systems (Companion)*, 65-68.
- Fitts, P. M. (1954). The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Exp.Psy.*, 47, 381-391.
- Fitzmaurice, G., Balakrishnan, R., & Kurtenbach, G. (1999). An Exploration into Supporting Artwork Orientation in the User Interface. *Proceedings of the CHI '99 Conference on Human Factors in Computer Systems*, New York, NY, 167-174.
- Fitzmaurice, G., Khan, A., Pieké, R., Buxton, B., & Kurtenbach, G. (2003). Tracking Menus. *Proceedings of the UIST '03 Symposium on User Interface Software and Technology*, Vancouver, Canada, 71-79.
- Forlines, C., Vogel, D., & Balakrishnan, R. (2006). Hybridpointing: Fluid Switching between Absolute and Relative Pointing with a Direct Input Device. *Proceedings of the UIST '06 Symposium on User Interface Software and Technology*, Montreux, Switzerland, 211-220.

BIBLIOGRAPHY

- Furnas, G. W., & Bederson, B. B. (1995). Space-Scale Diagrams: Understanding Multiscale Interfaces. *Proceedings of the CHI '95 Conference on Human Factors in Computing Systems*, Denver, Colorado, United States, 234-241.
- Gescheider, G. A. (1997). *Psychophysics : The Fundamentals* (3rd ed.). Mahwah, N.J. ; London: L. Erlbaum Associates.
- Goldstein, E. B. (2002). *Sensation and Perception* (6th ed.). Australia ; Pacific Grove, CA: Wadsworth-Thomson Learning.
- Goularte, R., Camacho-Guerrero, J. A., Valter R. Inacio, J., Cattelan, R. G., & Pimentel, M. d. G. C. (2004). M4note: A Multimodal Tool for Multimedia Annotations. *Proceedings of the WebMedia & LA-Web 2004 Joint Conference 10th Brazilian Symposium on Multimedia and the Web 2nd Latin American Web Congress - Volume 00*, 142-149.
- Goularte, R., Cattelan, R. G., Camacho-Guerrero, J. A., Jr., V. R. I., & Maria da, G. C. P. (2004). Interactive Multimedia Annotations: Enriching and Extending Content. *Proceedings of the ACM symposium on Document engineering*, Milwaukee, Wisconsin, USA, 84-86.
- Gray, H., & Lewis, W. H. (1918). *Anatomy of the Human Body* (20th ed.). Philadelphia: Lea & Febiger.
- Greenberg, S., & Fitchett, C. (2001). Phidgets: Easy Development of Physical Interfaces through Physical Widgets. *Proceedings of the UIST '01 Symposium on User Interface Software and Technology*, Orlando, Florida, 209-218.
- Gross, M. D., & Do, E. Y.-L. (1996). Ambiguous Intentions: A Paper-Like Interface for Creative Design. *Proceedings of the UIST '01 Symposium on User Interface Software and Technology*, 183-192.

BIBLIOGRAPHY

- Grossman, T., Hinckley, K., Baudisch, P., Agrawala, M., & Balakrishnan, R. (2006). Hover Widgets: Using the Tracking State to Extend the Capabilities of Pen-Operated Devices. *Proceedings of the CHI '06 Conference on Human Factors in Computing Systems*, Montreal, Quebec, Canada, 861-870.
- Guiard, Y., Beaudouin-Lafon, M., Bastin, J., Pasveer, D., & Zhai, S. (2004). View Size and Pointing Difficulty in Multi-Scale Navigation. *Proceedings of the AVI '04 Working conference on Advanced visual interfaces*, Gallipoli, Italy, 117-124.
- Guiard, Y., Beaudouin-Lafon, M., & Mottet, D. (1999). Navigation as Multiscale Pointing: Extending Fitts' Model to Very High Precision Tasks. *Proceedings of the CHI '99 Conference on Human Factors in Computing Systems*, Pittsburgh, Pennsylvania, United States, 450-457.
- Guimbretière, F., Martin, A., & Winograd, T. (2005). Benefits of Merging Command Selection and Direct Manipulation. *ACM Transactions on Computer-Human Interaction*, 12(3), 460-476.
- Guimbretière, F., Stone, M., & Winograd, T. (2001). Fluid Interaction with High-Resolution Wall-Size Displays. *Proceedings of the UIST '01 Symposium on User Interface Software and Technology*, New York, 21-30.
- Guimbretière, F., & Winograd, T. (2000). Flowmenus: Combining Command, Text, and Data Entry. *Proceedings of the UIST '00 Symposium on User Interface Software and Technology*, 213-216.
- Harrison, B., Fishkin, K., Gujar, A., Mochon, C., & Want, R. (1998). Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces. *Proceedings of the CHI '98 Conference on Human Factors in Computing Systems*, New York, NY, 17-24.

BIBLIOGRAPHY

- Harrison, B. L., & Baecker, R. M. (1992). Designing Video Annotation and Analysis Systems. *Proceedings of the GI '92 Graphics Interface Conference*, 157-166.
- Herot, C., & Weinzapfel, G. (1978). One-Point Touch Input of Vector Information for Computer Displays. *Proceedings of the SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, 210-216.
- Hinckley, K., Baudisch, P., Ramos, G., & Guimbretiere, F. (2005). Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in Scriboli. *Proceedings of the CHI '05 Conference on Human Factors in Computer Systems*, Portland, USA, 451-460.
- Hinckley, K., Czerwinski, M., & Sinclair, M. (1998). Interaction and Modeling Techniques for Desktop Two-Handed Input. *Proceedings of the UIST '98 Symposium on User Interface Software and Technology*, New York, NY, 49-58.
- Hinckley, K., Guimbretiere, F., Agrawala, M., Apitz, G., & Chen, N. (2006). Phrasing Techniques for Multi-Stroke Selection Gestures. *Proceedings of the GI '06 Graphics Interface Conference*, Quebec, Canada, 147-154.
- Igarashi, T., & Hinckley, K. (2000). Speed-Dependent Automatic Zooming for Browsing Large Documents. *Proceedings of the UIST*, San Diego, California, United States, 139-148.
- Ikehara, C. S., & Crosby, M. E. (2005). Assessing Cognitive Load with Physiological Sensors. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, 295-304.
- Jacob, R., Sibert, L., McFarlane, D., & Mullen, M. (1994). Integrality and Separability of Input Devices. *ACM Transactions on Computer-Human Interaction*, 1(1), 3-26.
- Jul, S., & Furnas, G. (1998). Critical Zones in Desert Fog: Aids to Multiscale Navigation. *Proceedings of the UIST '98 Symposium on User Interface Software and Technology*, New York, NY, 97-106.

BIBLIOGRAPHY

- Kandel, E. R., Schwartz, J. H., & Jessell, T. M. (1991). *Principles of Neural Science* (3rd ed.). New York: Elsevier.
- Kara, L. B., & Stahovich, T. F. (2004). Hierarchical Parsing and Recognition of Hand-Sketched Diagrams. *Proceedings of the UIST '04 Symposium on User Interface Software and Technology*, Santa Fe, NM, USA, 13-22.
- Kattinakere, R. S., Grossman, T., & Subramanian, S. (2007). Modeling Steering within above-the-Surface Interaction Layers. *Proceedings of the CHI '07 Conference on Human Factors in Computing Systems*, San Jose, California, USA, 317-326.
- Komerska, R., Ware, C., & Plumlee, M. (2002). Haptic Interface for Center-of-Workspace Interaction. *Proceedings of the IEEE Virtual Reality - Haptics Symposium*, Orlando, FL, 352-353.
- Kurtenbach, G., & Buxton, W. (1993). The Limits of Expert Performance Using Hierarchical Marking Menus. *Proceedings of the CHI '93 Conference on Human Factors in Computing Systems*, New York, USA, 35-42.
- Latulipe, C., Kaplan, C. S., & Clarke, C. L. A. (2005). Bimanual and Unimanual Image Alignment: An Evaluation of Mouse-Based Techniques. *Proceedings of the UIST '05 Symposium on User Interface Software and Technology*, Seattle, WA, USA, 123-131.
- Lécuyer, A., Coquillart, S., Kheddar, A., Richard, P., & Coiffet, P. (2000). Pseudo-Haptic Feedback: Can Isometric Input Devices Simulate Force Feedback? *Proceedings of the Virtual Reality 2000 Conference*, New Brunswick, New Jersey, USA, 83-90.
- Levine, S. R., & Ehrlich, S. F. (1995). The Freestyle System: A Design Perspective In Baecker, Grudin, Buxton & Greenberg (Eds.), *Human-Computer Interaction: Toward the Year 2000* (pp. 871-880): Morgan Kaufmann Publishers.

BIBLIOGRAPHY

- Li, Y., Hinckley, K., Guan, Z., & Landay, J. A. (2005). Experimental Analysis of Mode Switching Techniques in Pen-Based User Interfaces. *Proceedings of the CHI '05 Conference on Human Factors in Computing Systems*, Portland, Oregon, USA, 461-470.
- Lin, J., Newman, M. W., Hong, J. I., & Landay, J. A. (2000). Denim: Finding a Tighter Fit between Tools and Practice for Web Site Design. *Proceedings of the CHI '00 Conference on Human Factors in Computing Systems*, 510-517.
- Lowe, S. R. (1999). *The Complete Idiot's Guide to Handwriting Analysis*: Alpha Books.
- Marshall, C. C. (1997). Annotation: From Paper Books to the Digital Library. *Proceedings of the ACM International Conference on Digital Libraries*, 131-140.
- McGuffin, M., & Balakrishnan, R. (2002). Acquisition of Expanding Targets. *Proceedings of the CHI '02 Conference on Human Factors in Computer Systems*, New York, NY, 57-64.
- McGuffin, M., Burtnyk, N., & Kurtenbach, G. (2002). Fast Sliders: Integrating Marking Menus and the Adjustment of Continuous Values. *Proceedings of the GI '02 Graphics Interface Conference*, Calgary, Alberta, 35-42.
- Mizobuchi, S., Terasaki, S., Keski-Jaskari, T., Nousiainen, J., Ryyanen, M., & Silfverberg, M. (2005). Making an Impression: Force-Controlled Pen Input for Handheld Devices. *Proceedings of the CHI '05 extended abstracts on Human factors in computing systems*, Portland, OR, USA, 1661-1664.
- Mizobuchi, S., & Yasumura, M. (2004). Tapping Vs. Circling Selections on Pen-Based Devices: Evidence for Different Performance-Shaping Factors. *Proceedings of the CHI '04 Conference on Human Factors in Computer Systems*, Vienna, Austria, 607-614.
- Moscovich, T., & Hughes, J. F. (2004). Navigating Documents with the Virtual Scroll Ring. *Proceedings of the UIST '04 Symposium on User Interface Software and Technology*, Santa Fe, NM, USA, 57-60.

BIBLIOGRAPHY

- Mynatt, E., Igarashi, T., Edwards, W., & LaMarca, A. (1999). Flatland: New Dimensions in Office Whiteboards. *Proceedings of the CHI '99 Conference on Human Factors in Computing Systems*, New York, 346-353.
- Oshita, M. (2005). Pen-to-Mime: Pen-Based Interactive Control of a Human Figure. *Computers & Graphics*, 29(6), 931-945.
- Pederson, E., McCall, K., Moran, T., & Halasz, F. (1993). Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings. *Proceedings of the CHI '93 Conference on Human Factors in Computing Systems*, New York, 391-398.
- Perlin, K., & Fox, D. (1993). Pad: An Alternative Approach to the Computer Interface. *Proceedings of the SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, 57-64.
- Pook, S., Lecolinet, E., Vaysseix, G., & Barillot, E. (2000). Control Menus: Execution and Control in a Single Interactor. *Proceedings of the CHI '00 Conference on Human Factors in Computing Systems (Extended Abstracts)*, New York, NY, 263-264.
- Potter, R. L., Weldon, L. J., & Shneiderman, B. (1988). Improving the Accuracy of Touch Screens: An Experimental Evaluation of Three Strategies. *Proceedings of the CHI*, Washington, D.C., United States, 27-32.
- Raisamo, R. (1999). Evaluating Different Touched-Based Interaction Techniques in a Public Information Kiosk. *Proceedings of the Conference of the Computer Human Interaction Special Interest Group of the Ergonomics Society of Australia*, 169-171.
- Ramos, G., & Balakrishnan, R. (2003). Fluid Interaction Techniques for the Control and Annotation of Digital Video. *Proceedings of the UIST '03 Symposium on User Interface Software and Technology*, 105-114.

BIBLIOGRAPHY

- Ramos, G., & Balakrishnan, R. (2004). *Visual Features and Interference in Pressure Widgets*: Technical Report - DGP-TR-2004-003, University of Toronto.
- Ramos, G., & Balakrishnan, R. (2005). Zliding: Fluid Zooming and Sliding for High Precision Parameter Manipulation. *Proceedings of the UIST '05 Symposium on User Interface Software and Technology*, Seattle, WA, USA, 143-152.
- Ramos, G., Boulos, M., & Balakrishnan, R. (2004). Pressure Widgets. *Proceedings of the CHI '04 Conference on Human Factors in Computing Systems*, Vienna, Austria, 487-494.
- Ramos, G., Cockburn, A., Balakrishnan, R., & Beaudouin-Lafon, M. (2007). Pointing Lenses: Facilitating Stylus Input through Visual-and Motor-Space Magnification. *Proceedings of the CHI '07 Conference on Human Factors in Computing Systems*, San Jose, California, USA, 757-766.
- Ramos, G. A., & Balakrishnan, R. (2007). Pressure Marks. *Proceedings of the CHI '07 Conference on Human Factors in Computing Systems*, San Jose, California, USA, 1375-1384.
- Rekimoto, J. (2002). Smartskin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. *Proceedings of the CHI '02 Conference on Human Factors in Computer Systems*, New York, NY, 113-120.
- Rubine, D. (1991). Specifying Gestures by Example. *Proceedings of the SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, 329-337.
- Saund, E., & Lank, E. (2003). Stylus Input and Editing without Prior Selection of Mode. *Proceedings of the UIST '03 Symposium on User Interface Software and Technology*, Vancouver, Canada, 213-216.
- Sawhney, N., Balcom, D., & Smith, I. (1996). Hypercafe: Narrative and Aesthetic Properties of Hypervideo. *Proceedings of the ACM Hypertext Conference*, 1-10.

BIBLIOGRAPHY

- Schilit, B. N., Golovchinsky, G., & Price, M. N. (1998). Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations. *Proceedings of the CHI '98 Conference on Human Factors in Computing Systems*, 249-256.
- Shen, C., Vernier, F. D., Forlines, C., & Ringel, M. (2004). Diamondspin: An Extensible Toolkit for around-the-Table Interaction. *Proceedings of the CHI '04 Conference on Human Factors in Computer Systems*, Vienna, Austria, 167-174.
- Shneiderman, B., & Kang, H. (2000). Direct Annotation: A Drag-and-Drop Strategy for Labeling Photos. *Proceedings of the IEEE Conference on Information Visualization*, 88-98.
- Smart Technologies Inc. from <http://smarttech.com/>
- Smelser, N. J., & Baltes, P. B. (2001). *International Encyclopedia of the Social & Behavioral Sciences* (1st ed.). Amsterdam ; New York: Elsevier.
- Snibbe, S. S., MacLean, K. E., Shaw, R., Roderick, J., Verplank, W., & Scheeff, M. (2001). Haptic Techniques for Media Control. *Proceedings of the UIST '01 Symposium on User Interface Software and Technology*, 199-208.
- Song, M., Elias, T., Müller-Wittig, W., & Chan, T. K. Y. (2005). Using the Chinese Calligraphy Brush as a Tangible User Interface Tool in Virtual Heritage Scenarios. *Computers & Graphics*, 29(1), 41-48.
- Srinivasan, M., & Chen, J. (1993). Human Performance in Controlling Normal Forces of Contact with Rigid Objects. *ASME*, 49, 119-125.
- Stevens, S. S. (1975). *Psychophysics: Introduction to Its Perceptual, Neural, and Social Prospects*. New York: Wiley.
- Subramanian, S., Aliakseyeu, D., & Lucero, A. (2006). Multi-Layer Interaction for Digital Tables. *Proceedings of the UIST '06 Symposium on User Interface Software and Technology*, Montreux, Switzerland, 269-272.

BIBLIOGRAPHY

- Sutherland, I. (1963). Sketchpad: A Man-Machine Graphical Communication System. *Proceedings of the AFIPS Spring Joint Computer Conference*, 329-346.
- Sutherland, W. R. (1966). *The on-Line Graphical Specification of Computer Procedures*. Ph.D. Thesis, Massachusetts Institute of Technology.
- Tan, H., Srinivasan, M., Eberman, B., & Chang, B. (1994). Human Factors for the Design of Force-Reflecting Haptic Interfaces. *Proceedings of the Proceedings of the 3rd Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, ASME/IMECE, Chicago, IL.*, 353-359.
- Tian, F., Ao, X., Wang, H., Setlur, V., & Dai, G. (2007). The Tilt Cursor: Enhancing Stimulus-Response Compatibility by Providing 3d Orientation Cue of Pen. *Proceedings of the CHI '07 Conference on Human Factors in Computing Systems*, San Jose, California, USA, 303-306.
- Toshiyuki, M., Kouichi, K., & George R. Borden, I. V. (1995). Elastic Graphical Interfaces to Precise Data Manipulation. *Proceedings of the Human factors in computing systems*, Denver, Colorado, United States, 143-144.
- Tsang, M., Fitzmaurice, G. W., Kurtenbach, G., Khan, A., & Buxton, B. (2002). Boom Chameleon: Simultaneous Capture of 3d Viewpoint, Voice and Gesture Annotations on a Spatially-Aware Display. In *Uist '02 Symposium on User Interface Software and Technology* (pp. 111-120): ACM.
- Vogel, D., & Balakrishnan, R. (2004). Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users. *Proceedings of the UIST '04 Symposium on User Interface Software and Technology*, Santa Fe, NM, USA, 137-146.

BIBLIOGRAPHY

- Walker, N., & Smelcer, J. B. (1990). A Comparison of Selection Time from Walking and Pull-Down Menus. *Proceedings of the Proceedings of the SIGCHI conference on Human factors in computing systems.*, eattle, Washington, United States., 221-226.
- Wu, M., & Balakrishnan, R. (2003). Multi-Finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays. *Proceedings of the UIST '03 Symposium on User Interface Software and Technology*, Vancouver, Canada, 193-202.
- Zelevnik, R., & Miller, T. (2006). Fluid Inking: Augmenting the Medium of Free-Form Inking with Gestures. *Proceedings of the GI '06 Graphics Interface Conference*, Quebec, Canada, 155-162.
- Zelevnik, R., Miller, T., & Forsberg, A. (2001). Pop through Mouse Button Interactions. *Proceedings of the UIST '01 Symposium on User Interface Software and Technology*, 195-196.
- Zhai, S. (1995). *Human Performance in Six Degree-of-Freedom Input Control*. Unpublished Ph.D., University of Toronto, Toronto, ON.
- Zhai, S., & Smith, B. (1999). Multi-Stream Input: An Experimental Study of Document Scrolling Methods. *IBM Systems Journal*, 38(4), 642-651.
- Zhao, S., & Balakrishnan, R. (2004). Simple Vs. Compound Mark Hierarchical Marking Menus. *Proceedings of the UIST '04 Symposium on User Interface Software and Technology*, Santa Fe, NM, USA, 33-42.

9 Appendix A:

Pressure Widgets' Study

Survey Forms

APPENDIX A

Please tell us a little about yourself...

What is your gender?

Male

Female

What is your age?

18-24

25-34

35-44

45-60

Have you ever used a Tablet PC?

Yes

No

Have you ever used a PocketPC, Palm© or other pen-based device?

Yes

No

APPENDIX A

I believe I made many mistakes when performing tasks without visual feedback.

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

Today you tried a number of selections methods. Rate them according to how easy to use you found them, by putting a number beside each one

Click

Stroke

Time delay

Quick Release

In a couple of sentences tell us on what devices you can see pressure being used as input.

Please tell us your final thoughts and/or impressions you may have about your experience in this session.

Thank you! This completes the session.

APPENDIX A

Please tell us a little about yourself...

1. What is your gender?

Male

Female

2. What is your age?

18-24

25-34

35-44

45-60

3. Have you ever used a Tablet PC?

Yes

No

4. Have you ever used a PocketPC, Palm© or other pen-based device?

Yes

No

APPENDIX A

5. At first, it was difficult to control the pressure of the pen, but latter I improved

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

6. I had complete control of the pressure I applied with the pen at all times

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

7. It was very difficult to control the pressure I applied with the pen at all times

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

8. It was easy to control the pressure of the pen at the beginning, but latter it got worse

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

9. Today you were exposed to three main types of menus. Rate them, by putting a number beside each one, according to ...

...ease of use

Flag Moving Flag Bulls eye Pie Rotating Pie

...visual attractiveness

Flag Moving Flag Bulls eye Pie Rotating Pie

...overall impression

Flag Moving Flag Bulls eye Pie Rotating Pie

APPENDIX B

10. After a few tries, I knew how to use the ...

...Flag

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

...Moving Flag

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

...Bulls Eye

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

...Pie

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

...Rotating Pie

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

APPENDIX B

- 11. Please tell us your final thoughts and/or impressions you may have about your experience in this session.**

Thank you! This completes the session.

10 Appendix B:

Zlider's Study

Survey Form

Post Questionnaire

1. What is your gender?

Male

Female

2. What is your age?

18-24

25-34

35-44

45-60

3. Have you ever used a Tablet PC?

Yes

No

4. Have you ever used a PocketPC, Palm© or other pen-based device?

Yes

No

5. Today you were exposed to three different techniques. Please read the following sentences and rate the techniques by putting a number beside them (e.g. 1st, 2nd, and 3rd)

With which technique you think made the most errors?

Pen Pressure

Force Button

Shift/Ctrl Keys

APPENDIX B

Which technique you think is easiest to use?

Pen Pressure
Force Button
Shift/Ctrl Keys

Overall, which technique do you prefer to use?

Pen Pressure
Force Button
Shift/Ctrl Keys

6. After a few tries, I knew how to use the ...

Pen Pressure

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

Force Button

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

Shift/Ctrl Keys

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

7. Please tell us your final thoughts and/or impressions you may have about your experience in this study.

Thank you! This completes the session.

11 Appendix C:

Pressure Marks' Study

Survey Form

Pressure Marks - Post Questionnaire

User:

1. Sex. Male Female

2. Age 18-24 25-34 35-44 45-60

3. Have you ever used a Tablet PC or similar device? Yes No

4. Do you draw? Yes No

5. Overall, I found  easy to do

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

6. Overall, I found  easy to do

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

7. Overall, I found  easy to do

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

8. Overall, I found  easy to do

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

9. Overall, I found  easy to do

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

10. Overall, I found  easy to do

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

11. Overall, I found  easy to do

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

12. Overall, I found  easy to do

Strongly Disagree	Disagree	Somewhat Disagree	Neither Agree/Disagree	Somewhat Agree	Agree	Strongly Agree

13. Please tell us your final thoughts and/or impressions you may have about your experience in this study.

Thank you! This completes the session.