

Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces

Yang Li¹ Ken Hinckley² Zhiwei Guan³ James A. Landay^{4,5}

¹ Group for User Interface Research
Computer Science Division, EECS
University of California, Berkeley
Berkeley, CA 94720-1776
yangli@cs.berkeley.edu

² Microsoft Research
One Microsoft Way
Redmond, WA 98052
kenh@microsoft.com

³ Department of Technical Communication
University of Washington
Seattle, WA 98195-2195
zguan@u.washington.edu

⁴ DUB Group
Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
landay@cs.washington.edu

⁵ Intel Research Seattle
1100 NE 45th Street
Suite 600
Seattle, WA 98105
james.a.landay@intel.com

Abstract

Inking and gesturing are two central tasks in pen-based user interfaces. Switching between modes for entry of uninterpreted ink and entry of gestures is required by many pen-based user interfaces. Without an appropriate mode switching technique, pen-based interactions in such situations may be inefficient and cumbersome. In this paper, we investigate five techniques for switching between ink and gesture modes in pen interfaces, including a pen-pressure based mode switching technique that allows implicit mode transition. A quantitative experimental study was conducted to evaluate the performance of these techniques. The results suggest that pressing a button with the non-preferred hand offers the fastest performance, while the technique of holding the pen still is significantly slower and more prone to error than the other techniques. Pressure, while promising, did not perform as well as the non-preferred hand button with our current implementation.

Categories & Subject Descriptors: H5.2. [User Interfaces]: Interaction styles, I.3.6 [Methodology and Techniques]: Interaction techniques

General Terms: Design, Experimentation, Human Factors, Performance.

Keywords: Pen interfaces, mode switching, mode errors, ink, gestures.

INTRODUCTION

Pen interaction techniques have demonstrated much promise in two general areas: raw ink intended for interpretation by a person [18], and gestures intended for immediate interpretation by a computer [12]. Inking allows

data entry, while gestures are drawn to issue commands for manipulating the data. For example, a note-taking tool allows people to take notes via natural handwriting and to edit by making copy editing gestures to perform commands. Some note-taking tools, such as Windows Journal, support scratching out a word to delete it. However, if this gesture is performed on a system without any explicit signal for gestures, then naturally occurring ink strokes may be falsely classified as gestures, such as when the user shades part of a diagram. In some cases such problems can be averted, but in general it is extremely difficult, and it may often be intractable for computers to “understand” natural ink input and always “do the right thing” unless given some explicit guidance by the user.

Consequently, many pen interfaces in the literature [9, 12] support two distinct modes: ink mode and gesture mode. A mode switch includes switching into a gesture mode and switching back to inking mode after completing a gesture. Users may have to frequently transition between these modes, and a slow or ineffectual ink/gesture mode switching technique may become a major bottleneck in the usability of a pen-based interface.

Modes have long been recognized as a significant source of errors, confusion, unnecessary restrictions, and complexity in interfaces [15]. Efforts have been made to alleviate modes in pen-based interactions by processing both ink and gesture input in a single mode based on pure inference-based approaches [3] or with the help of user mediation [16]. However, as illustrated above, current technologies are still extremely limited in their ability to discern gestures from other ink strokes in freeform sketches, so typically modeless input of both gestures and ink can only be realized in narrow and highly-constrained situations in which sketches are well classified or where frequent user mediation is required, which can distract users and compromise performance. Other solutions rely heavily on particular assumptions about how the pen interface is structured or how basic events (tap, tap-and-hold, drag, etc.) are handled, making the techniques difficult to apply across a wide variety of pen-based applications. Simple and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2005, April 2–7, 2005, Portland, Oregon, USA.
Copyright 2005 ACM 1-58113-998-5/05/0004...\$5.00.

effective explicit mode switching techniques could provide pen users with consistent mechanisms that are applicable across a wide variety of applications.

Rather than trying to eliminate ink and gesture modes, we focus here on how to improve mode switching techniques, so that these mode transitions can be quick, predictable, and minimally disruptive. Previous work has investigated mode switching techniques in the context of text editing tasks [17], and a number of interaction techniques have been reported for switching between inking and gesturing [9, 11, 12]. However, there has not been a quantitative and systematic investigation into the performance of these techniques. Our study is the first to offer a quantitative analysis of methods for switching between ink and gesture modes. We implemented and examined five mode switching techniques for this analysis. Two of them, using a barrel button on a pen and holding a pen still, are standard mode switching techniques in existing pen-based user interfaces. The other three, using pen pressure, using a non-preferred hand and using the eraser end of a pen, have not been deeply explored for mode switching.

Our experimental design was based on the experimental paradigm of Dillon et al. for measuring the “true cost” of command selection [4]. Mode switching is analogous to selecting a “command” with the user incurring a cost to activate a mode, and a second cost to deactivate that mode and return to the previous state. We employ the same strategy of carefully comparing completion time for a baseline task without any mode transitions to an experimental task that does include mode transitions. In this way the experiment captures all influences of mode switching on task completion time for an experimental task.

We designed an experimental task of making a crossing stroke [1, 8] through an oriented pie slice (see Figure 2). Users were prompted to make each stroke as either an ink stroke or as a gesture stroke. This is an experimental task that is meant to capture the quick and fluid feel of a pen interface that includes transitions between ink strokes and gestures. For the case of ink, the task is similar to a user annotating a document or crossing out previously written notes with a simple straight line. For the case of gestures, the task is similar to activating a marking menu [8] command with a stroke in a particular direction.

In the following sections of this paper, we first describe the five mode switching techniques that were investigated in the experiment. We then describe the design of the experiment. This is followed by a discussion of experimental results, which includes the implications of improving mode switching techniques for pen-based UIs.

MODE SWITCHING TECHNIQUES

The techniques are discussed in order from the most common to the more novel.

Technique 1: Pressing Barrel Button

Pressing the stylus’ barrel button for gesturing is a standard technique in many existing pen-based applications [9, 11].

The barrel button also serves as a right-click equivalent. Before gesturing, users press the barrel button while the pen is in the air and keep it pressed until drawing is started. After a pen down event, the gesture mode is not disengaged until the later of a pen up or the barrel button is released.

Technique 2: Press and Hold

Similar to Technique 1, the action of *Press* and *Hold* is also a standard technique for gesturing or as a right-click equivalent [9-11]. This requires a user to press the tip of the pen onto the tablet, hold it still until some mode change feedback appears, and then while the mode change feedback still appears on the screen, the user can either lift the pen to bring up a popup menu or move the pen to draw a gesture. This technique leverages temporal information for switching modes, which is useful on devices where few input devices are available, e.g., a PDA or mobile phone.

We implemented this technique based on the Hold technique available on Tablet PCs [11]. To draw a gesture, there are three phases to performing a Press and Hold action for a mode switch. Once the pen touches the tablet, it enters the *Hold Detection* phase in which the bounding box size of the pen movement must be kept less than a *pen travel threshold* to be considered “still.” In our implementation, the Hold Detection phase was 1 second long and the pen travel threshold was 1.5mm. If the pen is held longer than 1 second, a red circle appears around the pen tip and it enters the *Hold Through* phase in which the user should move the pen to exceed the pen travel threshold and draw a gesture. The Hold Through phase was 800ms long in our implementation. This phase enables a user to stay in the ink mode by holding through this phase, if the user did not actually intend to draw a gesture.

The gesture mode is engaged when a pen is moved out of the pen travel threshold in the Hold Through phase and automatically disengaged when the pen is lifted after drawing a gesture.

Technique 3: Using Non-Preferred Hand

Two-handed interactions have been studied extensively [6, 7]. An appropriate design of two-handed interactions allows the division of labor across two hands and the reduction of task completion time by simultaneously carrying out two subtasks. Some of the previous work investigated using two-handed interactions for compound tasks such as drawing and selection [4, 7]. In these tasks, the non-preferred hand is used to set the context, e.g., the color, for drawing. Based on this previous work, we implemented a technique for using the non-preferred hand for selecting a gesture mode.

A physical button mounted at the top-left corner of a Tablet PC screen was employed as a *mode switching button*. Pressing the mode switching button using a non-preferred hand indicates a mode switch and the gesture mode is engaged when the pen is off the tablet. After a pen down event, the gesture mode is not disengaged until the later of a pen up or the mode switching button is released.

Technique 4: Pressure-Based Mode Switching

Pen pressure sensitive inputs are available on many tablet devices. However, except when used for drawing and image manipulation, such as adjusting the thickness of strokes, pen pressure has rarely been employed until recently. The LEAN [13] system employed pressure as a feature for gesture recognition. Ramos et al. indicate that dividing the pressure space into six levels or less produces the best user performance [14].

Here, we looked at using the stylus pressure to achieve an implicit mode transition. Under the assumption that inking occupies most of the user’s time in a pen-based interface, we intended to preserve the normal (middle) pressure space for inking and leave the heavy spectrum for gesturing. This was also based on the observation that the low end spectrum is harder to control [14].

To find out the right pressure spectrum for our gesture mode, we ran a preliminary experiment with four people, two males and two females. In the experiment, our participants were asked to draw with their normal pressure or with heavier pressure alternately on a Tablet PC, which has 256 levels of pressure. We collected the pressure data for 120 straight drawings varying along different directions from each participant. There was no visual feedback other than the pressure sensitive ink thickness.

Although the sample size was small, the data still gave us useful information for making a quick implementation of this idea. The collected data indicated there was a statistically significant difference on the *maximum average pressure* of a stroke between the normal and heavy pressure conditions ($t_3=6.486$, $p<0.007$). The maximum average pressure is defined as the maximum of the *average pressure* at each time while drawing the stroke. The average pressure at the time t_i is measured as:

$$AP_{t_i} = \frac{1}{i+1} \sum_{j=0}^i P_j, P_j \text{ is the pressure at the time } t_j$$

Consequently, as a rough implementation of this technique, we simply detect if the average pressure at a time exceeds a threshold, which was 190 based on the samples from this preliminary study (94% of the maximum average pressure in the normal condition was lower than 190 and 86% in the

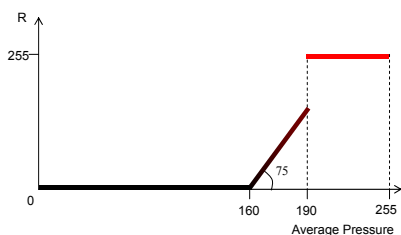


Figure 1. The red element R of the pen ink feedback is linearly interpolated so that the ink color continuously changes from Black, i.e., RGB (0, 0, 0) towards Red, i.e., RGB (255, 0, 0) once the average pressure exceeds the lower bound 160.

heavy condition was higher than 190). Once the average pressure reaches the threshold, the gesture mode is engaged. The gesture mode is automatically disengaged when the pen is lifted. We also observed that this simple measure was immune to quick “heavy” tapping, which we originally thought might cause heavy pressure values but it turned out that this resulted in relatively small average pressure values.

In addition to the pressure sensitive ink thickness, we designed continuous color feedback to reflect the change in pressure. Rather than directly jumping from black to red¹, a linear interpolated color was used once the average pressure exceeded a lower bound. This helps a user to realize that the pressure is getting close to the threshold. At the same time, an abrupt change of color is still kept as a salient signal so that a mode change can be easily perceived (see Figure 1).

Technique 5: Using the Eraser End of a Pen

The eraser end of a pen has been used for deleting objects [11] as well as other functions, such as music playback [5]. Here, we explored a more general use of the eraser end for drawing gestures. The gesture mode is engaged when a user inverts a pen and disengaged when the pen is reverted.

In the rest part of the paper, we will use the abbreviations of *BarrelButton*, *Hold*, *NonPrefHand*, *Pressure* and *Eraser* to refer to these five techniques, respectively.

EXPERIMENT

We conducted a quantitative experiment to investigate the performance of each of these five mode switching techniques.

Experimental Design

A pie-crossing task was devised as an abstraction of gesturing and inking. Based on this, the baseline task and the compound task were designed.

Pie-Crossing Task

To capture the nature of sketch-based interactions, which are normally informal and fluid, we designed a pie-crossing task as an abstraction of the action of gesturing and inking (see Figure 2). A pie slice is shown with one of eight orientations corresponding to the eight major geographical directions. A participant was required to quickly cross a slice from its inner edge towards its outer edge according to a target’s orientation. This design examines the drawing of various directions without requiring precise positioning and careful alignment by participants. This design also captures a realistic use scenario of gestures, i.e., marking menus [8], where users can cross a series of objects with marks to perform different commands.

¹ In our implementation, all five techniques display a red dot cursor for gesture mode and a black dot cursor for ink mode at the position of the pen tip. They also have the common visual feedback of a red pen trail for gestures and a black pen trail for ink.

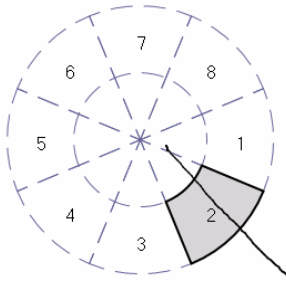


Figure 2. The pie-crossing task is a simplification of gesturing and inking. Each slice covers 45 degrees and eight orientations are allowed. For example, the orientation of the filled slice, Slice 2, is 45 degrees. A successful crossing requires a stroke to cross a target slice as well as to have the right orientation. For example, to cross Slice 2, the orientation of a stroke has to be between 22.5 and 67.5 degrees. The numbers and the dashed lines are not shown in the experiment (see Figures 3 and 4).

We also simplified the semantics of the ink mode and the gesture mode by providing no further semantic feedback than stroke color. This simplification allowed us to efficiently capture the effect of mode switching techniques on overall performance rather than being distracted by other issues, such as gesture interpretation.

Baseline and Compound Task

Based on this basic element, two kinds of tasks were devised as trial blocks: a *baseline task* and a *compound task*. In a baseline task, five unfilled black slices with a certain orientation were shown on each screen (see Figure 3). A participant only needed to perform a pie-crossing task for each slice and no mode switch was required.

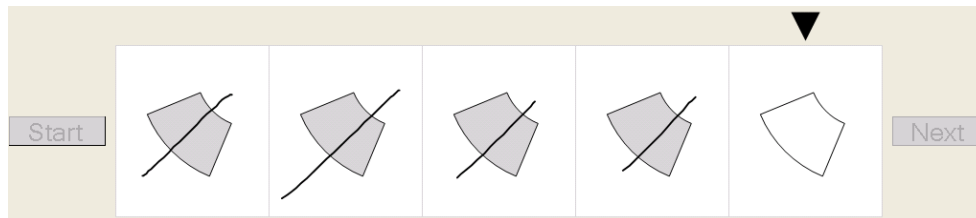


Figure 3. The fourth screen of a baseline task. All slices are in black. A participant starts drawing by first clicking the Start button (which is disabled once clicked) and the tool starts timing. After finishing crossing all five slices from left to right sequentially, timing stops and the Next button becomes enabled. A participant clicks the Next button to go to the next screen. A target slice is centered in a target box pointed at by the arrow. Once a target slice is successfully crossed, the pie becomes filled.

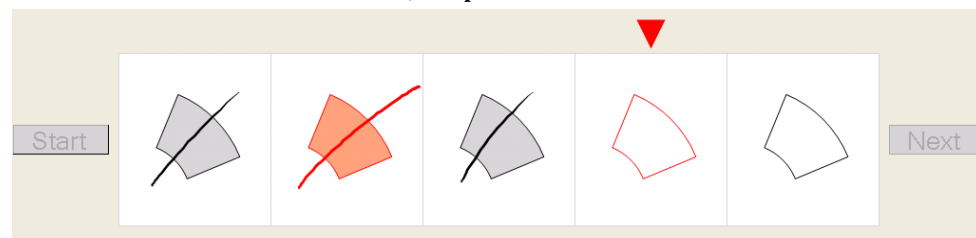


Figure 4. The eighth (last) screen of a compound task. Five slices are presented in black and red alternatively, which requires a participant to cross a slice with the slice's color. The participant needs to switch modes to draw lines with different colors. The gesture mode must be engaged to draw in red.

In a compound task, slices were presented alternatively in two different colors, black and red, starting with a black slice (see Figure 4). A participant was required to cross a black slice with a black line and a red slice with a red line. Thus, a participant was required to switch color according to the color of a target pie using a mode switching technique. Both the baseline task and the compound task included eight screens, each with the slices shown in one of the eight orientations of Figure 2. So in total, forty pie crossings were required in a baseline or a compound task.

Here, a mode switch includes switching into gesture mode and switching back to ink mode. In a compound task, each screen requires two mode switches. Thus, sixteen mode switches were required in a compound task.

Error Classification

Errors in this experiment were classified into three categories: *mode errors*, *crossing errors*, and *out-of-target errors*. Mode errors include *Mode-In* errors that happen when drawing black lines on red targets and *Mode-Out* errors that happen when drawing red lines on black targets. A crossing error takes place when a drawing does not cross a target slice, has the wrong orientation or both. An out-of-target error happens when a drawing falls entirely out of a target box, which can be caused when a participant draws on or taps the area outside of the target box.

If an error happens, the experimental tool asks participants to correct their drawing by beeping and highlighting the target box with a bold rectangle.

Procedure

The experiment included a training phase for the baseline tasks, five experimental sessions with one session for each

technique, and a post-study questionnaire. The experiment took about 80 minutes in total. A 5x5 Latin Square was used to counterbalance the order of the techniques. Each session was divided into two parts. The first part involved learning to use a mode switching technique and extensive practice. The second part was the experimental phase in which a participant was given 9 blocks of trials. The first block was a *baseline task* and then a *compound task*, alternating until the ninth block ended with a *baseline task*. A participant could take a break between blocks. In total, the experiment consisted of:

- 15 participants x
 - 5 mode switching techniques x
 - 9 block of trials x
 - 8 screens (8 orientations) x
 - 5 pie-crossing tasks
- = 27,000 pie-crossing tasks

4,800 mode switches had to be performed during these tasks. We collected data on all events in the experiment including pen entering/leaving range, pen down/up, pen draw, barrel button down/up, mode switching button down/up as well as the stylus' current position and pressure value and the time of each event. The high level events of gesture mode engaged or disengaged were also recorded.

Performance Measures

The dependent variables were the *mode switching time*, the total number of errors in a compound task, and the subjective preference of participants. The first two blocks in the experimental phase were for warming up and the data of the seven following blocks were used for analysis.

The timing for each screen is started when the Start button is clicked and automatically ended when the last pie is crossed and the pen is lifted. This duration is divided into three cycles. The first cycle starts when the Start button is clicked and ends when the first pie is crossed. The second cycle starts right after the first cycle and ends after the third pie is crossed. This is followed by the third cycle, which includes crossing the last two slices. Therefore, one target needs to be crossed in the first cycle and two targets need to be crossed in each of the second and the third cycles. We call cycle 2 and 3 *full cycles* and cycle one the *start cycle*. In a *compound task*, a full cycle contains a complete mode switch process including switching into gesture mode and switching back to ink mode.

The *mode switching time* for each of the three compound blocks was computed by subtracting the mean of the two adjacent baseline tasks' *average cycle durations* from the compound block's *average cycle duration*. Average cycle duration was the mean duration of all correct full cycles in a block.

A start cycle guarantees that there is a pie-crossing task performed in ink mode before a full cycle. Having a pair of gesturing and inking in a full cycle enabled us to investigate how a mode switching technique also affects the following inking after the gesture mode is disengaged. For

the BarrelButton, the Eraser and the NonPrefHand techniques, this can also detect if a mode is successfully switched back to the ink mode.

After participants finished all five sessions, they rated the five techniques in terms of ease of learning, ease of use, accuracy, operation speed, eye fatigue and hand fatigue.

Apparatus

The experiment was conducted using a Toshiba Portege Tablet PC running under Windows XP Tablet PC Edition. This Tablet PC has a Pentium III M 1.33GHz processor and 1GB of RAM. The device has a 12.1" LCD display and a digitizer with 256 levels of pressure. The experiment was conducted at a resolution of 1024x768 pixels. The stylus is equipped with a barrel button and an eraser.

Participants

Fifteen adult volunteers were recruited from a university campus. They were graduate and undergraduate students, two female and thirteen male, 20 to 30 years old. They received \$25 for their participation. None of these participants were involved in the preliminary experiment for determining the pressure threshold.

Two of the fifteen participants were left-handed and only one of them felt that using his non-preferred hand to press the mode switching button was a little inconvenient. Even so, he still gave a highest preference rating for the NonPrefHand. 20% of participants had experience using the BarrelButton, and 13% of participants had experience using the Hold, primarily from using PDA devices. 13% of participants had experience in adjusting stylus pressure to control the ink thickness while drawing, but none of them had experience using stylus pressure for mode switching. None of them had experience with the NonPrefHand technique.

Experimental Software

The experimental software was developed in C# using the Microsoft Tablet PC Platform SDK. The software presented tasks to participants and logged all primitive and high level events. It also automatically generated tables of durations and errors as pure text files so that they could be easily imported into SPSS and MS Excel for analysis.

RESULTS

Repeated measure variance analyses and post hoc pairwise comparisons were performed on the mode switching time and the subjective preference. The number of errors was analyzed using a chi-square analysis.

Performance Stability over Experimental Blocks

There was no obvious learning effect on the mode switching time and the error rate between the three experimental *compound* blocks. Participants not only kept their time performance stable ($F_{2,28}=.248, p=.782$), but also had no statistically significant change in their error rates ($\chi^2_2=3.137, p=.208$).

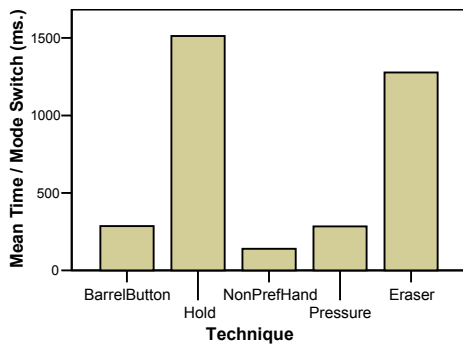


Figure 5. The mean mode switching time of five techniques.

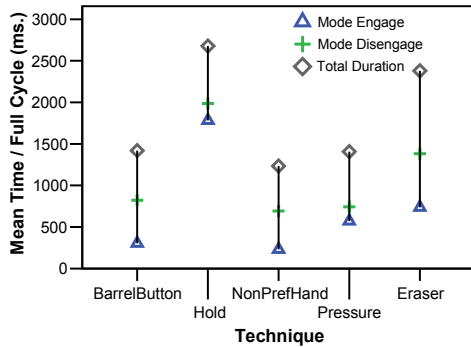


Figure 6. The mean duration of a full cycle in a compound task (before subtraction). The time of mode engagement and disengagement are also shown.

This result can be accredited to the extensive practice in the training phase and the beginning 2 blocks (one baseline block and one compound block) for warming up. After these preliminary blocks, participants had highly acquired the five mode switching techniques and stably applied them in the following experimental blocks.

There was also no significant joint effect of mode switching techniques and experimental blocks (technique \times block) on the mode switching time ($F_{8,112}=1.655$, $p=.117$) and the error rates ($\chi^2_8=5.165$, $p=.740$), respectively. We therefore used the aggregated mean mode switching times and error numbers over three compound blocks as the value of the mode switching time and the error number, respectively, to analyze the differences between techniques.

Mode Switch Time

The mean mode switching time was analyzed in an analysis of variance with five techniques (BarrelButton vs. Hold vs. NonPrefHand vs. Pressure vs. Eraser) as a within-subject factor². The sphericity assumption was not met so the Huynh-Feldt correction was applied. The main effect of mode switching techniques was significant with the

² Before analyzing the variance, an analysis of normality found that the data of three techniques were moderately skewed, which can be remedied by applying a square root transformation. The variance analysis on the transformed data showed the same statistical significance results as on the original data. For simplicity, we based our analysis on the original data space.

corrected degree of freedom, $F_{1,518,21,247}=97.78$, $p<.001$, (see Figure 5).

Post-hoc comparisons were performed using the Bonferroni adjustment for multiple comparisons. The mode switching time of using the NonPrefHand was the smallest with a mean of 139ms, which was found to have no significant difference ($p=.106$) from using Pressure to switch modes (Mean=284ms). There was no significant difference found on mode switching time between using Pressure and the BarrelButton (Mean=286ms). Although these three techniques can be grouped on their similar efficiency in switching modes, they are significantly faster than the other two techniques (Hold and Eraser) tested in the experiment ($p<.001$). The mode switching time of using the Eraser technique had a sharp increase of over a second with a mean of 1277ms, but it was still significantly lower than the mean of using the Hold technique (Mean=1513ms) ($p<.001$).

Figure 6 shows the completion time of a full cycle in a compound task, without subtraction of the baseline time, as well as the times when a gesture mode becomes engaged or disengaged. These measurements are discussed in detail in the following sections.

Error Analysis

The number of errors was analyzed by using the chi-square test and post hoc analysis based on standardized residuals. On the frequency of making errors, we found that there is a statistically significant relationship between the number of errors and the technique ($\chi^2_4=61.903$, $p<.001$) (see Figure 7). The NonPrefHand appears to support accurate mode switches resulting in very few errors with a standardized residual at -4 . The Eraser technique also resulted in few errors ($z=-2.8$). Many more errors ($z=6$) were made with the Hold technique.

We further analyzed the frequencies of each type of error when using the five different techniques. Since several techniques did not lead to any Out-Of-Target errors, to satisfy the chi-square's requirements that expected frequency should not be less than 5, we conducted the chi-square analysis of five techniques on only three types of errors (Mode-In, Mode-Out, and Crossing errors).

The results indicate that there is a statistically significant relationship between the five techniques and three types of errors ($\chi^2_8=81.039$, $p<.001$). As seen in Figure 7, the mode errors, including Mode-In and Mode-Out errors, were the main errors found for all techniques except the Eraser. The Hold and the Pressure techniques led to fewer Crossing errors, while the Eraser resulted in many more Crossing errors ($z=5.1$).

In terms of Mode-In and Mode-Out errors (see Figure 8), the Hold technique led to more Mode-In errors ($z=2.6$), but less Mode-Out errors ($z=-3.4$) than the expected frequencies. When using the Eraser, fewer Mode-In errors were made than would be expected. The Pressure technique led to more Mode-Out errors.

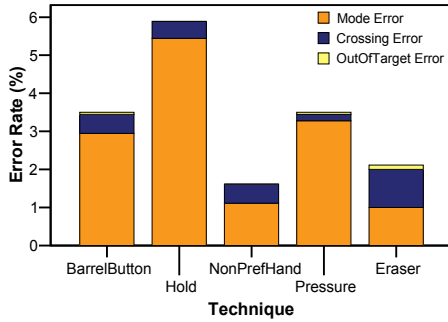


Figure 7. The error rate on each pie-crossing. The mode errors included both Mode-In and Mode-Out errors.

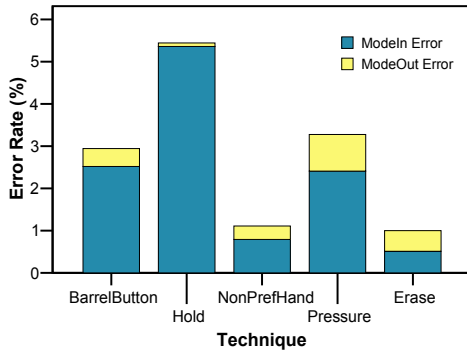


Figure 8. The occurrence rate of Mode-In and Mode-Out errors on each pie-crossing, which is weighted by the frequency of inking and gesturing, respectively. 40% of pie-crossings were for gesturing, while 60% were for inking.

Although the Eraser was not the most accurate technique, it led to the least number of mode errors, with Mode-In and Mode-Out errors evenly distributed. As mentioned earlier, the Eraser caused the most Crossing errors. It was not a surprise to see that the Eraser caused the most Out-Of-Target errors since it was observed that our participants dropped the pen or accidentally pressed the pen when flipping the pen.

The Hold technique has the highest mode error rate in our experiment, although 13% of our participants had some experience using this technique on PDAs, as mentioned earlier. The slippery LCD screen made it difficult to hold the pen in a 1.5mm range for a second. We often saw our participants slide the pen too much while performing this technique. Among these mode errors, most were Mode-In errors. This means that the temporal and spatial constraints in our implementation made it much harder to use this technique to switch to a gesture mode than switch back to an ink mode. However, on the other hand, simply loosening these thresholds might increase the Mode-Out errors while reducing Mode-In errors. We discuss this technique further in the following sections.

Unlike the other techniques, the Pressure technique is the only technique that allows a gesture mode to be engaged while drawing. By watching the continuous color feedback of ink while crossing, the participants can tell if the

pressure is increasing and if a gesture mode is engaged. This may have required a participant to focus on the pen trails while drawing, thus making Crossing errors less likely (see Figure 7).

Subjective Preference

The five techniques were rated by the participants on six dimensions: ease of learning, ease of use, accuracy, operation speed, and fatigue of the eyes and hands (see Table 1). Participants were required to rate these techniques on a 5-point scale (5 for best, and 1 for worst). There was a significant difference found in the effect of the five techniques on the overall preference rating ($F_{4,56}=18.63$, $p<.001$).

Table 1. The participants' preferences for each technique.

Dimension	Barrel-Button	Hold	NonPref-Hand	Pressure	Eraser
Learning	4.4	3.5	4.7	3.5	4.2
Use	3.7	2.2	4.1	3.4	2.4
Accuracy	3.7	2.9	4.6	3.3	3.6
Speed	4	1.7	4.5	4	1.9
Eye fatigue	4.1	3.3	4.4	3.9	4.2
Hand fatigue	3.5	3.3	4.1	3.3	2.1

Tukey's post hoc pairwise analysis showed that the BarrelButton, the NonPrefHand, and Pressure were rated significantly higher than the other two techniques ($p<.05$). The NonPrefHand was rated the best overall, followed by the BarrelButton, and there was a significant difference between the two. The preference level for the Pressure technique was found to be significantly lower than for the NonPrefHand, but Pressure had no significant difference with the BarrelButton. The Eraser was rated lower than all but the Hold technique, although there was no significant difference between them ($p>.05$).

The subjective preference of these five techniques is fairly consistent with the time and error performance, but has some inconsistency with respect to the error rate results of the Eraser. The Eraser has a comparatively low overall rating although the error rates were the second best. One reason was that our participants commented that they disliked flipping the pen frequently while drawing. The extremely low rating on hand fatigue of the Eraser also implies this.

DISCUSSION

We conducted keystroke level analyses on the results. The results are also discussed in terms of implications for the design of mode-switching techniques in pen-based applications.

Keystroke Level Analysis

The duration of a full cycle can be decomposed as the following:

$$T_{FullCycle} = T_{P1} + T_{C1} + T_{P2} + T_{C2}$$

where T_{P_i} was the positioning time before crossing the i th target and T_{C_i} was the time for crossing the i th target (see Figure 9 and Table 2).

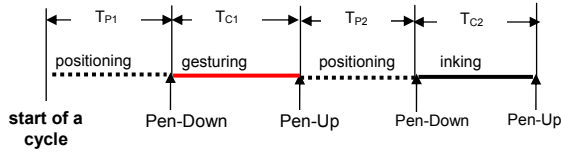


Figure 9. A decomposition of a full cycle.

Table 2. The cycle decomposition. The duration is in ms. T_{Engage} is the mean duration from the start of the cycle to the moment that a gesture mode is engaged. $T_{Gesture}$ is the mean duration of being in a gesture mode.

Task	Baseline Task	Compound Task				
Tech.	N/A	Barrel-Button	Hold	NonPref-Hand	Pressure	Eraser
T_{P1}	362	506	644	427	396	1023
T_{C1}	198	243	1343	213	347	193
T_{P2}	364	462	467	389	441	919
T_{C2}	198	208	223	205	223	243
T_{Engage}	N/A	304	1781	230	572	737
$T_{Gesture}$	N/A	518	205	463	172	645

The T_{C2} row shows that these five techniques had no significant impact on the inking after a gesture mode is disengaged because the T_{C2} of each technique was very close to the T_{C2} of the baseline task. The differences of each phase of a cycle across these columns are mostly caused by a mode switch using the different techniques. We analyzed this difference based on the temporal models of each technique.

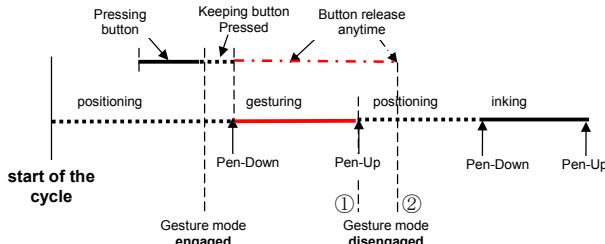


Figure 10. The temporal model of the NonPrefHand and the BarrelButton techniques.

Using BarrelButton and NonPrefHand

The BarrelButton and NonPrefHand share the same temporal model (see Figure 10). Neither of them requires significant hand movement to hit the mode switching button or the barrel button. The difference between these two techniques is that the BarrelButton uses one hand for two subtasks, drawing and mode switching, while the NonPrefHand uses two hands for these two subtasks. The result that NonPrefHand has better performance than BarrelButton can be explained as partly due to temporal overlap of two subtasks as described in [7]. By subtracting the T_{P1} of the baseline task from the T_{P1} of the compound task, we get an estimate of the time for engaging a gesture

mode, which in the case of the NonPrefHand took about 65ms and for the BarrelButton it took about 144ms.

Although the participants were informed that they could release the button once they started drawing, the decomposed durations indicated that they mostly kept the button pressed until a gesture was completely drawn.

The positive results of the NonPrefHand were based on a tablet button that was not in an optimal position. This should help motivate tablet manufacturers to design tablet buttons that are positioned properly for this type of situation and for use while holding a tablet. In fact, an early 90s tablet prototype from DEC (Lectrice [2]) provided buttons for just this scenario.

Using Hold

We originally expected that the T_{P1} of the Hold technique should be close to the baseline's T_{P1} because the action of engaging a gesture mode is started after the pen is pressed on the tablet. However, we found there was 282ms extra time in the Hold's T_{P1} compared to the baseline's. After re-analyzing the effort for a user to perform a Hold-based mode switch, we found it was consistent with the hand movement. When the hand moved quickly to the next target, some extra effort was needed to slow down the movement and to hold the pen steady (see Figure 11). This effort can probably take even longer if the preceding hand movement was very fast. We call this time effort the preparation time (282ms).

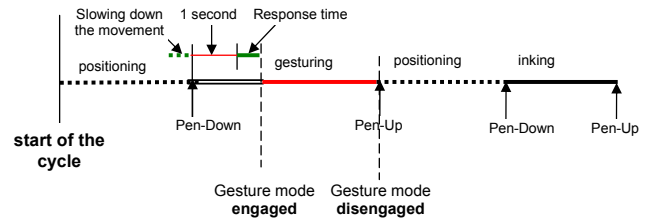


Figure 11. The temporal model of the Hold technique.

Table 2 also shows that there was a short interval after the Hold Detection phase (1 second long) and before the gesturing phase (see Figure 11). This was the time it took for a participant to respond to the feedback (a red circle) and move the pen. This response time can be calculated as:

$$T_{response} = T_{Engage} - T_{P1} - 1000 = 1781 - 644 - 1000 = 137ms$$

Using Pressure

From Table 2, we can see that Pressure took the least time before starting drawing a gesture among these techniques. The time difference with the baseline's T_{P1} was 34ms. In other words, there was little extra effort required in this phase while using the Pressure technique.

After the pen was down, two actions were performed simultaneously: crossing the pie and increasing the pressure to the threshold (see Figure 12).

By subtracting the Pressure's T_{P1} from its T_{Engage} , we get an estimate of the total time for increasing the pressure to the threshold and the time for drawing in this period, which is

176ms. By comparing the baseline's T_{C1} and the Pressure's T_{C1} , we can see it took longer to draw with higher pressure.

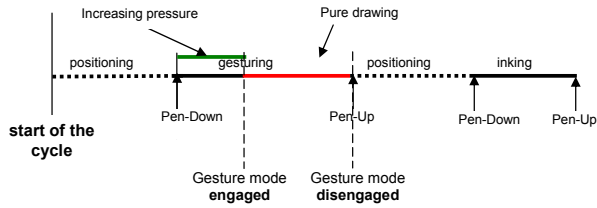


Figure 12. The temporal model of the Pressure technique.

Using the Eraser

The temporal model of the Eraser is shown in Figure 13. The Eraser's T_{C1} was very close to the baseline's T_{C1} (see Table 2). This means there was no extra effort required while drawing a gesture, although a participant needed to flip the pen, which took about 737ms for engaging the gesture mode. This tells us that the Eraser technique should be used in situations where mode switching is less frequent and the gesture mode needs to be maintained for a relatively long time.

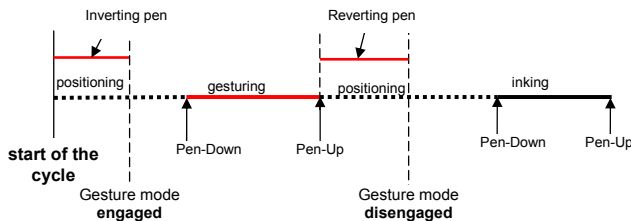


Figure 13. The temporal model of the Eraser technique.

We apply subtraction again to get an estimate for inverting and reverting the pen. The time for inverting the pen was 661ms based on the T_{P1} difference and the time for reverting the pen was 555ms based on the T_{P2} difference. At first glance, they should be the same since it requires a similar effort to flip the pen both ways. We found the 100ms difference was because the eraser end was too blunt for drawing. We often observed that the tablet did not detect the eraser's contact with the screen. Participants had to press the eraser extra hard, which consumed extra time.

Validity of Subtraction

Some of our analyses of the time performance were based on a subtraction methodology that is similar to that of Dillion et al. [4]. The subtraction was conducted under the assumption that both baseline and compound tasks required a similar amount of movement effort for crossing and positioning. Our pie-crossing task captures the informal and quick nature of sketch-based interactions. However, it may have relaxed the constraint on drawing of the point-connecting task in Dillion's experiment. But, the data of our experiment indicated that the difference on the mean total movement distance of a full cycle, including two positionings and two crossings, between baseline and compound tasks, is less than 20 pixels. This is very small compared to the mean total movement distance of 580 pixels. This indicates that our participants expended a

similar effort in terms of pure crossing and positioning in both the baseline and compound tasks.

Improving Pressure-based Mode Switching

Our preliminary design of the Pressure-based technique shows a moderate level of performance among the five techniques. The experimental data indicates this technique can be improved by using a personalized pressure space.

As mentioned earlier, we ran a preliminary study to collect pressure data from four participants, which helped us in determining the threshold for separating gestures and inking as well as a linear interpolation function for continuous color feedback. Although the pressure space of our experimental Tablet PC is from 0 to 255, we noticed that our participants have different pressure spaces. This is consistent with the intuition that different people might draw with different force. In particular, one female participant in the preliminary study had a much lower overall pressure than others although the pressure of her heavy drawing was still significantly different from the pressure of her normal drawing. This implies that the threshold should be adapted to the user. However, we used a uniform threshold of 190 in our experiment because we thought people would adapt to this threshold.

The mode switching experimental results indicated our participants did indeed adapt to this threshold. But, it was still not enough to overcome the differences in their inherent pressure spaces. We found that participants who had a lower pressure space tended to make more Mode-In errors and less Mode-Out errors, and vice versa. For example, a participant whose mean average pressure was 91 made the most Mode-In errors, while another participant whose mean average pressure was 164 made the most Mode-Out errors. It was further confirmed by the high negative correlation between the number of Mode-In errors and the number of Mode-Out errors ($r^2=.56$, $p<.01$). This result confirms our earlier concern with using a standard pressure threshold for all participants.

Improving Hold-based Mode Switching

The Hold technique had the worst performance in our experiment. However, as mentioned earlier, since it requires the least hardware support, it has been widely used in many mobile devices, such as the Palm Pilot and the PocketPC. Thus, it is worth improving this technique.

The keystroke level analysis above showed that there was a 282ms preparation time before the pen is pressed. To reduce this time, we devised a revised Hold called *Stroke+Hold*. This technique allows a user to perform a Hold at any point of a drawing rather than only at the starting point. This can reduce the preparation time. Although it may introduce some extra time for slowing down the pen in the middle, it is easier to hold the pen still in the middle or end of a drawing than at the moment of touching the slippery tablet. All of these factors can help to improve the overall performance of this technique. This new technique has recently been added to DENIM [9] for

selection. It seems to work better than the traditional Hold, although no formal evaluation has been done on it.

Improving Synchronization

Since the BarrelButton and the NonPrefHand techniques both require a button to be pressed while the pen is down, synchronization is a main cause of mode errors. While using the BarrelButton, the data showed that 93% of Mode-In errors were caused by late button pressing and 7% of them were caused by early button releasing. If a barrel button is allowed to be pressed within the first 37ms after the pen is down for gesturing, which is a small percentage of time compared to the whole drawing time of 243ms (see Table 2), 50% of the Mode-In errors can be eliminated. Thus, the BarrelButton technique can be greatly improved by adding a short detection phase after the pen is pressed. However, while using the NonPrefHand, when the two hands were not synchronized correctly, the mode button was usually pressed long after the pen was down, and in half of these, the button was pressed after the drawing was completed. Thus, using a 37ms detection phase can only reduce 14% of the NonPrefHand Mode-In errors.

FUTURE WORK

Because our goal was to reveal the performance attributes of different mode switching techniques, we designed and conducted this experiment in a lab setting that allowed us to efficiently collect a large amount of data and to conduct a quantitative analysis with high accuracy. However, information from actual deployment of these techniques in an application will complement these results. Based on these results, we have been further studying how well these techniques can perform in a more natural setting, e.g., the technique of using a non-preferred hand has been deployed in DENIM. We will continue to improve the pressure-based mode switching technique and we will also consider the impact of mode switching frequency as well as gesture complexity on the performance of mode switching techniques. It is also worth investigating how using different mode feedback, such as audio, can improve the performance of these techniques.

CONCLUSIONS

In this paper, we investigated five techniques for switching between ink and gesture modes in pen interfaces, including a pen-pressure based mode switching technique that allows implicit mode transitions. An experimental study was conducted to evaluate the performance of these techniques, which was the first study to offer a quantitative analysis of methods for switching between ink and gesture modes. The results suggest that pressing a button with the non-preferred hand offers the fastest performance, while holding the pen still is significantly slower and more prone to error than the other techniques. Pressure, while promising, did not perform as well as the non-preferred hand button with our current implementation. But the experiment did show that pressure-based mode switching can be improved by using a personalized pressure space. We also discussed how our

keystroke level analysis allowed us to propose improvements to the techniques. The experimental results and the methodology of this study can be employed for further exploring mode switching techniques for pen-based user interfaces. The results also give designers the tradeoffs to consider for particular types of applications as well as available hardware when using any of these techniques.

ACKNOWLEDGMENTS

We would like to thank Brian Youngstrom, Zile Wei, Jia-Chi Wu and Xin Dong for their help. This work was supported by NSF under Grant No. IIS-0205644.

REFERENCES

- [1] Apitz, G. and Guimbretiere, F., CrossY: A Crossing-Based Drawing Application. In UIST'04, *CHI Letters*, 2004. 6(2): pp. 3–12.
- [2] Compaq. Virtual Book: Lectrice Tablet. <http://research.compaq.com/SRC/virtualbook/>
- [3] Davis, R. Sketch Understanding in Design: Overview of Work at the MIT AI Lab. In AAAI Spring Symposium – Sketch Understanding. 2002: AAAI. pp. 24–31.
- [4] Dillon, R.F., Eday, J.D., and Tombaugh, J.W. Measuring the true cost of command selection: techniques and results. In CHI'90. New York, NY. pp. 19–26.
- [5] Forsberg, A., Dieterich, M., and Zeleznik, R. The Music Notepad. In UIST'98. San Francisco, CA. pp. 203–210.
- [6] Guiard, Y. and Ferrand, T., Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behavior*, 1987. 19(4): pp. 486–517.
- [7] Kabbash, P., Buxton, W., and Sellen, A.G. Two-Handed Input in a Compound Task. In CHI'94. Boston, Massachusetts. pp. 417–423.
- [8] Kurtenbach, G. and Buxton, W. User learning and performance with marking menus. In CHI'94. pp. 258–264.
- [9] Lin, J., et al., DENIM: finding a tighter fit between tools and practice for Web site design. In CHI'00, *CHI Letters*, 2000. 2(1): pp. 510–517.
- [10] Microsoft. Windows CE. www.microsoft.com/windowsce/
- [11] Microsoft. Windows XP Tablet PC Edition. <http://www.microsoft.com/windowsxp/tabletpc/>
- [12] Moran, T.P., Chiu, P., and Melle, W.v. Pen-based interaction techniques for organizing material on an electronic whiteboard. UIST'97. Banff, Alberta. pp. 45–54.
- [13] Ramos, G. and Balakrishnan, R., Fluid Interaction Techniques for the Control and Annotation of Digital Video. In UIST'03, *CHI Letters*, 2003. 5(2): pp. 105–114.
- [14] Ramos, G., Boulos, M., and Balakrishnan, R., Pressure Widgets. In CHI'04, *CHI Letters*, 2004. 6(1): pp. 487–494.
- [15] Raskin, J., Meanings, Modes, Monotony, and Myths, In *The Humane Interface*, Addison Wesley.
- [16] Saund, E. and Lank, E., Stylus Input and Editing Without Prior Selection of Mode. In UIST'03, *CHI Letters*, 2003. 5(2): pp. 213–216.
- [17] Sellen, A.G., Kurtenbach, G.P., and Buxton, W., The Prevention of Mode Errors through Sensory Feedback. *Human-Computer Interaction*, 1992. 7(2): pp. 141–164.
- [18] Stifelman, L., Arons, B., and Schmandt, C., The audio notebook: paper and pen interaction with structured speech. In CHI'01, *CHI Letters*, 2001. 3(1): pp. 182–189.