

Making Pen-Based Operation More Seamless and Continuous

Chuanyi Liu and Xiangshi Ren

School of Information Kochi University of Technology,
Kami-shi, 782-8502 Japan
{renlab,ren.xiangshi}@kochi-tech.ac.jp

Abstract. The feature of continuous interaction in pen-based system is critically significant. Seamless mode switch can effectively enhance the fluency of interaction. The interface which incorporated the advantages of seamless and continuous operation has the potential of enhancing the efficiency of operation and concentrating the users' attention. In this paper, we present a seamless and continuous operation paradigm based on pen's multiple-input parameters. A prototype which can support seamless and continuous (SC) operation is designed to compare the performance with MS Word 2007 system. The subjects were requested to select target components, activate the command menus and color the targets with a given flowchart in two systems respectively. The experiment results report the SC operation paradigm outperformed the standard ways in MS Word in both operation speed and cursor footprint length (CFL).

Keywords: Pen-based system, pressure, twist angle, continuous, seamless.

1 Introduction

Pen devices such as PDAs and Tablet PCs, have been used more and more widely because of the natural pen input. However, the current operation systems (OS) and applications for pen devices still remain the style of OS initially designed for Mice. There are various studies on exploring pen-suitable UI design. In these studies, how to improve the switch efficiency in selection-action patterns is an important research topic. Various techniques and paradigms on selection-action patterns have been presented lately (e.g., [1-3]). Most of these studies utilizing the same input channel for inking and gesturing. In some cases, it is rather difficult to eliminate the ambiguity of stroke recognition completely. And the use of these proposed techniques in pen-based systems is greatly limited for the lack of flexibility and ubiquity. On the other way, a commercial electronic pen commonly possesses multiple input channels. So our basic motivation is to find out an unambiguous and ubiquitously applicable method utilizing extra pen input channels with which users can perform selection-action patterns continuously, fluidly and unambiguously.

In this paper, we present a pen-suitable operation paradigm, under which fluid and continuous operations and seamless switch between different types of operation become possible throughout a computer task. To evaluate the proposed methods, a

drawing prototype system was implemented as a JavaTM program. And a comparative experiment was done to compare the operation paradigm and the corresponding ways in MS Word 2007 system. In the experiment, the subjects were asked to select the target components of a given flowchart, activate the command menus and color the targets. The results show that the proposed operation methods outperform MS Word in both speed and CFL, despite a little higher error rate.

2 Related Work

In this section, we discuss related work regarding both the studies on pen input parameters and these on seamless and continuous operations in pen-based systems.

2.1 Previous Work on Pen Input Parameters

To date, there are many studies on the utilization of pen input parameters. These studies can be roughly divided into two categories. One category investigates the general human ability to control pen input parameters; the other category aims at enhancing performance of human and computer interaction by implementing novel applications or techniques which exploit particular input parameters.

Up to now, pressure parameter has been explored extensively. Herot and Weinzapfel [4] studied the human capability of the finger to apply pressure and torque to a computer screen. Buxton [5] investigated the use of touch-sensitive technologies and the potential for interaction that they suggested. Ramos et al. [6] explored the human ability to vary pen-tip pressure as an additional channel of access to information. Ramos and Balakrishnan introduced *pressure marks* [1] and *Zliding* [7]. Pressure marks can encode selection-action patterns in a concurrent, parallel interaction. In pen strokes, variations in pressure make it possible to indicate both a selection and an action simultaneously. *Zliding* explores integrated panning and zooming by concurrently controlling input pressure while sliding in X-Y space. Li et al. [8] investigated the use of pressure as a possible method to delimit the input phases in pen-based interactions. Harada et al. presented a set of interaction techniques that leveraged the combination of human voice and pen pressure and position input when performing both creative 2D drawing and object manipulation tasks [9].

Input angles (i.e. tilt angle, twist angle and azimuth) are often used as UI clues for natural and intuitive interaction. Balakrishnan et al. [10] introduced the *Rockin'Mouse*. The *Rockin'Mouse* is a promising device for both 2D and 3D interaction that uses tilt input to facilitate 3D manipulation on a plane. Tian et al. [11] explored the *Tilt Menu*. The *Tilt Menu* is implemented by using 3D orientation information of pen devices for better extending selection capabilities of pen-based interfaces. Some other studies such as *TiltType* [12] and *TiltText* [13] focus on using the tilt information of mobile phones to affect text entry tasks in mobile devices. Bi et al. [14] explored rolling angle on general human being control ability. They suggested that both rolling amplitude and speed should be taken into account for rolling-based interact techniques.

As for sketch-based techniques, Davis et al. [15] introduced their *SketchWizard*, which is about wizard of Oz prototyping of pen-based user interfaces. Apitz and Guimbretire [16] presented their *CrossY*, in which pen stroke did all the drawing operations.

2.2 Previous Work on Seamless and Continuous Operations

Hinckley et al. [3] presented their pigtail delimiters, with which selection-action patterns can be performed in one continuous fluid stroke. A pigtail is created explicitly by intersecting one stroke itself and an action is specified or an object manipulated by the stroke's direction. Pigtails provide a way to integrate an explicit command invocation in a fluid stroke following the selection specification. But it is rather difficult to manipulate multiple targets in an irregular layout, since the targets are selected by a lasso. Furthermore, there is ambiguity between pigtail delimiters and freeform drawings.

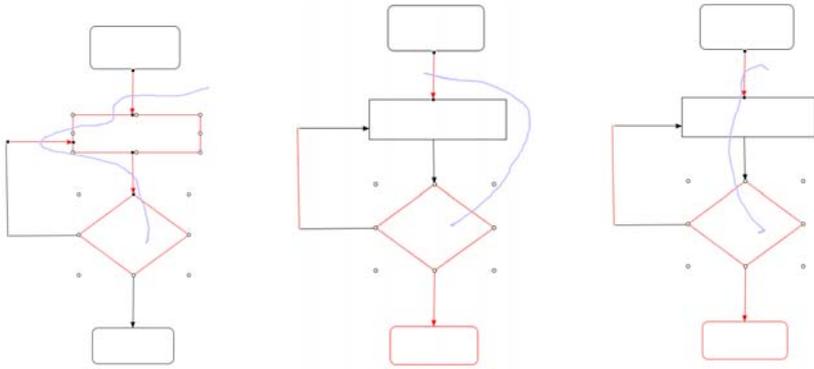
Baudisch et al. [17] introduced *marquee menus*, which are a technique where the selection-action pattern occurs concurrently. The marquee menu's selection is specified by a rectangular area, which is defined by the start and the end points of a straight stroke; its action is determined by one of four movement directions of the stroke. *Marquee menus* are sensitive to both a mark's point of origin and direction while providing a compact interaction phase. The technique is promising for web browsing in small screens. But it has not been elaborated to show whether and how this technique scales for non-straight strokes with arbitrary orientations. Regardless of these considerations, this kind of technique is not suitable for multiple targets in an irregular layout and ambiguity between gesture strokes and freeform drawings limits its practical applicability in other scenarios.

Ramos and Balakrishnan [1] introduced their *pressure marks*, where variations in pressure are used as metaphors for actions. The marks of pressure variation are integrated into selection strokes, and then the selection-action patterns can be performed concurrently and seamlessly. However, there are some limitations with *pressure marks'* variation, e.g. once the user begins to slide a pen slightly then the HL (pressure variation signature, high-low, defined in the original) or HH (high-high) pressure mark may not appear in the following stroke. Furthermore, the number of simple pressure marks is limited, and compound marks are difficult to memorize and control. Again, this kind of technique is only useful for targets arranged in a regular layout.

3 The Proposed Operation Methods

From the previous work, we can see that the selection-action patterns have been explored extensively, but the use of these techniques are limited to some specific narrow scenarios. Furthermore, it is rather difficult to eliminate the ambiguity between gesture strokes and freeform drawings, since both are based on the same input channel. In this paper, we present an operation paradigm with extra input channels, which allows fluid target selection and continuous and seamless switching from selection to action. Commonly, a computer task includes three phases, i.e. object selection, command selection and object property setting phases. Under the operation paradigm, a computer task can be performed in one continuous and fluid stroke. In the target selection phase, users are allowed to string and select the targets with a pen stroke. Pen pressure input is used as a delimiter to distinguish between selection strokes and freeform drawings. When all the targets have been selected by a pen stroke, users can activate a pie menu by rolling the pen. If the rolling angle and speed exceed the respective thresholds, the pie menu will be activated and displayed with its center under

the cursor. And then the user slides the pen tip, an action will be performed when the pen tip crossed a menu item. Throughout the whole process, the pen tip need not to be lifted from the screen. All the operations can be performed in one continuous and fluid stroke. The design of the three phases under the operation paradigm will be introduced in detail in the next section.



(a) String & select objects with one stroke (b) Steer clear of an object (c) Ignore an object crossed by the stroke

Fig. 1. Pressure-based line-string selection (the blue line is the cursor footprint; the objects with sizing handles are selected)

3.1 Target Selection

As suggested by [16, 18, 19] crossing performs better than pointing-and-clicking in UI design, especially for pen-based input devices. In the prototype system, we present a pressure-based line-string selection method. During a pen being slid on the screen, the objects stringed by the stroke are selected when the pen input pressure exceeds a given threshold.

Pressure Coupling Normal Stroke and Line-string Selection. In the application, pressure is used as a delimiter to couple normal stroke and line-string selection. A pilot study was done to determine the right pressure spectra for normal stroke and line-string selection. In our experiment, 12 participants were asked to draw with light pressure, normal pressure and heavier pressure alternately on a WACOM tablet combined display, which has 1024 levels of pressure. The results showed a statistically significant difference on the *maximum pressure scale* of a stroke between the light, the normal and the heavier pressure conditions. In our implementation, the heavy spectrum of pressure was employed for line-string selection, and the normal spectrum for normal stroke; for low end pressure, the spectrum is more difficult to control [7], therefore, it was omitted from the technique design.

Object Selection. The user strokes the pen starting from a blank area, where there is no object. If the pressure input exceeds the specified threshold, the stroke will be pressure-based line-string selection; otherwise it will be normal stroke. Under this

selection mode, the user only needs to stroke the pen on a screen and all the objects stringed by the pen will be selected (see Fig. 1a). A blue footprint line following the path of the pen is used as visual feedback for the selection state. If there are some objects that the user does not want to select in the path of the selection stroke, s/he can steer clear of them (Fig. 1b) or reduce the pressure on the pen to below the threshold without lifting the pen tip from the screen, until the blue footprint line disappears. Then the figure will be crossed by the stroke without being selected (Fig. 1c).

Undoing Selection. The user can stroke the pen back and cross the footprint line on a selected object to undo selecting it. If the user lifts the pen and taps in a *blank area*, selection of all the items will be canceled.

3.2 Activating the Menu

Although, there are various studies on the select-action patterns, but most of these techniques use the same pen input channel for both command gesture and freeform drawings. So it is rather difficult to eliminate the recognition ambiguity completely. In the following section, we introduce a smooth and unambiguous technique for switching smoothly between selection and action by introducing extra pen input channels.

Li et al. [8] investigated five different mode switching techniques in pen-based UI design. They suggested that *non-preferred hand* is the most promising mode switching technique. In their experiment, a physical button mounted at the top-left corner of a Tablet PC screen was employed as a *mode switching button*. It was called a *non-preferred hand* mode switch that users tapped on the *mode switching button* with their *non-preferred hands* to perform a mode switch. In their study, they did not explore angle input channels, e.g., tilt angle, azimuth or twist angle. To determine the most suitable extra input channel that can serve as a switching trigger to activate the menu, we performed a pilot study to investigate all the possible input channels of a pen for mode switching techniques. After the first block of tests using the non-preferred hand section of the trials, we noticed that the subjects tended to keep one finger of their *non-preferred hands* on the *mode switching button*. Taking into account the practical application scenarios, it is impossible to keep the non-preferred hand on a specific button all the time. And under most conditions, the keyboard or such a button is not available in a pen-based system. In our implement, twist angle of pen input was used as an extra input channel for mode switch.

Bi et al. [14] presented their study on rolling (twist) angle for pen input. They suggested that the rolling can be identified as incidental if the rolling speed of a data event is between $-30^{\circ}/s$ and $30^{\circ}/s$ or the rolling angle is between -10° and 10° . And -90° to 90° can be exploited as the usable rolling range. Based on their study results, rolling is employed in our experiment design to activate the pie menu if the rolling speed exceeds the range of $[-50^{\circ}/s, 50^{\circ}/s]$, and rolling angle exceeds $[-50^{\circ}, 50^{\circ}]$. After selecting all the targets, the user intentionally rolls the pen. If the rolling angle and speed exceed the specific thresholds, the pie menu will be activated and displayed with its center under the cursor.

3.3 Performing an Action

We employed crossing to activate a menu command, for crossing performs better than pointing-and-clicking in UI design [16, 18, 19]. When the pie menu is activated, the user slides the pen tip across a menu item, the corresponding action is performed.

4 Experiment

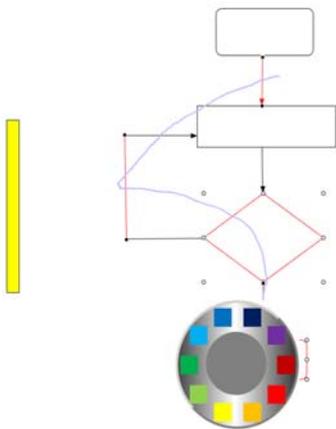
To investigate the performance of SC operation paradigm, a quantitative experiment was conducted, the corresponding operation in MS Word 2007 served as a baseline.

4.1 Apparatus

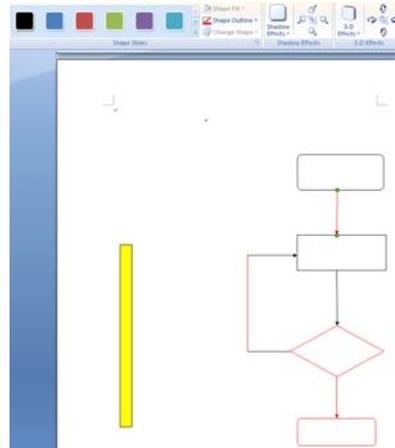
The hardware used in the experiment was a WACOM Cintiq 21UX flat panel LCD graphics display tablet with a resolution of 1,600×1,200 pixels (1 pixel= 0.297mm), using a wireless pen with a pressure, tilt angle, azimuth and twist angle sensitive iso-metric tip (the width of the pen-tip is 1.76mm). It reports 1024 levels (ranging from 0 to 1023, the minimum unit is 1) of pressure and 360° (ranging from 0° to 359°, the minimum unit is 1°) of twist angle. The experimental program was implemented with Java™ 6.0 running on a 3.2 GHz P4 PC with the Windows XP SP2 Professional operating system.

4.2 Participants

Six participants (two female and four male ranging in age from 27 to 36 years old, none paid) were all volunteers from the local university community. All of them were right-handed. One of them has two years of experience of using a digital pen and the other five have no such experience.



(a) The experiment UI of the proposed methods.



(b) The experiment UI in Word 2007

Fig. 2. The experiment UI design

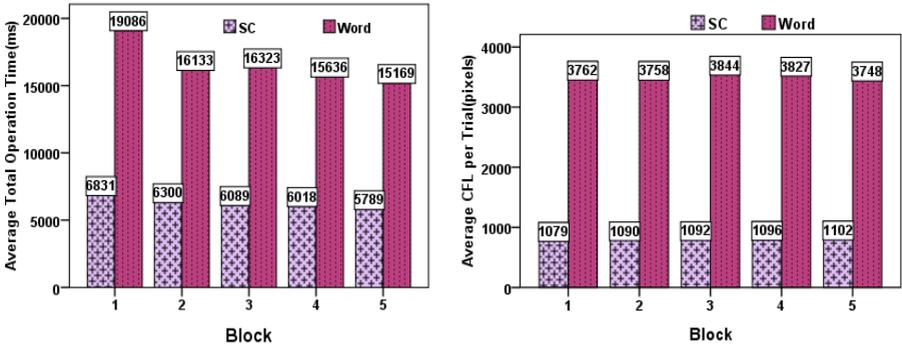
4.3 Task

In the experiment, the subjects were asked to perform for both types of interface (SC operation UI and Word operation UI). For each trial in both types of interface, the subjects were given a flowchart (Fig. 2) composed of 10 components. Five out of the 10 components were randomly chosen as targets (displayed in red). And the target color was shown as a rectangular bar to the left side of the flowchart. For each corresponding trial; the flowchart size, component number, location in the screen as well as the targets are kept the same in both kinds of interface. The subjects were requested to color the outlines of the target components with the given target color. Each trial includes three operation phases, object selection phase (called as *selection phase*), menu trigger phase (called as *trigger phase*) and object property setting phase (called as *setting phase*). Under the proposed paradigm, the subjects selected the targets using pressure-based line-string selection (this process is computed as its *selection phase*), rolled the pen to activate the pie menu (this process is computed as its *trigger phase*) and slid the pen tip across a menu item to color the targets (this process is computed as its *setting phase*). The experimental program recorded the time and accuracy of each phase, and the CFL per trial. With Word 2007, the subjects tapped the pen tip on each target to select it with the (*Shift* or *Ctrl* key being pressed, this process is computed as its *selection phase*), moved the pen tip from the last target and pointed to the toolbar (this process is computed as its *trigger phase*) and tapped the pen tip to color the targets (this process is computed as its *setting phase*). Running in the background, the experimental program analyzed and recorded the time and accuracy of each phase, and the CFL per trial.

4.4 Procedure and Design

Each subject was asked to complete 5 blocks of trials. Each block consisted of 6 selection-action trails. A trial was erroneous if there is any error caused in any of the three phases. The program recorded one *selection phase* error if any target component was omitted or any non-target component was selected. One *trigger phase* error happened when the menu was activated incidentally under SC paradigm, or when the wrong toolbar is tapped in Word 2007. If the target components were not colored with the target color, a *setting phase* error was generated. The errors caused in the *selection phase*, *trigger phase* and *setting phase* are called as *selection error*, *trigger error* and *setting error* respectively. And the time elapsed in the *selection phase*, *trigger phase* and *setting phase* was computed respectively as *selection time*, *trigger time* and *setting time*. A within-subject design was used. The dimensions of all the flowcharts were displayed at a resolution of 297×622 pixels. In the SC operation UI, there are ten standard colors arranged in the same order as the standard color arrangement in the color toolbar of Word 2007. Before the task in Word 2007 began, the subjects were conducted to activate the standard color toolbar as a quick access toolbar, and to scroll the Word page to keep the flowchart directly under the toolbar. The dependent variables were trial time, CFL, error rate and subjective preference. Prior to the study, the experimenter explained and demonstrated the task to the participants. The participants were asked to do the trials as quickly and accurately as possible. At the end of the experiment, participants were instructed to give their subjective comments by

completing a questionnaire, which consisted of four questions regarding “usability”, “fatigue”, “preference” and “attention concentration” on 1-to-7 scale (1=lowest preference, and 7 =highest preference). “Attention concentration” is a promising degree that takes into account the users' ability to focus on the targets themselves.



(a) The average total operation time

(b) The average CFL

Fig. 3. The average total operation time and CFL

4.5 Results

Trial time for each participant averaged thirty minutes. A RM-ANOVA (repeated measures analysis of variance) was used to analyze the performance in terms of operation time, CFL, accuracy and subjective preference.

Total Operation Time and CFL

There was a significant difference in the overall mean operation time ($F(1,5) = 41.832, p = 0.001$) and CFL ($F(1,5) = 50.394, p = 0.001$) between the two operation paradigms. The overall mean operation time per trial was 6309.945 ms of SC operation, 16562.46 ms of operation in Word 2007. And the overall CFL per trial was 1084.172 pixels for SC operation, 3805.964 pixels for the operation in Word 2007. There were no main effects for blocks on overall mean operation time for either SC operation ($F(4,20) = 1.718, p = 0.186$) nor Word operation ($F(4,20) = 1.663, p = 0.198$). There were no main effects for blocks on CFL for either SC operation ($F(4,20) = 0.247, p = 0.908$) or Word Operation ($F(4,20) = 0.058, p = 0.993$). However, as Fig. 3a illustrates, we observed a little improvement in speed. No significant effect was found for paradigm*block on overall mean time ($F(4,20) = 1.029, p = 0.417$) or overall CFL ($F(4,20) = 0.094, p = 0.983$), which indicated that the improvement in learning did not significantly affect relative performance on the two kinds of operation paradigm.

Selection Time. There was a significant difference in the overall mean selection time ($F(1,5) = 88.284, p < 0.0001$) between the two different kinds of operation paradigms. The overall mean selection time per trial was 3700.110 ms for SC operation and

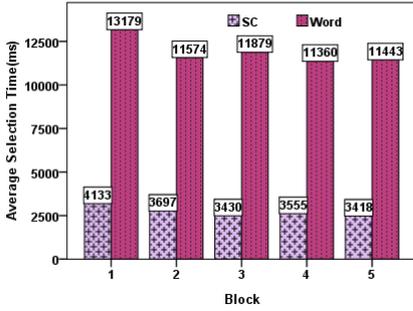


Fig. 4. The average selection time

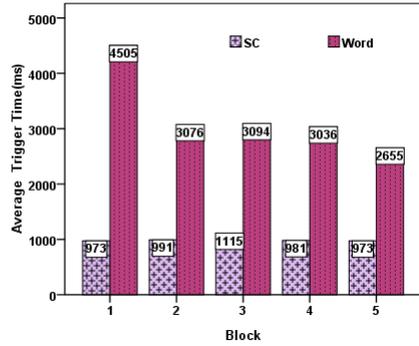


Fig. 5. The average trigger time

11955.45 ms for Word operation. There were no main effects for blocks for the operation of SC ($F(4,20) = 1.164, p = 0.356$) or Word 2007 ($F(4,20) = 0.625, p = 0.650$), on overall mean selection time. A small speed improvement in selection time for both SC and Word operation was also observed in Fig. 4. No significant effect was found for paradigm*block on the overall mean selection time ($F(4,20) = 0.307, p = 0.870$), which indicated the learning improvement did not significantly affect the relative performance of the two kinds of operation paradigm on selection time.

Trigger Time. There was a significant difference ($F(1,5) = 6.991, p = 0.046$) in the overall mean trigger time per trial between the two different kinds of operation paradigms. The overall mean trigger time per trial was 1030.373 ms for SC operation and 3297.632 ms for Word operation. There was no main effect for the operation of either SC ($F(4,20) = 0.885, p = 0.491$) or Word ($F(4,20) = 1.570, p = 0.221$), for blocks on overall mean trigger time. Fig. 5 also illustrates a small improvement in selection time for both SC and Word operation. No significant effect was found for paradigm*block on overall mean trigger time ($F(4,20) = 1.562, p = 0.223$), which indicated learning improvement did not significantly affect the relative performance of the two kinds of operation paradigm on trigger time.

Setting Time. There was a significant difference ($F(1,5) = 12.973, p = 0.016$) in the overall mean setting time per trial between the two different kinds of operation paradigms. The overall mean setting time was 1579.463 ms for SC operation and 1309.381 ms for Word operation. For the operation of both SC ($F(4,20) = 2.896, p = 0.048$) and Word ($F(4,20) = 2.994, p = 0.043$), there were main effects for blocks on overall mean setting time. Fig. 6 illustrates a little improvement in setting time for both SC and Word operation. No significant effect was found for paradigm*block on the overall mean trigger time ($F(4,20) = 0.417, p = 0.794$), which indicated the learning improvement did not significantly affect the relative performance of the two kinds of operation paradigm on setting time.

Errors. The results showed a significant difference in the overall mean error rate ($F(1,5) = 24.306, p = 0.014$) between the two different kinds of operation paradigm. The overall mean error rate was 2.458% of SC operation and 1.606% of Word operation. There were main effects for blocks on overall mean errors for SC operation

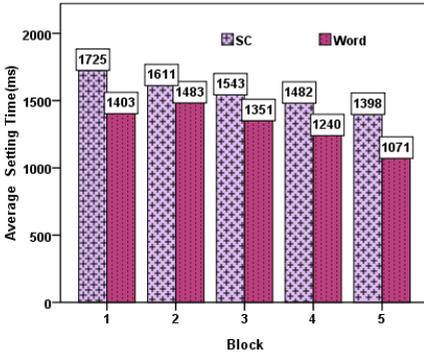


Fig. 6. The average setting time

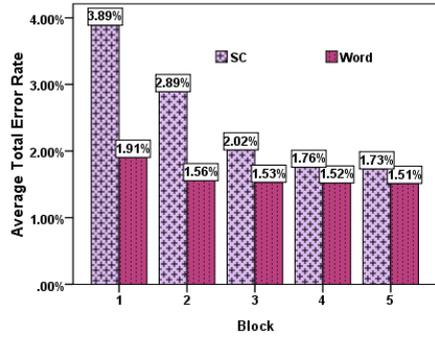


Fig. 7. The average total error rates

($F(4,20) = 6.332, p = 0.002$), but no main effects for blocks on overall mean errors for Word operation ($F(4,20) = 1.010, p = 0.043$). As Fig. 7 illustrates, we observed a significant decrease in errors for SC and a marginal one in Word operation. Significant effects were found for paradigms*block on the overall mean errors ($F(4,20) = 5.588, p = 0.003$), which indicated the learning improvement significantly affected the relative performance of the two kinds of operation paradigm regarding errors.

Selection Error. The experimental analysis reported a significant difference in the overall mean selection error rate ($F(1,5) = 9.423, p = 0.028$) between the two different kinds of operation paradigm. The overall mean selection error rate was 0.864% of SC operation, 0.540% of Word operation. There were main effects for blocks on overall mean selection error rate for SC operation ($F(4,20) = 1.650, p = 0.021$), but no main effects for blocks on the overall mean selection error rate for Word operation ($F(4,20) = 0.625, p = 0.650$). Fig. 8 illustrates a big improvement in selection errors for SC operation and a marginal improvement for Word operation. Significant effect was found for paradigm*block on the overall mean trigger time ($F(4,20) = 5.058, p = 0.037$), which indicated the learning improvement significantly affected the relative performance of the two kinds of operation paradigm on selection errors.

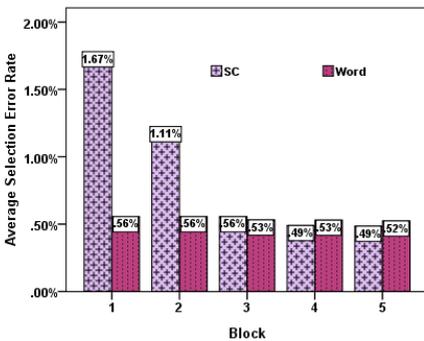


Fig. 8. The average selection error rates

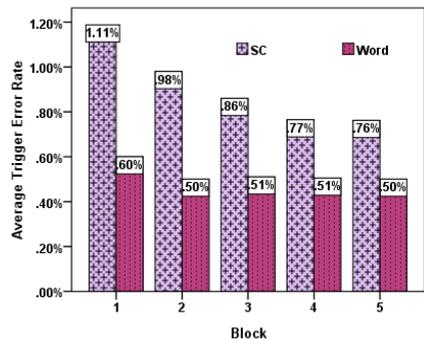


Fig. 9. The average trigger error rates

Trigger Error. There was a significant difference in the overall mean trigger error rate ($F(1,5) = 20.000, p = 0.007$) between the two different kinds of operation paradigm. The overall mean trigger error rate was 0.896% for SC operation and 0.524% for Word. There were main effects for blocks on overall mean trigger error rate for SC operation ($F(4,20) = 17.857, p = 0.001$), but no main effects for blocks on overall mean trigger error rate in Word 2007 ($F(4,20) = 0.250, p = 0.906$). Fig. 9 illustrates a significant decrease in trigger error rate for SC operation and a little decrease for Word 2007. Significant effect was found for paradigm*block on the overall mean trigger time ($F(4,20) = 9.062, p < 0.0001$), which indicated the learning improvement significantly affected the relative performance of the two kinds of operation paradigm on trigger error rate.

Setting Error. There was no significant difference in the overall mean setting error rate ($F(1,5) = 5.000, p = 0.076$) between the two operation paradigms. The overall mean setting error rate was 0.7% for SC operation and 0.534% for Word operation. There were main effects for blocks on overall mean setting error rate for SC operation ($F(4,20) = 5.000, p = 0.006$), but no main effects for operation in Word 2007 ($F(4,20) = 2.742, p = 0.057$). Fig. 10 illustrates the improvement in setting errors of both SC and Word operation. No significant effect was found for paradigm*block on the overall mean setting error rate ($F(4,20) = 2.619, p = 0.066$), which indicated the learning improvement did not significantly affect the relative performance of the two kinds of operation paradigm on trigger errors.

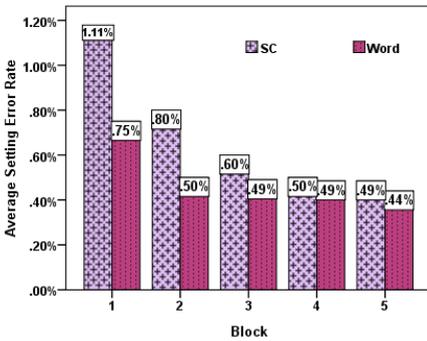


Fig. 10. The average object property setting error rates

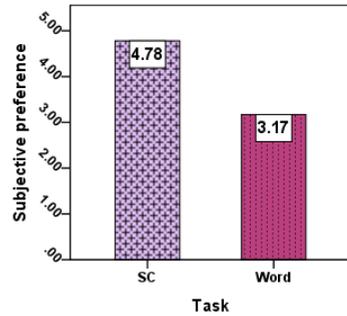


Fig. 11. The subjective preference

Subjective Comments. Fig. 11 shows the subjective ratings for the two kinds of operation paradigm. These ratings were based on the average value of the answers given by the subjects to the four questions. Significant main effects were observed between the two operation paradigms ($F(1,5) = 9.365, p = 0.028$). The average preference for SC operation paradigm is 4.8, and for MS Word it is 3.2.

5 Discussion

Various contrastive techniques (e.g., lassoing + pigtailling [3]) were taken into account, but none of the presented techniques for pen-based systems is suitable for the wide

range of common computer tasks. Thus, MS Word was chosen as the baseline because it is the most widely used semantic paradigm. At the beginning of the experiment, we noticed that the participants stroked the pen rather cautiously and slowly to select the targets, rolled the pen nervously to activate the pie menu, and wanted to lift the pen tip to tap the target menu item. But after several trials, they stroked and rolled the pen fluidly and confidently. They commented that the SC operation was enjoyable; some of them said that performing the SC operation was like playing games.

The results illustrate that the selection and trigger speed of SC operation are significantly faster than that of MS Word. But the setting speed of SC operation is a little slower than that of MS Word. This is probably due to that part of the pie menu was visually occluded by the hand in the setting phase. We observed that some of the participants tended to adjust their hands when crossing a target menu item, others tended to hold the pen at a little higher position to facilitate crossing the menu item after the first block. From the experiment results, we also noted that the error rates for the three phases of SC operation were much higher than for MS Word in the first two blocks. But the difference between SC and MS Word operation in error rates was not much different from the third block, except for the average trigger error rate. During the experiment, we observed that some participants tended to trigger the pie menu accidentally much more often than others. This is probably due to the participants' different ways of holding the pen. Fig. 3b illustrates that the CFL for SC is much shorter than for MS Word, which proves that the cursor needs to be moved less in SC operation than in MS Word. This can further indicate that, in SC operation, the participants can concentrate their attention on the targets much better than with the standard interfaces.

6 Conclusion and Future Research

In this paper, we present an operation paradigm that is suitable for seamless and continuous operation in pen-based systems. The results of SC operation are rather promising in both speed and CFL, and the accuracy is not significantly different to the standard operation in MS Word after the second block. In our future research, we will explore which combination of pen input parameters is most promising, and the possible maximum number of pen input channels that the subjects can comfortably cope with.

References

1. Ramos, G., Balakrishnan, R.: Pressure marks. In: Proc. CHI 2007, pp. 1375–1384 (2007)
2. François, G., Andrew, M., Terry, W.: Benefits of merging command selection and direct manipulation. *ACM Trans. Comput.-Hum. Interact.* 12(3), 460–476 (2005)
3. Hinckley, K., Baudisch, P., Ramos, G., Guimbretiere, F.: Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In: Proc. CHI 2005, pp. 451–460 (2005)
4. Herot, C.F., Weinzapfel, G.: One-point touch input of vector information for computer displays. In: Proc. SIGGRAPH 1978, pp. 210–216 (1978)
5. Buxton, W., Hill, R., Rowley, P.: Issues and techniques in touch-sensitive tablet input. In: Proc. SIGGRAPH 1985, pp. 215–224 (1985)

6. Ramos, G., Boulos, M., Balakrishnan, R.: Pressure widgets. In: Proc. CHI 2004, pp. 487–494 (2004)
7. Ramos, G., Balakrishnan, R.: Zliding: Fluid zooming and sliding for high precision parameter manipulation. In: Proc. UIST 2005, pp. 143–152 (2005)
8. Li, Y., Hinckley, K., Guan, Z., Landay, J.A.: Experimental analysis of mode switching techniques in pen-based user interfaces. In: Proc. CHI 2005, pp. 461–470 (2005)
9. Harada, S., Saponas, T.S., Landay, J.A.: Voicepen: Augmenting pen input with simultaneous non-linguistic vocalization. In: Proc. ICMI 2007, pp. 178–185 (2007)
10. Balakrishnan, R., Baudel, T., Kurtenbach, G., Fitzmaurice, G.: The rockin’mouse: integral 3d manipulation on a plane. In: Proc. SIGCHI 1997, pp. 311–318 (1997)
11. Tian, F., Xu, L., Wang, H., Zhang, X., Liu, Y., Setlur, V., Dai, G.: Tilt menu: Using the 3d orientation information of pen devices to extend the selection capability of pen-based user interfaces. In: Proc. CHI 2008, pp.1371–1380 (2008)
12. Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., Want, R.: Tilttype: Accelerometer supported text entry for very small devices. In: Proc. UIST 2002, pp. 201–204 (2002)
13. Wigdor, D., Balakrishnan, R.: Tilttext: Using tilt for text input to mobile phones. In: Proc. UIST 2003, pp. 81–90 (2003)
14. Bi, X., Moscovich, T., Ramos, G., Balakrishnan, R., Hinckley, K.: An exploration of pen rolling for pen-based interaction. In: Proc. UIST 2008, pp. 191–200 (2008)
15. Davis, R.C., Saponas, T.S., Shilman, M., Landay, J.A.: Sketchwizard: Wizard of oz prototyping of pen-based user interfaces. In: Proc. UIST 2007, pp. 119–128 (2007)
16. Apitz, G., Guimbretire, F.: Crossy: A crossing-based drawing application. In: Proc. UIST 2004, pp. 1–10 (2004)
17. Baudisch, P., Xie, X., Wang, C., Ma, W.Y.: Collapse-to-zoom: viewing web pages on small screen devices by interactively removing irrelevant content. In: Proc. UIST 2004, pp. 91–94 (2004)
18. Accot, J., Zhai, S.: More than dotting the i’s- foundations for crossing-based interfaces. In: Proc. CHI 2002, pp. 73–80 (2002)
19. Ren, X., Moriya, S.: Improving selection performance on pen-based systems: a study of pen-based interaction for selection tasks. *ACM Trans. Comput.-Hum. Interact.* 7(3), 384–416 (2000)