

Interface for Digital Notes Using Stylus Motions Made in the Air

Yu Suzuki[†]

Kazuo Misue[†]

Jiro Tanaka[†]

[†]Department of Computer Science, University of Tsukuba
{suzuki, misue, jiro}@iplab.cs.tsukuba.ac.jp

Abstract

Although digital notes have many functions intended to be operated with a stylus, the digital notes interface is not always user-friendly when a stylus is used. One reason for this shortcoming is that the interface was designed based on the WIMP interface. In this paper, we describe a technique to overcome this obstacle. This technique uses stylus motions made in the air (called stylus actions) as interactions. We applied Shortcut Actions to digital notes in an experiment to test this use of the stylus actions. The results of this experiment indicate that the Shortcut Actions were effective.

Keywords: Digital Note Interface, Shortcut Actions

1 Introduction

As the use of Tablet PCs, PDAs, and Smart Boards spreads, people are becoming accustomed to using computers to create memos. Although many tools for creating memos using a computer have been developed, most have had only simple functions (like drawing some handwritten characters or graphics). Recently, though, sophisticated multifunction tools for digital notes have been developed, such as Microsoft Office OneNote¹.

When using OneNote, as well as writing handwritten memos users can attach digital data such as pictures and audio files, and they can collect and manage these data in one document. This allows users to create a repository which contains all-digital data that can be reused. As a result, activities can be done more productively and creatively. In this way, OneNote is a digital note tool which supports creative activities.

Because the pen is a tool which almost everyone has used from an early age, it is one of the most familiar creative activity tools. Therefore,

we think that compared to devices used for computers only (like the mouse and keyboard), the pen-shaped stylus is a more suitable interface for creative activities.

While users can activate OneNote with a mouse, OneNote provides many functions, like handwriting recognition and digital ink, intended to be used with a stylus. The OneNote interface design was based on a traditional WIMP interface, though, so the interface is not very suitable for creative activities. For example, to select a menu, users have to follow a tree-structured hierarchical menu from a menu bar at the top of a window or click a small icon on a tool bar. These operations force users to control a mouse cursor carefully while watching the display, interrupting the users' mental concentration for creative activities. In addition, using a stylus forces users to point directly at an operational object on a display. The larger the display becomes, the more difficult it is for users to operate. These are obstacles to creative activities.

By resolving the above problem regarding stylus use and achieving our goal of creating a stylus interface suitable for creative activities, we want to generalize the use of creative activities using a computer.

In this paper, we describe an interface for digital notes which does not interfere with creative activities.

2 Problems and Proposed Technique

2.1 Digital Note Problems

Here we describe a problem of digital notes in terms of the interface used when using digital notes for creative activities.

We assume a situation where a user makes a copy of one object written on a digital note. The user selects the object, and then clicks the "Edit" menu on a menu bar. After that, the user clicks the "Copy" menu in a pull-down menu. Because this sequence of operations requires fine cursor controls, the user's mental concentration for creative activities and eye lines are interrupted. As

¹Microsoft Office OneNote
<http://office.microsoft.com/en-us/onenote/default.aspx>

this gets in the way of creative activities, this interface can be improved.

2.2 Introduction of Shortcut Actions

We developed Shortcut Actions as a way to improve the interface and reduce the problem described in the preceding section when a stylus is used as an input device. Shortcut Actions use hand motions in the air and invoke a certain function of application software. This allows us to interact with a computer without directly pointing with an operational object on a display.

Various hand motions in the air can be used for Shortcut Actions. However, the hand motions used for Shortcut Actions must satisfy the following conditions: they must be possible while holding a stylus, and they must be natural. In this research, we chose to use the following three motions.

- (a) Rolling a pen around the pen-axis direction
- (b) Shaking a pen in the pen-axis direction
- (c) Swinging a pen in the direction vertical to the pen-axis

We call these motions “actions”, and respectively refer to each as

- (a) Rolling
- (b) Shaking
- (c) Swinging

With Shortcut Actions, we can assign a frequently used function to an action or we can freely assign our favorite function to an action.

2.3 Related Operation Technique

Shortcut Actions can be considered analogous to the uses of shortcut keys in keyboard operations.

The keyboard has an operation method, called the shortcut key, that is an original interaction method. While we can invoke a frequently used function by pressing Ctrl+C or Ctrl+V as a shortcut key function, we must use a mouse and a keyboard at the same time.

On the other hand, Shortcut Actions allow us to invoke functions using only a stylus. Therefore, Shortcut Actions are simpler and has a better interaction style than the shortcut key.

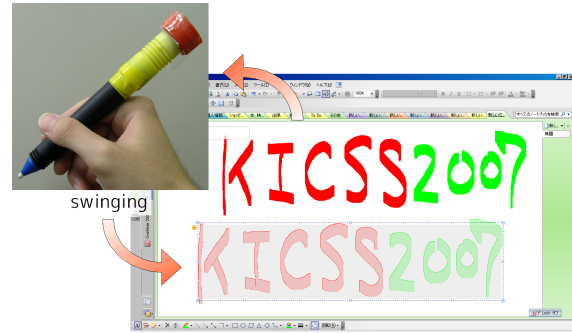


Figure 1. Copy and paste by swinging

3 Developed Interface

3.1 Digital Note Interface

We added Shortcut Actions to OneNote as a new operation method. Each action was assigned to the following functions:

- Rolling...The user can change thirteen pen types which are prepared in OneNote. Rolling the stylus in a clockwise direction causes the pen type to switch to the downward pen type. Rolling the stylus counter-clockwise causes the pen type to switch to the upward pen type.
- Shaking...The user can switch pen mode and selection mode. The pen mode allows characters and graphics to be written by strokes. The selection mode allows selection of objects in a note by encircling them with a bounding box. These modes can be switched by shaking. One shaking action can change between the pen mode and the selection mode.
- Swinging...The user can copy or paste (Figure 1) by swinging the stylus. We can identify swinging in four directions. In this application, we assigned the copy function and the paste function to two swinging directions.

By shaking and swinging, the user can copy and paste or change pen type in a completely different way. When a user copies and pastes an object with a traditional interface, the user first has to select the object, and then click the copy menu and the paste menu in a menu bar. In contrast, when using Shortcut Actions, the user can copy and paste by swinging after selecting the object.

3.2 Oh! Stylus Draw

In Section 3.1, we explained how we extended the OneNote interface, which is an existing form of digital notes. We could not change the interface design, though, because OneNote is off-the-shelf application software.

We are developing an original form of digital notes because we want to use Shortcut Actions in application software where we can freely change the interface design and then verify the advantage provided by Shortcut Actions. We will build several functions in our digital notes. In this paper, we describe “Oh! Stylus Draw”, which is one of our specific digital note functions and has some painting functions. Oh! Stylus Draw is an easy painting tool and has the following functions.

- change a drawing color
- switch a drawing mode
- switch a pen type
- undo

Oh! Stylus Draw has two ways to select these functions. One way is to select from a menu bar, the other is to select from the Shortcut Actions. Since the former is a well known method, we will not explain it here.

In the latter way, swinging calls some menus, rolling changes a drawing color and shaking switches a color palette. Menus which are invoked by swinging are shown to the user on a large panel that the user can easily operate using a stylus. Unlike an icon menu or menu bar interface, which are used for many applications, Oh! Stylus Draw’s interface is designed to be operated by stylus.

The relations between actions and menus are as follows.

- rolling...change a drawing color
- shaking...switch a drawing mode
- swinging...
 1. show a drawing mode panel
 2. show a pen panel
 3. undo
 4. switch display/hide a color palette

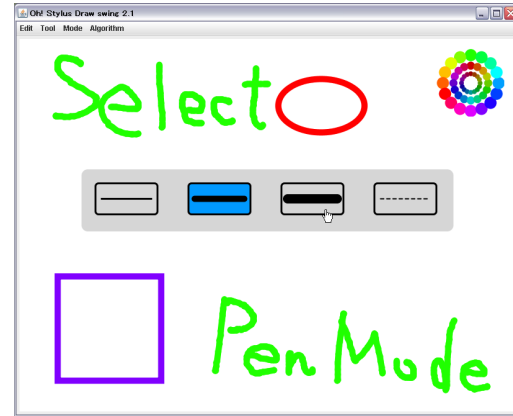


Figure 2. Pen panel displayed by swinging

Figure 2 shows a screenshot where a pen panel has been made to appear in the center of the screen by swinging. The user selects the desired pen type from the pen panel by tapping. In this way, users can select a function without their concentration being interrupted because the displayed menu panel is larger than the small icons which appear in the traditional interface. Thus, the user can concentrate on creative activities.

4 Implementation Method

4.1 Context Sensitive Stylus

We have to detect hand motions made in the air to implement Shortcut Actions. To do this, we attach an accelerometer to the stylus. We use a Cookie sensor² (Figure 3) as the accelerometer.

The Cookie can obtain three-axis acceleration by 10 Hz. In addition, the Cookie can communicate with a computer wirelessly because it has a Bluetooth module.

We show a stylus with an attached Cookie in Figure 4. We call this stylus a context sensitive stylus (CS stylus).

4.2 Action Recognition

To use three actions (rolling, shaking, swinging) for interaction with computers, our system has to recognize these actions by using three-axis (x, y, z) accelerations. The motion axis of shaking and swinging is the same axis as that which can be obtained from an accelerometer. Therefore, for example, the system can recognize these actions by comparing X_τ , which is an x-axis acceleration at a certain time τ (τ is discrete time in

²Cookie sensor from Nokia Research Center Tokyo.

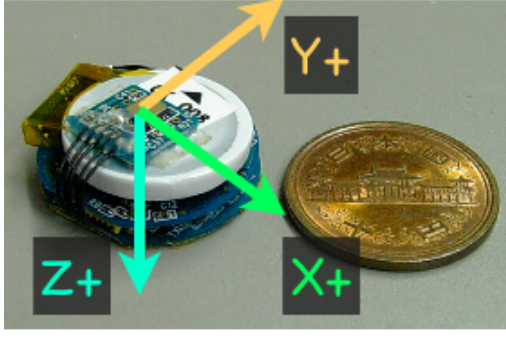


Figure 3. Cookie sensor and obtained three-axis direction



Figure 4. Context sensitive stylus

100-ms units), with $X_{\tau-1}$, which is an x-axis accelerometer at the previous time 100 ms earlier.

On the other hand, rolling cannot be recognized by the above method because rolling is a rotational motion. Therefore, we had to devise a method of rolling recognition.

4.3 Method of Rolling Recognition

First, we calculate pseudo-angle θ from accelerations along two axes (x, y) to recognize rolling action. We can represent θ_τ at a certain time τ by the following expression:

$$\theta_\tau = \tan^{-1}(Y_\tau/X_\tau)$$

We can represent $\theta_{\tau-1}$ at a time 100 ms before τ by a similar expression.

We can represent the difference of angles between the times τ and $\tau - 1$ by

$$\delta_\tau = (\theta_\tau - \theta_{\tau-1}) \bmod 360 - 180$$

However, we cannot realize practical rolling recognition accuracy from this difference of an-

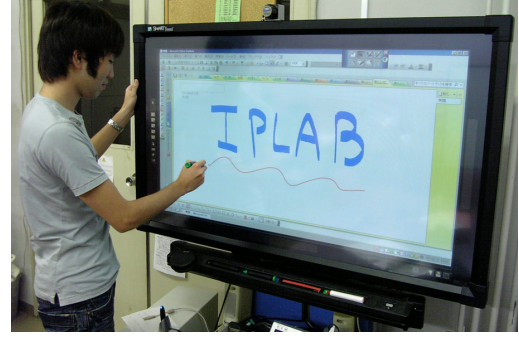


Figure 5. PDP with touch panel used in evaluation

gles. Thus, our aim is to enhance rolling recognition accuracy by improving the rolling recognition algorithm.

5 Evaluation

We evaluated the Shortcut Actions which we proposed as a method to overcome the problems that arise when digital notes are used.

5.1 Purpose of Evaluation

The purpose of the evaluation experiment is to obtain knowledge needed to achieve the goal of this research, which is to create a stylus interface suitable for creative activities.

This evaluation experiment was intended as a pilot study before we do a full-scale experiment.

5.2 Method of Evaluation

There were four participants in the evaluation (three students in our laboratory and one of the authors). The first three participants (the students) were not used to using the CS Stylus; the author is an application creator who had previously worked with the CS Stylus a number of times, so he was familiar with its operation. Therefore, we could examine whether familiarity with the device affects the device operability by comparing the results of the author with those of the other participants.

The devices used for the evaluation experiment were a plasma display panel (PDP) with a touch panel (SMART Board), the CS Stylus, and a general stylus. The PDP had a 50" screen and 1280*768 dot screen resolution (Figure 5). We used OneNote and the interface described in Section 3.1 as the application and the interface for the evaluation.

Table 1. Time needed for tasks (s)

	Participant A			Participant B		
	CS Stylus	Stylus	difference	CS Stylus	Stylus	difference
First Try	10.2	10.9	-0.7	20.9	26.0	-5.1
Second Try	16.9	10.8	6.1	13.6	23.4	-9.8
Third Try	9.0	10.4	-1.4	14.8	28.3	-13.5
	Participant C			Participant D		
	CS Stylus	Stylus	difference	CS Stylus	Stylus	difference
First Try	13.0	15.5	-2.5	13.2	15.0	-1.8
Second Try	16.8	16.0	0.8	13.1	14.3	-1.2
Third Try	12.8	15.8	-3.0	12.0	14.0	-2.0

The evaluation task was that participants had to write down their own name on OneNote, and then make a copy through copy & paste commands. Most people are used to writing their own name, so this simple task should not have created advantages or disadvantages affecting each participant’s performance.

Participants tried this task three times using the CS Stylus and three times using the general stylus. Two of the participants (A, D) first tried the task with the CS Stylus, and then with the general stylus. For the other participants (B, C), the order was reversed.

We did not consider the difference in the task achievement time attributable to differences in the number of strokes of the participants’ names, because we wanted to measure only the differences in task achievement times due to using the two kinds of stylus.

5.3 Results

The evaluation results are given in Table 1. As shown, operations were done more quickly with the CS Stylus than with the general stylus in most trials.

We can also see that each participant finished the three trials in almost the same time when using the general stylus. This suggests that all participants were already used to using a WIMP interface, which is the general GUI environment.

In contrast, when using the CS Stylus participants’ second and third trials tended to be done more quickly than their first trial, with the second trial results for participants A and C being a notable exception. The second trials of participants A and C involved operational failures caused by the participants not being familiar with the CS

Stylus. As users gain experience using the CS Stylus, though, we expect these operational failures to become less frequent. Participant D, who is an author, was an experienced user of the CS Stylus, and so avoided operational failures; his three trial times are almost the same. Thus, the evaluation results suggest that operations can be done more quickly with the CS Stylus than with the general stylus once the user becomes used to the CS Stylus.

6 Discussion

When we use a mechanical pencil in the real world, we habitually knock the top of the pencil or shake it up and down to push out the graphite lead. We can perform these actions without being conscious of controlling the graphite lead because the actions are already common practice.

Our experience with the CS Stylus indicates that the time from starting to use the CS Stylus and when the user becomes used to operating it is shorter than for other devices. We believe that one reason for this is that the CS Stylus is also a “pen”, and most users have handled pens and pencils throughout their life.

Therefore, the Shortcut Actions that we propose can become second nature to users, much like motions when one works with a mechanical pencil. We believe that the CS Stylus will be a suitable interface for digital notes by enabling users to operate menus without interrupting their concentration, even when using a stylus.

7 Related Work

We have focused on problems when using digital notes with a stylus. Our research is aimed

at overcoming the problems and we have already developed some fundamental techniques[7].

Like our research, other studies have aimed at improving pen-based interaction. Apitz et al. created CrossY[2] which is a drawing tool in which all operations can be realized by crossing. Crossing, which is a technique of command invocation, is advocated by Accot et al.[5]. In crossing, a command is invoked when a pointing device crosses an operational object. In CrossY, the user can operate GUI parts (e.g., radio buttons and scroll bars) by crossing. The functions of CrossY are selection of drawing color, selection of pen or brush, scrolling a page, and so on.

Hinckley et al. developed Scriboli[6]. Scriboli shows Marking Menus[3, 4] by using Pigtail and the user can select his favorite function in Marking Menus. Pigtail is a technique where the user can operate a computer by drawing strokes like the tail of a pig at the end of a pen stroke. Scriboli shows Marking Menus by drawing Pigtail after encircling operational objects.

Smith et al. developed the radial scroll tool[1] which can be used to control scrolling by drawing round gestures with stylus.

8 Conclusion and Future Work

We have explained a problem in operating digital notes with a stylus and proposed Shortcut Actions as a solution. We have developed two things that support this solution: a OneNote interface implementation to realize our proposed technique, and a technique to implement the interface.

The evaluation of Shortcut Actions operated with the CS Stylus or a general stylus indicated that Shortcut Actions are superior to general menu operations. It also seems increasingly likely that operations will be done more quickly with the CS Stylus than with a general stylus as user becomes more familiar with the CS Stylus.

In the future, we will do a full-scale evaluation to observe and analyze users' familiarity with devices and will apply Shortcut Actions to other applications.

Acknowledgment

We thank Nokia Research Center Tokyo for lending us the Cookie sensors.

References

- [1] G. M. Smith and m. c. schraefel. The radial scroll tool: scrolling support for stylus- or touch-based document navigation. *Proceedings of the 17th annual ACM Symposium on User Interface Software and Technology (UIST'04)*, pp. 53–56, 2004.
- [2] Georg Apitz and Francois Guimbretiere. CrossY: A Crossing-Based Drawing Application. *Proceedings of the 17th annual ACM Symposium on User Interface Software and Technology (UIST'04)*, pp. 3–12, 2004.
- [3] Gordon Kurtenbach and William Buxton. The Limits Of Expert Performance Using Hierarchic Marking Menus. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'93)*, pp. 482–487, 1993.
- [4] Gordon P. Kurtenbach and Abigail J. Sellen and William A.S. Buxton. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Journal of Human Computer Interaction*, Vol. 8, No. 1, pp. 1–23, 1993.
- [5] Johnny Accot and Shumin Zhai. More than dotting the i's - Foundations for crossing-based interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'02)*, pp. 73–80, 2002.
- [6] Ken Hinckley and Patrick Baudisch and Gonzalo Ramos and Francois Guimbretiere. Design and Analysis of Delimiters for Selection-Action Pen Gesture Phrases in Scriboli. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2005)*, pp. 451–460, 2005.
- [7] Yu Suzuki and Kazuo Misue and Jiro Tanaka. Stylus Enhancement to Enrich Interaction with Computers. *Proceedings of 12th International Conference on Human-Computer Interaction (HCI International 2007)*, pp. 133–142, 2007.