Stylus Enhancement to Enrich Interaction with Computers

Yu Suzuki, Kazuo Misue, and Jiro Tanaka

Department of Computer Science, University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan
{suzuki,misue,jiro}@iplab.cs.tsukuba.ac.jp

Abstract. We introduce a technique to enrich user interaction with a computer through a stylus. This technique allows a stylus to be manipulated in the air to operate applications in new ways. To translate the stylus manipulation into application behavior, we take an approach that we attach an accelerometer to the stylus. Such a stylus allows control through new operations like rolling and shaking, as well as through conventional operations like tapping or making strokes. An application can use these operations to switch modes or change parameters. We have implemented a number of applications, called the "Oh! Stylus Series," that can be used with our proposed technique.

Keywords: Stylus Enhancement, Interaction, Accelerometer, Behavior.

1 Introduction

Computer operations based on a stylus, compared to those with a mouse, have the advantage of being direct and easy to understand. In terms of the interaction style, though, most stylus operations simply replace mouse operations, so these operations do not take advantage of all the ways a pen-shaped device can be handled. Potential interface techniques have not been realized because computers cannot detect stylus positions and movements when a conventional stylus is not touching the display.

Our goal is to make pen-based applications more useful by enabling a computer to detect such manipulations of a stylus and use them to govern applications run on the computer. Our approach is to attach an accelerometer to a stylus so that it can obtain context information regarding the stylus; i.e., its positions and movements. Analysis of the obtained context information allows a computer to respond to user commands input through the stylus movements. In this paper, we describe an interaction technique based on this approach, the creation of a context sensitive stylus, applications which use the stylus, and our evaluation of the interaction technique.

2 Proposed Technique

2.1 Primary Idea

Our proposed interaction technique takes advantage of how a user naturally handles a pen-shaped device; that is, it is based on the various ways users manipulate a stylus as

J. Jacko (Ed.): Human-Computer Interaction, Part II, HCII 2007, LNCS 4551, pp. 133–142, 2007.

they use it, and makes use of these behaviors for interaction with a computer. These behaviors include the following:

(a) driving a quill
(b) dotting a point
(c) pressing a point
(d) tapping the head or grip of a pen
(e) shaking a pen up and down
(f) rolling a pen between the fingers

Of these behaviors, (a) – (d) can be done through the same behavior when people use a stylus. In other words, as stylus operations these behaviors are “drag”, “tap”, ”press”, “point”, “push of a barrel button” and combinations of these operations. On the other hand, (e) and (f) are behaviors which cannot be used with a conventional stylus because they cannot be detected when a conventional stylus is used; i.e., the positions and movements of a stylus in the air cannot be detected. However, we enable computers to detect such stylus behaviors by attaching an accelerometer to the stylus, and this makes it possible to use such movements for interaction between the user and the computer.

2.2 Behavior Assignment

We have to assign various behaviors of a stylus, detected from context information regarding the stylus, to particular computer operations. To realize a comfortable, easy to understand interface based on our interaction technique, it is important to assign appropriate behaviors to these operations. We divided this assignment into two parts:

(i) assigning typical behaviors when people use a pen
(ii) assigning another actions

The first type of behaviors allows the user of a stylus with an attached accelerometer to get feedback from an application by operating the stylus as he would use a pen. This enables easy and intuitive operation. However, the variety of such behaviors is not rich.

The second type allows us to increase the variety of behaviors almost infinitely. In this case, though, intuitive computer operation is more difficult to achieve.

We have to discuss another way of behavior assignment.

3 Implementation

3.1 Hardware

Our system consists of a tablet PC (HP Compaq tc 1100), an accelerometer (Cookie (Nokia Research Center Tokyo)), and a Bluetooth USB adapter (Microsoft). Cookie is a coin-shaped, battery-operated complex sensor (Fig. 1, upper-left), and its Bluetooth v1.1 Serial Port Profile enables wireless communication with computers. Attaching Cookie to a stylus thus allowed us to create a smart stylus.

We call this stylus a context sensitive stylus. It was difficult to attach Cookie to a stylus directly, so we created a holder from a marker pen cap and a plastic bottle cap.

While this stylus was a rather crude prototype, we plan to create a more refined version in the future.

Because our Tablet PC was not equipped with a Bluetooth module, we attached a Microsoft Bluetooth USB adapter to the tablet PC so that it could communicate with Cookie.

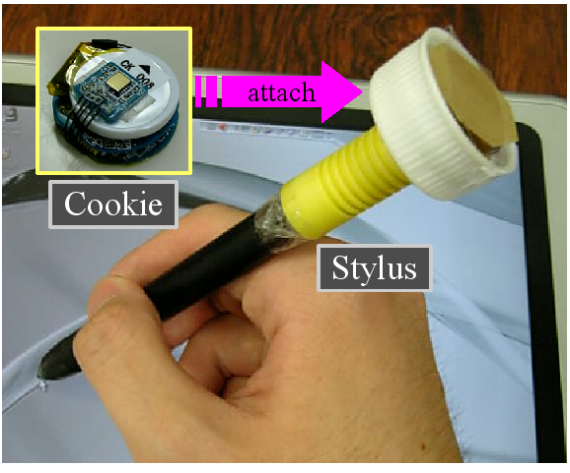

Fig. 1. Stylus with an attached Cookie

3.2 Converting Raw Accelerometer Data into Semantic Data

In this research, we sought to enable interaction with a computer through a "rolling" operation, meaning that the stylus was rotated around the pen axis, and a "shaking" operation, where the stylus was shaken up and down along the pen axis.

We had to convert low level data (acceleration data) from the accelerometer into semantic data to control applications through the rolling and shaking motions. First, three-axis acceleration data (roll, pitch, and yaw) was obtained from Cookie (Fig. 2). Cookie communicated with the computer every 100 ms. The range of acceleration for each direction was -3G to +3G. The rolling motion was detected from the roll and pitch accelerations, and the shaking motion was obtained from the yaw acceleration.

For rolling, we calculate the stylus angle of rotation from the accelerations on two axes (roll and pitch) and the system recognizes rolling from a change in the rotation

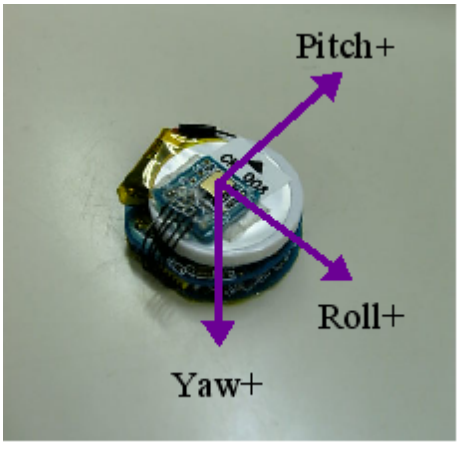

Fig. 2. 3-axis of Cookie

angle from the state 100 ms earlier. We calculate the angle of rotation from the arctangent. If the current roll acceleration is r_t and the current pitch acceleration is p_t, the current angle θ_t can be represented by the following expression.

$$\theta_t = \tan^{-1} p_t / r_t .$$

Likewise, the angle 100 ms earlier, θ_{t-1}, can be represented as

$$\theta_{t-1} = \tan^{-1} p_{t-1} / r_{t-1} .$$

By using θ_t and θ_{t-1}, we can calculate the stylus angle of rotation θ. The way that θ is used depends on each application. (We describe a detailed example of usage in the next section.) Other ways could also be used to recognize the rolling operation.

For shaking, we calculate the stylus angle of rotation from the acceleration on one axis (yaw). When the user raises the stylus, the yaw acceleration decreases, and when the user brings the stylus down, the yaw acceleration increases. If the yaw acceleration 100 ms earlier is y_{t-1} and the current yaw acceleration is y_t, the range for which the system will recognize shaking can be represented by

$$|y_{t-1} - y_t| >= 2G .$$

3.3 Application

We have created a group of applications, called the "Oh! Stylus Series," which are compatible with a context sensitive stylus. We used the following APIs and toolkits.

- Java 2 Platform Standard Edition Development Kit 5.0
- Java Communications API
- Blue Cove (Java Bluetooth API)
- SWT (Standard Widget Toolkit)

3.3.1 Oh! Stylus Draw

Oh! Stylus Draw is a paint tool that allows us to use a context sensitive stylus. Oh! Stylus Draw makes it possible to

(1) change a drawing color by rolling
(2) switch a color palette by shaking
(3) switch a drawing mode by holding down a barrel button + press + rolling.

A screenshot of Oh! Stylus Draw is shown in Fig. 3. There are five panels on the left of Fig. 3, and the upper four panels represent drawing modes (freehand line, straight line, rectangle, and ellipse). The user can switch between these modes by holding down a barrel button + press + rolling.

If the user rolls the stylus in a clockwise direction, the drawing mode switches to the mode one panel downwards (if the current drawing mode is that of the lowest panel, the next mode is the mode of the highest panel). If the user rolls the stylus in a counterclockwise direction, the drawing mode switches to the mode one panel upwards (if the current drawing mode is that of the highest panel, the next mode is the

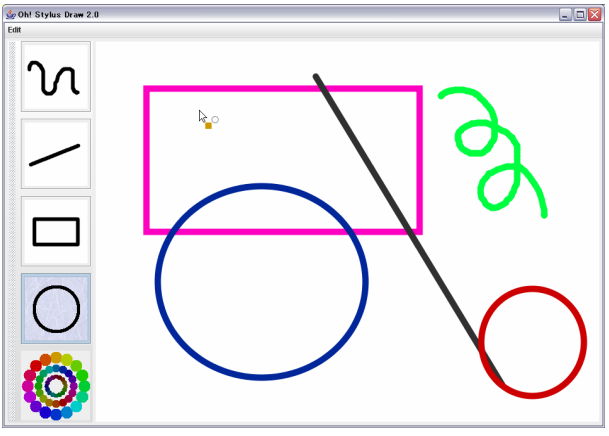

Fig. 3. Oh! Stylus Draw

mode of the lowest panel). This application recognizes clockwise rolling when the difference in the rotation angle from 100 ms earlier exceeds 50° and counterclockwise rolling action when the difference is under -50°. The user receives sound feedback from the application when he switches the drawing mode.

The lowest panel is the color palette panel, which has three zones called the first, second, and third zones (Fig. 4). There are five palettes, and when the user shakes the stylus, these palettes move through the zones. That is, the palette in the first zone is active and the user can choose a drawing color from this palette. When the user shakes the stylus, the palette from the second zone moves to the first zone and becomes active. At the same time, the palette in the third zone moves to the second one and the next palette in the rotation is displayed in the third zone. The motion of the palette switching is animated so that the user can easily see that the palette has changed. When the palette changes, the application also provides sound feedback.

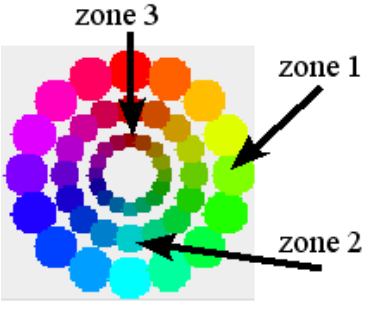

Fig. 4. Zones of the color palette panel

One color palette consists of 16 small circles, each representing one drawing color. Because there are five palettes, the total number of colors is 80. If the user's preferred color does not exist in any of the palettes, he can change one color into his preferred color by tapping one small circle. The drawing color is the color at the top of the current color palette, and the user can choose a drawing color from the current color palette by rolling. If the difference in the rotation angle from 100 ms earlier is over 50° when he rotates the stylus in the clockwise direction, the palette also rotates in the clockwise direction and the drawing color changes to the next color on the left. If the difference in the rotation angle is under -50° when he rotates the stylus in the counterclockwise direction, the palette rotates in that direction and the drawing color changes the next color on the right. The

palette rotation is animated, so the user can easily tell when the drawing color is changed. When the palette rotates, the application also provides sound feedback.

3.3.2 Oh! Stylus Scroll

Oh! Stylus Scroll is a scroll support tool that makes it easy for the user to scroll up or down (left or right) on a screen by rolling the stylus. Oh! Stylus Scroll can be used for most applications that have a scroll bar.

A screenshot of Oh! Stylus Draw is shown in Fig. 5. If the user rolls the stylus in a clockwise direction, the screen scrolls down (right), or vice versa. In this application, we provide a mouse wheel event as a target application.

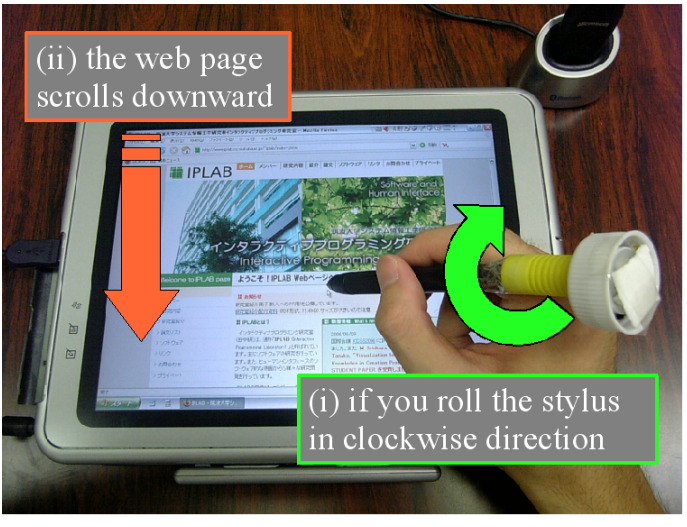

Fig. 5. Oh! Stylus Scroll

The amount of scrolling depends on the difference in the rotation angle from 100 ms earlier. If the difference is under 10°, no scrolling occurs. This prevents unintended scrolling. If the rotation angle difference, dif, is 10° to 100° degrees, the amount of scrolling, s, is calculated as

$$s = dif / 10 \text{ (rounded off to the nearest 10)} .$$

For example, when the change in the rotation angle is 73°, the amount of scrolling is seven lines. If the change exceeds 180°, though, no scrolling occurs; this is to prevent misrecognition of rotation in the opposite direction.

4 Evaluation

4.1 Purpose and Evaluation Method

We did an experiment to evaluate the usability of our proposed technique. The participants were nine students majoring in computer science. Each used Oh! Stylus

Draw and Oh! Stylus Scroll as they like, and then they completed a questionnaire about the usability.

4.2 Result

4.2.1 Oh! Stylus Draw

Typical comments obtained from the completed questionnaires included the following:

- shaking is easy to use
- the sound feedback is easy to understand
- it is easy to draw a gradation picture
- when the participants brought their stylus close to the screen, the drawing color which they had chosen by rolling changed without their intention
- it took a long time to choose a color on the opposite side of palette from the current color

We obtained many other comments, but all participants said that shaking was easy to use.

4.2.2 Oh! Stylus Scroll

Typical comments included the following:

- scrolling by rolling is convenient
- the participants took a long time to figure out the relationship between the difference in the rotation angle and the amount of scrolling
- when the participants tried clicking a hyperlink, the web page tended to scroll without their intention

5 Discussion

The evaluation showed that shaking is easy to use as an operation of a context sensitive stylus. Shaking seems to be a natural action for the user, because it is similar to a typical action when using a mechanical pencil. While some participants commented that rolling is convenient in Oh! Stylus Scroll, others commented that rolling is not an effective way to choose the drawing color in Oh! Stylus Draw. Rolling might not be a common action when we use a pen and the participants did not get used to this action. Because the evaluation results regarding rolling differed between Oh! Stylus Draw and Oh! Stylus Scroll, we think that assigning rolling to effective applications and operations will make it a more easily used interface.

We received many comments from the participants, some positive and others negative. However, most of the negative comments did not indicate that our interface was ineffective in concept; rather they indicated that there were problems in the implementation.

In the future, we plan to test how the display size affects the results of such an evaluation.

6 Related Work

Our goal is to improve computer interaction through stylus use. To achieve this, we want to use the various ways a user can manipulate a stylus by attaching an accelerometer to a stylus so that a computer can detect these manipulations and react to them. For this reason, we describe related work from two fields of research:

(1) research to improve computer operability with a stylus
(2) research on interface devices with an attached accelerometer.

6.1 Research to Improve Computer Operability with a Stylus

Miura et al. proposed RodDirect [1], which is an interaction technique where a stylus is used to resemble non-pen objects. RodDirect associates a stylus with physical metaphors. The user operates a stylus which is in a stylus holder. The amount of stylus rotation and parallel movement in the stylus holder is used as input to a computer. Dual Touch [2], proposed by Matsushita et al., detects multiple pointing points. The user interacts with a PDA by using a stylus and his thumb. In RodDirect and Dual Touch, because the user holds a PDA in one hand and operates the PDA with the other hand, both hands are needed for operation. In contrast, our proposed technique allows the user to operate a PDA with only one hand.

Smith et al. proposed the radial scroll tools [3]. In this system, when the user lays a stylus on a display, a guide appears at the pointing point. The user can then use gestures, like drawing a circle at the pointing point, to scroll a screen. While this system utilizes the advantage of a pen by gesturing, it uses only stroke information. In contrast, our technique uses additional information that cannot be obtained through a conventional stylus.

Siio et al. [4] proposed an interaction technique which uses Paperweight Metaphor. They attach a touch sensor electrode to the bottom or a corner of a PDA and use these sensors to provide seamless and intuitive switching between the scrolling mode and the editing mode. In this system, although the user can operate a PDA with one hand, the usage scenes are limited. In contrast, our technique has a wide variety of usage scenes.

6.2 Research on Interface Devices with an Attached Accelerometer

Rekimoto [5] described the effectiveness of using an accelerometer to operate computers. Since then, much research on this use of accelerometers has been done.

Harrison et al. proposed Squeeze Me, Hold Me, Tilt Me! [6]. They attached an accelerometer to a handheld PC and detected the tilt of the PC. They also developed some applications which can use the tilt information; for example, by tilting a handheld PC, the user can turn pages displayed on the PC. Hinckley et al. [7] attached an accelerometer to a PDA so that a user could scroll a screen by tilting the PDA. Rekimoto proposed Gesture Wrist and Gesture Pad [8], which use gestures for input to a wearable computer. These systems recognize hand gestures through sensors such as an accelerometer. Andrew et al. developed XWand [9], which enables the operation of electronic devices, such as a desk lamp, by gestures. By twisting or

shaking the XWand, the user can control an electronic device towards which the XWand is pointed. This system uses various sensors, such as an accelerometer.

As above, accelerometers are frequency used for perceptive user interface (PUI) research. Recent improvements in semiconductor technologies have allowed more compact accelerometers with greater performance. This has made it feasible to create small computer devices that incorporate an accelerometer.

7 Conclusion and Future Work

We have proposed a new interaction technique to enrich stylus-based interaction with computers. To apply various types of stylus manipulation, we attached an accelerometer to a stylus to create a context sensitive stylus. Through this stylus, our computer could recognize a rolling operation, where the stylus is rotated around the pen axis, and a shaking operation, where the stylus is shaken up and down along the pen axis. As applications which utilize a context sensitive stylus, we implemented the Oh! Stylus Series (i.e., Oh! Stylus Draw and Oh! Stylus Scroll). To control these applications, we found we could use rolling and shaking of the stylus. We then evaluated the usability of our proposal technique with respect to the Oh! Stylus Series.

In an evaluation experiment, we learned that shaking is easy for users to use as an operation of a context sensitive stylus. In contrast, the usability of rolling depended on how it was assigned to application operations. Because rolling and shaking are new interaction techniques, further work is needed to determine what kinds of application can best utilize rolling and shaking.

As our future work, we also plan to implement applications to improve the productivity of our proposed technique. In addition, we are developing and will soon implement the following additions to the Oh! Stylus Series:

- Oh! Stylus Map (map viewer)
- Oh! Stylus Popie (character input system by using Popie [10])
- Oh! Stylus Desktop Tools (to improve the desktop environment)

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

References

1. Miura, M., Kunifuji, S.: RodDirect: Two-Dimensional Input with Stylus Knob. In: Proceedings of the 8th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2006), pp. 113–120 (2006)
2. Matsushita, N., Ayatsuka, N., Rekimoto, J.: Dual Touch: a two-handed interface for pen-based PDAs. In: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST'00), pp. 211–212 (2000)
3. Smith, G.M.: schraefel, m.c.: The radial scroll tool: scrolling support for stylus or touch-based document navigation. In: Proceedings of the 17th annual ACM Symposium on User Interface Software and Technology (UIST'04), pp. 53–56 (2004)

4. Siio, I., Tsujita, H.: Mobile Interaction Using Paperweight Metaphor. In: Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST'06), pp. 111–114 (2006)
5. Rekimoto, J.: Tilting operations for small screen interfaces. In: Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology (UIST'96), pp. 203–204 (1996)
6. Harrison, B.L., Fishkin, K.P., Gujar, A., Mochon, C., Want, R.: Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'98), pp. 17–24 (1998)
7. Hinckley, K., Pierce, J., Sinclair, M., Horvitz, E.: Sensing Techniques for Mobile Interaction. In: Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST'00), pp. 91–100 (2000)
8. Rekimoto, J.: GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices. In: Proceedings of the Fifth International Symposium on Wearable Computers (ISWC'01), p. 21 (2001)
9. Wilson, A., Shafer, S.: XWand: UI for intelligent spaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2003), pp. 545–552 (2003)
10. Sato, D., Shizuki, B., Miura, M., Tanaka, J.: Menu-Selection-Based Japanese Input Method with Consonants for Pen-Based Computers. In: Proceedings of 6th Asia-Pacific Conference on Computer-Human Interaction (APCHI 2004), pp. 399–408 (2004)