Two-Handed Interaction on a Tablet Display

Ka-Ping Yee
Group for User Interface Research
University of California, Berkeley
Berkeley, CA 94720-1776 USA
ping@zesty.ca

ABSTRACT

A touchscreen can be overlaid on a tablet computer to support asymmetric two-handed interaction in which the non-preferred hand operates the touchscreen and the preferred hand uses a stylus. The result is a portable device that allows both hands to interact directly with the display, easily constructed from commonly available hardware. The method for tracking the independent motions of both hands is described. A wide variety of existing two-handed interaction techniques can be used on this platform, as well as some new ones that exploit the reconfigurability of touchscreen interfaces. Informal tests show that, when the non-preferred hand performs simple actions, users find direct manipulation on the display with both hands to be comfortable, natural, and efficient.

Author Keywords

Asymmetric bimanual interaction, tablet computing, touch-sensitive screens, commodity hardware.

ACM Classification Keywords

H5.2. [Information interfaces and presentation]: User interfaces — Input devices and strategies.

INTRODUCTION

The most common device for direct input on a computer display is the touchscreen. Touchscreens can be found everywhere, from banking machines to airports. No doubt this popularity is due in part to the simplicity of using a touchscreen: the only equipment needed to operate one is your finger, and pressing buttons on the screen with a finger is more direct and obvious than using pointing devices that are separated from the display, such as mice. Touchscreen-based computers and touchscreen add-ons for computer displays have been commercially available for many years. Most of these products detect one point of contact at a time and do not measure pressure.

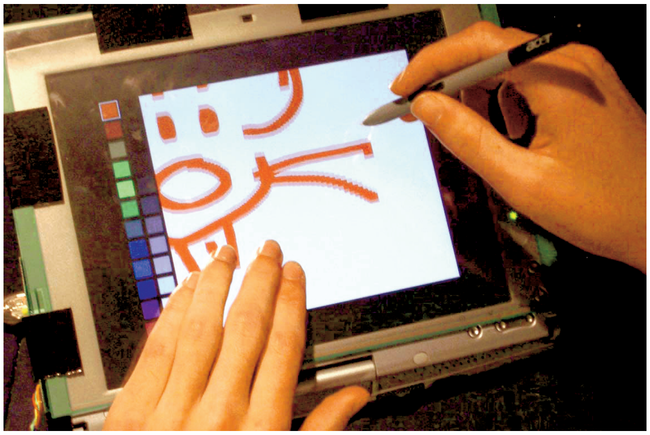

Figure 1. Two-handed tablet interaction example. The left hand positions the canvas for drawing by the right hand.

More recently, tablet displays have become much more available to computer users due to the introduction of Tablet PCs. Tablet displays combine an LCD with the same type of electromagnetic stylus technology found in Wacom graphics tablets. Tablets have very high resolution as well as the ability to detect pen pressure levels. A touchscreen can only sense position while being pressed, whereas a tablet can distinguish pointing from selection because the stylus can be sensed while it hovers over the screen. However, a tablet is useless without its stylus.

This work looks at the new possibilities afforded by combining these two input devices so that both hands can interact simultaneously with the display.

BACKGROUND AND RELATED WORK

The Case for Asymmetric Tasks

Many researchers have explored two-handed interaction techniques using a variety of input devices. Buxton and Myers [3] showed that both novice and expert users can use two hands together to perform compound tasks faster than with one hand. However, not all two-handed techniques are advantageous. Kabbash [8] showed that using both hands is most effective for "asymmetric dependent" tasks, whereas using both hands to simultaneously operate independent controls can be worse than one hand alone.

The definition of an asymmetric dependent task comes from the work of Guiard [4], who proposed a Kinematic Chain model of bimanual action. This model suggests that the preferred (P) hand moves within the frame of reference set by the positioning of the non-preferred (NP) hand; that the P hand engages in small, quick, frequent actions while the NP hand is used for larger, slower, less frequent actions; and that the NP hand precedes the P hand to set the frame of reference for its detailed actions. Guiard's hypothesis is supported by the results of several studies [4, 6, 8].

Combining a touchscreen with a tablet is a good fit for the Kinematic Chain model. The touchscreen's lower precision and tool-free operation are suitable for the NP hand's coarser and less frequent actions. The higher speed and precision of the P hand justify the use of a stylus.

Combining Reference Frames

In some two-handed systems, the two hands have separate physical reference frames, such as when using two independent pointing devices [7, 8]. In other systems, the hands have a combined reference frame, such as when using two sensors on a single tablet [9]. Hinckley et al. [5] showed that people have a strong kinesthetic sense of the positions of their two hands with respect to one another, which suggests that it is beneficial to use both hands in the same physical reference frame.

The touchscreen-augmented tablet presented here is an interesting platform because, unlike other two-handed systems, it places both hands and the display in the same reference frame using inexpensive commodity hardware.

Rekimoto's SmartSkin interactive table [13] shows how compelling an interaction environment can be when the reference frames of both hands are unified with the display. The table incorporates a custom sensor and a projected display, though its creator anticipates future versions of the SmartSkin with transparent electrodes. Combining an LCD tablet with a transparent SmartSkin sensor (instead of a touchscreen) would be an exciting future possibility.

APPARATUS

In this prototype, the tablet component is an Acer Travelmate C100. Any Tablet PC or tablet-enabled LCD would suffice. The tablet senses the stylus position based on electromagnetic resonance; contact and pressure are detected by the tip of the stylus, not the tablet surface. Thus, the tablet continues to function normally even if there is extra material between the stylus and the tablet surface.

The touchscreen component is an external resistive touch-sensing panel for laptop computers called the TouchNote from OneTouch. The frame of the panel was removed in order to bring the glass surface closer to the screen, and then the sheet of glass was taped in place over the LCD.

IMPLEMENTATION

The tablet reports the pen position accurately regardless of other touches on the screen. However, the touchscreen is affected by any touch, including the pen. To allow the pen and finger to independently convey positional input, the software driver has to distinguish three conditions:

- *Finger touching, no stylus*. The touchscreen reports a position and the tablet reports no stylus.
- *Stylus touching, no finger*. The tablet and the touchscreen both report the stylus position.
- *Both finger and stylus touching*. The tablet reports the stylus position. The touchscreen reports a position somewhere between the finger and stylus.

Position Determination

When both finger and stylus are touching the screen, the reported touch position is a consistent function of the two positions, and the stylus position is known, so the true finger position can be estimated. Dual Touch [10] applies the same principle to touchscreens on PDAs, with the assumption that the reported value is the midpoint between the two touch points. For a large touchscreen, the reported touch position can be quite far from the midpoint. Hence, we find the finger position by referring to a grid of previously measured calibration values.

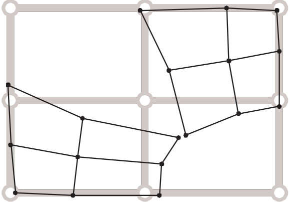

Figure 2. Examples of touch calibration grids. Grey circles are the touched points, black dots are the measured points. The two grids shown are for the pen at the bottom-left and top-right calibration points.

In this implementation, we calibrate with nine reported touch points in each grid. The grid of reported touch points is influenced by the pen, so we record nine touch grids – one for each calibration position of the pen. For an arbitrary pen position, we compute the appropriate touch grid by interpolating the calibration grids, and then use that touch grid to translate the touch point into the finger location in screen coordinates.

The precision of the measured finger position decreases when the pen is also down, in part because the reported position varies slightly depending on the finger pressure. However, the prototype demonstrates that even when the pen is down, it is possible to find the finger point to within about 25 pixels on a 1024-by-768-pixel screen, which is adequate for pressing buttons or making coarse motions.

Contact Determination

While the stylus is touching the screen, both the tablet and stylus report contact. To determine whether a finger is also touching the screen, we assume that the stylus and finger will not touch the same spot, and consider the finger to be touching only while the reported touch position is a little distance away from the reported stylus position.

If the pen makes contact with the screen while the finger is down, the reported touch position will fly away from the finger towards the pen, then settle to a point in between. Finger coordinates should not be reported to the application during this transition; they would make the finger position jitter when it should be stationary. Fortunately, the pen enters a hovering state before touching the screen, giving advance warning of where the touch position might jump.

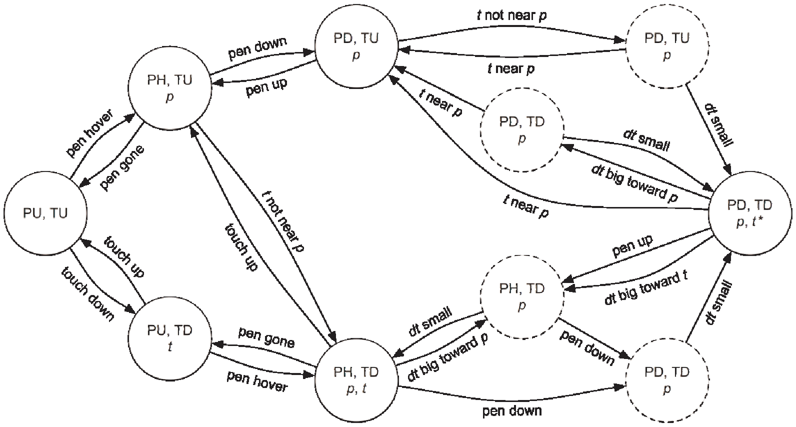

Figure 3. State diagram for the driver. The outputs PU, PH, PD, TU, and TD stand for "pen up", "pen hover", and "pen down", "touch up" and "touch down". The output positions *p* and *t* are in screen space. The dashed circles are "settling states", in which touch points are suppressed while waiting for the reported touch point to stabilize.

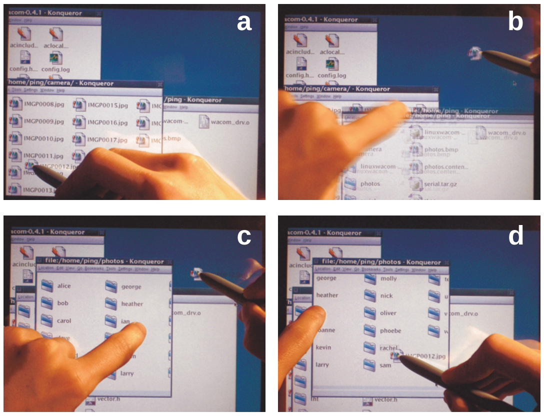

Figure 4. a. The user picks up a file icon with the stylus. **b.** While holding the file, the user can rearrange windows with the NP hand to find the drop target. **c.** The target folder is not visible in the window, so the NP hand is used to scroll. **d.** The target is revealed; P hand arrives to drop the icon.

Similar settling periods are required when the pen is lifted, leaving the finger on the screen, or when the finger makes or breaks contact while the pen is down. Figure 3 shows the state diagram for the software driver that translates events from both devices into pen and finger positions and contact states to report to the application.

INTERACTION TECHNIQUES

Three simple applications were written to test this device.

Painting Program

The painting program is depicted in Figure 1. The P hand holds the stylus and draws on the canvas. The NP hand can scroll the canvas by directly dragging it, much like the NP hand positions a sheet of paper for handwriting in real life. The NP hand can also select from the palette on the left, freeing the P hand from having to interrupt its task.

Zoom Viewer

This program tests the feasibility of the "stretch and squeeze" technique [7, 9] for navigating and orienting large spaces in a tablet context. The technique is based on the idea of sticking the pointers to the drawing. The pen or the finger can be used alone to grab and scroll the drawing. When both the finger and the pen touch the screen, the finger anchors the drawing and the pen movement rotates and zooms the drawing about the anchor point.

File Browser

The file browser, shown in Figure 4, displays overlapping windows like a typical GUI desktop. This program was written to test a possible solution to the problem of finding a drop target for a dragged object: since the mouse pointer is already occupied with an object, it can no longer distinguish pointing from selection, which leaves very little expressive control for navigating to the drop target. In this application, the NP hand can rearrange windows, open folders, and scroll the contents of folders to reveal the target for dropping the object "held" by the P hand.

EVALUATION

The three programs were tested informally with users who had no prior exposure to two-handed interfaces.

For the painting program, six users were asked to scribble all over a large canvas (four times the area of the screen). While doing so, they were asked to try scrolling using either hand, and selecting from the palette using the stylus or using their ND finger. When asked which was more comfortable, all of them preferred to select tools using the ND hand, and all but one preferred to scroll using the ND hand. All of them were comfortable using the painting program within a few seconds. The touch-based panning technique was described as "easy", "quick", and "natural".

For the zoom viewer, six users were asked to find a cluster of 20 red objects in a large field (initially 200 times the area of the screen) of 300 other multicoloured objects, and to arrange the view optimally to show the red objects as clearly as possible. All of the users were able to arrange the view fairly quickly (in 15 to 30 seconds), and all but one reported that they were comfortable with the zoom-and-rotate technique. When asked to comment, users said that it was "very fast" and that it "makes sense".

The file browser was tested by two people. The required task involved picking up and holding an icon with the stylus while opening and scrolling in another window using the NP hand to locate the drop target. One user was able to accomplish the task easily; the other had trouble operating the touchscreen with the NP hand. The operations required of the NP hand were probably too complex in this task. However, there may be other good uses for this type of interaction, in which the P hand suspends a task while the NP hand executes a subroutine to support the task.

DISCUSSION

In typical one-handed GUIs, the NP hand remains at the keyboard to activate application commands while the P hand controls the pointer. Balakrishnan and Patel [1] pointed out that many two-handed interfaces do not compensate well for taking the NP hand away from the keyboard, relying instead on the P hand to serially select a tool and then apply the tool. They suggest that the NP hand should have "an input device that allows rapid activation of commands and modifiers in addition to performing spatial positioning." The touchscreen has this property.

While the touchscreen lacks the tactile feedback of a keyboard, controls and buttons on a touchscreen can provide more informative cues as to what they do. Moreover, touchscreen controls have the advantage of being reconfigurable as appropriate to the current task. For example, part of the tool palette could change to provide modifier buttons appropriate for the current tool, such as constraints or snapping options. The NP hand can toggle or hold down these modifiers without interrupting the P hand.

The tradeoff for using both hands directly on the screen is that the hands can obscure the display. Also, as with any technique that puts both hands in the same reference frame, there can be the potential for the hands to collide or interfere with each other. Careful layout of touchscreen controls and judicious use of two-handed positioning techniques can help mitigate these problems. In particular, designing for the wrong preferred hand has a larger impact on the user, so touchscreen interfaces should support switching of the preferred hand.

The form factor of a tablet computer is an interesting distinguishing attribute. A tablet computer is made to be portable; adding a touchscreen does not impede its mobility. The result is an unusually portable two-handed interface; most other two-handed interfaces require an external input device for the NP hand or a separate display projector. Also, on a tablet computer, moving the NP hand between the keyboard and touchscreen is easier than between the keyboard and a mouse, because the screen is close to the keyboard, and because the fingers can operate a touchscreen while leaving the hand over the keyboard.

Many other two-handed interaction techniques are possible on this platform, including, for example, the toolglass and all kinds of see-through tools [2], Raisamo's alignment stick and alternative drawing techniques [12], two-handed stretchies [9], navigation/selection [3], positioning/scaling [3], buttons and scrolling controls for the NP hand [11], and marking keys [1].

A drawback of this system is that it assumes at most two points of contact on the screen – the stylus and one other point. Thus, if the application is tracking the movements of both hands simultaneously, the user must be careful not to touch the screen with the pen hand while drawing. However, many two-handed interaction techniques (e.g. all the techniques mentioned in the preceding paragraph except for positioning/scaling and two-handed stretchies) can work without requiring both hands to move simultaneously. For these techniques, the status of the finger point can be frozen while the pen is detected hovering over the tablet. It is then safe to rest the P hand on the tablet while using the pen.

CONCLUSIONS AND FUTURE WORK

The combination of touchscreen and tablet display is an effective platform for a wide variety of asymmetric bimanual interaction and direct manipulation techniques. The simplicity of its construction and the availability of its parts make it a good device for researchers who are testing and experimenting with new interface ideas.

Beyond merely re-implementing interaction techniques that were developed for other devices, there are also many opportunities for further research into new two-handed interaction techniques that exploit the particular strengths of a touchscreen, such as the reconfigurability of the interface, the direct interaction with the display, or the portability of the entire system.

ACKNOWLEDGEMENTS

I am grateful to Rebecca Middleton, Kragen Sitaker, and David Wallace for their help and support of this work, and to Dean Townsley for publishing helpful information about running Linux on the Acer Tablet PC. I would also like to thank all the people who participated in the informal user tests. This work is supported by an IBM Ph. D. Fellowship.

REFERENCES

1. R. Balakrishnan, P. Patel. The PadMouse: facilitating selection and spatial positioning for the non-dominant hand. *Proc. CHI 1998*, 9–16.
2. E. Bier, M. Stone, K. Fishkin, W. Buxton, T. Baudel. A Taxonomy of See-Through Tools. *Proc. CHI 1994*, 358–364.
3. W. Buxton, B. A. Myers. A Study in Two-Handed Input. *Proc. CHI 1986*, 321–326.
4. Y. Guiard. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *J. Motor Behavior* 19 (4), 1987, 486–517.
5. K. Hinckley, R. Pausch, D. Proffitt. Attention and Visual Feedback: The Bimanual Frame of Reference. *Proc. 1996 Symposium on Interactive 3-D Graphics*, 121–126.
6. K. Hinckley, R. Pausch, D. Proffitt, J. Patten, N. Kassell. Cooperative Bimanual Action. *Proc. CHI 1997*, 27–34.
7. K. Hinckley, M. Czerwinski, M. Sinclair. Interaction and Modeling Techniques for Desktop Two-Handed Input. *Proc. UIST 1998*, 49–58.
8. P. Kabbash, W. Buxton, A. Sellen. Two-Handed Input in a Compound Task. *Proc. CHI 1994*, 417–423.
9. G. Kurtenbach, G. Fitzmaurice, T. Baudel, W. Buxton. The Design of a GUI Paradigm based on Tablets, Two Hands, and Transparency. *Proc. CHI 1997*, 35–42.
10. N. Matsushita, Y. Ayatsuka, J. Rekimoto. Dual Touch: A Two-Handed Interface for Pen-Based PDAs. *Proc. UIST 2000*, 211–212.
11. B. A. Myers, K. P. Lie, B-C. Yang. Two-Handed Input Using a PDA and a Mouse. *Proc. CHI 2000*, 41–48.
12. R. Raisamo. An Alternative Way of Drawing. *Proc. CHI 1999*, 175–182.
13. J. Rekimoto. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. *Proc. CHI 2002*, 113–120.