

A Menu Interface for Wearable Computing

Gábor Blaskó, Steven Feiner

Department of Computer Science, Columbia University

New York, NY 10027

[gblasko, feiner]@cs.columbia.edu

Abstract

We present a menu interface designed primarily for head-worn displays that have a small field-of-view. To support interaction with a hierarchical menu, we logically divide an absolute positioning device into finger-operated strip segments, which we use as one-dimensional scrolling devices. Our menu system is intended to make user interaction faster by not requiring constant visual feedback. This is preferable for interaction in which the visual user interface elements occupy only a small portion of the eye's entire field-of-view and in which navigating in menus with a pointer would be awkward and time-consuming. With our approach, it is even possible for the user to use peripheral vision for interaction, since there is no need to precisely position a small pointer on the screen. Thus, the user can maintain eye contact with others or keep his or her focus of attention on the environment while using a wearable device.

Introduction

For some wearable computing applications that use head-worn displays, it is preferable to have a visual user interface that occupies only a very small portion of the eye's entire field-of-view. For example, the eyeglass display systems made by MicroOptical Corp. are relatively small and lightweight, but have a small field of view [1]. We are designing a menu system that takes into consideration this small screen size constraint. Furthermore, input devices for wearable computing systems should not only be as unobtrusive as possible [2], but should also take the least amount of time possible. Since controlling a pointer in two dimensions on a small display can be quite time-consuming and awkward, we have developed an approach that does not require the use of a screen pointer, but still allows fast multi-level menu navigation.

Interface Design

We have built a prototype using a Synaptics TouchPad™ device and have created a customized interface for it. Instead of using the touchpad to control a pointer on the display, we use it to report only the absolute coordinates of finger contact. Since the width of the touchpad is almost equal to the width of four fingers, using software we subdivided the entire touch-sensitive surface area into four independent vertical strips (similar to [6]), each corresponding to a finger of the hand, excluding the thumb, as shown in Figure 1. This design can also be

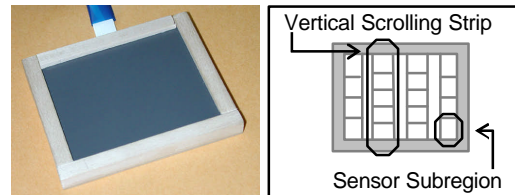


Figure 1: Touchpad divided into multiple scrolling strip regions.

applied to narrower touchpads that support fewer fingers. The horizontal coordinate is used only to determine the vertical strip within which a finger makes contact. These strips are further subdivided into smaller vertical regions, as described below.

Interaction Approach

We illustrate the use of this interaction device by describing a multi-level menu navigation system. Since our prototype has four vertical strip sensor regions, the screen can accommodate up to four top-level menus. Each top-level menu controls its own independent multi-depth menu hierarchy. The user opens up one of the four top-level menus by touching the pad in the corresponding vertical strip region. Sample menus, corresponding to the four vertical strips, are illustrated in their open states in Figure 2. A top-level menu can have multiple submenus, each representing a level in the hierarchy.

As shown in Figure 2, the vertical strips are subdivided into smaller sensor subregions, the height of which dynamically change, depending on how many menu elements there are to choose from at a particular depth level. As the user glides a finger along a vertical strip, a new menu item is highlighted upon crossing the boundary of two subregions.

When touchpads are used in laptops, a menu item is typically selected either by pressing an external button or by tapping repeatedly on the pad in the same spot. In contrast, if a touchpad is worn on the body, out of sight, it could be awkward and time consuming to blindly find the external button and then press it. Also, tapping in the same spot on the pad can be difficult when the body part onto which the touchpad is attached is in motion. One solution we considered for selection was to use the act of lifting the finger off the pad surface while a menu is highlighted. However, as pointed out by [6], we felt that the act of selection should be analogous to pushing a button rather than pulling it. Therefore, in our system a menu item is selected by touching the lower half of the adjacent vertical strip with an adjacent finger, as described below. Since two adjacent fingers of the hand

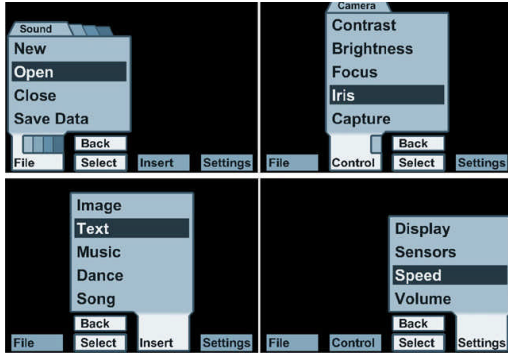


Figure 2: Sample menus corresponding to each linear sensor region and finger.

can move independently when both are in the proximity of the pad, we have found this to be an easily executable, less error-prone, and fast way to activate the highlighted menu item and take a step down in the menu hierarchy. It is possible to back up in the hierarchy by touching the upper half of the adjacent vertical strip. Unfortunately, current small touchpads are not capable of detecting multiple simultaneous finger contacts. Therefore, in our prototype, the finger that is used to highlight a menu item must be lifted off the pad before the adjacent finger makes contact. This would not be necessary with a multi-touch device. (We are currently experimenting with a larger multi-touch touchpad from Tactex Controls Inc.)

The current depth within a given top-level menu is indicated to the user by tabs at the top of the selection region. The top left image in Figure 2 shows the submenu at depth four of the first top-level “File” menu.

To minimize the amount of screen space occupied by the menu system, we needed to modify the shape of the menus to allow easy legibility with large font sizes. We decided to extend the menus horizontally. Depending on which top-level menu is opened, the direction of this extension needs to be flipped, as seen in the upper and lower portions of Figure 2. Thus, the leftmost two menus are extended to the right and the “adjacent finger” used for selection is the neighboring finger to the right of the selection finger, while the rightmost two menus are extended to the left and the “adjacent finger” used for selection is the neighboring finger to the left of the selection finger. The words “Back” and “Select” are displayed next to the label of the top-level menu to indicate the selection method described above.

Discussion and Conclusions

When the user reaches for the device, which can be mounted in different places on the body [3], the sides of the touchpad’s enclosure can be sensed by the outer fingers, automatically orienting all four fingers above the device, with each finger above its own vertical strip. Since the top and lower edges of the touchpad’s enclosure can also be felt, the user is also aware of the upper and lower boundaries of the device, all without the need to look at the device. In contrast, when a mouse-style device is used, even though the hand can be placed on it without looking at it, the user always has to first find the initial screen position of the pointer and determine its position

relative to the graphical interface elements. In some applications, the attention and time required to do this is undesirable and may even be dangerous. We avoid this initial step by using an absolute device, with a graphical user interface that relies on the natural positioning of the fingers. The VuMan3 system [7] also addresses this problem by using a physical dial to navigate a menu. However, we believe that with frequent use, an absolute device can be faster than a relative device, such as a dial.

We chose the text size so as to make the menu labels legible even in peripheral vision. For example, the MicroOptical CO-1 eyeglass-based display system has a vertical field-of-view of 8 degrees [4], making 40 pixel-high text subtend $2/3$ degrees. According to psychophysical research [5] it is possible to read 200 words per minute at 15 degree eccentricity when words are displayed at approximately the same screen position in quick succession. This is about $1/4$ normal foveal reading speed. This means that if the eyeglass display is placed off center in the visual field, it is still possible to read the menu labels and navigate the menu system using peripheral vision, glancing down only briefly at the display to confirm selections. This would not be possible with a relative and indirect input device that requires continuous visual feedback.

The need to change the eye’s focus between the display and real world objects causes problems when using peripheral vision for navigation. We have found that navigation speed improves when the display and real world objects focus at the same apparent distance.

Our initial experience with the input method and graphical user interface suggests that users are capable of selecting menu items quickly, which we believe is due to pointer navigation not being needed in the small field-of-view display.

Acknowledgements

This research is funded in part by ONR Contracts N00014-99-1-0249, N00014-99-1-0394, and N00014-99-0683; NSF Grants IIS-00-82961 and IIS-01-21239; and gifts from Synaptics, Intel and Microsoft.

References

- [1] M. Spitzer, N. Rensing, R. McClelland, and P. Aquilino. Eyeglass-based systems for wearable computing, *Proc. ISWC '97*, 48–51.
- [2] J. Rekimoto. GestureWrist and GesturePad: Unobtrusive wearable interaction devices, *Proc. ISWC 2001*, 21–27.
- [3] B. Thomas, K. Grimmer, D. Makovec, J. Zucco, and B. Gunther. Determination of placement of a body-attached mouse as a pointing input device for wearable computers, *Proc. ISWC '99*, 193–194.
- [4] MicroOpticalCorp. www.microopticalcorp.com/Products/CO1.html
- [5] S. Chung, J. Mansfield and G. Legge. Psychophysics of reading. XVIII. The effect of print size on reading speed in normal peripheral vision, *Vision Research*, 38(19), Oct. 1998, 2949–2962.
- [6] W. Buxton, R. Hill, and P. Rowley. Issues and techniques in touch-sensitive tablet input, *Proc. SIGGRAPH '85*, 215–223.
- [7] L. Bass, C. Kasabach, R. Martin, D. Siewiorek, A. Smailagic, J. Stivoric. The design of a wearable computer, *Proc. CHI '97*, 139–146.