

Single-Handed Interaction Techniques for Multiple Pressure-Sensitive Strips

Gábor Blaskó, Steven Feiner

Department of Computer Science, Columbia University
500 W. 120th St., 450 CS Building
New York, NY, 10027 USA
[gblasko, feiner]@cs.columbia.edu

ABSTRACT

We present a set of interaction techniques that make novel use of a small pressure-sensitive pad to allow one-handed direct control of a large number of parameters. The surface of the pressure-sensitive pad is logically divided into four linear strips which simulate traditional interaction metaphors and the functions of which may be modified dynamically under software control. No homing of the hand or fingers is needed once the fingers are placed above their corresponding strips. We show how the number of strips on the pad can be virtually extended from four to fourteen by detecting contact pressure differences and dual-finger motions. Due to the compact size of the device and the method of interaction, which does not rely on on-screen widgets or the 2D navigation of a cursor, the versatile input system may be used in applications, where it is advantageous to minimize the amount of visual feedback required for interaction.

Author Keywords

Single handed input device, touch/pressure-sensitive interaction, cursorless user interfaces, wearable computing, virtual- and augmented reality.

ACM Classification Keywords

H.5.2. Input devices and strategies.

INTRODUCTION

Computer applications continue to increase in complexity and functionality; however, the input devices that are typically used to control these applications are not changing and evolving at a comparable pace to help counteract the increase in the complexity of graphical user interfaces (GUIs). Instead, application developers usually create programs for the “lowest common denominator” of input devices, currently a keyboard and a mouse-like pointing device. Therefore, introducing new functionality into most applications is often handled by making menus deeper, creating new on-screen icons, and adding more on-screen control elements, which take up additional screen space. If these elements are made smaller in size, they require the

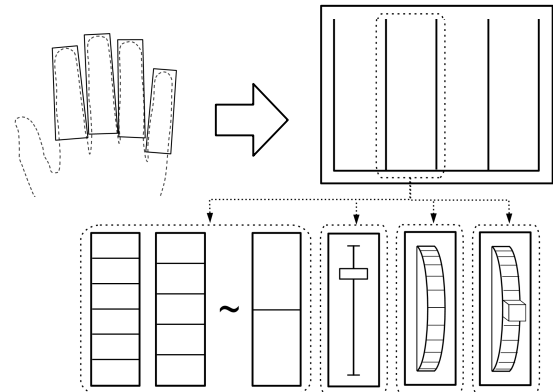


Figure 1. (a) Conceptual layout of four independent multifunctional strips. (b) Pressure-sensitive pad divided into four multifunctional strip segments: vertically resizing buttons, slider, spinning wheel, spring-loaded wheel.

user to navigate the cursor with higher precision, which results in slower interaction.

To address these problems, we have been developing a pressure-sensitive input device and associated interaction techniques that move a significant portion of the user interface off the screen and onto the input device. An important feature of our approach is that its interaction techniques do not use an on-screen cursor and are designed to make parsimonious use of the screen. Therefore, we feel that our approach is especially appropriate for wearable systems, which often have minimal screen real estate and are used by people who are busy with other tasks and who may function more effectively if they can avoid controlling a cursor.

In the remainder of this paper, we will present related work, describe our input device and a set of interaction techniques developed for it.

RELATED WORK

Current interaction techniques based on touch-sensitive pads include the use of a single finger to control a 2D cursor on many notebook computers, often supplemented with the ability to scroll windows using a designated thin strip along one edge of the pad [7]. The PadMouse [2], intended for use in the nondominant hand, replaces the buttons of a conventional mouse with a 2D touch-sensitive pad, and allows users to indicate commands and command modifiers

with 2D strokes of the index finger, using a variant of “marking menus” [6].

Inspiration for our work comes, in part, from the research conducted by Buxton and colleagues on touch-sensitive tablets and interaction techniques [5] especially involving multi-touch and multi-hand devices; in contrast, we use a single hand and one or two simultaneously touching fingers on a pressure-sensitive pad intended for single-touch use.

The interaction device described in the SpeechSkimmer system [1] uses a template to subdivide the surface of a touchpad into 8 linear regions (6 of which implement virtual sliders). This device has approximately three times the surface area (7x11cm) of ours, and requires the visual or tactile scanning of the device continuously to position the finger in the proper regions. Our system however has a one-to-one mapping of finger to linear strip, therefore no homing of the fingers is needed once the hand has been placed on the device and during interaction the hand does not need to move sideways at all. In [1] only a virtual slider and a few statically mapped virtual buttons are simulated, as opposed to our system which implements more metaphors, which are dynamically mapped. The device described in [1] has printed markings on it, since the functionality of the regions does not change. Our device is unmarked, since it was designed to be versatile, where each of the strips may have different functionality and each strip may implement a different metaphor for interaction. Our system may be arbitrarily configured in software to offer the best method of interaction for any given application.

MULTI-STRIP INPUT DEVICE

Conceptual Description

In concept, the multi-strip device is a set of physically independent, pressure-sensitive multi-functional strips, each approximately the width of a finger. Each strip corresponds to one of the user’s fingers and acts as an independent one-degree-of-freedom input device. Ideally, each of the user’s fingers should be able to reach the full extent of its strip, each strip should be shifted slightly vertically, relative to the other strips, and the set should be spread outwards, like a fan, for reasons of ergonomics, as illustrated in Figure 1(a).

In previous work [3,4], we introduced a one-handed, cursorless, menu-navigation approach, based on multi-strip interaction. Here, we extend this approach to make it possible for users to modify continuous, as well as discrete, parameter values by using the strips to simulate the traditional interaction metaphors: buttons, sliders, wheels, and spring wheels (commonly used on music synthesizer keyboards to bend sound pitch) as illustrated in Figure 1(b). In addition, while the full width of a physical pressure sensitive pad and the width of an individual human finger would appear to fix the maximum number of usable strips available across the pad to four, we later show how it is possible to overcome these physical limitations, and

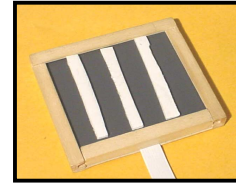


Figure 2. Photo of prototype multistrip input device

virtually more than triple the number of strips that may be used for application control.

Hardware Prototype

Our current prototype uses an electronically unmodified rectangular (6.4x4.2cm) Synaptics TM41P-200 TouchPad. It is a capacitance-based sensor that detects the effect of a finger on an array of capacitive lines integrated into the printed circuit board [7]. As shown in Figure 2, we placed physical dividers on the pad surface, to create four vertical linear segments and to offer tactile guides for placement of the fingers. These physical dividers create “grooves” that restrict finger motion to vertical movements and also visually convey that the user should not treat the device as a 2D cursor controller, but rather as a set of one-degree-of-freedom sensors. Unfortunately none of our prototype pads is large enough to allow for the vertical shifting of the strips or fan out as illustrated in Figure 1(a). Through the Synaptics TouchPad software interface, we were able to access the raw data stream that reports the 2D absolute coordinates of a single point of contact. We used the horizontal coordinate alone to determine in which of the four strips a finger had contacted the pad. The vertical coordinate and the pressure readings were processed for each strip independently, as described below.

INTERACTION TECHNIQUES

In the following subsections, we present the different interaction techniques that are possible with the linear strips.

Sliders

The simplest way in which the absolute nature of the linear strip can be used is as a slider, similar to [1]. The physical extents of a linear strip serve as predetermined thresholds for the upper and lower bounds. By sliding the fingertip up and down vertically within a strip, the user can set the value of a numeric parameter between two extremes (e.g., to scale a parameter from 0.0 to 1.0).

Dynamically Resizing Buttons

A linear strip may be divided vertically into multiple subregions. Tapping the strip in a desired subregion can execute a command like a button, or switch the state of a Boolean variable that a dual-state button would allow. Since the full strip is of constant length, the vertical size of the subregions scales to fit the strip, depending on the number of virtual buttons. Our informal experience has been that this method of command execution allows very quick access to functionality. After a very short training

period, while the user gets accustomed to the physical extent of the strip, direct execution of virtual button presses is easily accomplished for up to four subregions. This is expected since the prototype strips are approximately 40mm in length, making each sub-section 10mm in height, which is approximately the height of the contact area created by adult fingers on a pad surface.

Spinning Wheel with Discrete Increments

We define a substroke as the act of touching the strip in a subregion, moving into a neighboring subregion, and lifting the finger. If there are only two subregions, dividing the strip into two halves, an upward stroke would begin by touching the pad in the lower half and dragging the fingertip to the upper half, and lifting up the fingertip. Such a stroke motion is similar to spinning a small wheel. A single stroke can be used to increase or decrease the value of a parameter by a predetermined constant increment, each substroke being equivalent. If the strip is divided into more than two subregions, a single dragging motion can cross over multiple subregions, thereby allowing multiple substrokes to be executed, increasing or decreasing the value of a parameter by multiple discrete increments.

Spring Wheel with Continuously Varying Increments

It is useful to take advantage of the fact that within a strip we can measure the distance between the point where the fingertip makes contact and the point to which it is moved, while contact is maintained. The length of this dragging motion specifies the length of an upward or downward pointing vector. We can then change the value of a parameter, depending on the size and direction of this vector, until the finger is lifted off the pad. The motion is similar to that of a bidirectional spring-loaded modulator wheel, used on music synthesizer keyboards to bend the pitch up or down. The more the user drags the fingertip away from the point of initial contact, which may be anywhere on the strip, the longer the vector, and the larger the increment by which the parameter value changes.

INCREASING THE EFFECTIVE NUMBER OF STRIPS

Our prototype physical device is 64mm wide, which is approximately the width of four adult fingers. This physical constraint limits the number of directly accessible multifunctional strips to four, which we can extend to fourteen by taking advantage of the method by which the device reports sensor data, as explained below.

Pressure—Pop-Through Functionality

In addition to the horizontal X and vertical Y coordinates of the point of contact, the data stream reported by the TouchPad contains a third value, Z. The Z value relates to the total finger capacitance, which is a function of the finger's contact area, which is, in turn, affected by the finger's contact pressure and by the angle at which the finger is held [7]. As the angle between the finger and surface decreases, if firm pressure is applied, the soft portion of the fingertip

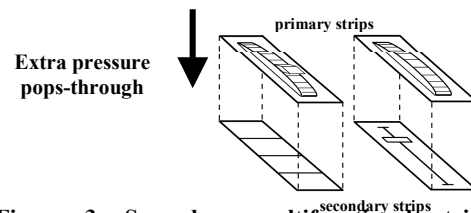


Figure 3. Secondary multifunctional strips are accessed by applying firm pressure to the strips. Function of primary and secondary strips may differ.

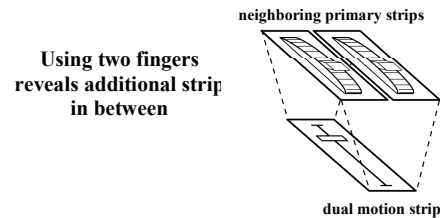


Figure 4. Dual motion strips are accessed by simultaneously touching neighboring strips with two fingers. Here, two single strips control separate spring wheels, but using two fingers simultaneously reveals a dual-motion strip that acts as a slider.

creates a larger contact surface, thereby producing higher Z values. The idea of accessing extra functionality by applying light and firm pressure to discrete buttons has been explored in the context of both desktop and VR interfaces [8, 9]. In these systems, after depressing a button lightly to access basic functionality, the user may depress it further, with additional tactile feedback, to “pop through” to either an expanded version of the basic menu or a different underlying menu. Inspired by this work, we adapted and extended this concept to introduce secondary multifunctional strips—conceptually underneath the primary multifunctional strips—that can be accessed by applying firm pressure, as illustrated in Figure 3. In our system, the user is not restricted only to buttons, but may pop through any of the four previously mentioned user interface elements, effectively doubling the number of strips. Unlike the physical pop-through button interface, our interface lacks the tactile feedback of the mechanical pop-through action. However, this can be compensated for, in part, by visual changes to on-screen elements or audio cues.

Dual-Finger Interaction

Even though the device reports only a single point of contact, it is possible to determine when two neighboring fingers make contact in two adjacent strips at approximately the same vertical position. In such cases, a higher pressure reading results and the position of the centroid of the two contacting fingers is reported. Newer versions of the TouchPad are also able to set a flag in the data stream, indicating that the width of the contact area is wide and, with high probability, that two fingers are contacting the pad [7]. To determine whether a dual-finger event has occurred, we can use either this flag bit or a preset pressure threshold, in addition to monitoring whether the horizontal position reported falls on the logical border of two adjacent

strips. These methods identify dual finger motions quite reliably.

A pair of adjacent fingers can together operate a single dual-motion strip, as shown in Figure 4; therefore, the four single-finger strips may be complemented by the three dual-motion strips to increase the number of one-degree-of-freedom input devices to seven. This may be doubled by the pop-through technique to increase the number of primary and secondary multifunctional strips to fourteen.

USER EXPERIENCES

During development we asked coworkers and visitors to our research lab to try out the prototype multi-strip device and the interaction methods described in informal preliminary tests. Most users very quickly picked up the concepts of multi-finger interaction, pop-through and dual-motion functionality. After a few minutes of experimentation they were able to exert the proper amount of pressure to switch between the primary and secondary strips by popping-through, with insignificant number of accidental switches. There are only two provisions that need to be made for pop-through actions. In case the primary strip implements virtual buttons or a virtual slider, popping-through to the secondary strip does not only switch the functionality of the strip but also invalidates any changes that were made when the finger initially contacted the strip, when the strip was in its primary state. Also, after the user pops-through, the strip remains in its secondary state— independent of exerted pressure—until the finger is lifted off the strip completely.

All users could easily execute dual-motions with a combination of their index&middle and middle&ring fingers, however due to the unergonomic shape of our prototype the ring&pinky finger combination was awkward for most, since the tip of the ring and the pinky finger had to be held side by side. We believe that this awkwardness can be overcome with a custom manufactured, more ergonomic prototype. We note that the concepts introduced may be used with less than 4 strips. For example, in the case when only the index and the middle finger is used with an even more compact 2 strip device, the introduced techniques still allow for the direct control of 6 parameters with arbitrarily configuration of interaction metaphors.

CONCLUSIONS

We have presented multiple novel uses for a pressure-sensitive pad, introducing multifunctional strips and dual-motion strips, and multiple methods of adjusting parameters, and activating commands. The 4 strip prototype, created from a commodity touchpad, without electrical modifications, allows for the direct control of up to 14 widgets with one hand. It doesn't require homing of the hand or fingers, nor the visual observation of the device or the display to move between widgets. The strip-to-widget mapping may be dynamically reconfigured without

the need to reposition fingers or to physically modify the device.

Since the complexity of the interface is implemented exclusively in software with the input device there is no need to show visual widgets in the GUI and no time consuming cursor navigation is required. Little or—for advanced users—no screen space is needed for interaction elements. These features make the compact input device very advantageous for systems where the input device needs to be worn/held such as virtual- and augmented reality systems and where displays are small like wearable computer systems.

ACKNOWLEDGMENTS

This research was funded in part by Office of Naval Research Contracts N00014-99-1-0249 and N00014-99-1-0394, NSF Grant IIS-00-82961 and IIS-01-21239, and gifts from Intel and Microsoft Research. We would like to thank Synaptics for donating to us pressure-sensitive TouchPads and Alias Inc. for donating 3D modeling and animation tools that we used in the production of the accompanying video material.

REFERENCES

1. B. Arons: SpeechSkimmer: Interactively Skimming Recorded Speech. Proc. UIST'93, 3-5 November, 1993, 187-196
2. R. Balakrishnan and P. Patel. The PadMouse: Facilitating selection and spatial positioning for the non-dominant hand. Proc. ACM CHI 1998 Conf. on Human Factors in Comp. Sys., 9-16.
3. G. Blaskó and S. Feiner. A Menu Interface for Wearable Computing Proc. 6th Int. Symp. on Wearable Comp. 2002, Seattle, WA, 7-10 October, 2002, 164-165.
4. G. Blaskó and S. Feiner. An Extended Menu Navigation Interface Using Multiple Pressure-Sensitive Strips. Proc. 7th Int. Symp. on Wearable Comp. 2003, White Plains, NY, 21-23 October, 2003, 128-129.
5. W. Buxton, R. Hill, and P. Rowley. Issues and techniques in touch-sensitive tablet input, ACM Computer Graphics, 19(3), (Proc. SIGGRAPH 85), 215-223.
6. G. Kurtenbach.. The design and evaluation of marking menus. Ph.D. Thesis. Dept. of Comp. Sci., University of Toronto, Toronto, Canada. 1993.
7. Synaptics TouchPad C++ API document and Interfacing Guide. <http://www.synaptics.com/>
8. R. Zeleznik, T. Miller, and A. Forsberg. Pop through Mouse Button Interactions. Proc. ACM Symp. On User Interfaces and Tech. (UIST '01), November, 2001, Orlando, FL, 195-196.
9. R. Zeleznik, J. LaViola, F. Acevedo, and D. Keefe. Pop through button devices for VE navigation and interaction. Proc. IEEE Virtual Reality 2002, 2002, 127-134.