DisplayCover: A Tablet Keyboard with an Embedded Thin-Film Touchscreen Display

Antonio Gomes
Human Media Lab
Queen's University
Kingston, ON, Canada
gomes@cs.queensu.ca

Tristan Trutna
Applied Sciences Group
Microsoft Corporation
Redmond, WA, USA
trist@microsoft.com

Roel Vertegaal
Human Media Lab
Queen's University
Kingston, ON, Canada
roel@cs.queensu.ca

ABSTRACT
Tablet computers aim to bridge the gap between portability and productivity, reducing the need for users to carry multiple devices. However, despite increases in resolution, their displays are limited in size. This commonly results in sequential rather than parallel options for screen navigation, a significant drawback when multitasking. In this paper, we present DisplayCover, a tablet cover that integrates a physical keyboard as well as a touch and stylus sensitive thin-film e-ink display. We developed example applications to demonstrate the ability to dynamically alter the cover display content based on usage context, as well as concurrent access to multiple applications, stylus annotation, gestures and trackpad interactions.

Author Keywords
Input Devices; Keyboard; Context-Aware; Secondary Display; Stylus Annotation; Peripheral Hardware; Tablet.

ACM Classification Keywords
H.5.2. [Information interfaces and Presentation]: User Interfaces – Input devices and strategies.

INTRODUCTION
The tablet computer has been a deep-rooted vision, with a number of abstract devices resembling current tablet technology depicted over the past 60 years [29]. In 1972, Alan Kay envisioned the Dynabook [16], a conceptual portable educational device for children of all ages that offered similar functionality to that of today's tablets. In addition to research explorations, several companies proposed commercial products in the following decades. However, tablet devices initially failed to gain popularity due to their bulky form factor, lack of accurate (multi)touch support, as well as their limited access to applications created for desktop computers.

MobileHCI '15, August 25 - 28, 2015, Copenhagen, Denmark
© 2015 ACM. ISBN 978-1-4503-3652-9/15/08…$15.00
DOI: http://dx.doi.org/10.1145/2785830.2785843

Figure 1. DisplayCover Prototype showing Windows Live Tiles in the secondary display

Recently, tablets have experienced a resurgence of interest as miniaturization and increases in display resolution enabled manufacturers to produce lighter and more powerful devices with precise multitouch displays. While their use has been popularized for email, reading, playing games, and social networking, productivity tasks are still not easily achievable on tablet devices [20]. For portability, tablet computers have displays that are limited in size. This has resulted in multiple documents or apps to typically be accessed in sequence rather than in parallel. Moreover, the absence of a physical keyboard poses significant challenges for extensive text entry [8,17].

Consequently, significant research and commercial development has gone into creating accessories that recapture many of the affordances associated with laptop or desktop computers. While physical keyboards, capacitive styli, and power covers have become commonplace, tablet screen size remains a limitation [24]. Support for external monitors cannot be regarded as a portable solution.

In this paper, we present DisplayCover (Figure 1), a tablet cover that combines both a physical keyboard and a thin-film multitouch display as a means to extend screen and interaction real estate. Our design allows applications to dynamically alter the cover display content based on usage context within a set of user application tasks, as well as stylus annotation, concurrent access to multiple applications, gestures and trackpad interactions.

RELATED WORK

We will first examine work related to *physical* and *display* keyboards. We will then discuss explorations of *multi-display* systems. Finally, we consider interaction techniques that inspired our design.

Physical Keyboards

Touch is now the most common interaction technique for tablet computers [12]. However, typing on flat glass does not perform well in comparison to physical keyboards [8], which remains the preferred input to perform tasks that require extensive typing.

Physical keyboards are offered in a variety of configurations, whether as standalone solutions or embedded in protective covers. With the exception of the Microsoft Type and Touch Covers [19], most keyboard accessories do not feature an embedded trackpad, requiring users to home their hands between the keyboard and the slate device touchscreen. To overcome this limitation, recent explorations aimed at augmenting physical keyboards with some of the interaction metaphors present in multitouch surfaces.

Touch&Type [7] combined a conventional keyboard with an extended touchpad where the touch area is formed by the surface of the keys themselves. Gu et al. [11] discussed an elongated touchpad that utilized the entire area below the keyboard of a laptop computer. Taylor et al. [28] proposed a mechanical keyboard capable of sensing rich and expressive motion gestures performed both on and directly above the device. GestKeyboard [32] demonstrated a novel technique for gesturing over ordinary, unmodified physical keyboards. These devices allow lightweight gestures to be performed without users having to move their hands away from the keyboard.

Display Keyboards

A series of adaptive display keyboards have been proposed by researchers and hardware manufacturers. Optimus Maximus [21] featured a keyboard with small programmable displays underneath each individual key. Microsoft Adaptive Keyboard [18] featured a large, touch-sensitive display atop a keyboard that continued underneath the keys. Observations suggested that extending applications toolbars or command icons to the adaptive keyboard increased productivity. Block et al. [5] demonstrated a touch-display keyboard with dynamic display and touch-sensing in each key. Their approach transformed the keyboard into an interactive surface, suggesting benefits for direct manipulation techniques that have previously been confined to adaptation of screen-based interface elements.

Multi-Display Systems

There is also a body of work featuring reconfigurable multi-display systems. Chen et al. [6] discussed an e-book reader featuring two displays mounted on two separate slates that can be used in side-by-side or detached configurations. Hinckley et al. [13] proposed a dual-screen tablet computer that could be oriented in a variety of postures. Their findings identified several benefits for multi-display systems (i.e. Division of Tasks between Screens; Quick Access to Toolbars and Command Icons; Lightweight Navigation). More recently, PaperFold [10] demonstrated a multi-segmented mobile device for triggering viewport transformations in its graphical interface.

Other projects explored pairing devices of distinct for factors to create distributed information displays. Bonfire [15] combined a laptop computer with two laptop-mounted micro-projectors that operated as a self-contained mobile computing system. DynamicDuo [23] explored the design space of distributed IO solutions that rely on and benefit from phone–tablet collaboration, both physically and digitally.

While the above explorations extended the desktop experience to peripheral hardware, there has been little to no research targeting devices whose limited screen size is a concern (e.g. tablet computers). Moreover, to our knowledge, there has been no exploration featuring an interactive display that spans the entire length atop a tablet keyboard accessory.

Contribution

Our approach explores the ability to dynamically alter the peripheral display content based on usage context, while extending the user experience and interaction model to the horizontal plane, where hands naturally rest. While context-aware auxiliary displays [18,21,25] and peripheral sensors [26] have been largely explored, DisplayCover is the first system that seamlessly combines a precise multitouch sensor and a high-resolution display in a peripheral accessory that closely resembles the form factor and portability of keyboard-covers available to consumers. Moreover, to our knowledge, stylus annotation on the horizontal plane is unique to our system, thus reducing the need for users to home their hands between the slate display and the physical keyboard.

While further research is necessary to ascertain the effectiveness of these changes in interaction patterns we believe this to be a promising approach to mimic paper interaction metaphors in digital devices. Finally, while previous explorations allowed users to display small icons on augmented keyboard keys [21,25], they required extensive user configuration and only mirrored command icons or application toolbars present on screen. Our approach dynamically places these elements on the secondary display without explicit input, while providing users with additional screen real estate in the slate device.

IMPLEMENTATION

DisplayCover is a peripheral cover designed for compact touch-enabled laptops. While our prototype is currently compatible with Microsoft Surface Pro and Pro 2 devices [19], our design guidelines are in principle applicable to

Figure 2. Example Application Scenarios: 1. Secondary Display Taskbar; 2. Dynamic UI Manipulation; 3. File Navigation and Manipulation; 4. Support for Concurrent Applications (e.g. Video call + Email Client)

other devices. A tactile keyboard affords users with the comfort and ease of use provided by physical keys. A thin-film Plastic Logic e-ink display [22] with a resolution of 1280 x 305 pixels extends the available screen real estate of the slate device by up to 8% (based on a Microsoft Surface Pro 2 with a 10.6", 1080p, 208ppi display). We chose this display due to the bistable nature of electrophoretic ink, reducing the secondary screen's impact on battery life. Additionally, the thinness and light weight of these displays could enable manufacturers to embed them in tablet covers while preserving portability. Our display is controlled by a Freescale driver board that is connected to the Microsoft Surface Pro 2, which handles applications and interface graphics. We borrow from the findings of Holman et al [14] to render interactive content at *5fps*, a suitable result for displaying toolbars, command icons or browsing folders. A capacitive touch sensor connected to an Atmel maXTouch controller [2] offers support for both multitouch and stylus input. DisplayCover can be magnetically connected to the slate device, following the existing design of the Microsoft Type and Touch Covers [19].

Form Factor

Our choice for placing the secondary screen directly under the primary display was two-fold. Firstly, hands naturally rest on the bottom of the cover; placing the display beneath the keyboard would occlude it or force users to change their pose during interaction. More importantly, we see the secondary display as a natural extension of the primary screen, allowing simultaneous access to information on both displays. While a small subset of tablet devices supports tiling windows as a means to concurrently access multiple documents or applications (i.e. each of two windows occupying half of the screen), this feature significantly decreases the available screen real estate for each individual window. Our approach mitigates this drawback by dynamically placing content on the secondary display and cannot be regarded as similar. Finally, significant effort was put into preserving the overall thinness of the device. While the Microsoft Type Cover is 6mm thick, our prototype is currently 7.5mm. We believe we could further approximate the thinness of commercially available covers with specialized manufacturing equipment.

INTERACTION TECHNIQUES

To illustrate the potential and immediate feasibility of our approach, we highlight a series of application scenarios to showcase interaction techniques and features enabled by DisplayCover (Figures 2-5). We apply existing interaction techniques [5,11,13,18] in a novel form factor, aimed at increasing productivity in tablet devices.

Context-Aware Display

While previous explorations required extensive user configuration [5,18,21], DisplayCover leverages ribbon frameworks and floating panels to dynamically place toolbars and command icons in the cover display (Figure 2.1). We have successfully integrated this feature in both Adobe Photoshop and Microsoft Word. Opening Adobe Photoshop will cause the active toolbars to automatically appear in the peripheral display (Figure 2.2). Similarly, Microsoft Word ribbons are displayed in the peripheral device, directly above the keyboard. Placing toolbars on the secondary display effectively extends the available screen real estate of the slate device. This feature could be readily implemented in the other Microsoft Office and Adobe applications.

Concurrent Access to Multiple Applications

Our prototype enables users to access documents or applications in parallel, thus providing better support for multitasking. To highlight multi-view operations, we developed a photo gallery application in which users can navigate thumbnails on the cover display. Tapping a thumbnail will maximize the corresponding image to the tablet screen. Furthermore, while composing an email, users can attach files by simply pressing and holding the corresponding thumbnail (Figure 2.3). To demonstrate concurrent access to multiple applications, we created an email client that operates entirely within the secondary display. Users can receive and respond to emails directly from the peripheral accessory (Figure 2.4), reducing the need to detract focus from a primary application on the slate device (e.g. respond to emails while on a video call). Other application scenarios could include a notification center, which would lessen the reliance on tablet screen real estate for notifications, while providing the user with immediate access to important information.

Figure 3. Gesture Interactions: 1. Zoom; 2. Pan; 2. Rotate

Gestures

When using a tablet device alongside a keyboard cover, users often need to home their hands between the slate display and the physical keys. Additionally, using the touch input on the large display generally leads to the occlusion of graphical objects of interest [3,30]. Our approach mitigates these limitations by allowing users to perform touch input in the horizontal plane, following the trackpad interaction model [8,11,31]. Furthermore, occlusion-free interaction is attainable by performing gestures using our multitouch sensor. We included support for both unimanual and bimanual gestures. Example interaction techniques include *zoom*, *pan*, and *rotate*. Dragging two fingers close together or apart from each other will respectively zoom in or out of a website, map, or picture (Figure 3.1). Panning is achievable by pressing and holding two fingers in close proximity while moving them around the sensor area (Figure 3.2). Lastly, pressing and holding one finger while moving the other around the surface of the sensor allows users to rotate objects (Figure 3.3).

Stylus Annotation

Many tablet devices support free-form inking, mimicking affordances associated with paper documents. However, the writing experience can be suboptimal when tablets are at a vertical angle, i.e., while reading. To mitigate the need for reorienting the device, we implemented handwriting input on the secondary display (Figure 4). Our approach leverages eyes-free annotation [4], reducing the need to produce spatially accurate writing. The handwriting input is filtered and displayed in the slate device, eliminating occlusion issues associated with pen computing [30].

Figure 4. Stylus Annotation on the horizontal plane

Trackpad

DisplayCover also implements a trackpad functionality, inspired by the findings of Gu et al. [11]. However, rather than placing the keyboard above the touchpad, we positioned the secondary input area directly underneath the primary display (Figure 5). Our touch sensor uses a relative mapping to create relative direct interaction supporting both touch and stylus input.

Figure 5. DisplayCover Trackpad Implementation

Limitation and Future Directions

While we believe DisplayCover to be a promising approach towards increasing productivity in compact devices, there are drawbacks that need to be addressed in future iterations. Significant effort was put into mimicking the form factor of existing keyboard covers. However, our secondary display is currently controlled by external driver boards. A more tightly integrated solution would be feasible if integrated during manufacturing. Moreover, our display is black and white, which prevented us from exploring application features relying on color graphics (i.e. in a color picker). In the next design revision we will include a thin-film full color display.

While the current version of our prototype features mechanical keys, we expect future iterations of our prototype to feature a soft keyboard with robust palm rejection mechanisms, allowing users to comfortable rest their hands on the keyboard. Moreover, we will develop an alternative form factor featuring the screen below the keyboard. We will then conduct a comparative study between the two configurations. While we are aware hardware manufacturers have shipped devices following our proposed configuration [1] and later changing their design following negative user feedback, we do believe having the secondary display as a natural extension of the slate device offers a subset of interaction patterns worth exploring. However, we recognize further studies are necessary to support this claim.

Our context-aware feature is limited by existing software solutions. Dynamically altering applications that do not feature floating panels or ribbon frameworks would require extensive changes to their core architecture. We plan to develop support for a larger subset of applications in future iterations of our prototype.

Finally, our hypothesis that DisplayCover is a viable solution to increase productivity in compact devices is yet to be verified. While informal user feedback suggested benefits over conventional keyboard accessories (i.e., stylus annotation on the horizontal plane; better support for multitasking; and occlusion-free interaction), we believe a qualitative study should provide definite data to verify our hypothesis.

CONCLUSION

In this work, we present a new tablet cover combining a physical keyboard and an electrophoretic display that supports dynamic UI manipulation, concurrent access to multiple applications, stylus annotation, and trackpad interactions on the horizontal plane. To illustrate these features, we demonstrated its use in a suite of example applications built around the proposed interaction techniques. DisplayCover extends the available screen real estate of tablet computers while mitigating occlusion issues associated with direct pen and touch input.

REFERENCES

1. Acer Aspire R Series www.engadget.com/2014/09/03/acer-aspire-r13-r14-switch-11/
2. Atmel maXTouch www.atmel.com/microsite/maxtouch-t-series
3. Benko, H., et al. Precise selection techniques for multi-touch screens. In *Proc. CHI '06,* 1263-1272.
4. Bharath A., Sriganesh M. Recognition of Eyes-free Handwriting Input for Pen and Touch Interfaces. Hewlett-Packard Labs, 2008.
5. Block, F., Gellersen, H., Villar, N. Touch-display keyboards. In *Proc. CHI '10*, 1145-1154.
6. Chen, N., et al. Navigation Techniques for Dual-Display E-Book Readers. In *Proc. CHI '08*, 1779-1788.
7. Fallot-Burghardt, et al. Touch&Type. In *Proc. NordiCHI '06*, 465–468.
8. Findlater, L., Wobbrock, J. O., Wigdor, D. Typing on flat glass. In *Proc. CHI '11*, 2453-2462.
9. Foucault, C., Micaux, M., Bonnet, D., Beaudouin-Lafon, M. SPad. In *Proc. CHI EA '14*, 1879–1884.
10. Gomes, A., Vertegaal, R. PaperFold. In *Proc. TEI '15*, 153–160.
11. Gu, J., et al. LongPad. In *Proc. CHI '13*, 1421-1430.
12. Hinckley, K., Sinclair, M. Touch-sensing input devices. In *Proc. CHI '99,* 223–230.
13. Hinckley, K., et al. Codex: a dual screen tablet computer. In *Proc. CHI '99,* 1933–1942.
14. Holman, D., et al. Flexkit. In *Proc. UIST '13 Adjunct*, 17–18.
15. Kane, S., et al. Bonfire: A Nomadic System for Hybrid Laptop-Tabletop Interaction. In *Proc. UIST '09*, 129-138.
16. Kay, A. C. A Personal Computer for Children of All Ages, 1972.
17. Kim, S., Son, J., Lee, G., Kim, H., Lee, W. TapBoard. In *Proc. CHI '13*, 553-562.
18. Microsoft Adaptive Keyboard www.microsoft.com/appliedsciences/projects/uist.aspx
19. Microsoft Surface http://www.microsoft.com/surface
20. Müller, H., Gove, J., Webb, J. Understanding tablet use. In *Proc. MobileHCI '12*, 1-10.
21. Optimus Maximus Keyboard www.artlebedev.com/everything/optimus/maximus
22. Plastic Logic Inc. (2014) - http://www.plasticlogic.com
23. Piazza, T., et al. Holy Smartphones and Tablets, Batman! Mobile Interaction's Dynamic Duo. In *Proc. APCHI '13*, 63-72.
24. Raptis, D., Tselios, N., Kjeldskov, J., Skov, M. B. Does size matter? In *Proc. MobileHCI '13*, 127-136.
25. Razer Blade Pro www.razerzone.com/gaming-systems/razer-blade-pro
26. Rendl, C., et al. FlexSense: a transparent self-sensing deformable surface. In Proc. UIST '14, 129-138.
27. Rick, J. Performance optimizations of virtual keyboards for stroke-based text entry on a touch-based tabletop. In *Proc. UIST '10,* 77-86.
28. Taylor, S., et al. Type-hover-swipe in 96 bytes. In *Proc. CHI '14,* 1695–1704.
29. Vannevar Bush. As We May Think, The Atlantic Monthly, 176(1), July 1945, 101-108.
30. Vogel, D., Casiez, G. Hand occlusion on a multi-touch tabletop. In *Proc. CHI '12*, 2307-2316.
31. Yee, K., Two-handed Interaction on a Tablet Display. In *Proc. CHI EA '04*, 1493-1496.
32. Zhang, H., Li, Y. GestKeyboard. *Proc. CHI '14,* 1675–1684.