

Towards Making Graphical User Interface Palettes Tangible

Martin Spindler, Michel Hauschild, Raimund Dachzelt
User Interface & Software Engineering Group
Otto-von-Guericke-University Magdeburg, Germany
{spindler, dachzelt}@ovgu.de, michel.hauschild@st.ovgu.de

ABSTRACT

In this work, we present techniques that aim at physically detaching the graphical user interface (GUI) from digital content that is displayed on a tabletop. For this purpose, we extended a pen-enabled tabletop with several spatially aware (6DOF) paper-like mobile displays. In contrast to previous work, we provide a fully dynamic solution. This means the GUI is not printed onto paper displays, but instead is dynamically projected by a ceiling-mounted projector. Based on this technical foundation, we developed a prototypic graphics editor that we use as a basis for several novel interaction techniques that particularly utilize the height above the tabletop for accomplishing common tasks typical for the domain.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors.

Keywords: Multi-surface UI, Tangible UI, Digital Paper.

INTRODUCTION

Today's graphics editing programs, such as Adobe® Photoshop, Adobe® Illustrator, or GIMP, are often crammed with an enormous set of features that is typically organized in graphical user interface (GUI) tool palettes, i.e., separate windows which overlay all regular windows. These palettes usually collocate similar aspects of functionality in a single place in order to make it easier for users to find and choose an appropriate tool.

Due to the wide spectrum of available features – e.g., layer handling, system control, selection management, zooming and panning, brush and color selection, slicing, transformation, etc. – a variety of tool palettes need to be displayed on the screen simultaneously. Therefore, valuable display space is often cluttered with UI-elements. In order to cope with the limited display resources, users have to spend too much time with organizing and rearranging UI-palettes, instead of just focusing on their actual task at hand. Obviously, one solution to this problem is to provide more room for GUIs and content by using larger (interactive) displays, such as tabletops. However, the main conflict still

remains: the strife between digital content and user interface over the very same display space. In addition, the larger the tabletop, the more difficult it is to reach out for desired tools.

RELATED WORK

To tackle these problems, various approaches have been developed that often aim at mimicking the traditional workflow of graphic artists. Examples are using different input modalities for different tasks such as finger/hand gestures for panning/zooming and digital pens for drawing [2], fading in and out occlusion-aware menus directly at the place of interaction [3], combining the physical and digital world into a single information space so users can benefit from the best of both domains [5], and to physically separate GUIs from content by introducing tangible paper-like ink-based (and thus essentially static) palettes that serve as physical proxies for the user interface (see the Anoto [1] – based VoodooSketch project [4]). Another interesting concept, although not related to the domain of graphics editing, is to exploit the height of spatially aware paper-like displays above the tabletop for the sake of interaction as described in the PaperLens project [6].

CONTRIBUTIONS

The aim of our work is to go a step further than the VoodooSketch [4] project. This is achieved by providing tangible paper-like interface palettes that on the one hand can change their image content *dynamically* and on the other hand are fully *spatially-aware*, i.e., they precisely know about their position and orientation in 3D space. Our contributions are: (I) addressing the problem of arranging GUI elements by transferring them onto paper-like *tangible GUI palettes*, (II) adjusting the complexity of a GUI depending on the available screen space by using *foldable GUI palettes*, (III) providing an intuitive way for browsing file content by interpreting the spatial location of the *tangible browser*, and (IV) navigating through local (editable) image details by detaching portions of the tabletop's main document to personal *tangible local views* and using their height above the tabletop for zooming/panning interaction.

TECHNICAL SETUP

We follow the technical approach of PaperLens [6] that enables us to project arbitrary (distortion-corrected) image content onto paper-like mobile displays by using a ceiling-mounted projector and an optical infrared (IR) marker-based tracking. We enriched this technical setup with Anoto [1] technology that allows us to use digital pens on both mobile lenses and the tabletop. In addition to that, we at-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ITS'10, November 7–10, 2010, Saarbrücken, Germany.

Copyright 2009 ACM 978-1-60558-745-5/09/10...\$10.00.



(a) TUIPs make interface palettes tangible and thus allow for physically separating UI and content. (b) By unfolding, the size of a tangible palette can be extended to show further GUI elements. (c) By lifting/lowering a tangible browser, users can easily browse through large image collections. (d) Tangible local views are a tool to zoom/pan through local (editable) image details.

Figure 1: Several examples of tangible user interface palettes (TUIP) in a pen operated tabletop environment. The paper-like mobile displays show arbitrary GUI content that is dynamically projected onto them by a ceiling mounted projector. Users interact with the system by either using digital pens directly on tabletop/mobile displays or by using the spatial location of the mobile displays.

tached *marker buttons* (self-made tactile IR-reflective buttons) onto some of the paper displays in order to let users explicitly switch the interaction mode between the states “*spatially aware*” and “*frozen*”. In contrast to PaperLens, the default state of tangible palettes is always “*frozen*”, i.e., moving a palette is not used as input.

TANGIBLE GUI PALETTES

With tangible GUI palettes, we extend the display space of a tabletop by additional paper-like displays. By allowing users to freely transfer arbitrary GUI elements between a tabletop and mobile displays, GUI and application content are physically separated (see Figure 1a). Thus, arranging the user interface is reduced to the problem of moving pieces of cardboard through the physical space above and around the tabletop. Hereby, both tangible displays and the tabletop respond to digital pen input in the same way.

FOLDABLE GUI PALETTES

By allowing users to fold/unfold tangible palettes, additional screen space is provided that can be used to display more GUI elements, e.g., a more sophisticated color selection tool, if required (see Figure 1b).

TANGIBLE BROWSER

With the tangible browser, we demonstrate how a collection of symbols, textures, icons, etc. can be explored by using the palette’s physical position. Similar to PaperLens [6], we span a zoomable space with all image files visible at a low height (see Figure 1c/left) and a zoomed-in view being displayed for a higher height (see Figure 1c/right). At the same time, panning is performed by horizontal translations of the palette. In contrast to PaperLens, users have to explicitly activate the “*spatially aware*” interaction mode by holding a marker button. Hereby, the interaction is always *relative* to the location, where the marker button was originally pressed. As soon as the button is released, the tangible browser will be “*frozen*” again, i.e., palette movements are not interpreted as input anymore.

TANGIBLE LOCAL VIEWS

A common task in the graphics editing domain is to view/edit small details while still maintaining the overall (global) view. Tangible local views tackle exactly this problem (see

Figure 1d). Here, users can link a portion of the main image document (tabletop) to their own personal TUIP. Zoom and pan is then performed similar to the tangible browser, i.e., by interpreting the display’s physical position in a *relative* manner. To avoid unintentional zoom/pan, users need to explicitly activate the “*spatially aware*” mode by holding a marker button. Editing/drawing is performed by pen input (and likely by using other palettes to select pen attributes such as color, width, etc.).

CONCLUSION AND FUTURE WORK

Currently, we are in the first cycle of an iterative design process and thus our system is work in progress. From an initial user evaluation, we received very positive feedback. For the next cycle we plan to incorporate some of the user suggestions and to extend the functionality.

REFERENCES

1. Anoto Group AB - Website. <http://www.anoto.com>
2. P. Brandl, M. Haller, M. Hurnaus, V. Lugmayr, J. Oberngruber, C. Oster, C. Schafleitner, and M. Billinghurst: An Adaptable Rear-Projection Screen Using Digital Pens and Hand Gestures. In *Proc. of ICAT '07*, 2007, pp. 49–54.
3. P. Brandl, J. Leitner, T. Seifried, M. Haller, B. Doray, and P. To: Occlusion-aware Menu Design for Digital Tabletops, In *Proc. of 'CHI 09 (Extended Abstracts)*, 2009, pp. 3223-3228.
4. F. Block, M. Haller, H. Gellersen, C. Gutwin, and M. Billinghurst: VoodooSketch: Extending Interactive Surfaces with Adaptable Interface Palettes. In *Proc. of TEI '08*, 2008, pp. 55–58.
5. M. Haller, P. Brandl, D. Leithinger, J. Leitner, T. Seifried, and M. Billinghurst: Shared Design Space: Sketching Ideas Using Digital Pens and a Large Augmented Tabletop Setup. In *Proc. of ICAT '06*, 2006, pp. 948–959.
6. M. Spindler, S. Stellmach, and R. Dachsel. PaperLens: Advanced Magic Lens Interaction Above the Tabletop. In *Proc. of ITS'09*, 2009, pp. 77–84.