Gesture-Supported Document Creation on Pen and Touch Tabletops

Fabrice Matulic[1]
matulic@inf.ethz.ch

Moira C. Norrie[1]
norrie@inf.ethz.ch

Ihab Al Kabary[2]
ihab.alkabary@unibas.ch

Heiko Schuldt[2]
heiko.schuldt@unibas.ch

[1]ETH Zurich
8092 Zurich, Switzerland

[2]University of Basel,
4056 Basel, Switzerland

CHI'13, April 27 – May 2, 2013, Paris, France.
ACM 978-1-XXXX-XXXX-X/XX/XX.

Abstract

We present an ongoing effort to design and implement a prototype system for pen and touch-operated digital workdesks supporting the activity of document creation. Our application exploits bimanual gestures to perform common document editing operations including element manipulations, text input, clipart retrieval and insertion in a mostly direct way. For many of our gestures, we rely on pen-mode switching actions triggered by postures of the non-dominant hand, which allows us to provide a largely widget-free yet efficient user interface.

Author Keywords

Pen and touch interaction; digital tabletops; document engineering

ACM Classification Keywords

H.5.2.

General Terms

Design, Human Factors

Introduction

Interactive surfaces with simultaneous pen and touch input capabilities have recently emerged as effective digital platforms that provide real added value compared to unimodal multitouch. Other than allowing users to smoothly interact with content using the appropriate medium (e.g. pen for drawing, touch for manipulating), the simultaneous availability of pen and touch affords richer interaction possibilities via powerful bimodal synergies [2, 4, 6, 7, 9]. Within the context of the ongoing quest towards the office of the future, digital workdesks that can be operated with a stylus and multitouch seem like natural candidates for the execution of document-based office work. Yet surprisingly, there are still only very few such productivity applications specifically designed for interactive tabletops, let alone for pen and touch systems. In particular, the essential task of document composition has not received much attention from HCI researchers and engineers.

The present work attempts to partially fill this relative void by providing intuitive and efficient tools to support document creation and editing on digital tabletops. We are creating a prototype application with a UI that follows a loose virtual desk metaphor. This UI is supported by a rich gesture set making use of combined pen and touch input, in particular pen-mode switching activated by postures of the non-dominant hand (i.e. the hand not holding the pen), which we call modifier postures in analogy with modifier keys on the keyboard. We describe these tools and gestures after a brief review of relevant prior work.

Background

The theory behind pen and touch interaction is based on Guiard's seminal work, which introduced the kinematic chain model for asymmetric bimanual interaction [5, 8]. According to this model, the roles of the two hands are such that the non-dominant hand (NDH) sets the frame of reference in which the dominant hand (DH) operates. In our pen and touch context, this generally translates into a more or less strict division of labour, as studied by several authors [2, 7-9]. In this role distribution, the pen executes fine-precision actions such as drawing and handwriting, touch performs coarser manipulations such as panning, zooming, tapping etc. and combined pen and touch provides "new tools" [7]. Those new tools can materialise in several ways. For instance, the NDH can be used to constrain or set the parameters of pen tracing [2] or to activate a particular function, whose expression is articulated by the stylus [8] etc.

A common characteristic of those bimanual actions is that they do not require simultaneous motions of the two hands. In most cases, and as per the kinematic chain model, the NDH fixes the operational context and the DH moves in that context producing a particular response, e.g. a finger of the NDH pins an object and the pen is used to drag off a copy of it [4, 7]. This suggests that NDH postures can be an effective technique to enhance the vocabulary of pen interactions [8].

On the document creation front, as has been mentioned above, tabletop applications developed so far have mostly limited themselves to supporting annotating and basic editing functions [9]. Furthermore, we observe that latest consumer office products (Microsoft Office at the forefront), while showing some level of adaptation to touchscreen devices, are still very apparently rooted in their WIMP[1] legacy [3].

Design Approach

We believe a tabletop affords an environment naturally conducive to document-centric work. In this context, a pen and touch-operated interactive surface seems to

1 http://en.wikipedia.org/wiki/WIMP_%28computing%29

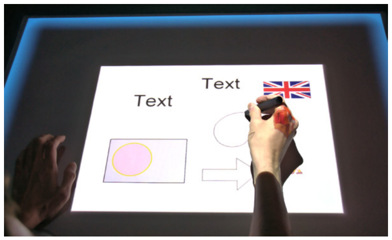

Figure 1: The main interface of our system, running on a DiamondTouch augmented with an Anoto overlay.

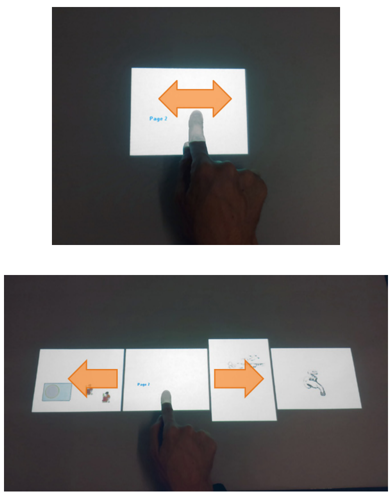

Figure 2: One-finger shake gesture to expand and collapse document pages

lend itself to the realisation of the desktop metaphor, even more so than a regular PC. Our design approach therefore considers a virtual work desk on which documents are placed and manipulated by users using appropriate multitouch and stylus-based interactions. As target media, we consider page-based documents consisting mainly of assembled elements, such as presentations, flyers etc. Intuitively, we feel the latter types of documents stand much more to gain from a pen and touch platform compared to, say, continuous, text-intensive documents such as reports and theses.

One of our main design goals is to try to construct a mostly clutter-free interface, i.e. largely devoid of widgets and other UI artefacts, but without compromising essential functionality. Depending on the breadth and nature of said functionality this is more or less achievable ([11] is a good example showing how this can be accomplished for equation writing). For instance, colour selection is hardly possible without a helper tool such as a palette. Another challenge is to fulfil the requirement of producing digital quality documents using close-to-natural pen interaction, that is, to be as near to a pen on paper experience as possible, but with resulting documents properly formatted and professional-looking. This discrepancy is especially difficult to bridge for text input, as trying to smoothly convert handwritten text into a coherent and meaningful ASCII rendition is not trivial. We explain how we address those various challenges further below.

Leveraging the pen and touch input paradigm
Many basic operations can be carried out through tried-and-true interactions governed by the appropriate modality. These include common manipulations such as panning, zooming, rotating etc. with multitouch and inking with the pen. We will present a few other unimodal gestures that we utilise in our application later on, but first we would like to introduce the concept of modifier postures as a functional equivalent to modifier keys on a keyboard.

Similar to modifier keys such as Control or Alt that, in combination with other regular keys or a pointing device, can be activated to perform a variety of actions, we see a potential for the NDH to be used in an analogous way in order to multiply the expressional vocabulary of the pen-holding DH. In the latter context, modification can be achieved by performing distinct postures with the NDH, such as placing one or more fingers on the surface to activate a particular mode. To some extent, this has already been exploited in prior work [6, 8], but we believe this ability can be further expanded, for instance by combining it with targeted gestures of the DH (e.g. three NDH fingers + a C-shaped tracing gesture of the pen triggers a screen capture). Moreover, if we carefully design our postures so that they do not interfere with other regular unimodal gestures, we need not require that these postures be performed in a specific area of the interactive surface (as in [6] for example). This not only saves valuable screen space, but allows for more flexibility, as users are not constrained in their positions and movements around the table.

Application
We now describe our prototype application, starting with basic page and element editing functions, followed by operations performed using modifier postures.

The workspace shows all open documents, by default a single page of each (Figure 1). Common drag,

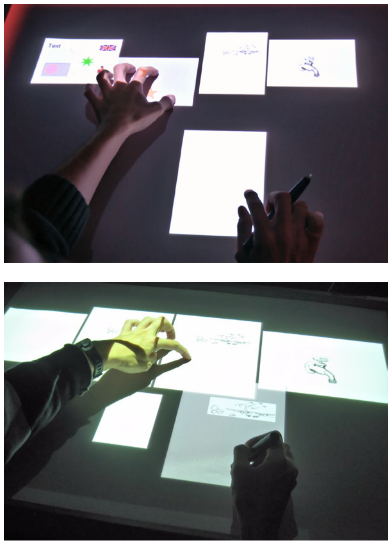

Figure 3: Three fingers on a page trigger a move operation (top) whereas a finger pin + pen drag causes a copy of the page to be created (bottom)

Figure 4: In situ text insertion with handwritten content. The right part of the string is gradually shifted as the user writes

pinch/spread gestures to pan and zoom documents and their constituent elements are available. Documents can be expanded and collapsed to respectively display and hide other pages via a short single-finger shaking gesture (Figure 2). Documents can be expanded horizontally or vertically depending on the direction of the shaking motion. A fit-to-screen zoom can be triggered with a simple double tap on the desired page for more precise editing. Lateral swipe gestures can then be used to flick from one page to another. It should be noted that no context switch is required to execute editing operations on a visible page, as those can be performed in any document state, collapsed, expanded or magnified.

Entire pages can be moved or copied within and between documents. Moving a page is achieved by placing three fingers on it and dragging it to a new location, inside an existing document or on the workspace to create a new document. Pages can be copied instead of moved by pinning them with a finger and dragging them away with the pen (Figure 3). This gesture is similar to [7] and [4], but we additionally support moving and copying of page ranges, where the two hands determine the start and end pages of the range.

Shape and text input

When the stylus is used on its own (i.e. without modifying), it should behave closely to a pen marking a physical piece of paper. However, we do not want produced documents to be scribbles and hence we introduce mechanisms to adequately beautify and format pen strokes. Currently, we distinguish two types of input: text and shape. When the user starts to mark a page with the pen, the handwriting recognition engine (VisionObject's MyScript [1]) assesses the likelihood that the input content is text. If the text recognition score is higher than a threshold value, the stroke is converted to print text after the pen is lifted and a short timeout expires. Otherwise, the strokes are considered to be shapes and the shape-recognition module formats them accordingly. Automatic distinction between text and shape does not necessarily yield the desired result. For example, an intended circle might be interpreted as the letter 'O' and vice versa. Hence, we include a means to force shape-recognition mode (as the stroke type recogniser tends to have a bias towards text) using an NDH posture, specifically the palm of the hand.

Pen strokes can be input in a blank area of a document page or in the vicinity of/inside other existing elements. In the first case, a new element is created and added to the document model. For text entered in such manner, the size of the font used for the recognised text is determined based on the user's script. In the second case, the result is added to the edited element. For text, the font size and style of the nearest letter in this element is adopted. Our text input scheme also allows handwritten words to be inserted in the middle of text elements. The user simply needs to start writing within the element for a gap to open at the insertion point. As the user writes her text, the right part of the edited string continuously shifts to create space (Figure 4). Highlighted portions of text can be edited and replaced by new user-written content as well.

Modifier touch postures and pen gestures

Following our design approach to provide a mostly clean UI thanks to gesture-supported interactions, we include a number of posture-based mode-switching triggers. In order to avoid conflicts with other unimodal gestures, specifically one-finger panning and two-finger

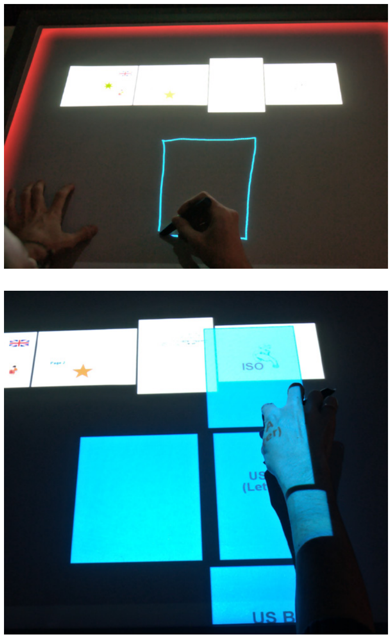

Figure 5: New documents can be created using the 3-finger command and drawing a rectangle (top). A choice of standard page dimensions then appears (bottom)

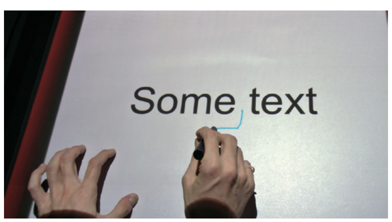

Figure 6: Inserting a line break using command mode and a pen gesture

zooming, we consider NDH postures using three or more fingers as well as blobs (contact areas larger than finger touches).

Other than the stroke-type forcing posture mentioned above, we include the following modifier postures: 3-finger for general command mode and 4-finger for selections. Visual feedback of the currently selected mode is given by light gradient borders around the screen (Figures 1, 5 and 8).

Command mode enables users to execute specific document operations determined by stroke-gestures executed with the pen. For example, a rectangle shape drawn on the workspace triggers the creation of a new document. The dimensions of the page can then be chosen among standard aspect ratios (ISO, US A and B) or the user-drawn rectangle can be selected (Figure 5). A line stroke deletes elements (pages, document components, text) crossed by it. We also include handy text editing actions for the insertion of white spaces directly in the formatted print text: a chevron (^) stroke gesture to add a space and an inverted L gesture to insert a line break (Figure 6).

The 4-finger selection mode is mostly used to select objects as well as highlight phrases in text elements. These phrases can then be removed by a delete gesture (which in this case will remove the selected text regardless of the length of the line stroke), replaced by new text, or copy-pasted in a new text element, using the finger pin + pen drag gesture of page duplication (Figure 7). The NDH holding the 4-finger posture can also be dragged up or down on the surface to increase or decrease the font size of the highlighted text (Figure 8). When an element is selected, dragging the NDH in selection mode changes the element's z-index to make it appear above or under overlapping objects.

Retrieval and insertion of external resources

Rich documents are seldom composed without, at some point, making use of external resources, such as cliparts, diagrams, pictures, copy-pasted text etc. The interface includes a virtual "drawer" panel that can be pulled out from the bottom bezel of the screen, on which users can input keywords and draw sketches to perform queries using our query-by-sketching engine [10]. For instance, if a user would like to insert an arrow in her document, she can sketch an arrow shape on the pad and additionally (or instead) enter keywords to specify the query. We provide two methods for keyword input: a small virtual keyboard or handwriting recognition. Search results are displayed in a scrollable ribbon that appears above the query pad (Figure 9). Those results are updated after each pen stroke or key input. The user can then drag desired elements from the ribbon to insert them in opened documents on the workspace. We believe this method to retrieve and insert external resources is relatively quick and easy, but we are also considering alternatives, such as allowing users to draw query sketches directly on the target document and displaying the results in situ.

Limitations

A criticism that is often levelled at minimalist interfaces that rely extensively on gestures is that their functionality is not exposed to the user and hence it is not immediately clear how a particular operation can be executed. Methods to tackle this issue of discoverability and accessibility have been proposed in the literature and we do not wish to discuss them here. In our experience, gestures are efficient and manageable if kept to

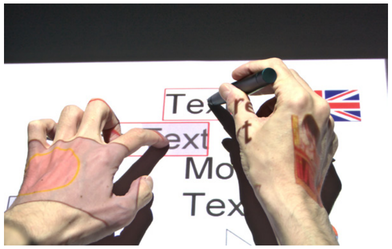

Figure 7: Creating a new text element as a copy of highlighted text using the pin + drag gesture

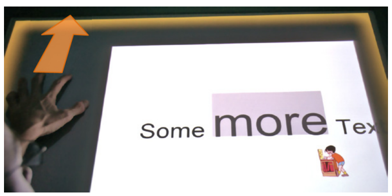

Figure 8: Font size modulation of selected text using a 4-finger vertical dragging motion

Figure 9: Dragging a clipart from the search results obtained by sketching a shape on the query pad

a core set of commonly used functions, which users can learn quickly. Our intention is not to show they can entirely replace widgets and other traditional tools (especially for a feature-rich application such as a document composer), rather that they can be a useful asset, much like keyboard shortcuts complement GUI controls of PC applications and not replace them.

Future Work

There are many avenues to explore, considering the breadth of functionality typically available in professional publishing software. Other than adding further standard editing features (especially for text styling and formatting), we would like to look next at tables and templates, which we think could also greatly benefit from a multimodal gesture approach. Tables are essential components of office documents and hence should be handled by our editor. Besides, we anticipate that many interaction techniques developed for tables are likely transferable to other types of office documents such as spreadsheets and forms. As for document templates, we believe they can function as constraint structures that can help solve some of the ambiguities, which our application tries to resolve through algorithmic inference. For example, a constraint-defining template would allow text input in a title box to be automatically converted to the correct font and size.

Finally, our application will have to be extensively tested by users in order to gain feedback about the pertinence of our interaction model. We hope our work will eventually help inform the design of future document-centric applications for pen and touch tabletops.

References

[1] http://www.visionobjects.com/en/myscript/

[2] Brandl, P., Forlines, C., Wigdor, D., Haller, M. and Shen, C. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proc. AVI 2008*, 154-161.

[3] Bright, P. Why bother? The sad state of Office 2013 touch support. *Ars Technica*, 2012, http://arstechnica.com/information-technology/2012/07/why-bother-the-sad-state-of-office-2013-touch-support/

[4] Frisch, M., Heydekorn, J. and Dachselt, R. Diagram editing on interactive displays using multi-touch and pen gestures. In *Proc. Diagrams 2010*, 182-196.

[5] Guiard, Y. Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model. *Journal of Motor Behavior*, *19*, 4 (1987), 486-517.

[6] Hamilton, W., Kerne, A. and Robbins, T. High-performance pen + touch modality interactions: a real-time strategy game eSports context. In *Proc. UIST 2012*, 309-318.

[7] Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H. and Buxton, B. Pen + touch = new tools. In *Proc. UIST 2010*, 27-36.

[8] Matulic, F. and Norrie, M. Empirical evaluation of uni- and bimodal pen and touch interaction properties on digital tabletops. In *Proc. AVI 2012*, 143-152.

[9] Matulic, F. and Norrie, M. C. Supporting active reading on pen and touch-operated tabletops. In *Proc. AVI 2012*, 612-619.

[10] Springmann, M., Kabary, I. A. and Schuldt, H. Image retrieval at memory's edge: known image search based on user-drawn sketches. In *Proc. CIKM 2010*, 1465-1468.

[11] Zeleznik, R., Bragdon, A., Adeputra, F. and Ko, H.-S. Hands-on math: a page-based multi-touch and pen desktop for technical work and problem solving. In *Proc. UIST 2010*, 17-2