TextTearing: Expanding Whitespace for Digital Ink Annotation

Dongwook Yoon
Cornell University
Information Science
Ithaca, NY 14850
dy252@cornell.edu

Nicholas Chen
Microsoft Research Cambridge
21 Station Road
Cambridge CB1 2FB, UK
nchen@microsoft.com

François Guimbretière
Cornell University
Information Science
Ithaca, NY 14850
francois@cs.cornell.edu

ABSTRACT
Having insufficient space for making annotations is a problem that afflicts both paper and digital documents. We introduce the TextTearing technique for *in situ* expansion of inter-line whitespace and pair it with a lightweight interaction for margin expansion as a way to address this problem. The full system leverages the dynamism of digital documents and employs a bimanual design that combines the precision of pen with the fluidity of touch. Our evaluation found that a simpler unimanual variant of TextTearing was preferred over direct annotation and margin-only expansion. Direct annotation in naturally occurring whitespace was least preferred.

ACM Classification Keywords: H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces

Author Keywords: e-reading, documents, annotation, whitespaces, margins, bi-manual interaction, layout

INTRODUCTION
Written annotations are a crucial aspect of engaged reading activities. Among the many roles that annotations play include demarking important passages, tracking reading progress, and recording interpretations about the text [10]. Paper documents provide a number of affordances that cater to annotation activities [14]. For instance, blank spaces in the document—margins, within-text spaces, and blank pages—offer readily available regions where annotations can be created with minimal interruption to the reading process. Moreover, the spatial proximity of these regions to the text provides context [5] and implicitly connects annotations with the text to which they refer [11]. Paper materials are not perfect, however. Pages have a set amount of space for markup, which can be insufficient at times [12].

Support for annotation is less robust when it comes to digital documents. Although many software tools support free-form inking with touchscreens or pen digitizers, writing on electronic screens tends to place more demand on the available annotation space [1]. However, the dynamic nature of digital documents provides an avenue for workarounds. Existing strategies such as comment boxes, digital Post-it™ notes [13], and the ability to insert blank pages expand the space for annotations. Unfortunately, these come at the cost of fluidity and can disrupt the reading process [14]. Moreover, displaying these types of annotations (in overlaid windows alongside the text, for example) can result in the main text being obscured or unpredictable annotation layouts.

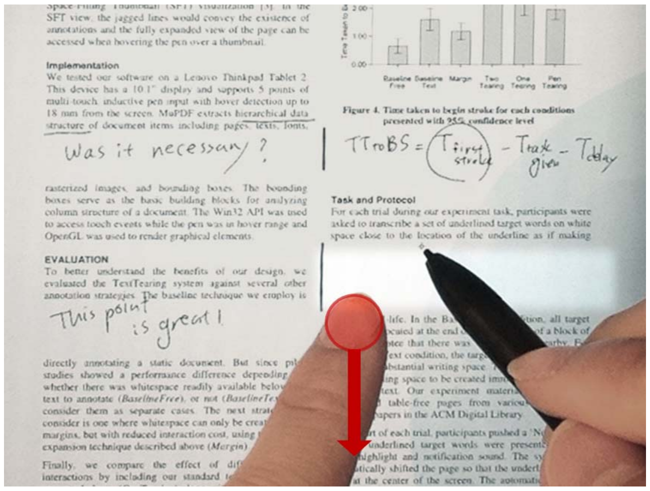


Figure 1. The TextTearing system allows additional writing space to be created between lines of text using a downward tearing gesture. The expanding region is highlighted. The red arrow indicates touch gesture opening a space. The blue shadow is a dynamic palm rejection region.

We introduce an interaction called TextTearing that addresses these problems. With this technique, users can tear open (i.e., expand) the whitespace between adjacent lines of text. This allows users to create blank space where it is needed, while maintaining the overall logical structure of the text and avoiding occlusions. We describe an initial design of our system, which includes our approach for rearranging document elements, a two handed tearing interaction based on Zeleznik et al.'s work [18], and a palm rejection system that makes it possible to implement this technique on commodity hardware. A complementary technique for expanding the page margins is also described.

Our evaluation of TextTearing compares it against a baseline where there is ample whitespace to directly write, a condition with expandable side margins, a two-handed version using an alternative tearing gesture, and a pen-only tearing technique based on pigtail gestures [7]. Our results showed that margin expansion was comparable to the baseline conditions, which were fastest because they required no interaction. However, the two highest ranked techniques in terms of user preferences employed tearing. In contrast, the baseline condition came in last, suggesting that the placement of the writing space makes a difference. Among the tearing techniques, the pen-only pigtail technique had a preference advantage over our initial design and could potentially be faster in practice.

RELATED WORK

Agrawala and Shilman [1] explained that differing physical characteristics between electronic displays and paper means that digital ink annotations are often larger and sloppier. As a result, what whitespace is available is more impacted. Their DIZI approach is to offer a zoomed-in writing pad to counteract the natural tendency to write larger, but because DIZI preserves document layout it provides no facilities for creating additional space.

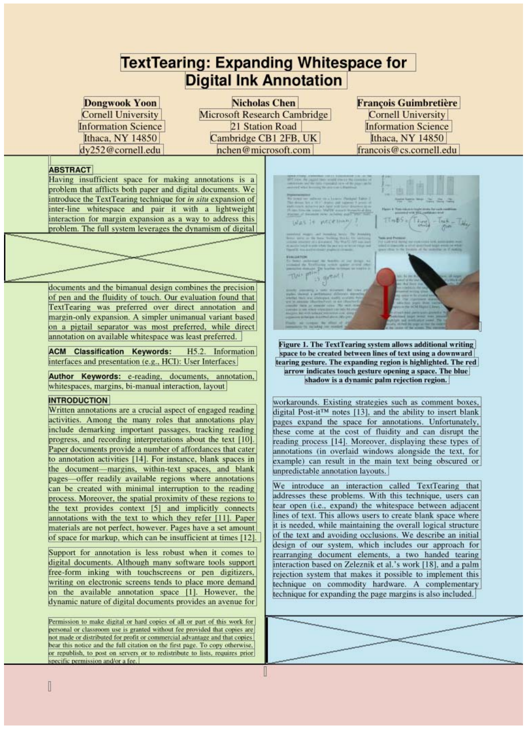
Digital documents offer a unique capability to bypass the physical constraints of paper documents, however. Chang et al. introduced the notion of fluid documents, which dynamically adjust their layout in order to display secondary information [2]. We follow this strategy but apply it in the context of annotation creation.

Zeleznik et al. [18] created a bimanual gesture for inserting and removing whitespace within a digital canvas to solve similar problems of limited writing space encountered when working through math problems. In their system, this feature is activated when a touch following a pen stroke manipulates a feed forward widget. We used a simplified version of this approach in the initial design of our system. Our study provides performance data about different variations of this interaction technique.

LiquidText used various multi-touch gestures to collapse, rearrange, highlight and extract portions of a document [16] and GatherReader [8] explored how these tasks could be further enhanced using pen + touch interactions [9]. Although the interactions in these systems bear a resemblance to TextTearing, they serve fundamentally different purposes. In these other systems the interactions assist in juxtaposing or gathering different parts of a document whereas the interactions in TextTearing are designed to help the user make *in-situ* ink annotations.

TEXTTEARING

Spatial proximity between annotations and the primary text provides context [10], helps maintain continuity of attention, and serves as an implicit connection between the annotation and the text [11]. Thus, the central goal of our system is to give readers access to writing space anywhere beside printed text. To accomplish this, we let users create an expandable region of whitespace by adjusting the spacing between lines of text. As the region grows, the content below it is shifted lower in the page. In the following sections, we first describe the algorithms we use to make the system applicable to a wider variety documents; specifically, those with multiple columns. Then, we present the interaction for performing the space expansion.


TextTearing: Expanding Whitespace for Digital Ink Annotation

Dongwook Yoon
Cornell University
Information Science
Ithaca, NY 14850
dy252@cornell.edu

Nicholas Chen
Microsoft Research Cambridge
21 Station Road
Cambridge CB1 2FB, UK
nchen@microsoft.com

François Guimbretière
Cornell University
Information Science
Ithaca, NY 14850
francois@cs.cornell.edu

ABSTRACT
Having insufficient space for making annotations is a problem that afflicts both paper and digital documents. We introduce the TextTearing technique for *in situ* expansion of inter-line whitespace and pair it with a lightweight interaction for margin expansion as a way to address this problem. The full system leverages the dynamism of digital documents and the bimanual design combines the precision of pen and the fluidity of touch. Our evaluation found that TextTearing was preferred over direct annotation and margin-only expansion. A simpler unimanual variant based on a pigtail separator was most preferred, while direct annotation on available whitespace was least preferred.

ACM Classification Keywords: H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces

Author Keywords: e-reading, documents, annotation, whitespaces, margins, bi-manual interaction, layout

INTRODUCTION
Written annotations are a crucial aspect of engaged reading activities. Among the many roles that annotations play include demarking important passages, tracking reading progress, and recording interpretations about the text [10]. Paper documents provide a number of affordances that cater to annotation activities [14]. For instance, blank spaces in the document—margins, within-text spaces, and blank pages—offer readily available regions where annotations can be created with minimal interruption to the reading process. Moreover, the spatial proximity of these regions to the text provides context [5] and implicitly connects annotations with the text to which they refer [11]. Paper materials are not perfect, however. Pages have a set amount of space for markup, which can be insufficient at times [12].

Support for annotation is less robust when it comes to digital documents. Although many software tools support free-form inking with touchscreens or pen digitizers, writing on electronic screens tends to place more demand on the available annotation space [1]. However, the dynamic nature of digital documents provides an avenue for

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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Figure 1. The TextTearing system allows additional writing space to be created between lines of text using a downward tearing gesture. The expanding region is highlighted. The red arrow indicates touch gesture opening a space. The blue shadow is a dynamic palm rejection region.

workarounds. Existing strategies such as comment boxes, digital Post-it™ notes [13], and the ability to insert blank pages expand the space for annotations. Unfortunately, these come at the cost of fluidity and can disrupt the reading process [14]. Moreover, displaying these types of annotations (in overlaid windows alongside the text, for example) can result in the main text being obscured or unpredictable annotation layouts.

We introduce an interaction called TextTearing that addresses these problems. With this technique, users can tear open (i.e., expand) the whitespace between adjacent lines of text. This allows users to create blank space where it is needed, while maintaining the overall logical structure of the text and avoiding occlusions. We describe an initial design of our system, which includes our approach for rearranging document elements, a two handed tearing interaction based on Zeleznik et al.'s work [18], and a palm rejection system that makes it possible to implement this technique on commodity hardware. A complementary technique for expanding the page margins is also included.

Figure 2. The dynamic document layout model. Boxes with an "X" are movable regions of the document. The area above and below the columns are each a single block. The left column has two text block and one region produced using TextTearing; the right, one text blocks and one tearing region.

Handling Multi-Column Layouts

To reduce undesirable effects of layout modification, such as introducing spurious spaces in adjacent columns, or collisions of shifted text into other elements, it is important for the system to understand the visual structure of the document and to enforce some modification constraints.

Our first task is to identify the position of text columns. Since our system targets PDF documents, we leverage layout information about the start and end points of the lines of text on a page. We first remove short lines of text from our analysis. Then, we use the midpoint between minimum and maximum horizontal line positions to identify the position of the alley between columns. Once that is done, we assign each line and figure into either of the left column, the right column, or leave it alone if it doesn't fall cleanly on either side. Once lines have been classified into the columns, the unclassified text is placed into either the header or footer depending on whether they are above or below the column bounding boxes. A typical structure our algorithm produces is shown in Figure 2. We currently only support two-column page layouts but this algorithm can be easily extended to handle layouts with more than two well-defined columns.

The extracted document structure dictates how the document expands. Tearing operations will always maintain the relative relationships between the header, columns, and footer. For instance, if space is created in one column, then the text below it will be pushed downward, which will in turn push the footer downward, creating whitespace at the bottom of the adjacent column if needed (Figure 2). Enforcing this high-level structure isolates expansion to a single column, which results in more predictable behavior and helps preserve spatial relationships in the document. In this first prototype we did not consider the case in which a tearing operation crosses over a long stroke such as the line in a callout, for example. This could be addressed by augmenting our system with the annotation reflow technique proposed by Golovchinsky and Denoue [4].

Annotations are anchored to neighboring pieces of text to accommodate subsequent movements of that text. This is done by first grouping strokes that are created in rapid succession (< 0.5s separation interval) together and then setting the stroke coordinates to be relative to the location of the closest piece of text so that when the text moves, so do the strokes.

Tearing Interaction

A guiding principle for the tearing interaction is that it should be precise (since lines could be closely spaced) and introduce minimal overhead compared to ordinary inking. Hinckley et al. showed that pen + touch input combines the lightweight nature of touch with the accuracy of pen input [8, 9]. For this reason, we adapt the pen + touch approach Zeleznik et al. employed to create blank space in Hands-on-Math [18]. In our system, users create additional space by hovering the pen tip over a target location and then use the finger on their other hand to "tear" the surrounding text apart. To avoid false positives that can occur when users are panning with the pen near the screen, we require that the initial finger touch must be within 16.5 mm of the line of text the pen is hovering over. Since we do not overload the hovering state, it is not necessary to have a feed forward widget like the one in Hands-on-Math.

Dynamic Palm Rejection

To provide pen + touch input along with a comfortable writing environment, we needed some way to reject touch events generated when the palm of the hand holding the pen inadvertently comes in contact with the screen. We noticed that Vogel et al.'s occlusion silhouettes model [17] provided a conservative estimate of the projection of the hand on the screen. Therefore, this model can be used to compute the area where the palm could possibly make contact with the screen. The palm rejection system can then reject any contact points in this area. Remaining touch points are passed to our application to be used for the tearing interaction as well as for panning and zooming when the pen is over the screen.

Margin Expansion

The tearing interaction described above could conceivably be used for expanding page margins as well. We discovered, however, that it was more straightforward to simply expand the margins when users pan past the current page extents. The expansion behavior does not interfere with page turning since our system uses a side tap to turn pages. Support for page turning with swipe gestures can be achieved using either a swipe starting on the bezel or a threshold mechanism to differentiate panning from swiping. Although we only enable this capability for the horizontal margins, this simplified interaction could be used for vertical margins as well.

Viewing the Original Document Layout

A three-finger touch gesture shows an alternative view of the document in which the dynamically generated whitespace regions are collapsed. This collapsed view of the document can be useful for checking the original document appearance or for ensuring that a full page can be displayed on the screen.

In the collapsed view, each collapsed area is represented by a jagged underline that reflects the projection of ink strokes onto the tear line. While maintaining this quasi-mode, hovering the pen over a jagged line overlays the associated annotation region over the page in a semi-transparent window. We also prototyped a "spring loaded" version of the software that collapses newly created regions by default. In this alternative design, keeping the pen tip in hover range keeps the newly expanded region open, and exiting the range restores the original document layout. Our experience with the technique suggested that a more effortless and stable way to maintain the quasi-mode is needed, however.

The collapsed view of the document may also be useful for maintaining a constant page aspect ratio for alternative visualizations such as Space-Filling Thumbnails (SFT) [3]. In the SFT view, the jagged lines would continue to convey the existence of annotations and the fully expanded view of the page can be accessed when hovering the pen over a thumbnail.

Implementation

We tested our software on a Lenovo Thinkpad Tablet 2. This device has a 10.1" display, multi-touch input supporting up to 5 simultaneous points of contact, and inductive pen input with hover detection up to 18 mm from the screen. We used the MuPDF library in conjunction with

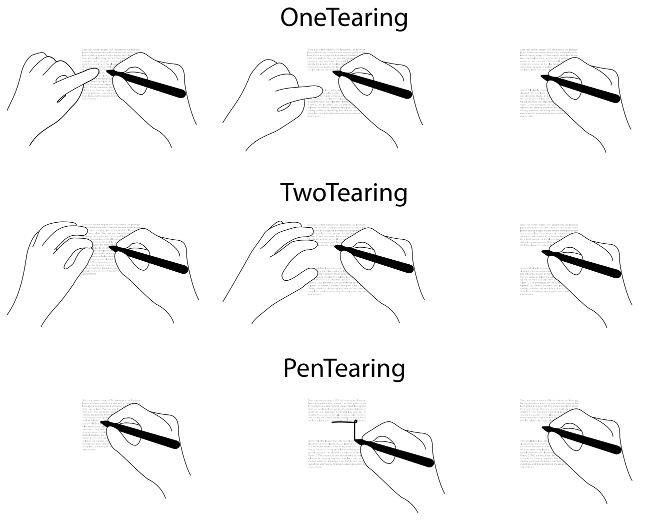


Figure 3. Tearing gestures considered in the experiment

OpenGL to render PDF documents onto the screen. MuPDF provided the hierarchical data structure of document items including page boundaries, text, raster images, and bounding boxes. The bounding boxes served as the basic building blocks for analyzing the column structure of a document. The palm rejection scheme currently used in Windows 8 does not support simultaneous bimanual interaction as it disables all touch input events when the pen is in hover range. We were able to work around this issue by using the RawInputDevice Win32 API to access touch events while the pen was in hover range.

EVALUATION

To better understand the benefits of our design, we evaluated the TextTearing system against several other annotation strategies. The baseline technique we employ is directly annotating a static document. But since pilot studies showed a performance difference depending on whether there was whitespace readily available below the text to annotate (*BaselineFree*), or not (*BaselineText*), we consider them as separate cases. The next strategy we consider is one where whitespace can only be created in the margins, but with reduced interaction cost, using the margin expansion technique described above (*Margin*).

Finally, we compare the effect of different tearing interactions by including our standard tearing technique presented above (*OneTearing*) along with two alternatives (Figure 3). The first alternative (*TwoTearing*), replaces the single finger non-dominant hand (NDH) interaction with a two finger NDH interaction resembling the gesture used for zooming on multi-touch devices. While hovering the pen, spreading one's fingers from a pinching position increases the size of the whitespace region. A possible benefit of this approach is that the positions of the NDH fingers map directly to the expansion region. The second alternative (*PenTearing*) uses a single handed pen interaction based on pigtail delimiters [7]. In this condition, a long horizontal line (> 12.0 mm) followed by a pigtail starts the tearing operation. The centroid of the stroke points preceding the crossing point indicates where the whitespace expansion should occur. The pen stroke is then extended downwards to specify the size of the expansion. The trigger is distinctive enough that it does not seem to interfere with writing activity. We considered but ruled out using the pen barrel button as a trigger since it can be error-prone [15].

Task and Protocol

For each trial during our experiment task, participants were asked to transcribe a set of underlined target words on white space close to the location of the underline as if making notes in real-life. In the BaselineFree condition, all target words were located at the end or the beginning of a block of text to guarantee that there was writing space nearby. For the BaselineText condition, the target was at least four lines away from substantial writing space. The tearing conditions required writing space to be created immediately below the underlined text. Our experiment materials consisted of picture and table-free pages from various two-column formatted papers in the ACM Digital Library.

At the start of each trial, participants pushed a 'Next' button and the underlined target words were presented with a visual highlight and notification sound. The system then automatically shifted the page so that the underlined words were at the center of the screen. The automatic centering removed page adjustment performance as a factor so that the measured time better reflects the amount of time it takes the participant to find or create writing space. Our implementation required a one-time iterative calibration procedure to manually tune the 5 parameters of the hand occlusion model (e.g., hand radius, forearm angle, etc.). This process could be made automatic, however [17].

Each technique was tested in one block. A block consisted of an introduction to the technique, a practice phase with 18 trials, and an experimental phase with 13 trials. The order of the techniques and materials was counterbalanced using a 6×6 Latin square. At the end of each block, participants were interviewed and completed a paper-based NASA Task Load Index survey (TLX). After all blocks were finished, the participant sorted 5 cards, representing the baseline and the four other techniques we tested, by preference. There were 12 participants in total (10 females, 2 males, average age 23.2 years old). Participants were paid $10 for the hour-long experiment.

RESULTS

Measurements of the elapsed time were computed starting from when the underlined target words are shown to the beginning of the first inking stroke. We accounted for the skewed distribution of human reaction time by taking the median value of a block's trials. We used Greenhouse-Geisser correction when we could not assume sphericity and the Bonferroni correction for pairwise comparisons.

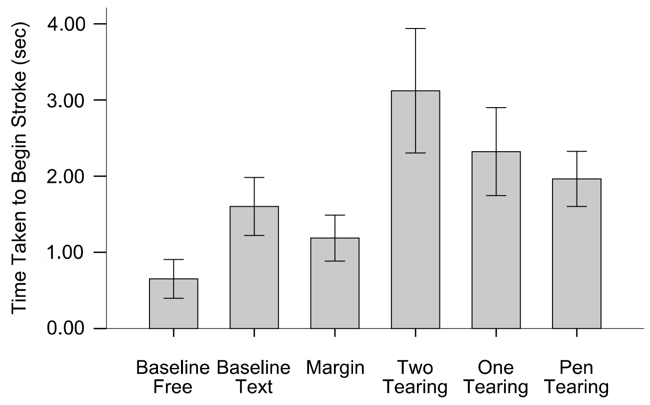


Figure 4. Time taken to begin stroke for each conditions presented with 95% confidence level

Errors

We counted an error whenever participants opened a space below a line other than the one that was underlined. A one-way repeated measure ANOVA on error rate showed a marginally significant effect of technique ($F(1.56, 17.17) = 3.73$, $p < .054$, partial $\eta^2 = .25$). TwoTearing (M = 13.69%, SD = .044) showed a significantly higher error rate than PenTearing (M = 2.78%, SD = .054, $p < .05$), due to small panning motions induced by the first finger prior to the second finger touching down. OneTearing fell in the middle (M = 7.63%, SD = .024) but the differences were not significant. We also measured the occurrence of false tearing activations for the pigtail gesture and found a single case across all trials (0.6%) that occurred when the participant was writing in cursive.

Total Space Opening Time

A one-way repeated measure ANOVA on the total space opening time showed a significant effect of technique ($F(2.47, 27.18) = 19.5$, $p < .001$, partial $\eta^2 = .64$). BaselineFree was the fastest condition (M = .65 sec, SD = .4, $p < .004$) followed by the Margin condition (M = 1.19 sec, SD = .48) that was marginally faster than BaselineText (M = 1.60 sec, SD = .60, $p < .067$) with both significantly faster than all tearing techniques ($p < .037$). It was somewhat surprising that the Margin technique performed in between the two baseline conditions that required no interaction at all. We believe that even though extra time was required to perform the margin expansion it is offset by the savings from not having to find a place to write.

For the tearing-enabled interactions, TwoTearing was the slowest (M = 3.12 sec, SD = .37) but the difference was only significant versus PenTearing (M = 1.96 sec, SD = .57, $p < .005$), but not OneTearing (M = 2.32 sec, SD = .91, $p < .098$). Given the large difference in means, we looked more carefully at our data and discovered that an overly cautious participant caused a large increase in the variance. Re-running the analysis without this participant resulted in TwoTearing being significantly slower than both of the other techniques ($p < .005$). The difference between PenTrearing and OneTearing was not significant ($p < .18$). To better characterize differences between OneTearing and PenTearing, we decomposed the total space opening time into 3 components: the time it took to begin tearing (T_{begin}), the time for the tearing action (T_{Tear}) and the time it took to start writing after the tearing action was completed ($T_{1^{st}\ stroke}$). We found that most of the difference between the two techniques were from $T_{Tear,}$ with PenTearing being faster ($t(11) = 5.43$, $p < .0001$).

Subjective Preferences

We performed a Friedman test on the ranking data we gathered at the end of each session. We found a significant difference in ranking ($\chi^2 = 46.80$, $p < .001$), with PenTearing (1.25) coming first followed by OneTearing (2.25), Margin (3.17), and TwoTearing (3.75). The baseline technique was the last (4.58). The TLX showed that TwoTearing required the highest effort ($F(2.81, 30.87) = 5.78$, $p < .003$, partial $\eta^2 = .34$). No other significant differences were observed.

DISCUSSION

PenTearing was the most preferred, and rated higher than the Baseline techniques even though it was 3 times slower than direct annotations. We believe this effect illustrates that directly writing in naturally occurring space is not necessarily desirable even though it can be done quickly. This interpretation is in line with Agrawala and Shilman's observations about the difficulties of writing on electronic documents [1]. Additional studies are needed to determine how our expansion-based techniques compare to a zooming scheme like DIZI.

The fact that two of the tearing techniques were preferred over margin annotation reinforces the notion that the spatial positioning of the annotation is important. One participant noted that creating a space below the target text for writing resulted in less occlusion from the writing hand compared to annotating beside the target text (i.e. writing in the margin). We believe that reduced occlusion, along with ergonomic and semantic benefits from being able to position annotations flexibly, are advantages that TextTearing would bring to real-world annotation tasks.

Participants' preference for the one-handed, pen-only, tearing technique over the two-handed ones was surprising given the expected advantages of bimanual interaction [6]. Participants told us that PenTearing was a simpler one-step gesture. They also pointed out that without hover as a mode delimiter, PenTearing allowed the pen tip to remain on the screen, which resulted in more stable targeting. Finally, participants mentioned that performing the tearing interaction using a single hand produced less occlusion than using two hands and freed the ND hand for other, unrelated, tasks. From an implementation standpoint, the simpler input hardware required is an additional advantage of the one-handed technique. We believe, however, that additional investigations of bimanual versus unimanual modalities in the context of real-world activities are needed to definitively confirm these advantages.

Palm Rejection

Despite the fact that users were required to have the pen in the hover zone before resting their palm on the screen, no participant thought avoiding these false-positives was burdensome. One situation where the occlusion model failed was when users rotated the screen prior to writing, such as to squeeze text into a narrow space during a baseline interaction. This situation could potentially be remedied by using on-board motion sensors. It bears noting that the palm rejection functionality is still useful even with a technique like PenTearing because it enables other touch gestures, such as navigation, while the pen is in the hover volume.

CONCLUSION AND FUTURE WORK

In this paper, we presented TextTearing, an interaction technique for expanding the whitespace between text lines. We also described a palm rejection technique for enabling pen + touch input. We conducted an experiment comparing three variations on the TextTearing technique against each other as well as against margin expansion and a baseline system with no expansion capabilities. Participants rated TextTearing techniques highly, which suggests that TextTearing is a desirable method for supporting annotation. The pen-only and one-finger tearing variants were faster than the two-finger variant and the pen-only technique was the most preferred.

In future work, we hope to see TextTearing used as part of a deployable document viewer to observe its effectiveness for a full range of real-world practices against methods such as DIZI [1] or sticky notes. Furthermore, studying the impact of TextTearing on annotation consumption tasks such as reading, searching, maintaining, and sharing would complement the research presented in this paper.

ACKNOWLEDGEMENTS

Yoon gratefully acknowledges support from the Kwanjeong Educational Foundation. This work was made possible by a generous gift from FX-PAL. We would like to thank Ken Hinckley and Michel Pahud at Microsoft Research for their technical assistance in managing pen and touch events simultaneously.

REFERENCES

1. Agrawala, M. and Shilman, M. DIZI: a digital ink zooming interface for document annotation. Human-Computer Interaction-INTERACT 2005, (2005), 69-79.
2. Chang, B.-W., Mackinlay, J.D., Zellweger, P.T., and Igarashi, T. A negotiation architecture for fluid documents. In Proc. UIST 1998, ACM Press (1998) , 123–132.
3. Cockburn, A., Gutwin, C., and Alexander, J. Faster document navigation with space-filling thumbnails. In Proc. CHI 2006, ACM Press (2006), 1-10.
4. Golovchinsky, G. and Denoue, L. Moving markup: Repositioning Freeform Annotations. In Proc. UIST 2002, ACM Press (2002), 21-30.
5. Golovchinsky, G., Price, M.N., and Schilit, B.N. From reading to retrieval: freeform ink annotations as queries. In Proc. SIGIR 1999, ACM Press (1999), 19–25.
6. 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.
7. Hinckley, K., Baudisch, P., Ramos, G., and Guimbretiere, F. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In Proc. CHI 2005, ACM Press (2005), 451-460.
8. Hinckley, K., Bi, X., Pahud, M., and Buxton. B. Informal information gathering techniques for active reading. In Proc. CHI '12 (2012), 1893–1896.
9. Hinckley, K., Yatani, K., Pahud, M., et al. Pen + touch = new tools. In Proc. UIST 2010, ACM Press (2010), 27-36.
10. Marshall, C.C. Annotation: from paper books to the digital library. In Proc. DL 1997, ACM Press (1997), 131–140.
11. Marshall, C.C. Toward an ecology of hypertext annotation. In Proc. HYPERTEXT 1998, ACM Press (1998), 40–49.
12. Pearson, J., Buchanan, G., and Thimbleby, H. Improving annotations in digital documents. In Proc. ECDL 2009, Springer-Verlag (2009), 429-432.
13. Pearson, J., Buchanan, G., and Thimbleby, H. The reading desk: applying physical interactions to digital documents. In Proc CHI 2011, ACM Press (2011), 3199-3202.
14. Sellen, A. and Harper, R. The Myth of the Paperless Office. MIT Press (2003).
15. Song, H., Benko, H., Guimbretiere, F., Izadi, S., Cao, X., and Hinckley, K. Grips and gestures on a multi-touch pen. In Proc. CHI 2011, ACM Press (2011), 1323-1332.
16. Tashman, C.S. and Edwards, W.K. LiquidText: a flexible, multitouch environment to support active reading. In Proc. CHI 2011, ACM Press (2011), 3285-3294.
17. Vogel, D. and Balakrishnan, R. Occlusion-aware interfaces. In Proc. CHI 2010, ACM Press (2010), 263-272.
18. 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, ACM Press (2010), 17-26.