SpaceInk: Making Space for In-Context Annotations

Hugo Romat
INRIA & TKM
Université Paris-Saclay, France

Emmanuel Pietriga
INRIA, Univ. Paris-Sud, CNRS
Université Paris-Saclay, France

Nathalie Henry-Riche
Microsoft Research
Redmond, WA, USA

Ken Hinckley
Microsoft Research
Redmond, WA, USA

Caroline Appert
Univ. Paris-Sud, CNRS, INRIA
Université Paris-Saclay, France

ABSTRACT
When editing or reviewing a document, people directly overlay ink marks on content. For instance, they underline words, or circle elements in a figure. These overlay marks often accompany in-context annotations in the form of handwritten footnotes and marginalia. People tend to put annotations close to the content that elicited them, but have to compose with the often-limited whitespace. We introduce SpaceInk, a design space of pen+touch techniques that make room for in-context annotations by dynamically reflowing documents. We identify representative techniques in this design space, spanning both new ones and existing ones. We evaluate them in a user study, with results that inform the design of a prototype system. Our system lets users concentrate on capturing fleeting thoughts, streamlining the overall annotation process by enabling the fluid inverleaving of space-making gestures with freeform ink.

Author Keywords
Annotation, e-reading, whitespaces, document layout, reflowing, pen+touch.

CCS Concepts
•Human-centered computing → Interaction techniques; *Touch screens; User studies;*

INTRODUCTION
Annotations are central to active reading. They are a means to emphasize and memorize specific pieces of information, to facilitate re-reading, to identify passages to revise and suggest edits, or to support sense-making [24, 19, 37, 27]. When annotating a document, people make ink marks directly on the content: they underline words, they draw links between related items, they circle elements in figures. These *overlay marks* are often accompanied by *in-context annotations*, that typically take the form of handwritten footnotes, marginalia, or sketches. In-context annotations are not overlaid on top of the document's text and figures, but rather placed close to the content that elicited them, wherever there is free space to accommodate them.

On paper, of course, such space is extremely limited and constrained by physical pages. Digital devices enable people to mark up documents in ways that may go beyond what is possible with physical pen and paper [23]. However, while annotating a printed document is just about jotting down brief comments or ideas and anchoring them to a particular passage (*e.g.*, by call-out lines), annotating an electronic document is often much more cumbersome. People have to create digital Post-It notes and fill them in, or create comments that will appear in a dedicated sidebar. These interactions with the interface impact active reading as they interfere with the capture of fleeting thoughts, driving many people to instead rely on pen and paper [27].

Researchers have proposed techniques to relax the physical constraints of paginated documents (*e.g.*, TextTearing [36]). However, digital documents are not necessarily paginated for printing on standard size paper. They are often designed to support different rendering contexts, for instance adapting the layout to different devices or screen orientations. This adaptability opens a wide range of possibilities to embed annotations within the document's content. In this work, we seek to articulate some of these options, explore their implications, and thereby sketch out the design space of digital annotations that are not necessarily constrained by physical space.

The goal of our research is to leverage the power of digital inking for active reading, reducing interface friction to offer fluid interactions for interleaving overlay marks and in-context annotations. We introduce SpaceInk, a design space of pen+touch input techniques that make room for in-context annotations by dynamically reflowing a document's content. The design space organizes techniques according to *when* additional space is created: before, while or after the user handwrites annotations; and *where* additional space is created: on the paragraph's side, wrapped inside the paragraph, between lines, or between words.

SpaceInk encompasses existing techniques such as TextTearing [36] – which enables the creation of white space between paragraphs in paginated documents to write annotations – and uncovers variations – *e.g.*, space is made *a posteriori*, avoiding

UIST 2019

ISBN 123-4567-24-567/08/06. . . $15.00

DOI: http://dx.doi.org/10.475/123_4

users' premature commitment. It also suggests new techniques that differ in the level of user agency regarding the size and location of the space to be made. Users may have full control over it, the system may automatically adjust it, or it may offer a mixed approach in which users retain some of the control. We gather feedback about these techniques in a user study, the results of which inform the design of a prototype system that lets users concentrate on capturing fleeting thoughts, streamlining the overall annotation input process by enabling the fluid interleaving of space-making and inking actions.

To summarize, our contributions are the following:

- SpaceInk: a design space of pen+touch input techniques for in-context annotations;
- insights from a study on 5 techniques in this design space;
- a design rationale and implementation of a prototype system for fluidly interleaving space-making and inking actions.

RELATED WORK

Digital Active Reading

Researchers from the XLibris project [32, 12, 11] were the first to propose an "active reading machine", with which users could annotate text using freeform marks on a pen+tablet. The prototype made it possible for handwritten notes to coexist with regular text, offering an experience close to taking notes with an analog pen on a printed document. As opposed to text input with a keyboard, handwriting with a stylus allows readers to write on the material while keeping their marks clearly distinguishable from the original content, a feature that is especially important to readers [24, 19].

Active reading with stylus-equipped devices has been further investigated since then. Matulic and Norrie [22] describe an application for active reading that uses pen+touch input on a tabletop. Pen input is dedicated to annotations, while multi-touch gestures support navigation: flipping pages, jumping to a specific page using space-filling thumbnails [9]. Another example is LiquidText [35, 34], an application that builds upon the XLibris notebook's design, where users could paste annotated text segments as clippings [11]. LiquidText splits the viewport into two areas: one shows the main document while the other shows a workspace that stores excerpts from it, dropped there by users. This workspace features advanced interactions for grouping annotations, linking them to several parts of the document, and even using them as navigational cues into the paginated document. LiquidText also enables users to collapse portions of a document in the spirit of the Mélange technique [10] to better support the sort of side-by-side comparisons that active readers often make [24, 23].

Active Annotations

As mentioned earlier, digital annotations can be leveraged to perform actions their analog counterparts are incapable of. For example, the MATE system [14] allows users to turn some specific handwritten marks into actual content edits. The digital world is also particularly effective at indexing content, as demonstrated in XLibris [12] and InkSeine [17]. Both systems infer queries from freeform annotations in order to bring up results that are likely to be of interest to users. XLibris builds queries from the content to which annotations are anchored, while InkSeine directly uses the handwritten words. In both, the system makes the hypothesis that annotations capture users' interests, and that they can act as starting points for further investigation. Very recently, the ActiveInk system [28] has gone one step further, making it possible to turn annotations made on data visualizations into analytical actions on the underlying data, such as filtering items out.

A frequent problem with annotations on printed paper is the lack of available space to accommodate them. Electronic documents offer opportunities to overcome this limitation. The DIZI system [1] partially addresses this problem by facilitating pen input in small spaces, thanks to a magnifying lens that pops up when users start annotating. Space between lines gets larger, making handwriting with a stylus more comfortable. However, the available space for annotations remains limited, and handwritten text can be very difficult to read when not magnified. Chang *et al.* [7] consider the problem from a navigation perspective. Frequent movements between pages are often poorly supported when reading on a digital device [33], and they introduce an architecture that facilitates navigation between *primary* material and *supporting* material in order to minimize such movements. They describe several techniques such as moving blocks and compressing interlines of primary material, or adding an overlay to make space for supporting material in the context of the primary material. While their system has not been implemented for ink-based annotations but rather for supporting material such as footnotes, it is relevant here as it transiently alters the layout of the primary material to make space for additional, related material.

Existing software implement more radical solutions, such as inserting user interface components floating over the text (*e.g.*, digital Post-It notes) or embedded within the text (*e.g.*, text boxes); or switching to a different view mode displaying a dedicated comment sidebar. However, these techniques interrupt the users' workflow as they require navigating menus and, in some cases, moving and resizing interface components. This disruption may impact the active reading process and interfere with the capture of fleeting thoughts. TextTearing techniques by Yoon *et al.* [36] offer pen+touch interactions that enable a more fluid workflow. Users explicitly create white space boxes between paragraphs to accommodate their annotations, thus actively changing the document's layout. The canvas of the page is resized, altering the *page aspect ratio* across the document, so as to preserve its pagination and ensure that all pieces of content remain on their original page. In their study, Yoon *et al.* found that participants preferred annotating in the extra space created with the technique than in the white space that was there in the original layout. The approach lets users integrate in-context annotations tightly with the document's content, which is particularly efficient when collaborating with other users. RichReview [37] implements TextTearing techniques coupled with the capability to record speech and deictic gestures associated with their annotations to facilitate communication between co-workers.

Our design space includes the above TextTearing techniques, but explores a wider range of possibilities to make space for in-context annotations. It relaxes contraints on document layout much further, taking advantage of content reflowing techniques, that provide opportunities not explored so far. Beyond space-making strategies, it also considers the different moments *when* it might be appropriate to make space for annotations from an interaction perspective, as we detail later.

Overlay Ink and Reflowable Documents

The HCI community has already started studying the problem of embedding annotations into reflowable documents such as Web pages. It has investigated several questions: how to store annotations and access them later on [26]; how to reconcile them with their scope when a Web page's content changes [4]; how to share them with other people [5, 37].

Closer to our work, other projects have looked at the problem of making annotations behave properly in case of content reflowing. This typically occurs when looking at the same page on another device, but on the same device as well, for instance when the browser window gets resized or when the font size gets changed [4, 11].

Systems such as *u*-Annotate [8] and iAnnotate [25] let users annotate Web pages with handwritten annotations. Both are implemented as plug-ins that put annotations on a transparent layer on top of the page. Annotations are anchored to the closest HTML element, so that their position can be properly restored no matter the actual layout rendered on screen.

Golovchinsky *et al.* also investigated the problem of making freeform annotations robust to content reflowing for non-paginated documents in XLibris [11]. The system computes the scope of the ink marks (the words that users have circled or underlined with the pen) so as to not only restore the position of those marks, but properly adapt their shape as well: they can get stretched or split to remain aligned and coherent with their initial scope, which might end up distributed on different lines or even successive pages. Bargeron and Moscovich [3] studied how users react to such techniques. They found that users like the automatic adaptation to the reflowing when the system does it properly; and that users prefer it when the system beautifies their freeform annotations, as opposed to re-rendering them with their original drawing style. While these projects have focused on making overlay ink reflow with the content, the SpaceInk design space proposed here is about techniques to make room for in-context annotations.

To summarize, previous work explored multiple aspects of digital annotation, suggesting strategies to reflow overlay ink as the document content evolves, as well as devising interaction techniques for making room for annotations within the content. Our research advances our understanding of in-context annotations, providing a space for reflecting on design dimensions that impact the user experience, and reporting on a study shedding light on their implications. Our work is also the first to consider how both overlay marks and in-context annotations coexist during active reading (Figure 10) with the goal of identifying a set of considerations about the design of interactions to fluidly interleave them.

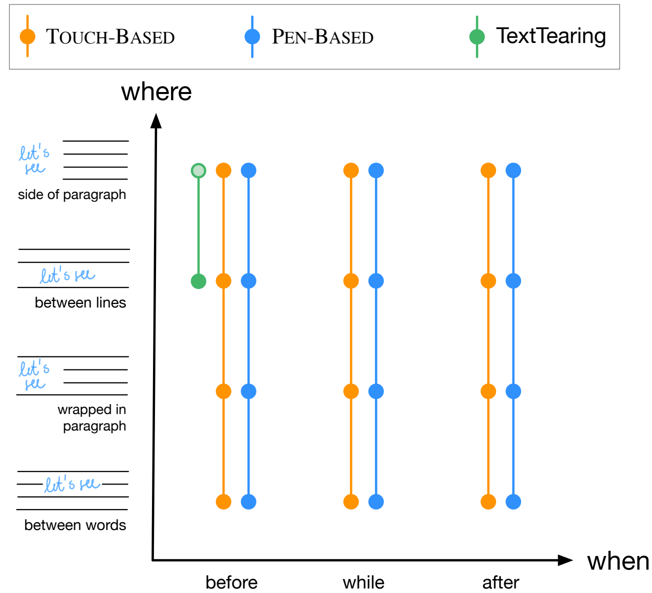


Figure 1. SpaceInk is organized along two dimensions: *when* and *where*. It includes the six techniques tested in our study (orange and blue), as well as TextTearing techniques [36] (green).

SPACEINK

In this section, we first map the space of interaction techniques for in-context annotations, organizing them around two salient dimensions. We describe our rationale, as well as a set of representative techniques that we implemented for further empirical study.

Design Space

Interaction techniques for in-context annotations consists of space-making actions (causing the document content to reflow) and inking actions (freeform input of the annotation content). SpaceInk organizes them along two dimensions, as illustrated in Figure 1:

- *where* the annotation is inserted relative to the corresponding content – annotations can be inserted *between words*, *between lines*, or they can be *wrapped in a paragraph* or *put on the side of a paragraph*;
- *when* do users interact to make space for the annotation – they can push content *before* annotating, *while* annotating, or *after* annotating.

Considering *where* the annotation is inserted is important, as it defines the spatial proximity with its scope – the content to which the annotation is anchored to. Allowing users to precisely adjust their annotations' position ensures that in-context annotations are rendered as close as possible to the content they refer to.

Considering *when* the space is made for the annotation is also important, as this interaction may interefere with active reading. Deciding of the location and size of the space needed beforehand takes the focus away from the capture of fleeting thoughts, while doing it after may obfuscate key content that could be necessary to finish formulating an idea.

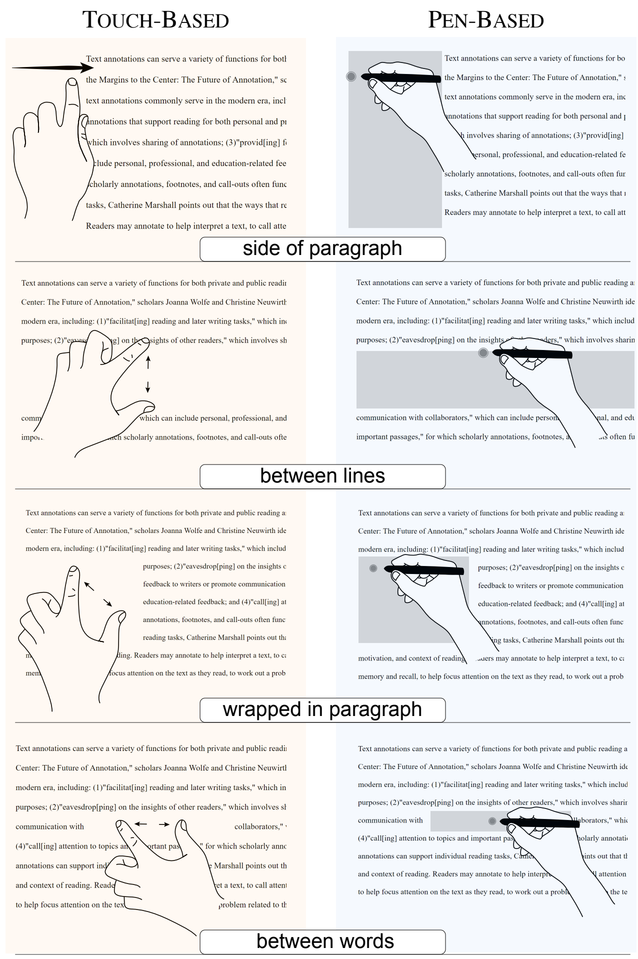


Figure 2. Both TOUCH-BASED and PEN-BASED techniques let users specify *where* they want annotations to be included relative to the content.

Figure 1 shows where TextTearing techniques [36] are situated in this space. They mostly fall in cell *where* = *between lines* and *when* = *before*, as they require users to first create some space between paragraphs (*i.e.*, between the last line of a paragraph and the first line of the following paragraph), and only then put ink in that space. Yoon *et al.* also decribe a *margin* technique that allows users to widen a document's margin to get more space for annotations. We position this technique in the {*side of paragraph* × *before*} cell, as it corresponds to the case where users create extra space on the side of paragraphs. It is not an exact fit though, as the technique from [36] affects all paragraphs that belong to the page.

This design space opens up new possibilities regarding strategies to make space for in-context annotations, that we start to explore with examples of techniques in the next section.

Implemented Techniques

We used SpaceInk to generate two sets of techniques: those that are TOUCH-BASED, and those that are PEN-BASED (Figure 2). Each set is comprised of three techniques (the vertical lines in Figure 1), one per value of *when*. For each technique, users choose *where* to make room for annotations among four options, as detailed below:

- TOUCH-BASED (Figure 2, left column): Given the ubiquitous use of single-finger panning gestures to scroll Web pages while reading them, we decided to co-opt the pinch-to-zoom gesture instead. We reasoned that zoom level would likely be set once before the active reading started and seldom adjusted. Co-opting it thus seemed less disruptive to users' workflow. Diagonal pinch gestures[1] create white space wrapped inside text; vertical gestures create space between lines; and horizontal gestures create space between words. However, as pinch-to-zoom proved often awkward to perform in tight spaces close to the screen bezel, we decide to use a swipe gesture for margin expansion, as proposed in [36]. This gesture can be distinguished from horizontal scrolling in web pages or page flipping in paginated documents by recognizing a swipe starting from the bezel [29].

- PEN-BASED (Figure 2, right column): different types of white space are generated depending on *where* the pen is when users start annotating. If starting in the margin of a paragraph, the paragraph gets pushed towards the right. If starting between two lines, the lines below the pen's position get pushed downward. If starting on the first word of a line, an empty box is added there and the paragraph's text gets wrapped around it. Finally, if starting on another word, or between two words of the same line, words that follow the pen's position are pushed toward the right, reflowing all sentences that follow (if need be).

Figure 3 illustrates how gestures (or pen-down events) and actual inking actions are temporally organized in each of the six techniques:

- Top row: users create space *before* inking. TOUCH-BASED case: the size of the white-space box is defined by the amplitude of the pinch gesture. PEN-BASED case: a box with a predefined size is created when the pen hovers the surface. In the current proof-of-concept implementation, assuming an average font size of 3mm (ascent+descent), a box on the side of a paragraph is 19mm wide; a box wrapped in a paragraph is 19×19mm; a box between lines is 9mm high; a box between words is 38mm wide.

- Middle row: users create space *while* inking. TOUCH-BASED case: white space is created by means of a pinch gesture. PEN-BASED case: white space is automatically added as the annotation gets longer, so as to prevent it from overlapping the document's content.

- Bottom row: users directly ink above the text and create white space to accommodate their annotations only *after* they are done inking; using a pinch gesture (TOUCH-BASED case) or tapping on the ink to tell the system to create the necessary amount of white space (PEN-BASED case).

We implemented all six techniques and iterated on them through pilot studies.[2] We eventually decided to discard the

[1] whose direction $d \in [20°, 70°]$ in our current implementation.
[2] Short videos demonstrating each technique are available at http://ilda.saclay.inria.fr/spaceink.

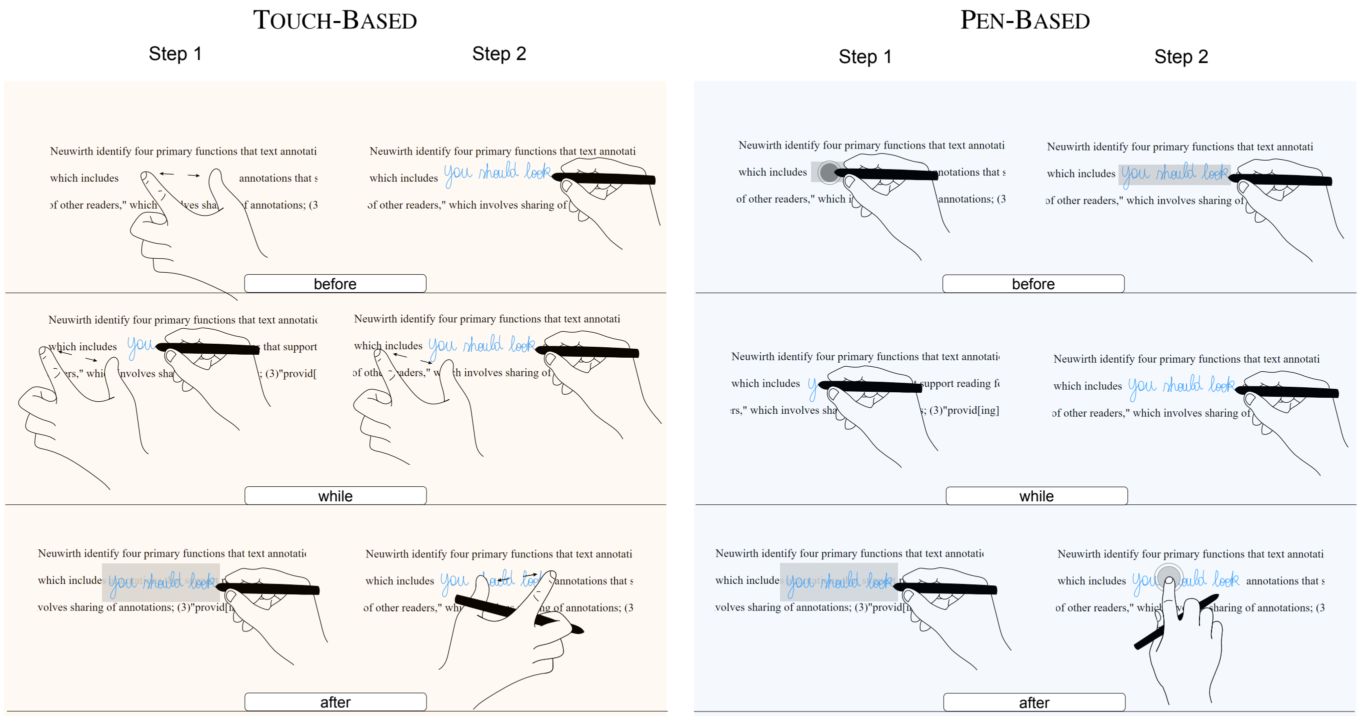


Figure 3. Both TOUCH-BASED and PEN-BASED techniques let users specify the strategy for creating white space at different moments (*when*): either *before*, *while*, or *after* annotating. With TOUCH-BASED techniques, users were free to both gesture *and* write with a single hand (as illustrated in the *after* condition) or with two hands (as illustrated in the *before* condition).

TOUCH-BASED technique that lets users adjust white space *while* inking, as it proved too difficult for users to perform a pinch gesture while writing at the same time. We identified properties of the remaining five techniques that might affect user experience. In particular, the extent to which content adaptation is under the user's control is higher for TOUCH-BASED techniques than for PEN-BASED techniques. With TOUCH-BASED techniques, the pinch gesture acts as a rubber-band interaction that defines the location and size of the created space. With PEN-BASED techniques, which rely on a single input channel, the control of the white-space area is merged with the act of annotating. The system plays a more active role in the specification of the area's dimensions, which is either predefined (*before*) or automatically computed by the system (*while*/*after*).

USER STUDY

The aim of our experiment is to gather qualitative feedback about these representative techniques, and to relate their performance with the contexts in which annotations are made. Based on the definition of an annotation as a marking made of some *content* and an *anchor* from [4], we introduce the following two factors to operationalize different contexts in which annotations are made:

- annotation *Length*: SHORT, LONG, or EXPANDING; and
- annotation *Scope*: INLINE or BLOCK.

The annotation's *Length* corresponds to how many handwritten letters and symbols it contains. This number can vary with the annotation context. For example, personal annotations tend to be tacit and short, as opposed to annotations to be shared, which are more verbose and explicit [20]. In addition to these two cases (SHORT and LONG), we also consider EXPANDING annotations in order to operationalize the situation where users are incrementally structuring their thoughts, or where they want to add more ink to an annotation, as a result of further investigation. This contrasts with the study reported in [36] where users were always able to anticipate the content, and thus the length, of their annotation. We believe that this is an important factor to consider, as long or expanding annotations make it difficult or impossible for users to anticipate how much space they will need to fit their markings. We expect this factor to impact *before* techniques particularly. Indeed, users have to anticipate how much space to create so as to make the annotation fit, which induces much premature commitment [13]. This leads to our first hypothesis:

- H_1: the usability of *before* techniques is affected by the length of the annotation.

The *Scope* factor can take the two values described by Marshall [19]: BLOCK corresponds to her *margin* anchor (*i.e.*, the annotation's scope is a paragraph), and INLINE to her *range* anchor (*i.e.*, the annotation's scope is a highlighted portion in a paragraph). As people like to ensure proximity between an annotation and its anchor [20], *Scope* might have an impact on their annotation strategy. This leads to our second hypothesis:

- H_2: users want to embed the annotation inside the content when the anchor is a portion of a paragraph (INLINE).

The hypotheses formulated above suggest that there is no *a priori* clear winner, but rather that each technique might have strengths and weaknesses. Our study aims at identify-

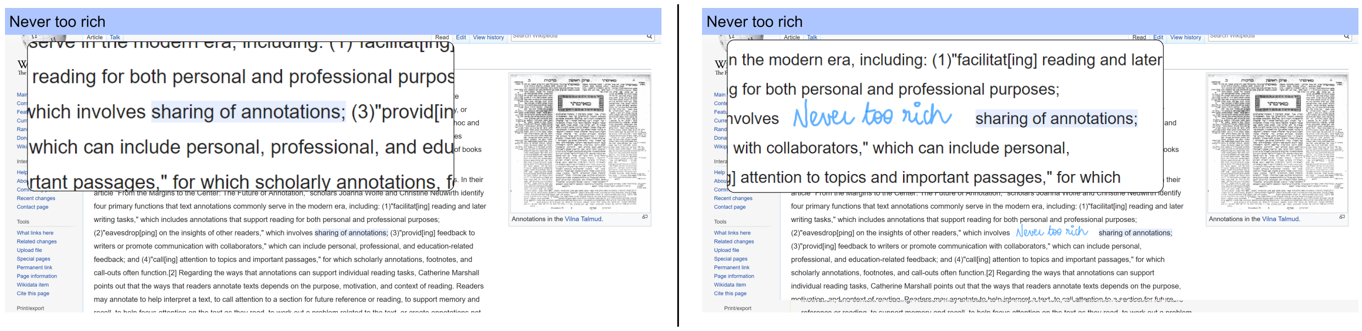


Figure 4. Experimental task. (left) stimulus: a participant is instructed to annotate the highlighted text `sharing of annotations` with words `Never too rich`. (right) response: she annotates in-context, making space for her ink using a *between-words* strategy.

ing what strategies were effective and in what context. We follow a within-subject design where participants have to perform annotation tasks under the above six conditions, with each of the five techniques introduced in the previous section: TOUCH-BASED+*before*, TOUCH-BASED+*after*, PEN-BASED+*before*, PEN-BASED+*while*, and PEN-BASED+*after*. In all conditions, participants are instructed to put annotations as close as possible to the highlighted anchor, and to minimize the amount of wasted white space.

Task

The experimental task is illustrated in Figure 4. Participants are presented with *i)* a document in which a specific text fragment is highlighted (the annotation's anchor), and *ii)* the text of the annotation they have to write. We deliberately designed a task focused on low-level aspects of interaction (visual perception, motor control), avoiding higher-level cognitive processes that would have added noise due to inter-user variability in the handling of different kinds of annotations. The document to annotate was derived from the Wikipedia page about text annotation. The contents of annotations come from MacKenzie and Soukoreff's phrase set [18], and are not semantically related to their scope. SHORT annotations consist of 3 words; LONG ones of 7. In the case of variable-length annotations (random length between 5 and 10 words), the system starts by showing the first 3 words, revealing one more word every two seconds until all words are shown. An ellipsis (. . .) is shown to indicate when more words are yet to be revealed.

Participants and Apparatus

Twelve unpaid volunteers (4 female), daily computer users, age 23 to 43 year-old (average 27.3, median 25), served in the experiment. It was conducted on a Microsoft Surface Book 2, equipped with a 13" screen (resolution 3000×2000 pixels) that supports multitouch and pen input. Participants were encouraged to take the device and adjust their hold for comfortable pen and touch interaction.

Procedure

Participants first sign a consent form and fill out a demographic questionnaire. The experiment is then split in two blocks, one per set of techniques (TOUCH-BASED and PEN-BASED). Each block is then divided into sub-blocks, one for each technique (two for TOUCH-BASED, three for PEN-BASED). The presentation order of blocks and sub-blocks is counterbalanced across participants. In each sub-block, the operator briefly introduces the technique using a short video clip, and then lets participants train with the technique.[3] After this training phase, participants complete a series of 12 trials, 2 replications presented in a row for each *Length*× *Scope* condition. The presentation order of these conditions is counterbalanced across participants as well. At the end of each sub-block, participants are asked to rate the technique along five 6-point Likert scales for: physical comfort, cognitive load, enjoyment, efficiency (for SHORT, LONG and EXPANDING annotations separately), and system predictability. At the end of the experiment, participants are further asked to rank the five techniques along these same axes (with the exception of system predictability, which might sound too abstract to participants). We use this final ranking in order to sanity-check that individual scores given after using each technique actually reflect participants' relative ranking of techniques (as individual scores might have been influenced by the presentation order).

Participants are encouraged to verbally share their impressions all along the experiment. In addition to audio recordings, the operator writes down a summary of each participant's feedback. The system logs participants' strategy for annotating (*i.e.*, where they create space), and takes screenshots at the end of each trial. The whole procedure lasts around 75 minutes.

Results

Figure 5 shows what strategies (*where*) participants used to annotate and Figure 6 illustrates examples resulting from these different strategies.[4]

People use the full range of strategies

Overall, participants mostly wrapped their annotations in paragraphs (36.4%) or inserted them between lines (34.2%) (Figure 5-a). Strategies that consist of annotating between words (16.9%) or on the side of a paragraph (12.6%) were used less frequently. They favored the three strategies that embed annotations inside the content over annotations on the paragraph's side. This is consistent with findings reported in [36]. A breakdown of the distribution of strategies per annotation *Length*

[3]This training phase consistently lasted about 5 minutes for all participants, even though they were not given any time limit.

[4]Data and screenshots collected during the study, as well as additional charts, are available at http://ilda.saclay.inria.fr/spaceink.

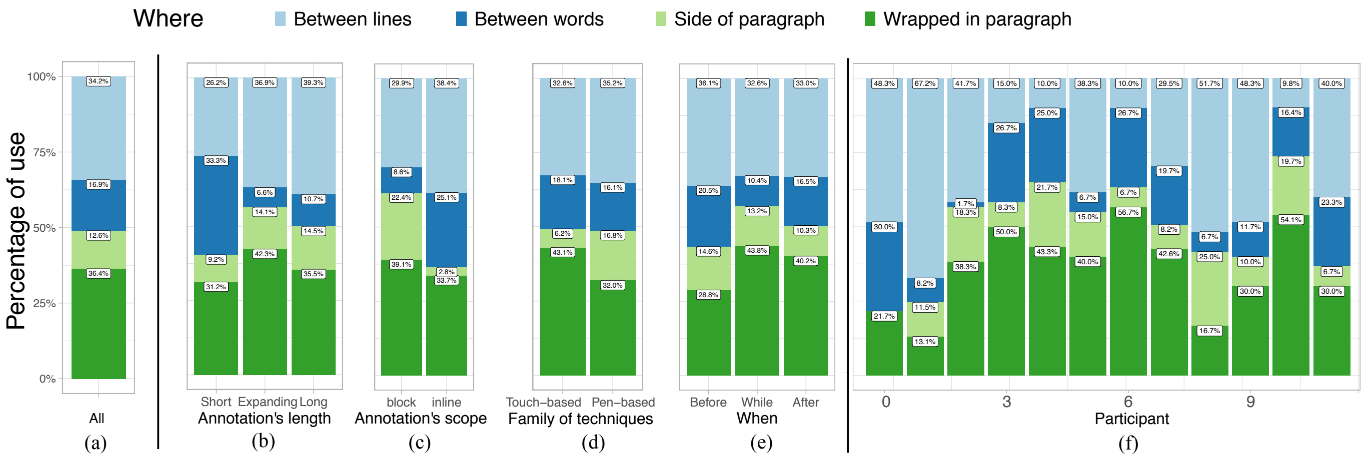


Figure 5. Distribution of participants' space-making strategies (*where*): (a) for all trials, (b) per *Length*, (c) per *Scope*, (d-e) per technique conditions, and (f) per participant.

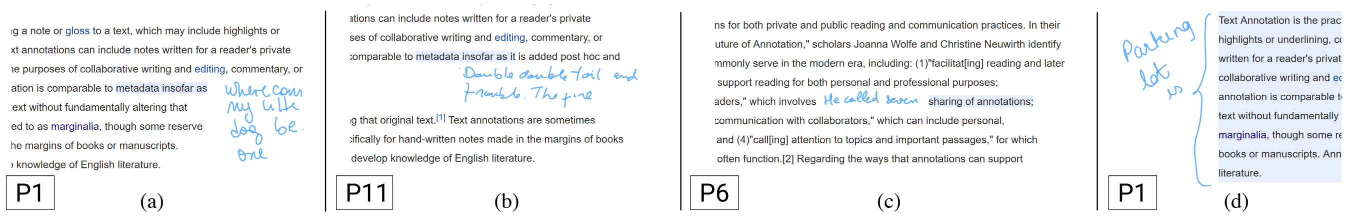


Figure 6. Sample participant annotations illustrating different strategies.

(Figure 5-b) also reveals that participants tend to embed annotations between words when they are SHORT (33.3%), but not when they are EXPANDING (6.6%) or LONG (10.7%). This effect is not very surprising, as our implementation of the *between-words* strategy is limited to a single line. As soon as the whitespace area is taller than a line, it becomes wrapped in paragraph.

What is particularly interesting to constrast with previous findings is that participants did adopt the *between-words* strategy. They used it to closely integrate annotation and content, especially to annotate a portion of text within a paragraph (INLINE). Figure 5-c shows that the *between-words* strategy was often chosen (25.1%) in the INLINE condition. Comparing the two *Scope* conditions, we can see a clear inversion in the distribution between *side-of-paragraph* and *between-words* strategies. This is in line with our expectations about users' will to minimize the distance between an annotation and its anchor (H_2).

The technique itself does not seem to influence the choice of strategy (Figure 5-(d-e)). What is noteworthy, however, is the variability between participants (Figure 5-f). During the experiment, the operator noticed that participants had their personal preferences regarding their strategy, which remained rather consistent all along the experiment. Figure 5-f reports the distribution of strategies per participant, and Figure 6 shows sample trials that illustrate those different strategies. For instance, participant P11 liked using vertical expansion. She used it both when the anchor was the BLOCK and when the anchor was INLINE, by creating some space below the paragraph or below the line containing the highlighted portion of text. In contrast, participant P6 wrapped almost all his annotations in the text, whatever the anchor, without worrying about consequences in terms of content reflowing. Finally, although the annotations' anchor was always highlighted, two participants chose to visually represent their scope (see P1's bracket in Figure 6-d). These observations support the fact that the *strategy for annotating is personal*, and that techniques should offer as much flexibility as possible to accommodate a wide variety of users.

People do not estimate required space accurately

The screenshots taken by the system at the end of each trial reveal that participants were often unable to estimate the length of their handwritten annotations, and ended up annotating over the text with *before* techniques in some cases. This is in line with hypothesis H_1 but, surprisingly, this happened even in the case of SHORT annotations (Figure 7). The cost of premature commitment and its impact on the perceived usability of *before* techniques is reflected in participants' ratings. They found *before* techniques much less efficient than *while* and *after* techniques. They also often complained about the predefined size of boxes with the PEN-BASED+*before* technique.

Participants' ratings, reported in Figure 8, suggest that TOUCH-BASED techniques were rated slightly higher regarding predictability, as users explicitly define the space for annotating with a pinch gesture. But PEN-BASED techniques were enjoyed much more. Participants found them more comfortable, and easier to use. This suggests a trade-off between predictability and fluidity: while TOUCH-BASED techniques are more predictable, users might favor the fluidity offered by

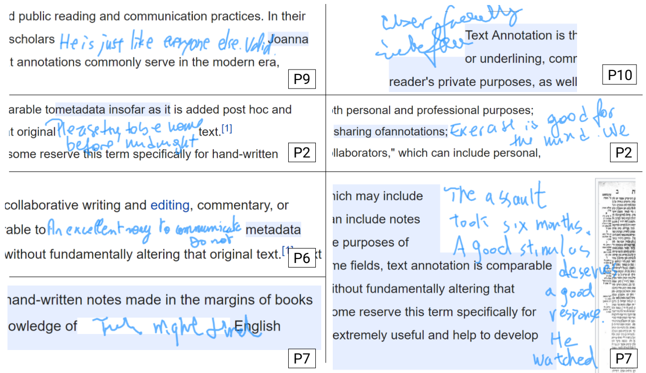


Figure 7. Sample trials with techniques making space *before* inking, in which participants had difficulties anticipating the length of annotations.

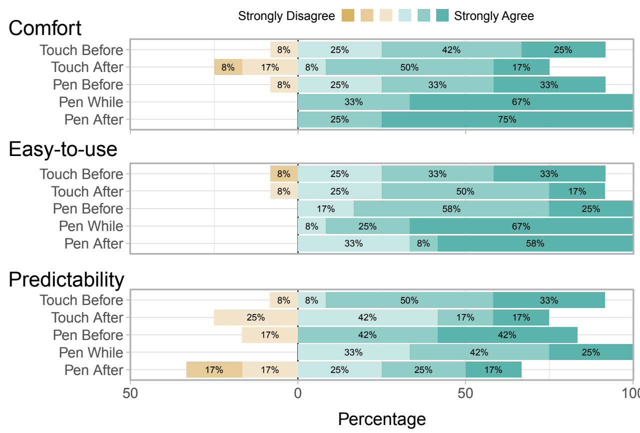


Figure 8. Participants' ratings of the five techniques.

PEN-BASED techniques, which rely on a single input channel for completing all steps in the annotation process. Interestingly, the difference in terms of predictability is quite small. Although there were some cases where participants got surprised by the space-making strategy the system applied when using PEN-BASED techniques – especially so with the *after* version – participants quickly got accustomed to the fact that the choice of strategy was driven by the location of the first pen-down event. We even observed that 10 out of 12 participants tapped on the predefined area before actually annotating when using the PEN-BASED+*before* technique. The initial tap was a means to explicitly tell the system where to make room for the annotation.

Less agency may prove more comfortable

A few participants commented that they did not like writing over the text when using *after* techniques. However, participants' average ratings about comfort do not strongly reflect this: both PEN-BASED+*while* and PEN-BASED+*after* received positive scores. Participants actually rated these two techniques high, as they really enjoyed not having to perform gestures. Interestingly, the fact that the text was dynamically reflowing as they were inking (in the PEN-BASED+*while* condition) did not seem to disturb them. This suggests a trade-off between visual interference and the number of actions: users might find it somewhat disturbing to annotate on top of content, or to see the document getting continuously reflowed as they write, but they find this preferable to performing more interaction steps. However, this observation might not hold in more realistic contexts where users check some elements downstream in the text while annotating. In such cases, the reflowing of content might be a nuisance.

PEN-BASED+*while* and PEN-BASED+*after* seem to be the favored techniques overall, but results remain contrasted. For instance, in terms of predictability, TOUCH-BASED+*after* fares better than PEN-BASED+*after*. This is reflected in participants' comments that praise the quality of all techniques. Five participants actually commented that they would like to have a tool that integrates the best of each approach. For example, P3 said: *"I could clearly see where each technique performed well. If you merge them into one, I would use such a tool."*

IN-CONTEXT ANNOTATIONS IN PRACTICE

The above user study sheds light on different aspects of in-context annotations, that require interleaving space-making interactions and inking. So far, we investigated two techniques in each cell of the design space (pen-only and pen+touch), favoring easy-to-learn and -use techniques based on common gestures. Our main goal was to understand what strategies were effective and in what context. But our general intent is to better support users over the entire document annotation process: when they are making in-context annotations, but when they are inking overlay marks as well. Aiming at streamlining these different types of annotations and considering insights from our study, we identify three key design requirements:

R_1: enable lightweight, seamless transitions between overlay ink and in-context annotations, as users make use of both, simultaneously;

R_2: support several space-making strategies for in-context annotations, as the choice of strategy is both context- and user-dependent;

R_3: make overlay ink reflow with content, as making space for in-context annotations causes such content reflowing.

We describe our rationale for a Web-based annotation environment that meets these requirements. Implemented in Javascript as a Google Chrome extension, this proof-of-concept allows us to test SpaceInk techniques with various devices, annotating arbitrary single-column HTML pages as well as fixed page size formats after conversion (for instance, PDF documents), as demonstrated in the companion video.

Seamless Transitions between Inks (R_1)

During active reading, people interleave marks such as circles or underlines to highlight some portion of content to explicitly *anchor* their thoughts, which they express through the *body* of their annotation [21]. Letting users make both overlay ink marks *and* space for handwritten in-context annotations is thus essential. It alleviates constraints imposed by the document's layout and lets them focus on capturing their thoughts. Figure 9 illustrates such simple cases. However, we must be careful about the cost of mode-switching between the two types of ink.

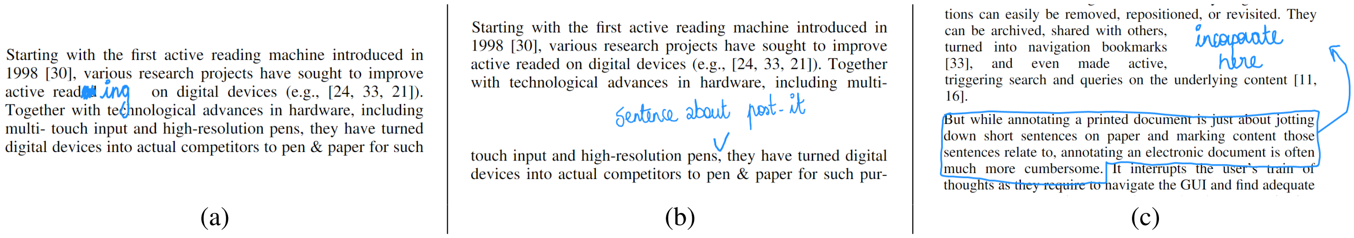


Figure 9. Using both overlay ink and in-context annotations to fix a typo (a), or to suggest higher-level revisions (b-c).

The mode-switch problem is ubiquitous in UI design, and HCI researchers have already faced it in similar contexts, for instance when designing systems where freeform ink coexists with ink-based commands (or stroke shortcuts [2]). The Inferred-Mode protocol [31] addresses the problem by attempting to infer users' intent from the context. By doing so, it removes the need for an explicit delimiter to switch between modes. This is an elegant solution, but because of the high variability in handwriting and annotating, it is unclear in our context which criteria could effectively disambiguate between the two types of inks. The very same variability makes pigtail delimiters [15], as used in TextTearing [36, 37], prone to false activations. For instance, a curly bracket might get confused with a pigtail gesture.

We rather opted for prefix flicks [38]: by default, the pen overlays ink on top of the content, without changing the layout; but when performing a flick gesture, the pen turns into an in-context annotation tool. This delimiter is particularly interesting because it allows users to specify the space-making strategy while switching mode: the strategy is inferred from the relative position of the initial point of contact and direction of the flick gesture, as detailed next. When the flick gesture is initiated in a margin, the subsequent annotation will push the paragraph, widening that margin. When performed over text, *i)* a diagonal flick gesture sets the strategy to *wrapped in paragraph*; *ii)* a horizontal one sets it to *between words*; and *iii)* a vertical one sets it to *between lines*. Pen-based flick gestures avoid breaking the fluidity of interaction, as they only involve one input modality and avoid artificial pauses. The pen automatically reverts back to overlay-ink mode when its location is more than 1cm from the bounding box of the in-context annotation. This solution avoids both bezel swipe gestures, which are usually dedicated to existing commands [29], and pinch gestures, which are often already assigned to navigation actions [30]. An alternative to prefix flicks for mode switching could come from [6]. For example, users could slide their index down the barrel of the pen to touch the surface with their finger while inking with the pen, which could be discriminated from pen-only input.

In addition to the above, a SpaceInk icon, displayed in the bottom left corner, slowly pulses when SpaceInk is activated. Users can rely on this explicit representation to check the current mode, and can tap the icon to force a mode switch. They can also hold their finger still on the icon, using it as a spring-loaded control [16] to change the current mode temporarily.

Combining Strategies for Making Space (R_2)

Observations from our study can inform the design of input techniques that make space for in-context annotations:

G_1: rely, as far as possible, on pen input only;

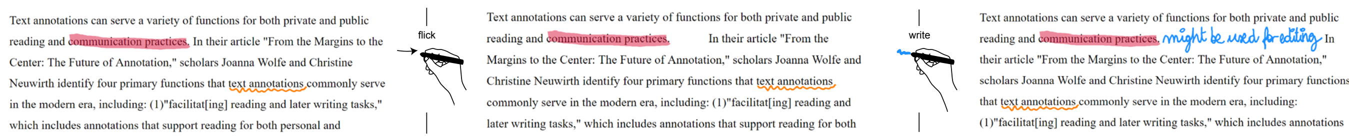
G_2: avoid premature commitment;

G_3: enable users to easily switch between different space-making strategies.

We address G_1 and G_2 by combining prefix flicks to specify the space-making strategy (described earlier) with techniques that adjust the white-space area *while* inking (Figure 3, middle row). The created white space incrementally expands to accommodate the annotation, as the user writes it. We also enable users to make more space on demand. Selecting an annotation activates it, enabling users to adjust the underlying white space using a pinch gesture, or manipulating the handles of its resizing box. That box is transient, progressively fading out when not interacted with, and instantaneously disappearing as soon as users start annotating again. Giving users control over the white space area's dimensions is important. For instance, it lets them prevent dynamic content from reflowing if they would rather not have it move while they are annotating. It also lets them create some white space for, *e.g.*, sketching, and then freely draw strokes anywhere in that space, since those strokes do not have to remain inside the 1cm-distance threshold introduced earlier.

We address G_3 by adding TOUCH-BASED-*after* to the suite of techniques available in the environment. At any moment, users can grab an existing in-context annotation with a finger and move it. The associated space-making strategy gets automatically updated based on the finger's location (in the margin, between lines, on the first word of a line, anywhere else on a line). When the strategy is updated, white space around the annotation gets cropped, minimizing wasted space. This also provides users with a way to quickly optimize space *a posteriori* when the extra white space is deemed unnecessary.

In addition to addressing the above guidelines, the system lets users turn in-context annotations into overlaid ink and *vice versa*. Annotation bounding boxes (which appear when tapping on them) are decorated with the SpaceInk icon. Tapping that icon toggles between states (overlaid or in-context).



(a) (b) (c)

Figure 10. In-context annotations coexist with overlay ink. (a) A user highlights and underlines some content (overlaid ink). (b) She performs a horizontal flick gesture to add some white space between two words. (c) She inserts an in-context annotation. White space automatically expands to accommodate the annotation, causing the content to reflow. All ink marks overlaid on content remain spatially-aligned with their scope.

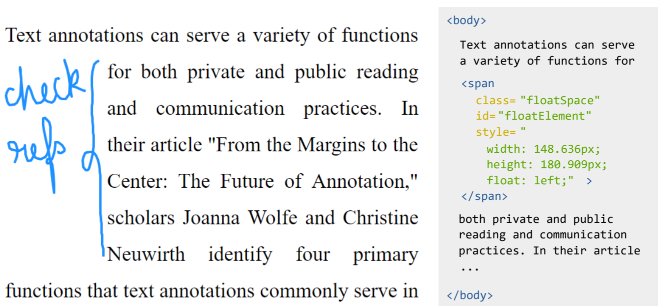


Figure 11. An annotation's spatial scope is encoded as a <span> element in the DOM of the Web page.

Finally, users can restore the document's original layout at any time by performing a hand swipe (*i.e.*, a swipe gesture using three or more fingers). The document's outer margins grow wider, and all in-context annotations get moved there, the document's content reflowing back to its original layout. Performing the same gesture again puts all annotations back in context, with all user-created white-space areas restored.

In summary, our prototype is designed to support a workflow in which users fluidly interweave both overlay marks and in-context annotations. As shown in Figure 10, users only require a pen, which they use both to ink and to make room for annotations. The latter action is triggered by flick gestures, the system automatically creating some white space and expanding it according to the specific gesture made. Users can also explicitly take control on-demand and adjust space precisely, either *before* or *after* the annotation has been written.

Aligning Ink(s) and Content (R_3)

As already mentioned, prior work has studied solutions to keep overlay marks consistent with their scope when the document gets reflowed [3, 11]. In our case, this is of primary importance as in-context annotations can themselves cause the content to reflow. Figure 10 illustrates this on a simple example.

To ensure that in-context annotations preserve their position relative to the content, and that overlay ink remains spatially-aligned with its scope, our proof-of-concept implementation relies on a dual-layer UI canvas. Interaction with, and rendering of, annotations are handled in a transparent layer on top of the document using Paper.js (`http://paperjs.org`). White space for annotations is then inserted in the content by adding `<span>` elements of the appropriate size to the HTML page's DOM (Figure 11). These `<span>` element are inserted as children of the closest-ancestor block (`<div>`, `<p>`, *etc.*), right after the word that is closest to the annotation's starting point.

Dummy `<span>` elements also get inserted for overlay ink. Their size is null, as their purpose is not to push content, but rather to act as spatial anchors for the overlaid ink marks. As the `<span>` nodes get reflowed along with all other inline CSS elements in the block, ensuring that overlay marks remain consistently aligned with their scope is straightforward. It can be achieved just by listening for DOM reflow events and requesting the upper layer to re-render annotations in the right place. This approach works in most cases, and could be extended with techniques described in [3, 11] to handle more advanced cases where overlay marks should be stretched or split.

FUTURE WORK

One limitation of our approach lies in the technique for positioning annotations relative to the document's content: it assumes that this content does not change. Reconciling annotations with content that does change, as is often the case with Web pages, can be a difficult problem [25]. Additional studies in the spirit of the one by Brush *et al.* [4] are needed to design more elaborate repositioning strategies. Such strategies could prevent, *e.g.*, the orphaning of annotations, which is perceived as an important issue by users [5].

Another question worth further investigation is the potential adverse impact of content reflowing on users' mental map of the document. When performing active reading on paper, people sometimes use annotations to build a spatial representation of the document [24]. If reading the document in a non-linear way, in-context annotations might interfere with this mental process, as they could be changing the document's layout frequently. At the same time, people are increasingly used to switching between different devices that render documents differently depending on factors such as, *e.g.*, screen size. It is thus actually unclear whether they build such spatial maps with digital content. Nevertheless, studying this type of annotations in higher-level tasks than the ones we considered in our study would help understand what are the benefits and drawbacks of content reflowing in terms of cognition.

Finally, we believe that an approach complementary to the one explored in SpaceInk is also worth investigating. Elaborating upon what was initiated with overlay ink in [3, 11], in-context annotations could be transformed dynamically to make them fit inside the available white space while optimizing spatial proximity with their scope. Techniques could include simple affine transforms applied to vector-based ink, morphing annotations (treated as textures) into arbitrary shapes, and reflowing handwritten text, for instance turning a long line into several shorter ones.

REFERENCES

[1] Maneesh Agrawala and Michael Shilman. 2005. DIZI: A Digital Ink Zooming Interface for Document Annotation. In *Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction (INTERACT'05)*. Springer-Verlag, 69–79. DOI: http://dx.doi.org/10.1007/11555261_9

[2] Caroline Appert and Shumin Zhai. 2009. Using Strokes As Command Shortcuts: Cognitive Benefits and Toolkit Support. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, 2289–2298. DOI: http://dx.doi.org/10.1145/1518701.1519052

[3] David Bargeron and Tomer Moscovich. 2003. Reflowing Digital Ink Annotations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, 385–393. DOI: http://dx.doi.org/10.1145/642611.642678

[4] A. J. Bernheim Brush, David Bargeron, Anoop Gupta, and J. J. Cadiz. 2001. Robust Annotation Positioning in Digital Documents. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM, 285–292. DOI: http://dx.doi.org/10.1145/365024.365117

[5] J. J. Cadiz, Anop Gupta, and Jonathan Grudin. 2000. Using Web Annotations for Asynchronous Collaboration Around Documents. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW '00)*. ACM, 309–318. DOI: http://dx.doi.org/10.1145/358916.359002

[6] Drini Cami, Fabrice Matulic, Richard G. Calland, Brian Vogel, and Daniel Vogel. 2018. Unimanual Pen+Touch Input Using Variations of Precision Grip Postures. In *Proceedings of the 31st Symposium on User Interface Software and Technology (UIST '18)*. ACM, 825–837. DOI:http://dx.doi.org/10.1145/3242587.3242652

[7] Bay-Wei Chang, Jock D. Mackinlay, Polle T. Zellweger, and Takeo Igarashi. 1998. A Negotiation Architecture for Fluid Documents. In *Proceedings of the Symposium on User Interface Software and Technology (UIST '98)*. ACM, 123–132. DOI: http://dx.doi.org/10.1145/288392.288585

[8] M. A. Chatti, T. Sodhi, M. Specht, R. Klamma, and R. Klemke. 2006. u-Annotate: An Application for User-Driven Freeform Digital Ink Annotation of E-Learning Content. In *IEEE International Conference on Advanced Learning Technologies (ICALT'06)*. 1039–1043. DOI: http://dx.doi.org/10.1109/ICALT.2006.1652624

[9] Andy Cockburn, Carl Gutwin, and Jason Alexander. 2006. Faster Document Navigation with Space-filling Thumbnails. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, 1–10. DOI: http://dx.doi.org/10.1145/1124772.1124774

[10] Niklas Elmqvist, Nathalie Henry, Yann Riche, and Jean-Daniel Fekete. 2008. Melange: Space Folding for Multi-focus Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, 1333–1342. DOI: http://dx.doi.org/10.1145/1357054.1357263

[11] Gene Golovchinsky and Laurent Denoue. 2002. Moving Markup: Repositioning Freeform Annotations. In *Proceedings of the Symposium on User Interface Software and Technology (UIST '02)*. ACM, 21–30. DOI: http://dx.doi.org/10.1145/571985.571989

[12] Gene Golovchinsky, Morgan N. Price, and Bill N. Schilit. 1999. From Reading to Retrieval: Freeform Ink Annotations As Queries. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*. ACM, 19–25. DOI: http://dx.doi.org/10.1145/312624.312637

[13] Thomas R. G. Green and Marian Petre. 1996. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages & Computing* 7, 2 (1996), 131–174. DOI: http://dx.doi.org/10.1006/jvlc.1996.0009

[14] Gary Hardock, Gordon Kurtenbach, and William Buxton. 1993. A Marking Based Interface for Collaborative Writing. In *Proceedings of the 6th Symposium on User Interface Software and Technology (UIST '93)*. ACM, 259–266. DOI: http://dx.doi.org/10.1145/168642.168669

[15] Ken Hinckley, Patrick Baudisch, Gonzalo Ramos, and Francois Guimbretiere. 2005. Design and Analysis of Delimiters for Selection-action Pen Gesture Phrases in Scriboli. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. ACM, 451–460. DOI: http://dx.doi.org/10.1145/1054972.1055035

[16] Ken Hinckley, Francois Guimbretiere, Patrick Baudisch, Raman Sarin, Maneesh Agrawala, and Ed Cutrell. 2006. The Springboard: Multiple Modes in One Spring-loaded Control. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, 181–190. DOI: http://dx.doi.org/10.1145/1124772.1124801

[17] Ken Hinckley, Ken Hinckley, Shengdong Zhao, Raman Sarin, Patrick Baudisch, Edward Cutrell, Michael Shilman, and Desney Tan. 2007. InkSeine: In Situ Search for Active Note Taking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, 251–260. DOI: http://dx.doi.org/10.1145/1240624.1240666

[18] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, 754–755. DOI: http://dx.doi.org/10.1145/765891.765971

[19] Catherine C. Marshall. 1997. Annotation: From Paper Books to the Digital Library. In *Proceedings of the*

Second ACM International Conference on Digital Libraries (DL '97). ACM, 131–140. DOI: http://dx.doi.org/10.1145/263690.263806

[20] Catherine C. Marshall. 1998. Toward an Ecology of Hypertext Annotation. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia (HYPERTEXT '98)*. ACM, 40–49. DOI: http://dx.doi.org/10.1145/276627.276632

[21] Catherine C. Marshall. 2009. *Reading and Writing the Electronic Book*. Morgan and Claypool Publishers.

[22] Fabrice Matulic and Moira C. Norrie. 2012. Supporting Active Reading on Pen and Touch-operated Tabletops. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. ACM, 612–619. DOI:http://dx.doi.org/10.1145/2254556.2254669

[23] M. R. Morris, A. J. B. Brush, and B. R. Meyers. 2007. Reading Revisited: Evaluating the Usability of Digital Display Surfaces for Active Reading Tasks. In *IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07)*. 79–86. DOI:http://dx.doi.org/10.1109/TABLETOP.2007.12

[24] Kenton O'Hara and Abigail Sellen. 1997. A Comparison of Reading Paper and On-line Documents. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '97)*. ACM, 335–342. DOI: http://dx.doi.org/10.1145/258549.258787

[25] Beryl Plimmer, Samuel Hsiao-Heng Chang, Meghavi Doshi, Laura Laycock, and Nilanthi Seneviratne. 2010. iAnnotate: Exploring Multi-user Ink Annotation in Web Browsers. In *Proceedings of the Australasian Conference on User Interface - Volume 106 (AUIC '10)*. Australian Computer Society, Inc., 52–60. http://dl.acm.org/citation.cfm?id=1862280.1862289

[26] Sriram Ramachandran and Ramanujan Kashi. 2003. An Architecture for Ink Annotations on Web Documents. In *Proceedings of the Int. Conference on Document Analysis and Recognition (ICDAR '03)*. IEEE, 256–260. DOI:http://dx.doi.org/10.1109/ICDAR.2003.1227669

[27] Yann Riche, Nathalie Henry Riche, Ken Hinckley, Sheri Panabaker, Sarah Fuelling, and Sarah Williams. 2017. As We May Ink?: Learning from Everyday Analog Pen Use to Improve Digital Ink Experiences. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, 3241–3253. DOI: http://dx.doi.org/10.1145/3025453.3025716

[28] Hugo Romat, Nathalie Henry Riche, Ken Hinckley, Bongshin Lee, Caroline Appert, Emmanuel Pietriga, and Christopher Collins. 2019. ActiveInk: (Th)Inking with Data. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, Article 42, 13 pages. DOI: http://dx.doi.org/10.1145/3290605.3300272

[29] Volker Roth and Thea Turner. 2009. Bezel Swipe: Conflict-free Scrolling and Multiple Selection on Mobile Touch Screen Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, 1523–1526. DOI: http://dx.doi.org/10.1145/1518701.1518933

[30] Vít Rusnák, Caroline Appert, Olivier Chapuis, and Emmanuel Pietriga. 2018. Designing Coherent Gesture Sets for Multi-scale Navigation on Tabletops. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, Article 142, 12 pages. DOI:http://dx.doi.org/10.1145/3173574.3173716

[31] Eric Saund and Edward Lank. 2003. Stylus Input and Editing Without Prior Selection of Mode. In *Proceedings of the Symposium on User Interface Software and Technology (UIST '03)*. ACM, 213–216. DOI:http://dx.doi.org/10.1145/964696.964720

[32] Bill N. Schilit, Gene Golovchinsky, and Morgan N. Price. 1998. Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '98)*. ACM Press/Addison-Wesley Publishing Co., 249–256. DOI: http://dx.doi.org/10.1145/274644.274680

[33] Hirohito Shibata, Kentaro Takano, and Kengo Omura. 2014. Comparison of Paper and Computer Displays in Reading Including Frequent Movement Between Pages. In *Proceedings of the Australian Computer-Human Interaction Conference (OzCHI '14)*. ACM, 549–558. DOI:http://dx.doi.org/10.1145/2686612.2686700

[34] Craig Tashman, Cristiano Ghersi, Stephen Dukker, and Dalas Verdugo. 2010. LiquidText. https://www.liquidtext.net/. (2010). Accessed: 2019-03-21.

[35] Craig S. Tashman and W. Keith Edwards. 2011. LiquidText: A Flexible, Multitouch Environment to Support Active Reading. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, 3285–3294. DOI: http://dx.doi.org/10.1145/1978942.1979430

[36] Dongwook Yoon, Nicholas Chen, and François Guimbretière. 2013. TextTearing: Opening White Space for Digital Ink Annotation. In *Proceedings of the Symposium on User Interface Software and Technology (UIST '13)*. ACM, 107–112. DOI: http://dx.doi.org/10.1145/2501988.2502036

[37] Dongwook Yoon, Nicholas Chen, François Guimbretière, and Abigail Sellen. 2014. RichReview: Blending Ink, Speech, and Gesture to Support Collaborative Document Review. In *Proceedings of the Symposium on User Interface Software and Technology (UIST '14)*. ACM, 481–490. DOI: http://dx.doi.org/10.1145/2642918.2647390

[38] Robert Zeleznik and Timothy Miller. 2006. Fluid Inking: Augmenting the Medium of Free-form Inking with Gestures. In *Proceedings of Graphics Interface 2006 (GI '06)*. Canadian Information Processing Society, 155–162. http://dl.acm.org/citation.cfm?id=1143079.1143105