Eyes-Free Touch Command Support for Pen-Based Digital Whiteboards via Handheld Devices

Fabrice Matulic
Interactive Media Lab,
Technische Universität Dresden
fabrice.matulic@tu-dresden.de

Maria Husmann **Seraiah Walter** **Moira C. Norrie**
Department of Computer Science,
ETH Zurich
husmann@inf.ethz.ch, walteser@student.ethz.ch, norrie@inf.ethz.ch

ABSTRACT

Digital whiteboards that only sense pen input are limited in their interactive capabilities. One way to artificially add touch support is through personal mobile devices, which people carry with them. This work investigates how smartphones can be used as portable quick-access toolboxes held by the non-dominant hand to provide assistive touch commands for pen-driven whiteboard tasks. We developed two interface designs, one based on a classic remote with standard GUI controls and another optimised for eyes-free operation to eliminate gaze shifts between the two devices. In a controlled evaluation based on an established mode-switching study protocol, we compare the two phone interfaces and a baseline technique consisting of a pen-triggered popup menu on the whiteboard. Our results show a superior efficiency of the phone UIs over the popup. The eyes-free UI only partially performed better than the classic interface at the subtask level after subtracting the costs of errors.

Author Keywords

Interactive whiteboards; mode switching; eyes-free interface; bimanual pen and touch coordination.

ACM Classification Keywords

H.5.2 [User Interfaces]: Input devices and strategies, Interaction styles.

INTRODUCTION

Whiteboards are popular tools for a wide range of activities including brainstorming, collaborative sketching, classroom work and personal note-taking. Interactive whiteboards offer the affordances and convenience of regular whiteboards and augment them with tools from the digital world to enhance their capabilities. Depending on the underlying hardware and software technologies, those capacities can vary tremendously. Modern products such as the SMART Board support touch in addition to pen input. This allows the non-dominant hand (NDH) to fulfil assistive tasks, such as manipulating the workspace, operating widgets or performing complementary actions. If the system is able to differentiate the two modalities, they can be assigned distinct roles, typically inking for the pen and command input for touch [10]. However, not all whiteboards have that capability. In many cases, either touch is indistinguishable from pen input or the device only supports pen interaction, e.g. whiteboard systems based on external IR digitisers or on Anoto technology [6]. With such setups, interaction mechanisms have to be devised to enable command triggering while preserving the natural experience of direct inking with the pen. The two most common solutions for tackling that problem are widgets, i.e. specific areas of the interface within which the pen behaves as a different tool (e.g. a menu or a toolbar), and gestures, i.e. pre-defined strokes that the system recognises as commands instead of regular ink. A gesture can also be used to summon a widget, from which operations can then be selected and executed (e.g. marking menus). Both of these methods have drawbacks, as either valuable UI space is consumed or users have to learn special strokes and potentially cope with misinterpretation errors. Some pens are equipped with barrel buttons that enable different mode activations but these also come with their own issues [5, 14].

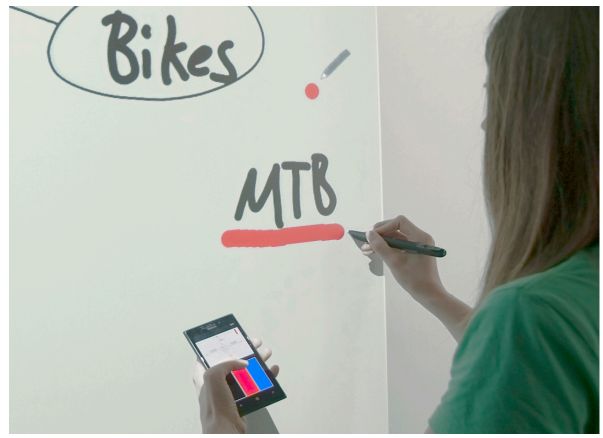


Figure 1. A user drawing with her dominant hand on a digital whiteboard, while operating controls on a smartphone with her non-dominant hand.

Beyond the surface of the whiteboard, practitioners have sought to explore extended interactive spaces created by the introduction of other devices, especially personal mobile devices such as tablets and smartphones. Examples include

ITS '15, November 15 - 18, 2015, Funchal, Portugal

ACM 978-1-4503-3899-8/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2817721.2817728

the use of handhelds to transfer content to and from the whiteboard [6, 20], to remotely point at or control objects on the display [2, 17] and even as physical objects used for tangible input [8, 21]. In most of the cases studied so far, the mobile device is also an active centre of (visual) attention, even if temporary, and users have to shift their focus between the board and their devices to manipulate content, locate a control or perform a gesture. This is generally not a problem in scenarios with little cross-device interaction, but for more intense whiteboard activities, smoother flows with lighter context switches might be desirable.

In this paper, we study the use of personal smartphones as low-attention touch-based assistive devices to efficiently support fluid pen-driven whiteboard tasks. We created two phone interfaces integrating a core set of whiteboard tools activated by the thumb of the user's NDH holding the device. One design resembles a classic remote or palette with standard GUI controls, while the other is optimised for eyes-free use and includes large buttons as well as a gesture pad. The latter is aimed at reducing the demands on visual attention by eliminating the to-and-fro gaze shifts between the mobile device and the whiteboard. In a set of controlled experiments involving various patterns of alternating atomic inking and command subtasks, we compare the efficiency, accuracy and qualitative merits of those interfaces as well as a baseline pen-only technique with a pen-triggered popup menu on the whiteboard.

RELATED WORK

There is a sizeable body of work related to interactive whiteboards in the literature. From the first Liveboard prototypes developed at Xerox PARC in the early 90's [5] to modern displays incorporating multiple sensing capabilities, a broad range of systems and dedicated applications have been created and studied. As motivated in the introduction, our main scope of interest concerns the integration of personal mobile devices into the interactive space of basic pen-based digital whiteboards, not as independent working and content-sharing devices, but mainly as controls facilitating the operation of the whiteboard. This concept falls under the category of multimachine user interfaces (MMUIs), a term coined by Myers et al., who extensively explored the interoperation of handhelds with desktop computers within the Pebbles project [17]. Around the same period, Rekimoto investigated the combination of mobile devices with whiteboards both for content-sharing and UI control [20]. While the first version of his system involved weighty tablets that required the DH to interact with both the whiteboard and the handheld, a second version based on PDAs made activations by the NDH possible via the PDA's physical buttons.

Tool palettes held in the NDH in a painter-like fashion need not be computing devices. For their NiCE discussion room, Haller et al. utilise acrylic and magnet palettes with Anoto patterns to provide a mobile and more direct tool-selection alternative to their digital pie menus triggered by pen long presses on the whiteboard [6]. Those physical palettes have the advantages and disadvantages of inert tangibles: they are convenient and always "on", but they only allow one fixed design and do not provide dynamic feedback.

Pushing further the spatial cross-device relationship are techniques in which phones are used as tangibles that physically contact the display surface to trigger particular actions [8, 21]. Interaction with the whiteboard via mobile devices can of course also be done in a contactless fashion, not only by the main whiteboard operator, but also by other participants in the background. For instance, Code Space is a system in which members of the audience can remotely point at and drag objects on a display as well as execute gestural commands [2]. A similar effort for TV sets proposes a set of gestures to remotely control the screen [7]. Those gestures range from easy flicking motions for navigation to more complex shape interactions to trigger commands.

As mentioned above, with the exception of when it is used as a pointer, the handheld engaged in a cross-device interaction with the whiteboard or large display necessitates the user's visual focus. The user, therefore, has to divide their attention between the multiple devices involved, which can have a notable performance and cognitive cost, if context switches are frequent [19]. In their third experiment on bimanual pen and touch coordination, Matulic and Norrie show that blind pen mode switching using NDH postures can reduce that cost, provided mode errors are few [15]. Eyes-free interfaces have been considered by other authors as well, but more as alternatives to operate the devices themselves in the absence of visibility rather than as a means to reduce attention shifting. Thus, most prototypes developed so far rely on I/O techniques that require the user's full attention such as audio feedback [4, 13, 25] and compound gestures [22]. Other methods based on simpler gestural motions such as jerks and tilts [1, 11] have also been proposed and while such device gestures could potentially be applied in bimanual MMUI contexts, we think they are less common than regular touch-based manipulations and so do not consider them at this stage of our work.

Looking at prior experimental evaluations that are relevant to our approach, Li et al. compared techniques to switch from pen inking to gesture mode on tablets and found that a physical button pressed by the NDH performs best [14]. In [9], Hinckley et al. propose and test various spring-loaded techniques allowing function selections to remain enabled for repeated actions. They report that such quasimodal activations are both effective and also preferred by people over status quo alternatives. While they reveal relevant results, those two studies utilised only single actuators to trigger or maintain one mode switch and were conducted on tablets. These conditions do not apply in our case as we focus on a combination of multiple modes and commands in a multi-device environment. However, the methodology used in those experiments could be suitably adapted for our purpose, especially [9], hence we based our protocol on it, as described in the Experiment section. Finally, in [3], Bragdon et al. experimentally determine that gestures are superior to soft buttons under conditions of high environmental distraction. Based on eye-gaze data, they also note that gestures are much more suited to eyes-free usage compared to soft buttons, which users have to look at to properly operate. However, their UI comprising 12 buttons laid out at the top of the phone screen was not designed with eyes-free use in mind and so that result is not generalisable. Wang et al., for instance, show that for a 3×4 grid of touch areas covering the whole screen, people achieve an eyes-free targeting accuracy of about 90% [23]. For lower densities, such as 3×2 or 2×2 layouts, the reported accuracy is even close to 99% and 100%. Those findings suggest that soft buttons can also be effective eyes-free controls.

APPROACH

We first explain the rationale that motivated our approach and formulate our hypotheses before describing the techniques used in our experiment.

Rationale

Our starting point is a digital whiteboard based on Anoto technology and front-projection, similar to that of the NiCE room [6]. Such a setup allows high-precision pen sensing and wide interactive surfaces, but on the other hand, it does not support touch input, which limits its interactive capabilities. Hence, one of our goals is, to some extent, to compensate for the lack of touch input on such whiteboard systems through the agency of smartphones that people always carry with them. However, we also think that personal handheld devices can make sense for pen and touch displays as well, for instance, as personalised or private palettes or to reduce occlusions caused by widgets on the board.

A second aspect concerns the efficiency of pen tasks on the whiteboard. As we have seen, there are many examples showing how new functionality is created to manipulate and exchange content in novel ways, but performance aspects are often not considered, both in terms of task productivity and cognitive demand.

A further factor that differs from performance studies made on tablets and tabletops (such as mode-switching) is the arm fatigue caused by protracted in-air interaction with a vertical display. The active use of a device held up in the NDH while writing with the pen might even amplify that adverse effect. These considerations motivated us to explore the potential of smartphones essentially as quick-access and low-attention assistive toolboxes that can be held in a casual and relaxed manner rather than as other primary devices that also tax the user's attention and stamina. This requirement suggests the need for the device to also be usable without having to look at it, that is, it should be operable in an eyes-free manner.

In order to frame our target scope more precisely, we enumerate the practical and theoretical benefits of smartphone-aided pen interaction on whiteboards using visible and eyes-free interfaces, as we see them:

General advantages of handheld-based controls:

1. Provides separate touch support for whiteboards that only support pen input
2. Allows menus and widgets to be relocated to the handheld, thereby reducing clutter on the whiteboard
3. Enables personalised controls
4. Presumably reduces mode-switching and command activation time

Additional advantages of eyes-free controls:

5. In meeting situations, the presenter's gaze is always on the board or the audience, not on their personal device
6. Enables private controls (if UI is hidden)
7. Presumably reduces mode-switching and command activation time even more
8. Presumably reduces cognitive and physical effort (but probably only minimally and after much training)

Analysing the items in the above list and relating the identified theoretical advantages of eyes-free interfaces to Yi et al.'s taxonomy of motivations for eyes-free interaction [24], we notice that: 1 is system-specific. 2, 3 and 6 depend on the target application (with 6 falling in the M5 category of Yi et al.'s classification). 5 (an M4 case) is conditioned by the type of whiteboard activity considered. This leaves 4, 7 and 8, the "presumable" performance aspects, (with 7 and 8 falling in the M10 category) that we are interested in and that we seek to validate or invalidate experimentally.

To summarise: within our considered usage scope of eyes-free interfaces, we contextually cover but do not study some of the social benefits identified by Yi et al. (M4 and M5) and we closely examine perceived improvements to personal efficiency (M10). In that respect, our focus is similar to that of Matulic and Norrie's third experiment [15] but with different environments: a vertical display and separate surfaces/devices for the two hands, where the NDH also holds a device instead of a single tabletop with which both hands freely interact (and are able to rest on the surface). We further do not want to restrict ourselves to purely quasimodal pen mode switching, as we would like to also include simple modal activations as well as general independent commands that do not involve the pen.

Another important aspect that Matulic and Norrie emphasise in their report is the importance of visual feedback to maintain correct awareness of the currently active mode or selected properties. While our study does not involve multiple users interacting at the same time, whiteboards are often used collaboratively and so we should not ignore those situations when designing awareness mechanisms. Therefore, large visuals such as surface-wide coloured borders are likely not appropriate in our case.

With regard to the operations to support, we select a set of core whiteboard functions commonly used for note-taking and brainstorming activities: pen colours, a stroke style change (thickness), delete and undo/redo. This choice reflects our main driving concern to find a satisfactory balance between simplicity (to keep eyes-free manipulations tractable) and integration of standard tools typically available in whiteboard applications. Furthermore, it covers properties that are activated and modified in different ways and thus, we believe, form an ecologically valid mix of tools in our context: in a reduced palette, such as the one we are considering, colour and style options are assigned discrete values that remain active once selected (normal mode), because we assume that in most cases those changes are meant to be permanent, until explicitly changed again. Delete can also be implemented as a system-maintained mode, but it is usually only needed on a temporary basis, so it is a good candidate for quasimodal activation in an eyes-free interface, i.e. the user would have to continuously maintain a pose, such as a finger pressed, to keep the mode active (as in [16]). Finally, undo/redo are independent and instantaneous commands that do not affect the pen status.

We now describe the designs of our phone interfaces based on that set of requirements.

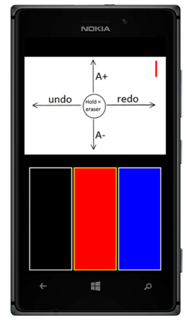
Phone UI Designs

Eyes-Free Interface

The main challenge of designing an efficient eyes-free interface is to create controls that can be quickly located and triggered with a minimal number of mistakes. In the case of a mobile device held and operated by the NDH, the controls have to be easily accessible by the user's thumb. Furthermore, in our particular case, the user's attention is also consumed by the pen-holding DH, not to mention by the main whiteboard activity and potentially other users. We therefore aim for simple controls that can be easily and quickly triggered rather than more sophisticated interactions such as path-based gestures, which are reportedly difficult to execute with one hand [3].

Our functions to be mapped can be divided into three categories: functions with a directional or ordinal quality, i.e. that convey a sense of movement (undo/redo) or corresponding to a value that can be increased or decreased (stroke thickness), a function associated with a categorical value (pen colour) and a mode (delete). The first group can be conceptually mapped to the two screen axes, which suggests as suitable UI component a virtual D-pad or directional gestures such as flicks. The delete mode can be assigned to an isolated modifier button (which needs to be kept pressed to maintain the non-default mode active) and the colours to regular soft buttons activated by single taps.

Following these considerations, we created the design shown in Figure 2 comprising two separate areas. At the top is a gesture pad on which flicking gestures mapped to undo/redo (horizontal flicks) and stroke thickness changes can be executed (vertical flicks). The pad also functions as the modifier button for the quasimodal activation of the delete mode (thumb pressed down). When the delete mode is engaged, users can strike out strokes with the pen to remove them from the screen.



Physical dimensions on Nokia Lumia 925	
Gesture pad	5.8 × 4 cm
Colour button	1.7 × 4.2 cm

Figure 2. The Eyes-Free phone UI with the physical dimensions of its components

In the lower part of the UI are a group of three mutually exclusive buttons corresponding to the three colours that we (arbitrarily) select: black, red and blue. We chose this number of divisions and layout with components filling the entire screen space based on the most efficient tile arrangements for eyes-free use reported by Wang et al. in [23]. Their analysis shows that there is a higher decrease in eyes-free targeting accuracy in the vertical direction than in the horizontal one. According to their experimental data, a division into two parts along the Y-axis and into two or three sections along the X-axis yields accuracies of, or very close to, 100%. We further note that the physical dimensions of the UI components when rendered on the screen of a Nokia Lumia 925 (see table in Figure 2) exceed by far Parhi et al.'s recommended size of 9.6mm for eyes-free targets [18].

With regard to user awareness of selected properties and active mode, we integrated visual feedback both on the phone screen and on the whiteboard. On the phone, it is located in the top right corner (Figure 2). On the whiteboard, we have to deal with the problem touched upon earlier of providing user-specific feedback on a large surface potentially used by several people. Our solution consists of an icon placed near the last drawn stroke (see Figure 1). This method is satisfactory as long as the user's strokes remain relatively close to each other, but is obviously less helpful if they move to other areas of the board between two strokes. We did not include any tracking that would allow icons to "follow" the user's movements but this feature could easily be added to our system.

The feedback icons give a representation of the currently selected stroke width and colour in inking mode and show an eraser in delete mode. A brief notification is also shown when an undo or a redo is triggered. Users can therefore remain aware of their actions and the selected stroke style only by monitoring the whiteboard. For the colours in particular, the buttons are such that they are triggered as soon as a dragged touch contact enters its active area. Thus, users can blindly slide their thumb across the lower area of the screen while checking the visual feedback on the board to keep track of changes. For delete mode toggling, haptic feedback is also integrated in addition to the visual aid: the phone shortly vibrates when the thumb is pressed down on the pad to indicate that the mode has been engaged.

Classic "Visible" Interface

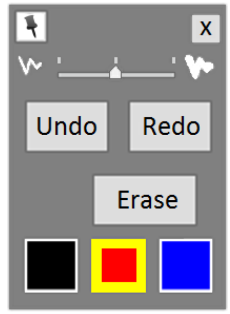
Physical dimensions on Nokia Lumia 925	
Slider (interactive area)	4 × 0.4 cm
Undo/Redo buttons	2 × 2 cm
Erase button	5.3 × 2.4 cm
Colour button	1.7 × 1.7 cm

Figure 3. The classic "visible" phone UI with the physical dimensions of its components

While there is naturally nothing precluding non-blind usage of this first design, we decided to create another interface without the eyes-free concern in mind following classic UI patterns and controls. The reasons for this choice are threefold: first, we wanted to compare design strategies based on different usage intents. If the experimental analysis revealed no significant differences, then we could make a stronger statement on the equivalence of two different UI designs rather than only of operational or contextual factors. Secondly, a condition where users would forcibly be required to look at an interface using an artificial mechanism would be unrealistic and therefore have little ecological validity. The third reason is to counteract possible learning effects that would likely skew the results of our within-subjects study, which also includes a whiteboard-only technique (see next section). We however also had to take into account the opposite, i.e. possible adverse learning effects caused by a totally different UI layout with different button arrangements, which would likely have confused participants when testing the other phone technique.

Our classic "visible" interface, shown in Figure 3, is our attempt at satisfying the aforementioned concerns while still following standard design practices for such kinds of UIs. Since, in effect, we were creating a simple toolbar or palette, we assigned one widget in a dedicated screen space to a given function. Thus, our classic design has four vertical layers of components with individual soft buttons for colour, erase and undo/redo, and a slider for stroke thickness. Compared to the eyes-free design, there are more widgets with slightly larger gaps between them so that blind usage is more difficult and therefore less of an option for users. Their arrangement and position are however similar to the eyes-free interface in order not to confuse users and their sizes are still sufficiently large to allow precise targeting. All buttons follow a standard tap-once-to-trigger behaviour. Hence, the erase button needs to be tapped to engage and disengage the delete mode and the colour buttons do not activate when dragging in from an external position.

Whiteboard Menu



Physical dimensions on whiteboard	
Pin button	2.2 × 2.2 cm
Slider (interact. area)	7.7 × 1.3 cm
Undo/Redo buttons	5.6 × 3.4 cm
Erase button	7 × 3.4 cm
Colour button	3.4 × 3.4 cm

Figure 4. The whiteboard popup menu with the physical dimensions of its components

The whiteboard menu is our baseline "phoneless" technique. It consists of a popup menu with a component layout resembling that of the classic phone UI (Figure 4). It is summoned by a pen long-press on the whiteboard, which is a standard technique to trigger an action differing from the default pointer function (here inking). As hold timeout we used the system value of 1000 ms, which is slightly more conservative than the delay used in [12]. The menu appears so that its centre is located at the tip of the pen to minimise subsequent arm movements to select items.

To reduce UI clutter and possible hindrances, by default, the menu disappears as soon as the pen taps outside it. This behaviour is however clearly not suitable in the case of frequent menu calls and so to avoid repetitious pen dwelling, we endowed the menu with a pin feature allowing it to remain visible. The menu can be moved to other locations by dragging its background or by calling it again with a long press, regardless of its pinned status.

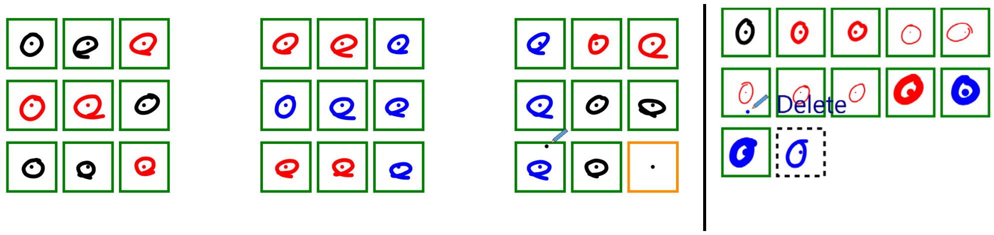
Enabling hardware and software

As stated above, our interactive whiteboard is based on an apparatus similar to that used in the NiCE discussion room [6]. It features a large front-projected vertical surface with an Anoto pattern for the detection of the optical pen. The dimensions and resolution of our display are respectively 360 × 112.5 cm and 2560 × 800 pixels. Our smartphone is a Nokia Lumia 925 with a 4.5" screen measuring 5.9 cm × 9.7 cm and with a resolution of 768 × 1280 pixels.

The software was implemented in C# on Windows 8 and Windows Phone 8.0. As communication protocol between the phone and the server we used TCP over Wi-Fi.

EXPERIMENT

To compare the usability, performance and subjective value of our phone techniques, we conducted a controlled within-subjects study in our lab. We used the popup menu introduced above as baseline condition. While there are, of course, several other valid pen-only techniques that we could have included in the comparison, e.g. marking menus or path gestures [12], we selected only this one pen technique to keep the experiment tractable in terms of time and effort. Pilot studies indeed revealed that arm fatigue was an issue that had to be taken into account and so we decided to



Physical dimensions of the stimuli on the whiteboard	
Box edge length	16.8 cm
Gaps adjacent boxes, X and Y axes	X=1.6 cm Y=3.2 cm
Gap between blocks	29 cm

a) *Distributed* Scene 1 b) *Local* Scene 4

Figure 5. Examples of the two types of scenes used in the experiment

compare only three techniques to keep the physical demand at acceptable levels.

For the experiment, the screen of the eyes-free UI was made completely black, in order to ensure that participants operated the device in the intended manner.

Design

Our study design was modelled after the paradigm developed by Hinckley et al. for their evaluation of spring-loaded controls [9]. It consists of a series of dot-circling tasks (imitating the act of writing) with different patterns of pen modes that users have to match. The costs of those mode switches are then calculated by subtracting the mean completion times of baseline tasks from that of the switched tasks. Our experiment followed a similar methodology but with notable differences that we point out in the following description.

Tasks

As in the Springboard study, our tasks were composed of blocks of atomic actions (subtasks) to be executed from left to right (but in our case also from top to bottom, since we use multiple rows), with basic inking subtasks involving circling a dot stimulus contained in a box using a single stroke. The colour and thickness of the dot indicated the style that had to be matched by the input ink. Participants used the tested technique to select the correct stroke colour and thickness before marking the box. To remain in manageable territory and facilitate equality checks in the analysis, we limited the range of the stroke thickness to three discrete values: small, medium and large.

Where we departed from the Springboard study is that we also included command stimuli to be acted upon. Hence, in between circling subtasks participants were required to delete strokes and issue undo/redo operations, where the latter concerned both inking and deleting actions. Another important difference is that our commands actually performed their actions on the UI instead of just leaving a trail representing the selected tool without doing anything. When a command subtask, say delete, was correctly executed, both the stroke and the box were erased so that the next subtask stimulus appeared at the same location as the one that was just removed. This mimicked a real correcting action, in which one stroke is replaced by another. Commands were enabled at all times, even when not prompted by the stimulus. This means that participants could make mistakes and also correct them using the tools at their disposal. If, for instance, a participant inadvertently triggered two stroke undos instead of one, they were required to re-input the mistakenly removed circle before they could move on. For that, they had the possibility to either redo the action or redraw the stroke. We believe that this kind of task model, where commands and errors have actual consequences, as in a real whiteboard activity, increases the ecological validity of our study design. Furthermore, by factoring those corrective manipulations into the completion times, we can determine realistic costs of making mistakes compared to simply playing a sound and registering an error.

A further condition that differs from tablet environments and whose influence we wanted to investigate is display real estate. When writing or sketching on whiteboards people often make use of the large space available to them, for example, if they want to spatially group different categories of notes. We attempted to model that aspect through tasks with several subtask blocks separated by wide gaps (Figure 5a). We hypothesised that this task type would be particularly unfavourable to the popup menu, as participants would either have to move it along or reach far away for it.

Protocol

We created a protocol composed of five series of subtask blocks which we called scenes. These included scenes of the two types described above: *Distributed* scenes containing three separated 3×3 blocks (Figure 5a) and *Local* scenes containing one compact 3×5 block (Figure 5b). *Distributed* scenes focused on the whiteboard width aspect and therefore only consisted of circle subtasks, where the stroke style had to be changed once for every segment of three subtasks. *Local* scenes concentrated on switching and command input and therefore required the user to change stroke styles and execute commands with increasing frequency.

The five scenes with their respective numbers and types of subtasks were the following:

1. *Distributed*: 27 circles with 9 colour changes
2. *Distributed*: 27 circles with 9 thickness changes
3. *Local*: 19 circles with 3 thickness changes, 1 delete, 6 undos and 3 redos
4. *Local*: 18 circles with 3 thickness and 3 colour changes, 6 deletes and 3 undos
5. *Local*: 21 circles with 5 thickness and 7 colour changes, 9 deletes, 6 undos and 3 redos

This yields a total of 112 circles with 20 thickness and 19 colour changes, 16 deletes, 15 undos and 6 redos per technique. Thus, with 3 techniques and 12 participants, a grand total of 4032 circles with 720 thickness and 684 colours changes, 576 deletes, 216 undos and 108 redos were performed for the main tasks (i.e. not counting training).

To counteract learning effects, we created three protocols following the above pattern with subtasks appearing at different indexes in the sequences but with the same type order, i.e. if a delete subtask followed a redo subtask in protocol A, that order was maintained in protocols B and C. These protocols were then used for each technique to be tested, where the order of techniques was permuted between participants.

Each scene started and ended with a circle subtask in the default style, which was medium stroke width and black colour. The completion times for those subtasks were not included in the analysis. Before each scene, a dialog specifying the type of task that the participant was about to carry out was shown. The dialog was to be tapped with the pen to start the scene. At the start of the first training task, a window picturing a mock 3×5 block of subtasks was displayed on the whiteboard. Participants were then given the opportunity to adjust the position and particularly the height of the window to their liking so that they could comfortably reach the three rows.

Participants were instructed to execute the tasks in a natural way, in particular, the circles were to be drawn in a pace and manner matching that of their handwriting.

Participants

We recruited 12 volunteers, 9 males and 3 females with a mean age of 29.7 years old (SD=5.38) for our study. One of the participants was left-handed. In the pre-study questionnaire, all participants indicated that they owned a smartphone. Four people stated that they had some experience with digital pens, whereas all others had none at all.

A session with a participant consisted of the following steps: introduction, filling-in the pre-study questionnaire, height adjustment on the whiteboard, training and main task execution using the 3 techniques, post-study questionnaire and interview. Participants were given the possibility to take breaks between technique trials and between scenes.

Training phases, in which participants familiarised themselves with each technique included example scenes of both types. For the eyes-free UI, participants trained with the visible controls first and then with the black screen, as used during the main tasks.

RESULTS

A full session with a participant took about one hour.

Repeated measures ANOVA and post-hoc pairwise comparisons were performed on scene completion times, subtask completions times, mode switching times and number of extra actions. The latter represent actions, which participants performed in addition to the minimum operations required to complete the given subtask, in other words, actions due to hesitations, errors and their corrections.

Scene Completion Times

Figure 6 shows the mean scene completion times. We conducted a 3×5 repeated measures ANOVA on Technique (*Eyes-Free*, *Classic*, *Popup*) by Scene (1 to 5). We obtained main effects for Technique ($F(2, 22) = 53.975$ $p < 0.001$) as well as for Scene (which is obvious since the five scenes are different). Post-hoc comparisons revealed that both *Eyes-Free* (mean time 59.0 s) and *Classic* (59.7 s) were significantly faster than *Popup* (77.5 s) with $p<0.001$. In particular, the phone UIs were more efficient than *Popup* not only in *Distributed* scenes but also in *Local* scenes. To rule out that the *Popup* technique was disadvantaged by the time spent dwelling or dragging to bring up or relocate the menu, we subtracted the time used for those operations from the total time. This resulted in a mean scene completion time of 72.8 s, which is still significantly higher than the times of both phone UIs ($p < 0.001$). Thus, the differences cannot be attributed only to the mechanical effort of having to move or re-call the menu. Other possible factors are increased cognitive effort to locate and position the popup menu or more time spent physically moving to reach the menu instead of repositioning it. We had expected *Eyes-Free* to perform better than *Classic* at least in Scene 5 where switches and commands were frequent, but this was not borne out by the data ($p=1.0$). Thus, it seems that the two types of phone UIs were not significantly different overall in terms of performance. We now deepen our analysis by examining the individual subtasks more closely.

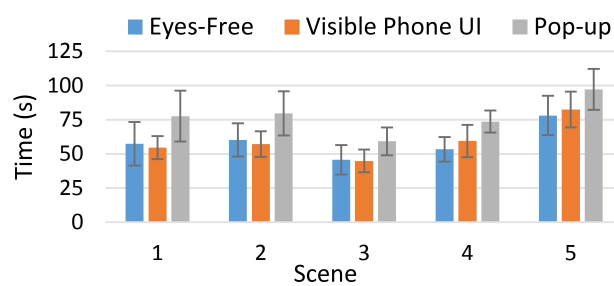


Figure 6. Mean scene completion times.

Subtask Completion Times

We considered our four types of subtasks: Circle, Delete, Undo and Redo. For each type, we discarded the two (three for Circle) best and worst observations from the dataset, in order to minimise outlier effects. This resulted in 3816 observations for Circle, 432 for Delete, 440 for Undo and 172 for Redo. We then calculated the mean task completion time for each task type with the remaining data points.

Repeated measures ANOVA on Technique by Task revealed a significant effect of Task ($F(2, 22)=162.007$, $p<0.001$), Technique ($F(2, 22)=43.431$, $p<0.001$), and Technique*Task ($F(2.069, 22.756)=6.087$, $p=0.0214$) on mean task completion time. A further ANOVA performed on each task type revealed that for all three subtasks, *Eyes-Free* and *Classic* were both faster than *Popup* ($p \leq .013$), but there was no significant difference between the two phone UIs ($p \approx 1$ in all cases). Therefore, we were also not able to detect any significant differences at the subtask level. We continued breaking down the individual actions and next looked at switching costs between subtasks.

Cost of Switching

We first considered circle tasks and calculated the cost of changing the colour and the stroke thickness by subtracting the mean task completion times of consecutive circle tasks requiring no style modification from those requiring one. After finding significant main effects for Technique, Style Change (Colour, Thickness) and Technique*Style Change interaction, we found no significant difference between *Eyes-Free* and *Classic* both for Colour (p=0.584) and for Thickness (p=1.0). When comparing the two types of style switches for *Eyes-Free*, we found that changing Colour (735ms) was significantly faster than Thickness (979 ms) (p=0.004), which is a result that we did not expect. Some of the stroke thickness switches involved changing from small to large and vice versa, which required two flicks on the gesture pad instead of one. To analyse the effect of that factor, we removed those instances from the dataset (leaving 305 data points out of 440 originally) and performed the test again. The result is an increased p value of 0.044, but which is still under the cut-off threshold, albeit only slightly. Hence, it appears that soft buttons were more efficient than flicking gestures as a trigger method.

We then compared the cost of switching from a delete to a circle subtask. An ANOVA on Technique revealed a significant effect ($F(2, 22)=32.843$, $p<.001$) on that cost. Post-hoc comparisons showed that *Eyes-Free* (mean cost 790 ms) was significantly faster than both *Classic* (1507 ms) (p=0.001) and *Popup* (1875 ms) (p<0.001). *Classic* was however not significantly faster than *Popup* (p=0.082). The most likely explanation for the significant difference between the two phone UIs is that reverting to inking mode with *Eyes-Free* only requires releasing the thumb, whereas in the other two conditions the button needs to be located and pressed again. While the delete action itself does not benefit from the quasimode when it is engaged, its disengagement enables rapidly segueing to the subsequent inking action.

Extra Actions

The number of extra actions was calculated by subtracting the minimum number of operations required to fulfil the different subtasks from the number of actions actually performed by participants, i.e. with inclusion of errors and their corrections. The types of errors that we observed were:

- Stroke errors: circle not containing the dot stimulus, not entirely inside the box or drawn with multiple strokes
- Dwelling errors: pen moved while holding to summon the popup, thereby accidentally inking the canvas
- Style errors: circle drawn in an incorrect colour or with the wrong stroke thickness
- Command errors: wrong command executed, e.g. doing an undo instead of changing the stroke size or colour
- Mode errors: pen used in the incorrect mode (erasing instead of inking and vice versa)

Repeated measures ANOVA on Technique revealed a significant effect ($F(1.385, 15.233)=20.568$, $p<0.001$) on the number of extra actions. Post-hoc comparisons confirmed that *Eyes-Free* (Mean=9.7 extra actions per scene) resulted in significantly more extra actions than *Classic* (4.8 extra actions) (p=0.002) and *Popup* (3.6 extra actions) (p=0.008). *Popup* and *Classic* did not exhibit a significant difference (p=0.377). This shows that despite training, participants had not completely mastered the controls of the Eyes-Free UI that they had to operate blindly. Further analysis of the data of *Eyes-Free* revealed that switching thickness in Scene 2 resulted in 54% more extra actions compared to switching colour in Scene 1, indicating that soft buttons could be easier to use than vertical swipe gestures. However, ANOVAs revealed no significant differences ($F(1, 11)=4.121$, $p=0.067$). A comparison of the same two scenes showed roughly the same number of undos and redos, thus, participants appeared to have no problems distinguishing horizontal swipes for undo/redo from vertical swipes for thickness.

Error-Corrected Analysis

Seeing that participants made significantly more mistakes with *Eyes-Free*, we wanted to investigate the possibility that those errors and the extra actions performed to correct them might be the cause of the two phone UIs not exhibiting any significant performance differences. We therefore compared the techniques in ideal theoretical conditions with no errors. For that, we filtered out subtasks with extra actions and repeated the analysis of their completion times using the remaining error-corrected data (3276 circles, 401 deletes, 345 undos and 172 redos). The results were somewhat surprising, as only circle subtasks exhibited a significant difference between the two phone UIs. Specifically, *Eyes-Free* was faster than *Classic* for Circle (p=0.005), but not for Delete (p=0.690), Undo (p=0.149) or Redo (p=0.909). We surmised that this might be due again to the quasimodal delete switch in *Eyes-Free* favouring quick transitions back to inking for the next circle subtask so we performed a new analysis without unmodified circle subtasks occurring immediately after a delete (4 instances per technique) and the difference remained statistically significant, albeit with a higher p-value (p=0.029). We are therefore left to conjecture that this divergence might be due to the different number of data points for each subtask and that possibly Delete, Undo and Redo might also become significantly faster with more data.

We also repeated the procedure for the cost of switching. In comparison to the analysis including errors, the switching costs were reduced on average by 24% for colour (SD=14%), 24% for thickness (SD=19%) and 24% for switching from delete to a circle task for *Eyes-Free* (SD=19%). For *Classic*, the respective reductions were 6% (SD=13%), 14% (SD=11%), and 4% (SD=17%). As *Popup* had the smallest number of extra actions, it naturally had the smallest reduction in the error-corrected case with all values below 5% (SD=13%, 8%, 8% respectively). Despite the larger reductions for *Eyes-Free*, we did not find any new significant effects compared to the previous analysis.

Subjective Preferences

We asked participants to rank each technique on a 5-point Likert scale in terms of perceived efficiency and concentration and effort (from 1="not at all", to 5="very much") for each type of task, *Distributed* and *Local*. A Friedman test showed a statistically significant difference in efficiency for Technique with the *Distributed* tasks ($\chi 2(2)=9.282$, $p=0.010$). Median values were 4.5 for *Eyes-Free*, 4 for *Classic*, and 3 for *Popup*. A Wilcoxon signed-rank test showed a significant effect of higher perceived efficiency of *Eyes-Free* over *Classic* ($Z=-2.496$, $p=0.013$) as well as *Popup* ($Z=-2.448$, $p=0.014$). *Classic* and *Popup* did not exhibit any statistically significant difference ($Z=-2.456$, $p=0.145$). While the higher rating of *Eyes-Free* over *Popup* is justified, it is interesting to notice that participants felt more productive with *Eyes-Free* than with *Classic*, even though the data do not vindicate that impression. For the *Local* tasks, participants seemed to prefer the phone UIs over the Popup as well, with median values of 4.5 for *Eyes-Free*, 4 for *Classic* and 3 for *Popup*, however the differences were not significant ($\chi 2(2)=4.343$, $p=0.114$). In the *Local* condition, the mobility of the UI was likely not as big a factor. The variation in the data was quite large for *Popup*, however. In particular, one participant rated all UIs with a 5. Excluding him from the analysis yielded a significant preference of *Eyes-Free* over *Popup* ($p=0.035$).

With regard to perceived concentration and effort, the obtained median values of 2.5 for *Eyes-Free*, 3 for *Classic*, and 3 for *Popup* for the *Distributed* tasks and 3 for all techniques for the *Local* tasks were not statistically significant. One user commented that *Eyes-Free* required higher concentration, but that the physical effort was lower. Conversely, three participants noted that *Popup* required less concentration but more physical effort. One person said that they would prefer a phone interface in a classroom setting, as it would provide them with more freedom to move about. Another participant stated that "it is better when the attention can be focused on the screen". Yet another remarked that "the pop-up is always in the way".

Participants rated the suitability of each technique for regular whiteboard activities (e.g. brainstorming, in meetings) on a 5-point Likert scale. Their answers are shown in Figure 7. All of the techniques had a median value of 4. All but one participant rated *Eyes-Free* with a 4 or a 5 and did not rate any other technique higher. The one participant who rated it with a 2, said that he preferred *Classic* because he "felt more in control". Three participants orally expressed a preference for *Eyes-Free* over *Classic*, but gave them equal ratings on the questionnaire. Despite being clearly the slowest interface, nine participants gave *Popup* a 4 or a 5. One participant mentioned that it could make sense to use the popup menu in combination with a phone UI and depending on the task, use one or the other.

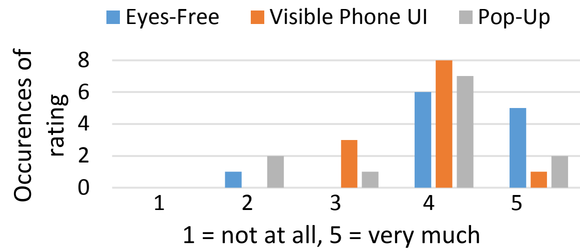


Figure 7. Suitability for general whiteboard usage.

DISCUSSION

The results of the experiment show a clear performance superiority of the phone UIs over the whiteboard-only popup menu in both types of task, which validates the first presumption from our list (item n°4). On the other hand, comparisons of the two phone UIs yielded mixed results. In general, participants made more errors with *Eyes-Free* than with the two other techniques. The error-corrected data suggest those errors are mainly responsible for *Eyes-Free* not performing significantly better than its *Classic* counterpart. Thus, if people are able to master the eyes-free UI they can likely surpass the other techniques, which tend to have lower improvement potential. Consequently, our second hypothesis (item n°7) is only valid in this ideal case.

Our participants did not cope with the eyes-free UI in the same way, as evidenced by the large variance in the data. The gains obtained from blindly manipulating the phone therefore highly depend on the user's skill and experience with the UI. A further observation that we made is that, despite its slightly more crowded design, some participants also occasionally used the classic phone UI in an eyes-free manner. We did not use any eye-tracking monitoring and so are not able to quantify to what extent that behaviour might have had an influence on the results and potentially reduced the performance gap between the two phone interfaces.

At the lower subtask and switching levels, our main findings concern the ability of quasimodes to speed up the transition back to the default mode upon release and the superior effectiveness of soft buttons over vertical flicking gestures to trigger an action. While the first result is logical, we find it more difficult to explain the second. A possible reason might be that the gesture pad was located at the top of the phone screen and the colour buttons at the bottom. It would be interesting to see if switching the positions of those two items has an impact.

With regard to the subjective feedback and ratings given by participants, in particular those concerning cognitive and physical effort, we did not obtain statistically significant results that allow us to make any conclusions. Hence, we cannot validate our last hypothesis (n°8), although we suspected that any possible difference might have been too small to elicit in our conditions.

CONCLUSION AND FUTURE WORK

In this paper, we presented a set of phone-based techniques to assist tasks performed on pen-enabled whiteboards. We introduced two phone interfaces with a set of standard controls, one optimised for eyes-free use and another following a classic remote design. In a controlled experiment, we compared the real and perceived performance of the two phone interfaces and that of a baseline whiteboard-only technique consisting of a pen-triggered popup menu. We established the superior efficiency of the phone UIs over the popup and the theoretical benefits of the eyes-free UI in ideal conditions when no errors are made. We further confirmed the performance benefit of quasimodes for temporary switching and showed that it manifests itself at the moment when the transitory mode is released to return to default mode. Finally, we demonstrated the suitability of soft buttons as eyes-free controls, which even performed better than simple flicking gestures.

Finally, even though our study was less about testing the chosen whiteboard tools themselves than the way they were activated via their controls, our results were obtained using specific UI layouts with a fixed number of functions. The natural next step of this work would therefore be to investigate how those results scale with different designs and with various kinds of whiteboard functions and compare those interfaces in visible and eyes-free conditions.

REFERENCES

1. Baglioni, M., Lecolinet, E. and Guiard, Y. JerkTilts: using accelerometers for eight-choice selection on mobile devices. In *Proc. ICMI 2011*.
2. Bragdon, A., DeLine, R., Hinckley, K. and Morris, M. R. Code space: touch + air gesture hybrid interactions for supporting developer meetings. In *Proc. ITS 2011*.
3. Bragdon, A., Nelson, E., Li, Y. and Hinckley, K. Experimental analysis of touch-screen gesture designs in mobile environments. In *Proc. CHI 2011*.
4. Dicke, C., Wolf, K. and Tal, Y. Foogue: eyes-free interaction for smartphones. In *Proc. MobileHCI 2010*.
5. Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., Tang, J. and Welch, B. Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration. In *Proc. CHI 1992*.
6. Haller, M., Leitner, J., Seifried, T., Wallace, J. R., Scott, S. D., Richter, C., Brandl, P., Gokcezade, A. and Hunter, S. The NiCE Discussion Room: Integrating Paper and Digital Media to Support Co-Located Group Meetings. In *Proc. CHI 2010*.
7. Hao, L., Matei, N. and Yang, L. Gesturemote: interacting with remote displays through touch gestures. In *Proc. AVI 2014*.
8. Hardy, R. and Rukzio, E. Touch & interact: touch-based interaction of mobile phones with displays. In *Proc. MobileHCI 2008*.
9. Hinckley, K., Guimbretiere, F., Baudisch, P., Sarin, R., Agrawala, M. and Cutrell, E. The springboard: multiple modes in one spring-loaded control. In *Proc. CHI 2006*.
10. Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H. and Buxton, B. Pen + touch = new tools. In *Proc. UIST 2010*.
11. Kajastila, R. A. and Lokki, T. A gesture-based and eyes-free control method for mobile devices. In *Proc. CHI EA 2009*.
12. Leitner, J. F., Perteneder, F., Liu, C., Rendl, C. and Haller, M. Kolibri: tiny and fast gestures for large pen-based surfaces. In *Proc. CHI 2013*.
13. Li, K. A., Baudisch, P. and Hinckley, K. Blindsight: eyes-free access to mobile phones. In *Proc. CHI 2008*.
14. Li, Y., Hinckley, K., Guan, Z. and Landay, J. A. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proc. CHI 2005*.
15. Matulic, F. and Norrie, M. Empirical evaluation of uni- and bimodal pen and touch interaction properties on digital tabletops. In *Proc. ITS 2012*.
16. Matulic, F. and Norrie, M. Pen and Touch Gestural Environment for Document Editing on Interactive Tabletops. In *Proc. ITS 2013*.
17. Myers, B. A. Using handhelds and PCs together. *CACM*, *44*, 11 (2001).
18. Parhi, P., Karlson, A. K. and Bederson, B. B. Target size study for one-handed thumb use on small touchscreen devices. In *Proc. MobileHCI 2006*.
19. Rashid, U., Nacenta, M. A. and Quigley, A. The cost of display switching: a comparison of mobile, large display and hybrid UI configurations. In *Proc. AVI 2012*.
20. Rekimoto, J. A multiple device approach for supporting whiteboard-based interactions. In *Proc. CHI 1998*.
21. Schmidt, D., Seifert, J., Rukzio, E. and Gellersen, H. A cross-device interaction style for mobiles and surfaces. In *Proc. DIS 2012*.
22. Serrano, M., Lecolinet, E. and Guiard, Y. Bezel-Tap gestures: quick activation of commands from sleep mode on tablets. In *Proc. CHI 2013*.
23. Wang, Y., Yu, C., Liu, J. and Shi, Y. Understanding performance of eyes-free, absolute position control on touchable mobile phones. In *Proc. MobileHCI 2013*.
24. Yi, B., Cao, X., Fjeld, M. and Zhao, S. Exploring user motivations for eyes-free interaction on mobile devices. In *Proc. CHI 2012*.
25. Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R. and Baudisch, P. Earpod: eyes-free menu selection using touch input and reactive audio feedback. In *Proc. CHI 2007*.