A Comparison of Techniques for In-place Toolbars

Andre Doucette, Carl Gutwin, and Regan L. Mandryk

University of Saskatchewan

ABSTRACT

Selections are often carried out using toolbars that are located far away from the location of the cursor. To reduce the time to make these selections, researchers have proposed in-place toolbars such as Toolglasses or popup palettes. Even though in-place toolbars have been known for a long time, there are factors influencing their performance that have not been investigated. To explore the subtleties of different designs for in-place toolbars, we implemented and compared three approaches: warping the cursor to the toolbar, having the toolbar pop up over the cursor, and showing the toolbar on the trackpad itself to allow direct touch. Our study showed that all three new techniques were faster than traditional static toolbars, but also uncovered important differences between the three in-place versions. Participants spent significantly less time in the direct-touch trackpad, and warping the cursor's location caused a time-consuming attentional shift. These results provide a better understanding of how small changes to in-place toolbar techniques can affect performance.

KEYWORDS: In-place toolbars, targeting time, direct input.

INDEX TERMS: H5.2 User Interfaces: Interaction styles

1   INTRODUCTION

Tasks in most computer applications require that users make multiple selections during the task, and one of the most common mechanisms for making these selections is through the use of toolbars. Palettes in drawing applications, navigation bars in web browsers, and Microsoft Office's Ribbon (office.microsoft.com) are all examples of the prominence of toolbars in current applications.

One of the problems with current toolbars is that they perform poorly for *mid-task* selections, which require that users leave their current location on the screen, move to the toolbar, select an item, and finally return to the original location to continue working. Because toolbars are often located at the top or sides of the screen, this means that toolbars are often the farthest targets for users to reach, making these selections time consuming.

Several techniques have been proposed for reducing the targeting time for selection. Most of these solutions involve in-place toolbars – palettes that are shown closer to the user's current cursor location; for example, Toolglasses [2], right-click context menus, and popup palettes. Despite this research activity, however, we still know very little about the details of in-place techniques. For example, there is little research to indicate whether it is better to have a toolbar pop up directly over the cursor (which could occlude the user's work), to warp the cursor to a distant toolbar (which requires a contextual shift), or to display the toolbar on an external input device (which allows

andre.doucette@usask.ca, {gutwin, regan}@cs.usask.ca
Department of Computer Science, University of Saskatchewan,
110 Science Place, Saskatoon, SK., S7N 5C9, Canada.

direct-touch techniques to be used, but requires an even larger context shift). Although these differences are subtle, they can have large impacts on effort and frustration. In addition, new hardware designs for laptops that include visual trackpads [10] are highlighting the relevance of these design issues.

To explore the subtle differences in the design space of mid-task toolbar techniques, we implemented three in-place toolbars that follow different design approaches. Warp Cursor automatically moves the cursor to a toolbar at the edge of the screen when a key is pressed, and moves it back again upon selection; Shadow Cursor pops up a toolbar at the current cursor location, and returns the cursor to its original location upon selection; and Visual Trackpad uses the display surface of the trackpad itself to show the toolbar, allowing the user to select items with direct finger input. All three techniques allow users to make a selection from a toolbar without having to move their cursor from their mid-task location.

We compared these three designs (and traditional toolbars) in an experiment where participants carried out repeated mid-task toolbar selections. As expected, all of the in-place approaches were substantially faster than Static Toolbar interaction. However, there were also differences between the three new techniques, and these differences were larger than expected. Our results show that Shadow Cursor and Visual Trackpad were the fastest, and people using Visual Trackpad spent significantly less time interacting with the toolbar. Warp Cursor did not perform as well as the other techniques, suggesting that large automatic movements of the cursor may make shifting attention more difficult.

2   RELATED WORK

2.1   In-place toolbar selections

It is well established that moving the toolbar closer to the cursor improves selection performance [2,5,6]. Toolglasses and magic lenses [2], marking menus [6], right-click context menus, and control menus are a few examples of techniques which take advantage of the idea of in-place selections. The menu-based solutions pop up at the cursor location, but all require the user to move their cursor back to the starting location after selection. Toolglasses, in contrast, can be brought to the cursor's location. A study comparing Toolglasses to Flowmenus and control menus [5] showed that Toolglasses have benefits for drawing tasks because the selections happen at the cursor location; however, these techniques require the use of a secondary pointing device operated by the non-dominant hand. One in-place technique of particular interest is the Hotbox [7]. The Hotbox combines many types of GUI elements and interaction techniques into one in-place pop-up element. It was implemented for Maya, a graphic design application, and initial results show that users preferred to use the Hotbox over the traditional interface with multiple menu bars.

2.2   Shifting attention

Toolbars require that the user shift visual context from their current work to the toolbar. Plumlee and Ware describe a general performance model for focus switching within an application [8] - the time required to complete a task is:

$$T = S + \sum_{1}^{v} (B_i + D_i)$$

where T is the total time, S is the setup time, v is the number of visits to each target, B is the time spent between targets, and D is the time spent at the targets. The techniques discussed in this paper replace cursor movement between targets (B) with faster eye movement. They also reduce D, as discussed below.

Zhai et al. [11] presented the MAGIC pointing technique that reduces D described above by warping the cursor to the location of the user's gaze on screen. They tried two techniques, one where the cursor followed the eye gaze whenever a threshold distance is met and another where the cursor only warped once the threshold was met and the mouse was moved. They found that users preferred the second technique, because moving the cursor constantly may be distracting.

2.3   Direct and indirect input

Direct input is possible when the input space and the display space are the same, and is by design absolute. Indirect input, in contrast, exists when input and display spaces are distinct, and can be relative or absolute. Buxton [3] states that types of input are good for different tasks, and that direct and indirect input are both good for different tasks: others have shown that touch input can be faster than other methods if targets are not too small [9].

Recent techniques allow hybrids of absolute and relative pointing [4]. The Visual Trackpad described below is a similar technique, except that we use an explicit mode switch (in [4], their implicit switch was defined by the hover height of the stylus).

3   OUR THREE TECHNIQUES

We implemented three techniques (Figure 1) for in-place, mid-task toolbar selections. All techniques used the same visual toolbar, but differed in the location of the toolbar and whether direct or indirect input was used to select an item. Also, our three techniques allow users to make a toolbar selection without moving their 'real' cursor point from its location in the task.
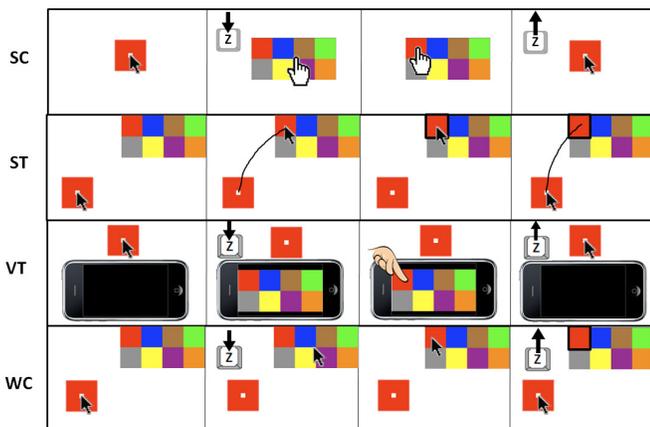

Figure 1. Mid-task selection: Shadow Cursor (SC), Static Toolbar (ST), Visual Trackpad (VT), and Warp Cursor (WC).

3.1   Shadow Cursor (SC)

In Shadow Cursor, the mode switch key activates the toolbar and displays it over the cursor's location. A secondary cursor with a different visual representation (a hand pointer) appears in the toolbar to make selections. Once an item is selected, the toolbar is hidden, and the system cursor re-appears at its original location.

3.2   Warp Cursor (WC)

Warp Cursor is a modification of a static toolbar that can be accessed quickly by pressing the mode switch key. This causes the cursor to warp to the center of the toolbar. Once a selection is made, the cursor returns to its mid-task location. Warp Cursor replaces the cursor movement across the screen with an attentional shift across the screen.

3.3   Visual Trackpad (VT)

Visual Trackpad is unique in that it is both a direct and indirect input device: it is a trackpad in indirect mode, and a touch screen in direct mode. Since VT overlays two types of input modalities on the same input surface, it can combine the benefits of both.

A mode switch key turns on the trackpad's screen and the toolbar is displayed. Users tap on a toolbar item, the toolbar disappears, and Visual Trackpad returns to being an indirect input device. Although Visual Trackpad requires users to move their attention to an off-screen toolbar, touching a toolbar item directly is known to be fast [9], and may capitalize on muscle memory.

3.4   Design considerations

3.4.1   Why these techniques

The location of in-place toolbar selection is one factor that has not been investigated. Our techniques use three locations for in-place toolbar techniques. From nearest to farthest, we have SC displayed above the cursor, WC displayed on the same screen, and VT displayed on an external display. Because of these three levels of location, each technique requires its own method to access the toolbar (we chose an explicit mode switch key).

The SC technique has much in common with the Hotbox technique. Kurtenbach et al. [7] state a motivation for developing the Hotbox was to clear the workspace of GUI widgets when they were not in use. SC tackles this problem in the same way as the Hotbox: only display the toolbar with an explicit mode switch.

WC expands on the MAGIC pointing technique, but replaces the gaze targeting with an explicit mode switch. Although this implies that WC only warps to one location, different hot keys could map to different toolbars, allowing users to warp to the various GUI elements on the screen. In Plumlee's and Ware's [8] model, this represents a reduction of B.

VT is a technology that is already available [10], so we are interested in what types of techniques are afforded to designers by this technology. In this case, we chose in-place toolbars, although this is only one of the techniques possible with this technology.

3.4.2   The trackpad and other input devices

Because the VT technique requires the use of a trackpad, we designed all of the techniques to use a trackpad as an input device. This limitation was a design choice in order to reduce the noise between the techniques. We believe that SC and WC would work with any relative input device that controls an on-screen cursor (including a mouse and a relative tablet). VT requires a device that affords both direct and relative input, but implementing VT on an absolute tablet with a display (such as a Tablet PC) would result in a technique that functions similar to SC. In this sense, SC is an implementation of VT using indirect input.

4   COMPARISON STUDY

We designed an experiment to consider three questions:
- Do the three designs perform differently?
- What are the attentional costs for the three approaches?
- Is there any advantage to allowing direct touch?

4.1 Apparatus, Tasks, and Participants

The four toolbar techniques (three in-place techniques plus Static Toolbar) were implemented on a Macintosh PC using a custom-programmed iPod Touch as a trackpad device. The iPod 'trackpad' used WiFi to communicate with the PC; the study used a custom Objective-C application to present the experimental tasks and record performance data. The trackpad was fixed in place in front of a wireless keyboard, and the study system was displayed on a 22-inch LCD monitor with 1600x1200 resolution. The application window was 1280x700 pixels with a 280-pixel sidebar in which the static toolbar appeared. The on-screen toolbars were the same physical size as the toolbar on the trackpad (3.5 inches wide).

The experimental tasks each involved a mid-task selection from a toolbar (see Figure 1). The task itself was a colour-filling activity: a 7x7-pixel white square target was surrounded by a 50x50-pixel box; the box was filled with one of the eight colors in the toolbar. Tasks had four parts: a moderately-difficult targeting action, a toolbar selection, a second targeting action, and a final click in the target. Participants were required to move their cursor into the square target, after which they would hear a sound and see the target turn black. They then selected the toolbar item corresponding to the box's color. Once a toolbar item was selected, participants were required to click on the square target again to fill the target with the colour. Note that only the Static Toolbar technique required users to target twice; the other techniques warped the cursor back to the target, allowing users to complete the task simply by clicking again after the selection.

Sixteen right-handed participants (aged 20-34) were recruited for the study. All were experienced computer users (> 2 hr/day), and all but one were experienced with a trackpad input device.

4.2 Procedure and Design

Each technique was demonstrated for the participant, and they then completed eight training trials with each technique. In the testing session, participants completed tasks at 64 target locations, with color selections chosen randomly. Target locations and color selections were standardized across the four conditions. Performance data was collected by the study system. After the experiment, participants rated and ranked the techniques.

The study used a single-factor repeated measures design, with the factor *toolbar technique* having four levels: Shadow Cursor (SC), Static Toolbar (ST), Visual Trackpad (VT), and Warp Cursor (WC). We collected three time measures: completion time for the toolbar-selection task, time spent in the toolbar, and pre-toolbar time before the toolbar appeared (or the cursor warped for WC). We also collected data about how far participants moved in each phase of the activity. The order of techniques was balanced using a Latin square, and the study took 45 minutes to complete.

5 RESULTS

We used RM-ANOVAs to test the effects of technique on completion time, toolbar time, pre-toolbar time, and path length. Pairwise comparisons were tested using Fisher's LSD method at $\alpha=0.05$. Outlier trials (completion time more than three standard deviations above the mean) were removed from further analyses (36 trials, 0.9%), while trials with selection errors were included.

5.1 Performance results

Completion time. There was a significant main effect of toolbar technique on completion time ($F_{3,45}=61.7$, $p\approx0.000$); see Figure 2. Pairwise comparisons showed that Static Toolbar was significantly slower than all of the in-place toolbars and that SC was significantly faster than WC.

Toolbar time. There was a significant main effect on time spent in the toolbar, defined as the time from the mode switch key being depressed to the correct toolbar selection ($F_{2,30}=22.1$, $p\approx0.000$); see Figure 3. Pairwise comparisons showed that our three techniques were significantly different; less time was spent in the VT toolbar than SC or WC; less time was spent in SC than in WC.

Pre-toolbar time. There was a significant main effect on time spent between the first target acquisition and the mode switch ($F_{2,30}=4.07$, $p=0.027$). Note that ST was not included in this analysis as it does not have a mode switch. Pairwise comparisons showed that less time was spent in SC than VT (Figure 3).

Path length. There was a significant main effect of toolbar technique on the Euclidean distance travelled by the cursor or the finger, between the mode switch and where users made the colour selection ($F_{2,30}=149.2$, $p\approx0.000$). ST was not included in this analysis as it does not have a mode switch. Pairwise comparisons showed that people moved less with SC than VT or WC (see Figure 4). Note also that Figure 4 shows the path length in pixels, which is the measure of distance for on-screen cursors, but distance for VT depends on DPI. The on-screen toolbars were the same absolute size as the VT toolbar (3.5 inches), with on-screen toolbars 280x140 pixels and VT toolbars 480x240 (the max size on the trackpad). The distance presented for VT in Figure 4 has been scaled to match the scale of the on-screen techniques.
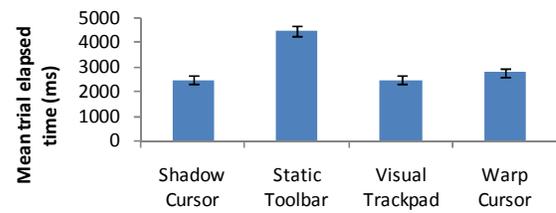

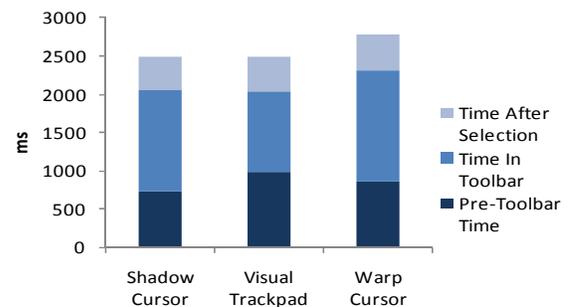Figure 2. Mean SE completion time (ms).

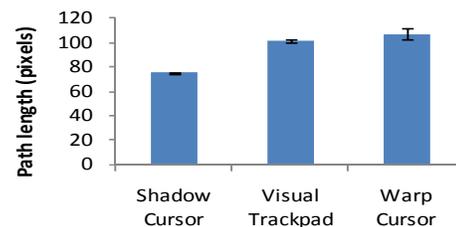Figure 3. Breakdown of mean times spent during trial

Figure 4. Mean SE path length in pixels.

5.2 Subjective results

Mean rankings of the techniques are shown in Figure 5. A Friedman test ($\chi^2_3=12.3$, $p=.006$) revealed significant differences

between the techniques and Wilcoxon signed ranks tests showed that ST was ranked lower than SC (Z=2.1, p=.033) or VT (Z=2.7, p=.007), but not WC (Z=1.8, p=.080). There were no differences between the three in-place techniques.
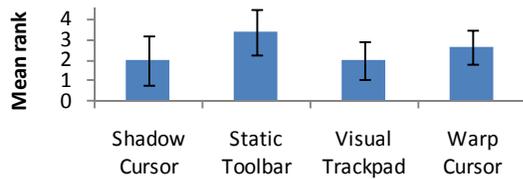

Figure 5. Mean ranks of techniques (1-4, lower is better).

6   DISCUSSION

All of the in-place techniques were faster than the Static Toolbar, but surprisingly, Shadow Cursor was not significantly faster than Visual Trackpad. We had expected that the off-screen attentional shift (VT) would be slower than both the in-place (SC) and on-screen (WC) attentional shifts. Although we believe this to be true (we plan to test this hypothesis using an eye tracker), the results suggest that the direct input afforded by Visual Trackpad makes up for the extra time users needed to shift their attention.

Figure 3 shows the proportion of time spent in different parts of the targeting action for each technique. Users spent more time preparing to do the mode switch (pre-toolbar time) in VT, even though this action is identical for the three in-place techniques. This may be because users are shifting their attention to the trackpad while making the mode switch. Once in the toolbar (time in toolbar), VT is significantly faster, suggesting that direct input is faster than indirect input for toolbar selections.

There is a surprising effect of attentional shift on path length with SC and WC (Figure 4). They both require the same path length, but warping the cursor to a distant toolbar (WC) caused users to click farther into the coloured target than when the toolbar was displayed in-place (SC). This may be because the sudden movement of the cursor caused users to jerk the cursor farther away, whereas the in-place popup encouraged users to click just inside the colour, or because the larger eye movements for WC encouraged clicking farther inside the target. This result, coupled with WC's slower performance, suggests that cursor warping may not be a good design choice. Another surprising effect is that the path length for VT was not significantly different from the path length of WC (Figure 4). On average, WC should have the optimal path length, since the cursor starts in the center of the toolbar. Since the path length depends on the location of the user's finger when they lifted it off the trackpad, VT's path length is variable. Trackpads are a relative input device, so the location of the finger at the end of the targeting motion could be anywhere on the trackpad; the results in Figure 4, however, suggest that, on average, the user's finger was near the correct toolbar item when the first targeting motion was completed. This suggests that users may have been anticipating which toolbar item they will select, or that most mouse movements happen in the center of the trackpad.

Last, in order to gain pointing advantages, our in-place toolbars must be relatively small; however, this affects all in-place strategies (including techniques such as Toolglasses). There are several design strategies (such as favorites toolbars) that make such an approach feasible.

7   FUTURE RESEARCH

We will continue our explorations in three ways. First, we will carry out eye-tracking studies to determine more details of the attentional shift for the in-place techniques. Second, we will investigate the one main problem with SC – that it occludes the target. Our participants stated that they often forgot which colour they were searching for because SC occluded the target, whereas the other techniques allowed them to shift their eyes back and forth between the target and toolbar. We will investigate whether transparency (the method Hotbox solves this problem) or multiblending [1] could be used to mitigate this problem.

Third, the results for Visual Trackpad are encouraging, and we will continue to investigate what types of interactions are appropriate for this new technology. We are interested in testing the effect of re-mapping hierarchical menus to the Visual Trackpad; these structures have similar problems to toolbars and could potentially benefit from direct input.

8   CONCLUSION

Traditional toolbars perform poorly for mid-task selections because users must move their cursor from their current location to the toolbar and back again, while in-place toolbars reduce the distance to the toolbar. To explore the subtleties of in-place designs, we implemented three in-place toolbar techniques that allow users to make mid-task selections without losing their cursor location. The two main differences between these techniques were the attentional shift required to make the selection (in-place, on-screen, and off-screen) and the interaction to select the item (indirect or direct input). We found that the in-place/indirect technique (Shadow Cursor) and the off-screen/direct technique (Visual Trackpad) were fastest, that on-screen toolbars are expensive with cursor warping, and that direct input for the off-screen technique made up for the additional time required to visually acquire the off-screen toolbar. These results show the value of widgets that minimize cursor movement, and suggest new opportunities for hybrid direct/indirect devices.

REFERENCES

[1]   Baudisch, P. and Gutwin, C. Multiblending: displaying overlapping windows simultaneously without the drawbacks of alpha blending. *Proc CHI* '04 367-374.

[2]   Bier, E., Stone, M., Pier, K., Buxton, W., DeRose, T., Toolglass and magic lenses: the see-through interface. *SIGGRAPH '93*, 73-80.

[3]   Buxton, W., Multi-Touch Systems that I have Known and Loved, www.billbuxton.com/multitouchOverview.html. (September 2009).

[4]   Forlines, C., Vogel, D., and Balakrishnan, R. HybridPointing: fluid switching between absolute and relative pointing with a direct input device. *Proc. UIST 2006*, 211-220.

[5]   Guimbretière, F., Martin, A., Winograd, T. Benefits of merging command selection and direct manipulation. *Proc. TOCHI 2006* 12, 3, 460-476.

[6]   Kurtenbach, G., and Buxton, W. User learning and performance with marking menus. *Proc. SIGGRAPH 1994*, 258-264.

[7]   Kurtenbach, G., Fitzmaurice, G. W., Owen, R. N., and Baudel, T. The Hotbox: efficient access to a large number of menu-items. *Proc. CHI 1999*, 231-237.

[8]   Plumlee, M., and Ware, C., Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. *Proc. CHI 2006*, 13, 2, 179-209.

[9]   Sears, A., Shneiderman, B. High precision touchscreens: design strategies and comparisons with a mouse. *IJMMS '01* 34, 4, 593-613.

[10]   Sharp   Mebius   NJ70a   Netbook.   www.sharp.co.jp/ /mebius/products/pcnj70a/index.html. (September 2009)

[11]   Zhai, S., Morimoto, C., and Ihde, S. Manual and gaze input cascaded (MAGIC) pointing. *Proc. CHI '99*, 246-253.

Graphics **Interface** 2010