

Faster Command Selection on Tablets with FastTap

Carl Gutwin¹, Andy Cockburn², Joey Scarr², Sylvain Malacria², and Scott Olson¹

¹Computer Science, University of Saskatchewan
Saskatoon, Saskatchewan, Canada

²Computer Science, University of Canterbury
Christchurch, New Zealand

carl.gutwin,scott.olson@usask.ca andy.joey@cosc.canterbury.ac.nz, sylvain@malacria.fr

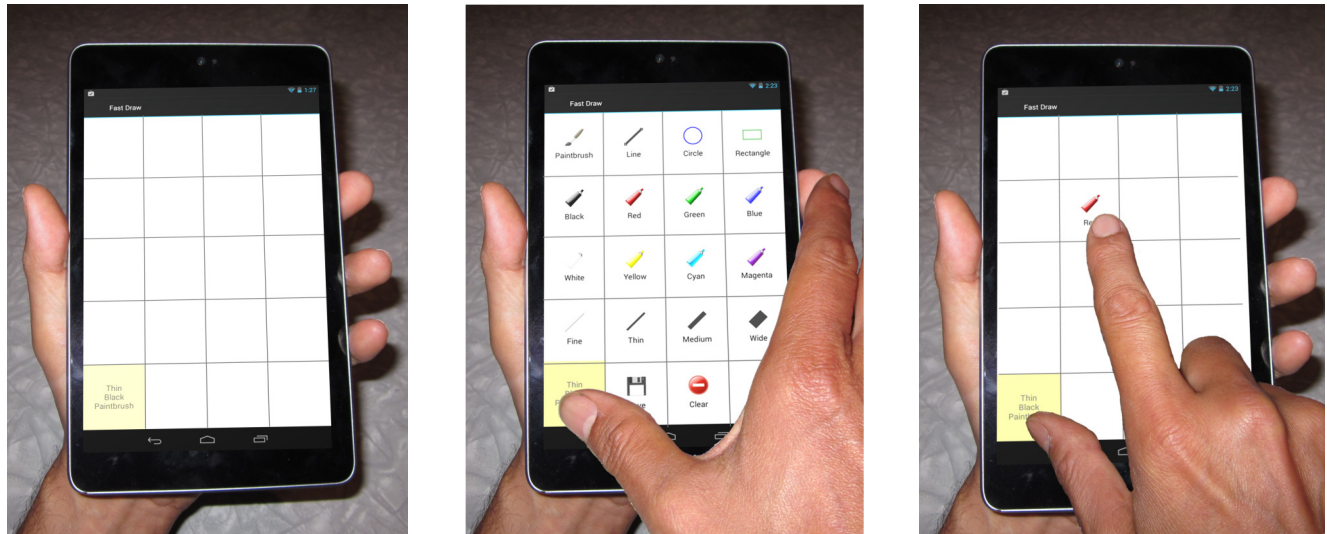


Figure 1. FastTap interface. Left: default state of the interface (gridlines enhanced). Center: FastTap grid overlay after touching the activation button. Right: FastTap selection by chording with the thumb and forefinger, without waiting for the overlay.

ABSTRACT

Touch-based tablet UIs provide few shortcut mechanisms for rapid command selection; as a result, command selection on tablets often requires slow traversal of menus. We developed a new selection technique for multi-touch tablets, called FastTap, that uses thumb-and-finger touches to show and choose from a spatially-stable grid-based overlay interface. FastTap allows novices to view and inspect the full interface, but once item locations are known, FastTap allows people to select commands with a single quick thumb-and-finger tap. The interface helps users develop expertise, since the motor actions carried out as a novice rehearse the expert behavior. A controlled study showed that FastTap was significantly faster (by 33% per selection overall) than marking menus, both for novices and experts, and without reduction in accuracy or subjective preference. Our work introduces a new and efficient selection mechanism that supports rapid command execution on touch tablets, for both novices and experts.

Author Keywords

Command selection; expertise; tablet UIs

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2473-1/14/04...\$15.00.

<http://dx.doi.org/10.1145/2556288.2557136>

ACM Classification Keywords

H.5.2. Information interfaces (e.g., HCI): User Interfaces.

INTRODUCTION

Desktop interfaces often offer multiple ways to select the same command, and the different mechanisms can have very different performance characteristics. For example, selecting commands from menu hierarchies is slow when compared to keyboard shortcuts and toolbars, which provide access to commands with fewer actions. Having these types of shortcuts in an interface can substantially increase a user's efficiency over time, by allowing the user to learn quicker methods of invoking the commands they use most often.

On portable touch-based devices such as tablet computers, however, interface shortcuts are seldom available. The lack of a physical keyboard means that there are no keyboard shortcuts for quick selection, and limitations on screen real estate leave little or no room for always-visible components such as toolbars. Touch-based command interfaces, therefore, often take the form of tedious menu hierarchies, with no way of making a transition to an expert method of interaction. This greatly decreases the utility of touch devices for productivity tasks.

One widely-studied interface that supports expertise on touchscreens is the *marking menu*, a type of radial menu that allows visual inspection of menu items for novices, and rapid gestural interaction for expert users [12]. However,

the contact-move-lift actions for marking menu gestures may take longer to perform than a simple tap action. With multi-touch capabilities widely available in modern tablet computers, there are opportunities for interfaces that support rapid command execution for experts, as well as smooth transitions from novice to expert use [3, 16, 23].

In this paper, we present FastTap, a new rapid-access interaction technique that allows fast command selection on multi-touch devices for both novice and expert users. As shown in Figure 1, FastTap uses the entire screen to present a spatially-stable grid of commands (based on the recent CommandMap interface [20]). The command overlay is hidden by default, and is shown when the user holds their thumb on the grid activation button. The interface can then be inspected, and commands can be selected with a finger; when the user lifts their thumb, the grid disappears.

Novices use the interface by showing the grid and visually searching for the commands they need. As users become familiar with commands, however, they remember item locations in the grid, leading to expert behavior – experts can select a command with a single ‘chorded’ tap using the thumb and forefinger, without waiting for the grid to appear. Similar to marking menus, this design follows Kurtenbach’s principle that *‘guidance should be a physical rehearsal of the way an expert would issue a command’* [15] – in other words, since the novice and expert interaction methods require similar motor actions, users can develop spatial and muscle memory of the action required for each command through natural use.

To assess the performance of FastTap, we carried out a controlled study comparing selection time and errors with FastTap and marking menus. Our study showed that selection time was significantly and substantially faster with FastTap (mean 1.6 seconds per selection) than with marking menus (mean 2.4 seconds). In addition, FastTap was significantly faster at all levels of expertise with the interface, and provided an additional speed benefit when multiple commands were carried out in sequence. We found no differences in terms of errors, effort, or preference.

Our work makes three main contributions. First, we introduce a new interaction technique for tablets that was significantly faster than marking menus in an initial study. Second, we further demonstrate the power of spatial memory as an organizing principle for visual interfaces, and demonstrate how spatial memory can be exploited together with multi-touch input to produce rapid command selection interfaces. Third, we provide empirical results about user performance with FastTap and marking menus.

DESIGN GOALS AND RELATED WORK

As shown in Figure 1, the FastTap interface works by displaying a CommandMap [20] over the main workspace when the user places their thumb on an on-screen activation area. In this section, we discuss the design goals behind FastTap and frame them in the context of related work.

Design Goal 1: Enable rapid command execution

Modern touch-screen applications typically use hierarchical menus and dialogs, with a few commands accessible on the main display, and others requiring several pointing actions that slow their selection.

Gesture-based systems, such as marking menus [12, 14], are one alternative to hierarchical linear menus, with some implementations appearing in commercial products (e.g., Autodesk Sketchbook Pro). Marking menus allow practiced users to traverse a radial menu hierarchy in a single gesture, speeding up interaction. However, navigating the hierarchy can still be slow, and marking menus are inherently limited in terms of the number of items that can appear at each hierarchical level (see Design Goal 3 for extensions to marking menus that can reduce this problem).

Multi-touch technology has created new opportunities for designing efficient command selection interfaces that exploit the higher bandwidth available with multiple concurrent contacts. For example, Wu and Balakrishnan [24] describe multi-finger and whole-hand interaction techniques for tables, including a selection mechanism that posts a toolglass with the thumb, allowing selection with another finger. Similar techniques are used and studied in Wagner et al.’s BiTouch system [23], but with a focus on handheld tablets where the thumb of the non-dominant hand is used to post interface components.

Multitouch marking menus [16] and finger-count menus [3] both allow users to specify a menu category by changing the number of fingers used to touch the screen (thus reducing the number of levels that must be traversed). Other techniques parallelize the hierarchy traversal: for example, Banovic et al.’s multi-finger pie menu [4] allows users to post the menu with one finger and select an item with another; Kin et al.’s two-handed marking menus [10] allow users to draw the marks for different menu levels simultaneously, by using both thumbs. However, these higher-bandwidth techniques do not always improve performance, since a more-complex control action may take more time to retrieve and execute [10].

Techniques utilizing device features other than the touch-screen have also been implemented. Jerktilts [1] allow quick selection of one of eight commands by quickly tilting the mobile device. Bezel Tap [22] uses the accelerometer to detect taps on the edge of the device, allowing shortcuts to commands while the device is asleep.

One key characteristic that determines whether a command selection interface is efficient is the number of separate actions needed to navigate to an item. Reducing this number is a main design goal of Scarr et al.’s CommandMap [20] technique, which uses a full-screen overlay to display as many commands as possible at once. These commands can be selected with a single action, which is faster than the multiple navigational steps needed for hierarchical menus and ribbons. Scarr et al. also showed

that navigational errors (e.g., choosing the wrong menu) substantially increased the time needed for hierarchical organizations. In this work, we adapt the CommandMap's flat and spatially stable design to work with mobile devices.

Design Goal 2: Support a transition to expertise

One of the primary advantages of marking menus is the way in which they support a smooth and rapid transition to expert use [12]. After activating a marking menu, a novice user can wait for a short time to see a labeled radial menu appear, from which they make their selection with a touch gesture; an expert user can make the same gesture without waiting for the menu to appear. Since the motor actions for the novice and expert uses of the menu are identical, users learn the expert gestures through normal interaction.

This principle of *rehearsal* is extremely important to the development of user expertise. FastTap is therefore designed to support rehearsal during novice use. In a similar manner to marking menus, the command grid only appears on-screen after a delay; users with spatial or muscle memory of the interface can interact instantly without waiting for the visual display, using the exact physical action they used as novices; intermediate users suspecting the desired command location but unwilling to execute it without confirmation can also benefit from FastTap, by anticipating the location of the target, positioning their finger over it while waiting for the grid to appear, and selecting the command after visual confirmation.

Design Goal 3: Support a large number of commands

While marking menus are generally limited to eight or twelve commands per level [14], various extensions significantly increase this limit. Polygon menus [25] and flower menus [2] both allow more commands by increasing the types of gestures available. More recently, Roy et al. [19] developed Augmented Letters, a system whereby users draw the first letter of a command on the screen, then select from a radial menu of resulting candidate commands. OctoPocus [5] recognizes gestures by shape, and provides visual suggestions for the remaining gesture based on the initial movements. While these systems increase the number of commands that marking menus can support, they still rely on gesture-based interaction.

Rapid execution is our priority for FastTap, but we also intend that it will support a wide command vocabulary. In FastTap, the number of items at each level is limited only by the size of the screen; our prototype uses a 5x4 grid, with one cell being used as the FastTap activation button. However, this number can be increased through the use of different activation buttons, or command tabs, which can be arranged along the bottom of the screen. We consider these design possibilities further in the Discussion.

STUDY: SELECTION PERFORMANCE

The aim of the study was to assess the performance of FastTap for command selection on tablets. We compared FastTap to marking menus [15], which allow fast command

selection for experts and support a smooth transition to expertise. We compared the two interfaces in a controlled experiment where participants selected a set of commands over several repeated blocks, allowing us to examine both novice and more expert selection behavior.

Experimental Conditions

Both FastTap and marking menus were implemented in a functional multi-touch drawing application (see Figure 2).

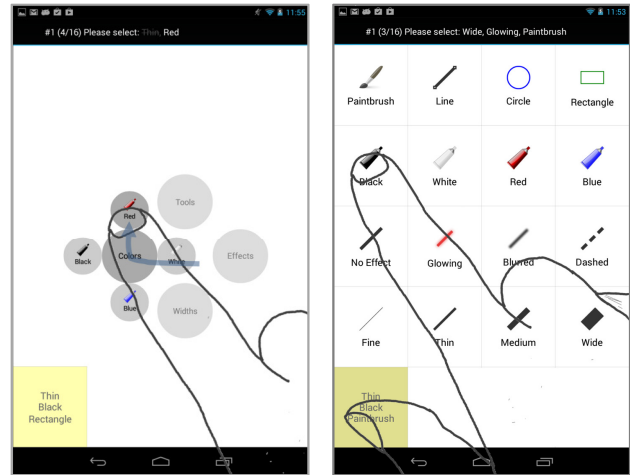


Figure 2. Study UIs. Left: marking menu (arrow shows gesture path). Right: FastTap. Cue appears at top of screen.

FastTap. As described above, FastTap provides modal access to a grid of command buttons. Selections are made by pressing a command button, either after invoking the grid display, or simultaneously with the invocation button (i.e., by chording). There is no difference in the selection mechanism for novice or expert use – experts who know the item locations simply tap the command before the interface is shown. After a chorded selection, feedback on the selected command is given by displaying the command icon for 500ms (Figure 1, right). The interface used in the experiment contained sixteen command buttons in a 4x4 grid, of which eight were used as study targets. The sixteen commands were organized into four rows that grouped similar commands together (see Figure 2).

Marking menus. We implemented a 16-item marking menu with a two-level hierarchy, adapted from Kurtenbach's previous work [15]. Once again, only eight of the 16 items were used as study targets. Upon invocation of the menu (described below), users move a finger towards one of four categories shown on screen (Shapes, Colors, Line Style, Line Width), and then to one of the items in that category (see Figure 2, left). The items in each category were the same as the row groups used in FastTap (Figure 2). We used a hierarchy because previous research indicates that selection errors increase significantly for menu levels containing more than eight items [15]. We chose commands for the study that fit well with the four categories, to reduce the chance that participants would select the wrong category. The categories are only needed when participants

are new to the menu, however; as users become familiar with item locations in the marking menu, they can shift to expert interaction simply by drawing the gesture that uniquely indicates an item (e.g., up-right).

Marking menus also need an invocation mechanism, and we used the same activation button as in the FastTap condition (see Figure 2). Users invoked the menu by touching the activation button with their thumb, then touching their finger on the screen to instantly post the menu. Moving the finger in one of the four directions then opens the corresponding submenu, and moving again selects one of four items. Some implementations of marking menus use a 'press timeout' invocation, but to avoid disadvantaging the menu condition, we chose a method that was equally fast to the invocation of FastTap.

Procedure

Participants completed a demographics questionnaire, and then performed a sequence of selections in a custom study system with both marking menus and FastTap. For each trial, a command stimulus (consisting of one, two, or three command names) was displayed at the top of the screen; the participant then selected the command(s) using the interface provided. Trials involved selecting a combination of one, two, or three individual tools and properties that could be used within the drawing application (e.g., 'Red', 'Red Line', 'Red Thin Line'). Trials were timed from the appearance of the stimulus until all targets were successfully selected. In the case of multiple-command targets, command names in the stimulus were crossed out as they were selected, and participants could select commands in any order. Participants were instructed to complete tasks as quickly as possible, and were told that errors could be corrected simply by selecting the correct item. Completion times included the time for correcting errors.

Of the sixteen commands in each interface, only eight were used as stimuli, in order to allow faster development of spatial memory and expertise for both interfaces. Two commands were used from each interface category (Shape, Color, Line Style, Line Width). Multiple-command targets were also composed from these eight commands.

For each interface, selection trials were organized into blocks of sixteen selections (eight one-command targets, four two-command targets, and four three-command targets). Participants first performed one practice block of sixteen trials (data discarded) to ensure that they could use the interfaces successfully. They then carried out 10 blocks of sixteen selections. Targets were presented in random order (sampling without replacement) for each block. Short rest breaks were given between blocks.

After each interface, participants filled out a NASA-TLX questionnaire [8]; at the end of the study, they also answered summary questions about their preferences.

Participants and Apparatus

Sixteen participants were recruited from a local university (8 female; mean age 26.2 years). The study was conducted on a Nexus 7 Android tablet (7-inch screen, 1280x800 resolution). The software was written in Java, and recorded all experimental data including selection times and errors.

Design and Hypotheses

The study used a 2x3x10 within-participants RM-ANOVA with factors *Interface* (FastTap, MarkingMenu), *NumberOfCommands* (1, 2, or 3 commands per trial), and *Block* (1-10). Dependent measures were command selection time, and errors per command selection. Interface was counterbalanced between participants. Hypotheses were:

- H1.** Mean selection times for *FastTap* will be faster than for *MarkingMenu*.
- H2.** *FastTap* will be faster than *MarkingMenu* both for novices and for experts.
- H3.** *FastTap* will show an added speed benefit for two- and three-command targets.
- H4.** There will be no evidence of a difference in error rates between the two interfaces.
- H5.** There will be no evidence of a difference in perception of effort for the two interfaces.
- H6.** Users will prefer *FastTap* over *MarkingMenu*.

Results

For significant ANOVA results, we report partial eta-squared (η^2) as a measure of effect size (where .01 should be considered small, .06 medium, and > .14 large [5]).

Selection Time per Command

We calculated the selection time for each command by dividing the total trial time by the number of commands in that trial. As shown in Figure 3 (rightmost bars), mean selection times were about 0.8 seconds faster per command with *FastTap* (1.60s, s.d. 0.52s) than with *MarkingMenu* (2.39s, s.d. 0.65s), giving a significant main effect of *Interface* ($F_{1,15}=84.37$, $p<.0001$, $\eta^2=0.85$). We therefore accept **H1** – FastTap selections were 33% faster overall than the marking menu.

Figure 3 also shows selection time by the number of commands per trial. ANOVA showed a significant main effect of *Number of commands* ($F_{2,30}=3.7$, $p<.05$, $\eta^2=0.20$), and also an interaction with *Interface* ($F_{2,30}=6.69$, $p<0.005$, $\eta^2=0.31$). As suggested by Figure 3, both effects can be attributed to *FastTap* permitting faster selections when commands are joined into groups of two or three. Post-hoc t-tests (Bonferroni corrected) show a significant difference between the two interfaces for each number of commands (all $p<.0001$). We therefore accept **H3**.

Learning Effects

As shown in Figure 4, selection times decreased across trial blocks; ANOVA showed a significant main effect of *Block* ($F_{9,135}=17.49$, $p<.0001$, $\eta^2=0.54$). There was no interaction with *Interface* ($F_{9,135}=1.43$, $p=.183$) or with *NumberOfCommands* ($F_{18,270}=1.29$, $p=.194$).

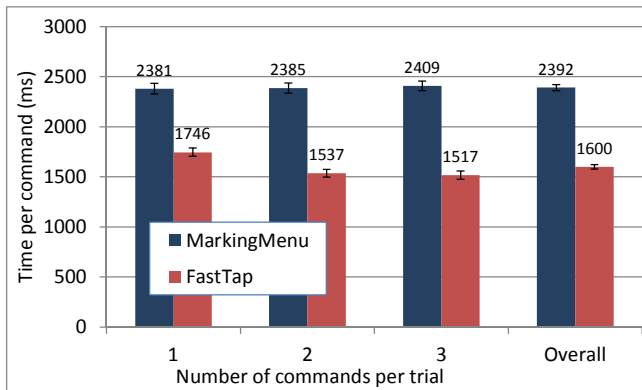


Figure 3. Mean selection times by Interface and Number.

We also analysed the performance difference between interfaces at each block. Bonferroni-corrected t-tests showed that *FastTap* was faster than *MarkingMenu* at all levels of expertise (all $p < .001$). Since the advantage of *FastTap* over marking menus is consistent across trial blocks, we accept **H2**.

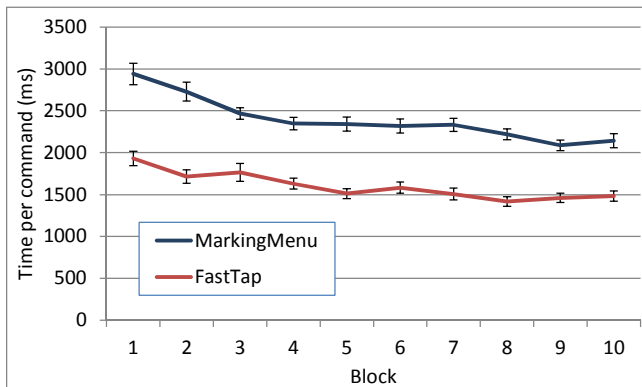


Figure 4. Mean trial times by Interface and Block.

Errors

As with selection time, we analysed errors per command, dividing the number of errors in a trial by the number of commands in that trial. Errors were counted as any incorrect selection (note that multiple-command trials could be carried out in any order). ANOVA showed no effect of *Interface* on errors, with *FastTap* at 0.10 errors/command, s.d. 0.13, and *MarkingMenu* at 0.08 errors/command, s.d. 0.11 ($F_{1,15}=2.96, p=.11$). We therefore accept **H4** (errors are considered further in the discussion section).

There was also no effect of *Block* ($F_{9,135}=0.41, p=.93$) or *NumberOfCommands* ($F_{2,30}=0.51, p=.61$) on errors per command; there were no interactions between any factors.

Subjective Responses: Effort and Preferences

User response was positive to both interfaces, but with no strong differences in NASA-TLX scores (compared using the Friedman test; see Table 1). No significant differences were found on any effort question, and the mean scores were close in all cases; therefore, we accept **H5**.

	FastTap	Marking Menu	χ^2	p
Mental demand	5.3 (2.2)	5.1 (2.0)	0.25	.62
Physical demand	3.8 (2.0)	4.4 (1.8)	0.25	.62
Temporal demand	4.6 (2.0)	4.7 (1.7)	0.0	1.0
Success	6.6 (1.3)	6.5 (1.4)	0.0	1.0
Hard work	5.3 (1.8)	5.5 (1.5)	0.25	.62
Frustration	3.6 (1.8)	3.5 (1.6)	0.0	1.0

Table 1. Mean (s.d.) NASA-TLX scores (0-9 scale, low to high).

We also asked participants which interface they preferred in terms of several qualities (see Table 2). As with the effort scores, counts were very close with no quality showing a significant difference. When asked about overall preference, seven participants preferred *FastTap*, eight preferred marking menus, and one had no preference. We therefore cannot accept **H6**.

	FastTap	Equal	Marking Menu	χ^2	p
Speed	8	2	6	0.071	.79
Accuracy	6	1	9	0.26	.61
Memorization	6	2	8	0.071	.79
Expert mode	5	3	8	0.31	.58
Comfort	8	1	7	0.0	1.0
Overall	7	1	8	0.0	1.0

Table 2. Counts of participant preferences.

Participant Comments

Like the preference results, participant comments were about evenly divided between the two selection techniques, and participants often mentioned the characteristic features of the designs when explaining their preferences. For example, participants made several comments about how spatial stability and quick activation helped the speed of *FastTap*: one person commented on “the simple and stable location of each icon”; another said “the grid is faster because it only requires a two finger tap, with one finger always in the same place. The menu, however, demands specific movements (swipes) which expend a few milliseconds more time.”

It was also clear that some participants liked the semantic categories of the *Marking Menu*, and others liked *FastTap*’s access to all commands at once. In favour of marking menus, participant comments included “the menu compartmentalized the options much better”; “the menu required less memory”; and “visually a grid with all available options is not as easily navigable as a menu with divisible submenus.” In favour of *FastTap*, comments included: “the options are all available at one time which makes it easier to pick, and memorization also helps a lot in the grid”; “it is easier to see the items that you want to select”; and “there is only one level to be memorized, where [the marking menu] has two levels.”

Some of the participants were initially concerned with the number of commands in *FastTap*, but ultimately preferred it: “after a while, the grid became easier to remember”; “I was able to catch on much quicker [with *FastTap*] and be able to visualize where everything was.”

Finally, one participant also mentioned that the sliding motions needed for the marking menu could present problems on a touch surface, and that *FastTap*’s tapping

action did not have this problem: “*sliding over a surface can be impeded by many factors like moisture [on the] hand and friction over the surface. [...] I feel the tapping action in the grid is more comfortable.*”

ADDITIONAL INFORMAL USAGE OBSERVATIONS

To gather additional information about whether FastTap would work outside of a study setting, we developed a simple multi-touch painting application with a FastTap interface and showed the system to five people in an informal test. We were interested in whether people would hold the device in a way that allowed correct FastTap operation (building on the work of Wagner et al. [23]), whether people would be able to use the interface with a minimum of instruction, and whether people would take initial steps toward expert selection behavior.

The five participants used the interface in a variety of settings (an office, an undergraduate lab, and three homes), and we asked them to play with the interface and to recreate a printed picture three times (to simulate increasing familiarity with a set of commands).

Holding the device. Wagner et al. [23] looked at the ways that people held an iPad, in order to determine whether the non-dominant hand could be used for bimanual input (e.g., to show toolbars or other interface elements). They observed several common postures involving the non-dominant palm or forearm. We also saw several different handholds in our informal sessions, some of which were seen by Wagner – for example, the tablet held in the left palm, with the FastTap activation button pressed by the non-dominant thumb. However, several other people held the device by gripping it between their left thumb and fingers – which was not possible in Wagner’s study due to the size of the iPad, but which was possible with our seven-inch Nexus tablet. This grip-based hold allowed operation of FastTap with the thumb and fingers of the dominant hand, which matched our intention in designing the method.

Novice and more expert behaviour. In our informal sessions, people found it easy to use FastTap – their hands fit the interface reasonably well, and people did not have any difficulty finding the commands that they wanted. On a few occasions, however, people inadvertently drew rather than selecting. In the short time of these sessions, we saw only a few uses of expert chording behaviour; however, it was clear that all users were gaining spatial memory of frequently-used commands. For example, people moved their selection finger towards or over the target even before displaying the grid, showing that they had an idea about where the desired command was located.

These preliminary observations are limited, but provide initial evidence that FastTap can work in natural use.

DISCUSSION

Our investigation of FastTap provides four main results:

- Selection with FastTap was significantly faster overall than with marking menus (0.8s per selection, more than 33%), and it was faster throughout the learning curve;
- Making multiple selections allowed a speedup of about 200ms in selection time for FastTap, which did not occur with marking menus.
- There was no significant difference in errors between FastTap and marking menus, although FastTap had a slightly higher error rate (0.10 errors/selection compared to 0.08 for Marking Menu);
- Participants were evenly divided between the two interfaces for perceived effort and overall preference; some participants particularly liked each interface;
- In limited usage observations, we saw that people were able to hold the device appropriately for FastTap, were able to find commands easily, and appeared to quickly build memory of command locations.

The following sections provide explanations for these findings, address design issues for real-world use of FastTap, and discuss its generalizability.

Explanation and Interpretation of Results

Why did FastTap work well for both novices and experts?

Our experiment showed that FastTap improved (by 33%) on a well-known technique (marking menus) for selecting commands on tablets; in addition, FastTap was faster at all stages of learning.

The performance advantage of FastTap can be explained through an analysis of the steps required for command selection, in both novice and expert cases. For novices, there are three steps required: activating the command mode, searching for the desired command, and executing a selection action. Activating the command mode was the same in both interfaces, but the interfaces differed in the second and third steps. Searching for a command in FastTap involves only visual search over all of the concurrently displayed items, whereas search in marking menus involves first choosing and selecting a semantic category (Colours, Tools, etc.) and then searching in a much smaller set of items. Although some participants found the semantic categories helpful, previous research has shown that the decision and selection time costs of traversing the hierarchy can exceed that of a broader visual search [20]. The visual search needed for FastTap could take longer initially, since there are more items to inspect, but the rapid development of spatial memory means that novice users will quickly make a transition from full visual search to remembering where items are located. The third step – execution of the selection action – is faster in FastTap, since a tapping action can be carried out more quickly than the sliding motion of a marking menu.

For experts, selection requires only two steps: retrieval of the command action (either a location or a gesture) from memory, and execution of that action. The performance advantage for expert use of FastTap most likely arises from

the speed of execution, since the memory-retrieval step is similar for both techniques. Execution of a thumb-and-finger tapping is a faster action than a drawn gesture, leading to a performance advantage for FastTap.

Why were multiple selections faster than single selections?

Multiple commands selected with FastTap were each about 200ms faster than single commands. We believe that this speed-up is due to people's ability to visualize multiple command locations within FastTap's grid interface, and optimize their movements for multiple selections. For example, we observed participants re-ordering the selections to reduce finger movement (e.g., ordering the selections by grid row allowed people to move their finger in a straighter line). With the marking menu, there is less opportunity for optimization (e.g., re-ordering the commands does not reduce overall movement).

Does FastTap lead to more errors?

Error rates were high overall in both techniques (10% and 8% for FastTap and marking menus). This high error rate is probably an artifact of our experimental protocol, which explicitly instructed participants to select items as quickly as possible, while noting that errors could be corrected afterwards. Although FastTap had a slightly higher mean error rate (10%) than marking menus (8%), the difference was not statistically significant ($p=.11$). However, if we assume that effect of higher errors with FastTap is actually a reliable one, we see three candidate explanations. First, the quick execution of a selection action in FastTap may have encouraged participants to view errors as amenable to rapid correction, thereby encouraging users towards a 'guess and correct' mode of operation. Second, participants may have found the post-selection visual feedback in FastTap more clearly communicative of the selected item than the marking menu, again encouraging faster but more error prone selections. Third, it is possible that people's memory of an item's spatial location was imperfect, and so participants may have experienced 'near misses' more often than with marking menus. Further work is needed to determine whether the difference in error rates is reliable, and to properly explain its cause if it is.

FastTap and Marking Menus with Larger Item Sets

Our study tested FastTap and marking menus with a small command set (sixteen items), and it is worth considering how the two approaches would compare in the case of a larger interface. The main performance differences between FastTap and marking menus are in initial visual search (novice behavior) and the number of actions needed for a learned item (expert behavior). These differences arise due to the properties of FastTap's single-level presentation, compared with marking menu's hierarchical organization.

With a larger command set, it seems likely that FastTap will require more visual search than marking menus – provided the hierarchical organization of the marking menu items is clear (which can be a non-trivial design problem [20]). However, several prior studies suggest that the novice

period of use is relatively short (e.g., [20, 21]) and users will soon be interacting with commands that they are familiar with. In this case, FastTap selection time will be similar to that observed in our study – since only a single invocation and selection action is needed, regardless of the size of the command set. Marking menus, however, must create hierarchies because they use less space, and so experts must continue to execute multiple navigational steps (even if these are executed as marks).

Recent work on multi-touch marking menus [3, 10, 16] suggests ways that these navigation interactions can be speeded up – e.g., by encoding top-level menu categories with different finger postures [16], or by parallelizing the execution of navigational marks [10]. Although we believe that FastTap will continue to compare well against other selection techniques as command sets increase in size, further work is needed to explore the potential differences between the spatial-memory approach of FastTap and the gesture-memory approach of marking menus.

Design issues for FastTap

In the following sections we consider issues likely to arise if deploying FastTap in real-world tablet interfaces.

How many commands can the interface accommodate?

Our drawing interface contained 19 commands in a 5x4 grid, but this is not the limit for the FastTap approach. In general, the number of commands in the interface is limited by the size of the device, the minimum desired size of the targets, and the size of the user's hand (to facilitate chording). Using the average width of an adult index finger (16-20mm [7]) as a guideline, it would be possible to have a grid of up to 40 buttons in an 8x5 layout on a 7-inch tablet. If we consider Parhi et al.'s recommendation that touch targets be no smaller than 9.6mm [17], a maximum density of 150 items in a 15x10 arrangement is possible. However, this size guideline is debatable – larger button sizes have been shown to provide higher success and satisfaction rates [17, 18], yet much smaller targets can also be used, as demonstrated by smartphone virtual keypads (which can be operated substantially eyes-free with sufficient practice). There are interesting research questions around users' ability to form spatial and motor memories for varying numbers of targets at different target densities.

Importantly, however, the possibilities for adapting FastTap to high functionality applications, or to small displays, is not necessarily limited to the 'all commands at once' designs examined in this paper. The capacity of the interface to display candidate targets can be multiplied through the use of multiple trigger buttons, or 'tabs', which organize the commands into multiple categories – each trigger button would then show a separate set of commands (see mockup in Figure 5). As discussed under Design Goal #3, this technique requires the allocation of additional trigger-button regions in the bottom row of the grid. Further research is required to determine how well people can remember these two-finger combinations.

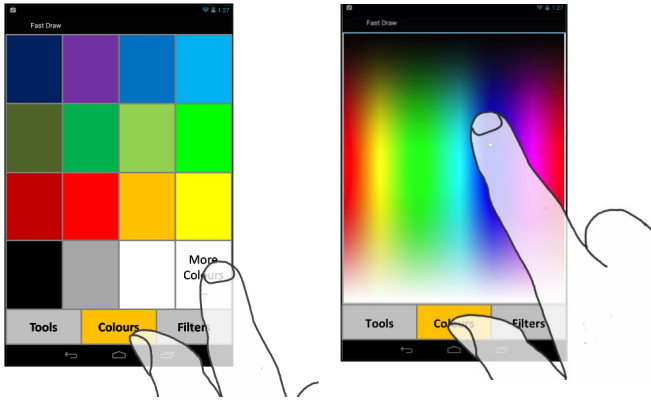


Figure 5. Left: FastTap UI with three tabs. Right: A color picker converted to work with FastTap.

Finally, it is also possible that if the grid interface cannot accommodate all of the application’s commands, it could still function as a ‘hotbox’ [13] for frequently-used items.

FastTap with different size factors

Our experiments were conducted on a device equipped with a 7-inch touchscreen. We used this relatively small size factor because tablets with screens smaller than 8 inches are the main proportion of the current tablet market, and this proportion is forecast to increase by 2017 [9]. We conducted pilot studies on several smaller touch-screen devices (e.g., Nexus 4, Galaxy S3), mostly testing the drawing system. The systems worked well, although the command buttons were smaller overall, which may have accuracy implications for expert users.

Porting FastTap to tablets with *larger* screens also raises interesting questions, especially whether the designer should use the full screen to fit more commands. If not, the FastTap interface can be limited to a comfortable size for the user’s hand. Using the full screen, however, would permit more commands to be displayed, but it would also create problems for single-handed operation due to hand-span limitations. A different triggering mechanism might be used for commands that are further away than a hand span – such as the bimanual thumb technique used in Wagner’s BiTouch system [23].

Issues regarding device orientation

Mobile devices such as tablet computers can be used in different orientations, which changes the aspect ratio of the screen. There are three possibilities to accommodate orientation changes. First, the grid could maintain its overall aspect ratio and scale to fit the smaller dimension of the new orientation, requiring that users adapt to a different-scale interface. Second, the grid could change its aspect ratio to fill the new orientation, requiring that users adapt to a stretched version of the grid. As shown by Scarr et al. [20], these two transformations would cause only a minor disruption to spatial selections. Finally, the grid could maintain its size regardless of the orientation, fitting the most constraining orientation.

Biomechanical constraints

Some of our participants cited difficulties with the hand positions required to select commands with FastTap. Specifically, participants noted that targets were more difficult to acquire when they were close to or far away from the trigger button, or when they were at the lower-right side of the grid. People with long fingernails had particular difficulty maintaining registered contact with the display when selecting targets near to the trigger button, (the natural posture for such selections involves the fingers being nearly perpendicular to the display, causing nails to lift fingers off the touch screen).

In all cases, however, problems with hand positions were addressed in two ways. First, users could re-orient the device with the holding hand in order to present a better presentation of the trigger and target buttons. For example, to more easily tap a target in the lower right part of the grid, the user could turn the device slightly anticlockwise with their holding hand, so that the target button (finger) was above and to the right of the trigger button (thumb). Second, users did not have to use the thumb and forefinger to make selections, and could use a combination that simplified the particular selection. For example, many users used their first and second fingers for the bottom row of commands instead of their thumb and first finger.

It would also be possible to change from a grid-based command organization to one that better matches the ergonomics of the hand. However, it is not clear whether a non-rectilinear grid would make it more difficult to acquire targets in expert mode (i.e., with no visual feedback).

FastTap and advanced interfaces

Modern UIs generally include widgets more advanced than push-buttons. For example, sliders or color spectrum pickers can be used to provide a finer degree of control over application parameters. These widgets can easily be converted to work with FastTap, as shown in Figure 5. FastTap provides quick access to these widgets and allows users to perform rapid and coarse slider positioning that can instantly be adjusted without having to switch modes.

Modal interface and occlusion of the document

All in-place interfaces (including marking menus) occlude parts of the document, but the overlay presentation of FastTap’s interface is different in that it completely fills the screen. In most cases, this is unlikely to cause a problem for the user, for three reasons: first, the grid is partially transparent and so the main visual elements of the document can be seen through the FastTap overlay; second, the overlay does not appear at all in expert mode, so there is no occlusion problem for experienced users; third, it is easy for the user to control the presence of the overlay (by lifting their thumb from the activation button), allowing rapid switching back and forth from the document to the UI.

One case where the overlay presentation requires a design solution is the situation where the user has selected an

object on the screen, and wants to see previews of a command's effects on that object (e.g., what it looks like in different colours). Although it is possible to simply select different colours using the standard FastTap process (expert mode makes this fast and easy), other designs could reduce the problem – such as placing a floating copy of the selected object above the grid, or fading out the overlay whenever the user's finger stops moving across the grid.

Last, the visual presence of the activation button is another potential cause of occlusion. This button is also partially transparent, and could easily increase in transparency level as the user becomes more experienced (although this does not happen in our current prototype).

Multiple simultaneous selection

Normal expert selection in FastTap involves two digits (normally the thumb and one finger; but sometimes two different fingers). This combination selects one command; however, there is no reason why additional commands cannot be selected simultaneously in the same chorded tap. For example, the user could select “black paintbrush” with the three-finger chord shown in Figure 6.

These kinds of selections already work in our prototype application. Added-finger selections work well only for certain combinations (because of constraints on positioning the fingers); therefore, if these are to be used in interface design, further work must determine which fingers combinations are possible, and which command combinations are desirable.

Error rates in memory-based UIs

Above we discussed error rates with FastTap and marking menus. However, a more general question concerns the errors that will always occur with a memory-based means for interaction. Errors are inherent in any memory-based technique, particularly in the transition from novice to expert use, since users must at some point depend on their memory rather than on visual feedback. This issue affects marking menus, keyboard shortcuts, and FastTap.

In general, there are several issues that must be considered when designing for the possibility of errors: the destructiveness of a command, the ease with which the error can be identified, the time that the command takes to execute, and the time required to correct the error. For non-destructive and instantaneous commands such as the property-setting commands of a drawing program (colour, line style, etc.), selection errors are a minor problem. If feedback on the selection is given (as it is in our implementation, even in expert mode), users are likely to realize their error and simply correct it.

For commands that take a long time to construct or execute (e.g., a complicated image filter), commands should be easily interruptible to avoid problems of erroneous selection. For destructive commands (e.g., deletions), systems may need a confirmation step or a well-designed ‘undo’ facility. It is possible that memory-based techniques

like FastTap or marking menus could be reserved for lighter-weight commands, with destructive or time-consuming commands accessed through a different mechanism (e.g., a traditional application menu).

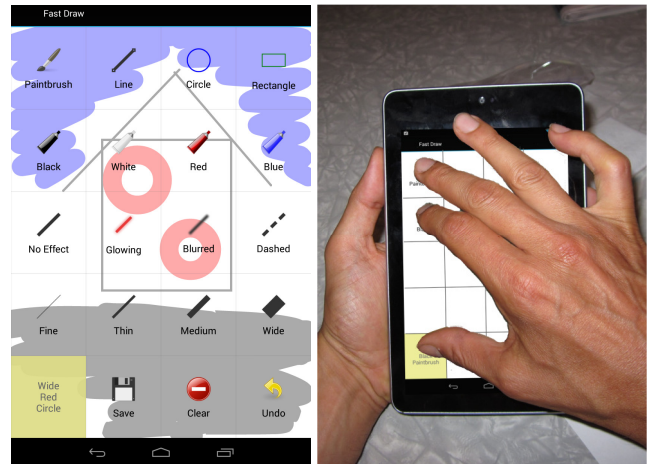


Figure 6. Left: transparent grid with drawing behind. Right: three-finger chord that selects two commands.

Single-handed use

Although we developed FastTap on a tablet that would be difficult or impossible to use with one hand, several people have noted that phone-sized mobile devices are often used in a one-handed fashion. The ideas underlying FastTap can be used in a one-handed context, providing some or all of the benefits of the interface. The spatial-memory advantages of FastTap could be achieved even without multi-touch chorded taps; a two-tap version of the interface would bring up the interface with one tap, and then allow selection with a second tap. For most single-handed operation, the user touches the screen with their right thumb, which means that the activation button would have to be moved to the right side of the screen.

It would also be possible to place an activation button somewhere else on the device to accommodate chording with one-handed use (e.g., on the back of the device, or on the bezel under one of the other fingers, although these would require hardware changes). Similarly, the activation functionality that currently uses a multi-touch selection could be changed to another modality – for example, double-tapping on a grid location could be a method for specifying both activation (the double tap) and selection (the location) with a single finger.

CONCLUSIONS AND FUTURE WORK

Although multi-touch tablets are now common, and are starting to be used for productivity work, there are few techniques for these devices to support rapid command selection. In this paper, we presented a new selection technique for multi-touch tablets called FastTap that uses thumb-and-finger touches to show and choose from a grid-based overlay interface. FastTap allows novices to view and inspect the full interface, but once item locations are known, FastTap also allows people to select commands

with a single quick thumb-and-finger tap. The interface helps users move toward expert use, since the motor actions carried out in novice mode rehearse the expert behavior. A controlled study with 16 participants showed that FastTap was significantly faster (0.8 sec/selection, 33%) than marking menus, both for novices and experts, and without reduction in accuracy or subjective preference.

Our research thus far with FastTap gives us strong initial results, and future work will continue in three directions. First, we will continue development of the drawing application and release a fully-functional version of the app, in order to gather real-world usage and performance data from a wide audience. Second, we will develop new prototypes that explore some of the design issues described above, including tabbed command sets, complex interface widgets, and interface scaling for larger devices. Third, we will develop FastTap prototypes for other applications that could benefit from fast access to commands and shortcuts (e.g., contacts and response options in a mail client, or favourites and page actions in a web browser). Our experience with FastTap suggests that the underlying ideas of spatial-memory-based expertise and quick tap-based access can be successfully and broadly applied across several application domains.

ACKNOWLEDGMENTS

This work was supported by the NSERC SurfNet Research Network and the GRAND NCE, and by the Royal Society of New Zealand Marsden Grant 10-UOC-020.

REFERENCES

1. Baglioni, M., Lecolinet, E. and Guiard, Y. JerkTilts: using accelerometers for eight-choice selection on mobile devices. *Proc. ICMI 2011*, 121-128.
2. Bailly, G., Lecolinet, E. and Nigay, L. Flower menus: a new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. *Proc. AVI 2008*, 15-22.
3. Bailly, G., Müller, J. and Lecolinet, E. Design and Evaluation of Finger-Count Interaction: Combining Multitouch gestures and Menus. *IJHCS 70(12):673-689*.
4. Banovic, N., Li, F., Dearman, D., Yatani, K. and Truong, K. Design of unimanual multi-finger pie menu interaction. *Proc. ITS 2011*, 120-129.
5. Bau, O., and Mackay, W., OctoPocus: a dynamic guide for learning gesture-based command sets. *Proc. UIST 2008*, 37-46.
6. Cohen, J. Eta-squared and partial eta-squared in communication science. *Human Communication Research 28(56):473-490*.
7. Dandekar, K., Raju, B. and Srinivasan, M. 3-D finite-element models of human and monkey fingertips to investigate the mechanics of tactile sense. *Journal of Biomechanical Engineering*, 125, 5, 2003, 682-691.
8. Hart, S. and Staveland, L. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. in Hancock, P. and Meshkati, N. eds. *Human Mental Workload*, 1988, 139-183.
9. IDC. IDC Forecasts Worldwide Tablet Shipments to Surpass Portable PC Shipments in 2013, www.idc.com/getdoc.jsp?containerId=prUS24129713, 2013.
10. Kin, K., Hartmann, B., and Agrawala, M., Two-Handed Marking Menus for Multitouch Devices. *ToCHI 18(3)*: article 16.
11. Kurtenbach, G. and Buxton, W. Issues of Combining Marking and Direct Manipulation Techniques. *Proc. UIST 1991*, 137-144.
12. Kurtenbach, G. and Buxton, W. User Learning and Performance with Marking Menus. *Proc. CHI 1994*, 258-264.
13. Kurtenbach, G., Fitzmaurice, G., Owen, R. and Baudel, T. The Hotbox: efficient access to a large number of menu-items. *Proc. CHI 1999*, 231-237.
14. Kurtenbach, G., Sellen, A. and Buxton, W. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Human-Computer Interaction 8(1):1-23*.
15. Kurtenbach, G.P. *The design and evaluation of marking menus*, PhD Thesis, University of Toronto, 1993.
16. Lepinski, G., Grossman, T. and Fitzmaurice, G. The design and evaluation of multitouch marking menus. *Proc. CHI 2010*, 2233-2242.
17. Parhi, P., Karlson, A. and Bederson, B. Target size study for one-handed thumb use on small touchscreen devices. *Proc. Mobile HCI 2003*, 203-210.
18. Park, Y., Han, S., Park, J. and Cho, Y. Touch key design for target selection on a mobile phone. *Proc. Mobile HCI 2008*, 423-426.
19. Roy, Q., Malacria, S., Guiard, Y., Lecolinet, E. and Eagan, J. Augmented letters: mnemonic gesture-based shortcuts. *Proc. CHI 2013*, 2325-2328.
20. Scarr, J., Cockburn, A., Gutwin, C. and Bunt, A. Improving command selection with CommandMaps. *Proc. CHI 2012*, 257-266.
21. Scarr, J., Cockburn, A., Gutwin, C. and Malacria, S. Testing the Robustness and Performance of Spatially Consistent Interfaces. *Proc. CHI 2013*, 3139-3148.
22. Serrano, M., Lecolinet, E. and Guiard, Y. Bezel-Tap gestures: quick activation of commands from sleep mode on tablets. *Proc. CHI 2013*, 3027-3036.
23. Wagner, J., Huot, S., and Mackay, W. BiTouch and BiPad: Designing bimanual interactions for hand-held tablets. *Proc CHI 2012*, 2317-2326.
24. Wu, M., and Balakrishnan, R. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *Proc. UIST 2003*, 193-202.
25. Zhao, S., Agrawala, M. and Hinckley, K. Zone and Polygon Menus: Using Relative Position to Increase the Breadth of Multi-Stroke Marking Menus. *Proc. CHI 2006*, 1077-1086.