HandMark Menus: Rapid Command Selection and Large Command Sets on Multi-Touch Displays

Md. Sami Uddin[1], Carl Gutwin[1], and Benjamin Lafreniere[2]
[1]Computer Science, University of Saskatchewan
Saskatoon, Canada
[2]Autodesk Research
Toronto, Canada
sami.uddin@usask.ca, gutwin@cs.usask.ca, ben.lafreniere@autodesk.com

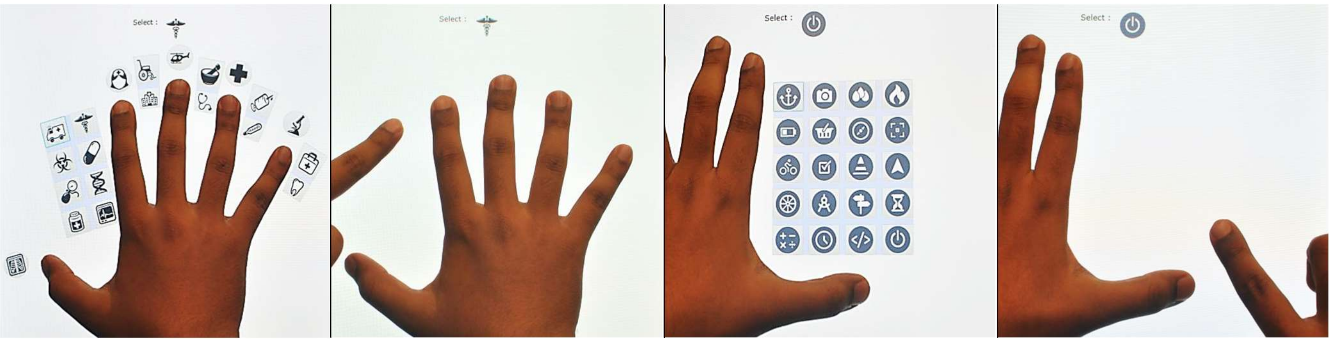

Figure 1. HandMark Menus. From left, 1: HandMark-Finger (novice mode). 2: HandMark-Finger chorded selection (expert mode), 3: HandMark-Multi (novice mode), 4: HandMark-Multi chorded selection (expert mode).

ABSTRACT

Command selection on large multi-touch surfaces can be difficult, because the large surface means that there are few landmarks to help users build up familiarity with controls. However, people's hands and fingers are landmarks that are always present when interacting with a touch display. To explore the use of hands as landmarks, we designed two hand-centric techniques for multi-touch displays – one allowing 42 commands, and one allowing 160 – and tested them in an empirical comparison against standard tab widgets. We found that the small version (HandMark-Fingers) was significantly faster at all stages of use, and that the large version (HandMark-Multi) was slower at the start but equivalent to tabs after people gained experience with the technique. There was no difference in error rates, and participants strongly preferred both of the HandMark menus over tabs. We demonstrate that people's intimate knowledge of their hands can be the basis for fast and feasible interaction techniques that can improve the performance and usability of interactive tables and other multi-touch systems.

Author Keywords

Command selection; landmarks; multi-touch; tabletops.

ACM Classification Keywords

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

CHI'16, May 07 - 12, 2016, San Jose, CA, USA
Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3362-7/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858211

INTRODUCTION

Command selection on large multi-touch surfaces, such as tabletops, can be a difficult task. Selection techniques and widgets from desktop interfaces are often a poor match for the physical characteristics of a table – for example, menus or ribbons are typically placed at the edges of the screen, making them hard to reach on large displays, and hard to see on horizontal displays (due to the oblique angle to the user). As a result, researchers have proposed several techniques that bring tools closer to the user's work area, such as moveable palettes and toolsheets controlled by the non-dominant hand [8, 26], gestural commands [31], finger-count menus [5], or multi-touch marking menus [33]. These techniques can work well, but are limited in the number of commands that they can show (e.g., finger-count menus are limited to 25 commands, marking menus to about 64 [29]).

Part of the difficulty in developing new high-capacity selection techniques for large surfaces is that there are few landmarks that can help people learn the tool locations. Once a widget such as a tool palette is displayed on the screen, people can learn the locations of items by using visual landmarks in the palette (e.g., corners or colored items), but if the selection widget is hidden by default, the user must first invoke the menu before they can make use of this familiarity.

There is, however, a well-known landmark that is always present and visible to the user of a touch surface – their hands. People are intimately familiar with the size and shape of their hands, and proprioception allows people to easily locate features (e.g., touching your right index finger to the tip of your left thumb can be done without looking). This intimate knowledge of hands, however, is not exploited for command selection. For example, widgets such as tool palettes [26] are held by the non-dominant hand, but the palette does not use the details of the hand as a reference frame. Although people can use proprioception to bring a palette close to the selecting finger, the palette can be held in many different ways relative to the hand, and so any detailed familiarity with the tool locations is based mainly on the visual display of the palette. One technique that does use detailed knowledge of the hands is finger-count menus [5], which select commands based on the pattern of fingers touching the surface. This allows the development of proprioceptive memory for command invocation, but is limited to 25 commands, and does not make extensive use of people's familiarity with the size and shape of their hands.

To explore the use of people's hands as a landmarking technique for command selection, we developed and tested two hand-centric menu techniques for multi-touch displays. The first, HandMark-Finger, places command icons in the spaces between a user's spread-out fingers (Figure 1.1). This technique uses the hand as a clear external reference frame – once the locations of different items are learned, people can use their hand as a frame for setting up the selection action even before the fingers are placed on the touchscreen. The technique can be used with both hands to increase the number of available items.

The second technique, HandMark-Multi, provides multiple sets of commands, where the set is chosen by the number of fingers touching the surface (Figure 1.3). The technique is therefore similar to finger-count menus in the way that a category is selected, but allows many more items per category because a larger menu is displayed between the thumb and index finger (20 items in a 4x5 grid). HandMark-Multi also allows people to prepare for their selection before the hands are placed on the screen, once they have learned what menu an item is in and its location in the grid.

We carried out a study that compared HandMark menus to equivalent tab widgets presented at the top of the display. The study showed that HandMark-Finger was significantly faster than standard tabs (0.6 seconds per selection) with a similar error rate. The study also showed that although HandMark-Multi was slower than a tab UI in the early stages of use, there was no difference between the techniques as people gained experience. For both menus, it was clear that people did use their hands as a reference frame that aided memory of tool locations (e.g., people increasingly prepared their two hands for a correct selection as they gained experience). Participants also strongly preferred HandMark menus over the tab interfaces. Our work shows that the hands, and people's intimate knowledge of them, are an under-used resource that can improve the performance and usability of interfaces for tables and multi-touch systems.

HANDMARK DESIGN GOALS AND RELATED WORK

HandMark menus display command sets in specific places on the touch surface based on the sensed position of the left or right hand and the specific combination of fingers (see Figures 1.1 and 1.3). They are a design descendant of early bimanual techniques such as Palettes and Toolglasses [26], which allowed users to control a menu of tools with the non-dominant hand, and make selections with a pointing device in the dominant hand. This division allows one hand to act in a supporting role to the other (e.g., following Guiard's Kinematic Chain model [19]).

However, although techniques such as Toolglasses can improve performance compared to traditional selection widgets [8], they only allow users to build up a coarse understanding of the locations of specific commands in relation to the hand, and only when used with an absolute input space. The intent of HandMark menus is to go beyond the design of other multi-hand selection techniques, and use the hands as a more detailed absolute reference frame for developing memory of specific item locations. This allows people to remember commands using features on their hands, and allows them to position their hands and fingers for a selection even before the hands have touched the surface.

Design Goal 1: Rapid multi-touch command selection

A well-established method for improving selection speed is to enable memory-based command invocation rather than visually-guided navigation [10, 21, 22]. Researchers have used several mechanisms to enable memory-based interaction, such as spatial locations [23], gestures [32], multitouch chords [18] or hotkeys [37]. HandMark menus associate command icons with locations around the user's hands, so they use a spatial-memory mechanism – as users learn command locations, they can make selections using recall rather than visual search. Spatial memory is built up through interactions with a stable visual representation [13], and as people gain experience with a particular location, they can remember it easily. Studies have shown that people can quickly learn and retrieve command locations [15, 23, 39].

Multi-touch surfaces provide new opportunities for rich interaction and proprioceptive memory. For example, Wu and Balakrishnan [48] 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. Multi-touch marking menus [33] and finger-count menus [5] both allow users to specify a menu category by changing the number of fingers used to touch the screen. However, since a more-complex control action may take more time to retrieve and execute, these techniques do not always improve performance [27].

The efficiency of a command selection interface depends on the number of separate actions needed to find and execute a command. Using a full-screen overlay to display all commands at once, Scarr et al.'s CommandMap [41] successfully reduced the number of actions for desktop systems, an approach also used by the Hotbox technique [30]. Similarly, FastTap [23] uses chorded thumb and finger touches on a spatially stable grid interface to accelerate command selection for tablets. However, some of these techniques are difficult to use on large touch tables because the user can be at any location and any orientation, making it difficult to accurately position a visual representation.

Design Goal 2: Use hands as landmarks

Landmarks play a vital role in retrieval by providing a reference frame for other objects' locations. For example, the FastTap technique uses the corners and sides of a tablet's screen as the reference frame for organizing a grid menu [23]. However, on a large surface, these natural landmarks are not readily available (because people may be working in the middle of the screen and not near an edge or corner). In these situations, artificial visual landmarks can be useful to support spatial memory (e.g., Alexander et al.'s Footprints Scrollbar [1]); in addition, the visual layout of a toolbar can also show implicit landmarks, such as the corners and sides of the palette. Artificial visual landmarks can only be used once the toolbar is displayed, however.

In touch-based systems, there is another set of natural visual landmarks that are readily available and well known to the user – their hands. Therefore, we may be able to use hands and fingers as landmarks to support the development of spatial memory for item locations. There is considerable space around each hand and its fingers; if we use that space to represent command items, people can use their knowledge of their hands' shapes and sizes to remember those locations. In addition, the hands are a natural reference frame that is always visible, meaning that users can prepare for a selection even before they touch the surface. For example, if a command is stored near the user's left thumb, they can move their selection finger near to the thumb as they touch down on the surface, potentially reducing selection time.

Numerous other selection techniques have also used the hands in some fashion. As described above, bimanual techniques like Toolglasses [8] and Palettes [26] use one hand to control a palette's position and other hand to select. However, these techniques differ from HandMarks in that they do not use the details of the hand as a reference frame. In the original version, the palette was controlled by an indirect pointing device [26], so the hand was not visible at all; and when used with touch surfaces, the way in which the user holds the palette can change (thus changing the frame). Users can use proprioception at a coarse level (e.g., to quickly bring the tools to the work area and orient them appropriately, but there is no detailed mapping between commands and specific hand locations.

Other techniques also use proprioceptive memory of the hands as a non-visual reference frame. For example, Finger Count menus [5] rely on people's memory of finger patterns, and other systems use multi-finger chords to represent commands [18, 46]. Finally, although not intended for table-based interactions, techniques such as Imaginary Interfaces [21], Body Mnemonics [2], and Virtual Shelves [35] also rely on proprioceptive memory for command selection.

Design Goal 3: Hand detection

In order to use hands as the landmarks for a menu, we need to know the shape and orientation of the hand once it has touched the surface. Earlier work has explored hand detection using several methods: computer vision approaches, specialized hardware, and glove-based tracking. Several systems use computer vision to track the position of hands and to identify fingers [3, 16, 34]. Another uses distance, orientation, and movement information of touch blobs to identify fingers and people [12, 47]. Schmidt et al's HandsDown [42] system allows hand detection on tabletops, and provides lenses for interaction [43]. The reliability and accuracy of vision-based recognition, however, remains a challenge for all of these systems.

Other methods use specialized hardware to distinguish between hand parts and between users. For example, the DiamondTouch system [14] uses capacitive coupling to identify different users. Other hardware approaches distinguish hand parts: for example, an EMG muscle-sensing armband identifies a person's fingers touching a surface [7], while fingerprint recognition could provide similarly precise touch information and user identification [25]. Other techniques distinguish a user's hands and their posture in space by using colored patches or markers on gloves [9, 48].

As described below, we developed a new hand identification technique for HandMark menus that does not use either vision or specialized hardware, and relies only on the touch points that are reported by a multi-touch surface.

Design Goal 4: Support a large number of commands

Many memory-based command selection interfaces provide a limited number of commands. For example, FastTap supports only 19 commands [23], and Finger Count menus [5] provide only 25. Several approaches have been used to increase the number of commands in selection techniques. Marking Menus uses multiple levels to provide more commands (allowing about eight items per level [32]); other techniques such as Polygon menus [49], Flower menus [4], Augmented letters [40], Gesture avatar [36], Arpège [18], FlowMenu [20] and OctoPocus [6] increase the command vocabulary by expanding the range of gestures.

For HandMark menus, rapid execution is our priority, but we also want to support a large command vocabulary. Our prototypes place as many items as possible around the hands, while still ensuring that hands and fingers can be used as landmarks to facilitate rapid development of spatial memory.

HANDMARK DESIGN AND IMPLEMENTATION

We developed two variants of the HandMark technique to explore different kinds of hand-based landmarks and different menu sizes.

Design 1: HandMark-Finger

This technique provides modal access to two different sets of commands, each belonging to one hand (Figures 1.1 and 2). To access commands, the five fingers of the left or right hand are touched down in any order, spreading the hand to provide space between the fingers.

Commands are displayed in the space around the hand and between the fingers (Figure 2), and selections are made by touching an item with the other hand. We place pairs of icons between fingers, and one command at the top of each finger. As the space between the thumb and index finger is larger, we place eight commands there in a 4x2 grid. The size of the grid was determined using the average width of an adult index finger (16-20mm [11]) as a guideline and considering Parhi et al.'s recommendation that touch targets be no smaller than 9.6mm [38]. In total, HandMark-Finger supports 42 items (21 in each hand).

The user can rotate and move the menu in any direction. Following a hand touch, the menu appears after a short 300ms delay, but selections can be made immediately. This enables two types of selections. Novice users can wait until the menu appears and use visual search to select a target. Expert users, who have built up spatial memory of the location of a desired item, can tap the location without waiting for the menu to be displayed (Figure 1.2). This follows Kurtenbach's principle of rehearsal, which states that novice actions should be a rehearsal of the expert mode [28].

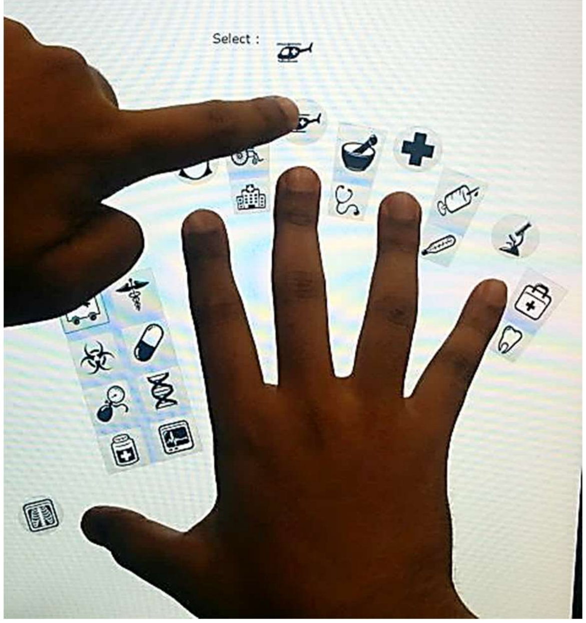

Figure 2. Making a selection with HandMark-Finger.

Design 2: HandMark-Multi

This interface also provides modal access to different sets of commands (Figures 1.3 and 3) and has a similar selection method to that described above. In HM-Multi, however, there are eight command sets (four in each hand) and each set can be accessed by touching on the screen with a specific number of fingers and thumb in an L-shaped posture (see Figure 3). The index finger and thumb are always used, and adding other fingers accesses other sets – e.g., to access the second set on the left hand, the index and middle fingers of the left hand are touched down along with the thumb.

A spatially-stable grid of items is then shown in the space between the thumb and index finger (Figure 3). We placed 20 commands (a 5x4 grid) in the space between thumb and index finger [11, 38]. Since these two fingers are always used to frame the grid, we can provide four sets in total (the first uses only thumb and index finger, and the others add the middle, ring, and pinky fingers). HandMark-Multi supports 160 items (20 in each tab, and 4 tabs in each hand). The menu follows the user's hand as it moves or rotates on the screen. Handmark-Multi also supports the novice and expert selection methods described for HandMark-Finger above.

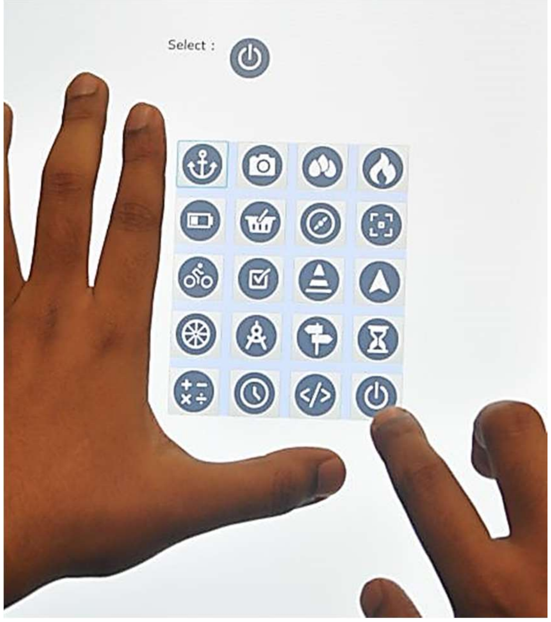

Figure 3. Making a selection with HandMark-Multi.

Hand identification

HandMark requires accurate identification of the left and right hand using only the fingers' touch points. We make use of the distinctive geometries of people's hands in terms of the position of the thumb compared to other fingers and the individual positions of the fingers compared to the thumb. For example, the position of thumb is always below the other fingers if the hand points upwards, and the rightmost touch is always the thumb for the left hand (and reversed for the right). Using these features, we are reliably able to differentiate the left and right hand. Other fingers (index, middle, ring, and pinky) can be found from the touch points once the hand and thumb are identified.

The algorithm we use is as follows. For each set of points touched down simultaneously, determine whether the rightmost or leftmost point is lower than the others in the set. Identify this as the thumb (which also determines the left or right hand). The remaining points can then be identified using left-to-right ordering for the right hand, and right-to-left for the left hand. This algorithm requires that users place the fingers of one hand (all five fingers for HM-Finger, and at least two for HM-Multi) on the surface in an approximately upright posture, and at approximately the same time (but in any order). Other finger-identification techniques exist that are more robust (see Vogel [45]), but our simplistic approach works well for the prototypes described here.

In-place tools and occlusion of content

All in-place interfaces occlude parts of the work surface [44] (e.g., pop-up menus) or the whole screen (e.g., FastTap). For HandMark menus we chose a hybrid overlay presentation – when used in novice mode, the menu covers part of the screen, but in expert mode, no visual presentation is needed. In addition, it is easy for the user to control the presence of the overlay (by lifting the fingers from the touch surface), allowing rapid switching between menu and content. It is also easy to move the menu hand after activating the menu, which allows the user to further manage occlusion.

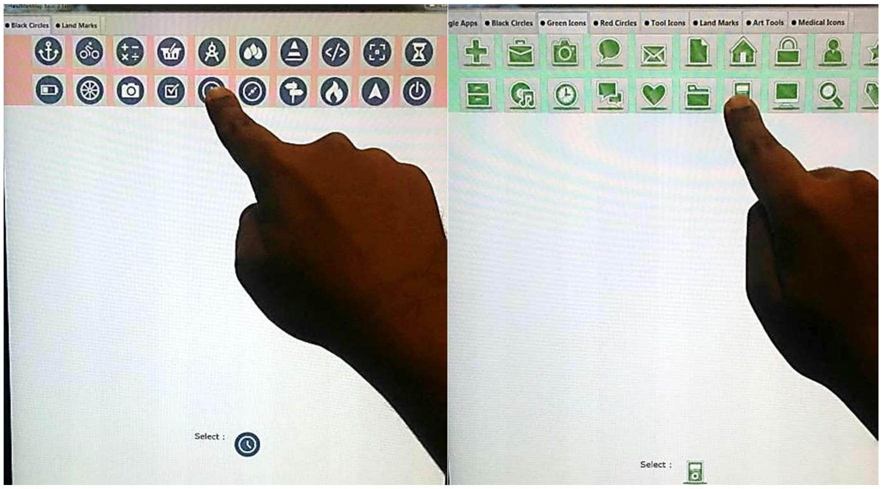

Figure 5. Left: Tabs-2, Right: Tabs-8.

EXPERIMENT

To assess the performance of command selection using hands as landmarks, we conducted a study comparing HandMark menus to standard tab-based menus. We compared the interfaces in a controlled experiment where participants selected a series of commands over several blocks, allowing us to examine selection behaviors and learning in each interface.

Experimental Conditions

Two versions of HandMark menus, and two equivalent versions of a standard tab interface were implemented in a tabletop environment (see Figures 2, 3 and 5).

HandMark-Finger was implemented as described above. The interface used in the experiment contained 21 commands in each hand's set, for a total of 42 items. Eight items were used as study targets – four from each hand (Figure 4).

HandMark-Multi was also implemented as described above. There were 20 command buttons in a 5x4 grid for each set. There were eight sets (grouped by color) for a total of 160 command buttons. Eight targets were used in the study, one from each set (Figure 4 shows command locations within the grid; note that each command was from a different set).

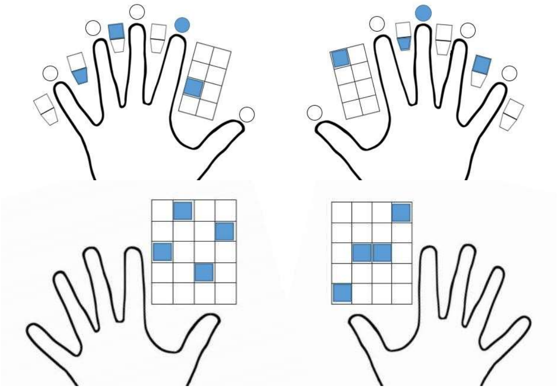

Figure 4. Target locations for HM-Finger and HM-Multi (collapsed across different command sets).

Standard tab interfaces. We implemented two versions of standard tabbed ribbon interface (Tabs-2 and Tabs-8) to compare with the two HandMark menus. Tabs-2 (Figure 5 left) had only two tabs (each consisting of 20 command buttons) to match HandMark-Finger. For Tabs-8 (Figure 5 right), there were eight tabs each with 20 items in a 2x10 grid (total of 160). Items were grouped by type and color, and the named tabs were placed side by side as a ribbon interface at the top left edge of the screen.

We compared HandMarks to Tabs rather than other research systems for several reasons: Tabs offer equitable command range to our prototypes (which is not provided by several research techniques), and they are the de facto standard UI; in addition, a main goal of the evaluation was to compare the strong landmarking and proprioceptive approach of HandMarks to a traditional visually-guided approach. In future work we will also extend the comparisons to other systems such as Marking Menus and other recent designs

For all interfaces (and both expert and novice mode), feedback was shown for 300ms after a command was selected by displaying the icon in its home location.

Procedure

The study was divided into two parts. Part 1 tested HandMark-Finger and Tabs-2, and part 2 tested HandMark-Multi and Tabs-8. Participants completed a demographics questionnaire, and then performed a sequence of selections in the custom study system with both interfaces. For each version, a command stimulus (one of eight icons, Figure 4) was displayed in the middle of the screen; the participants had to tap one large (easily accessible) button placed at bottom to view the command stimulus and start the trial. Trials were timed from the appearance of the stimulus until that icon was correctly selected. Participants were instructed to complete tasks as quickly and accurately as possible, and were told that errors could be corrected simply by selecting the correct item. In our analysis, we include error correction in completion times.

The study was carried out using a 24-inch multitouch monitor placed flat on a table in front of the participant in portrait mode. Although this is not a large-scale surface, it adequately simulated the combination of a local work area and a far edge that participants needed to reach in order to use. Participants were stationed at a fixed seat and allowed to lean forward for selecting items using both hands.

For both interfaces, only eight commands were used as stimuli, in order to allow faster development of spatial memory. For each interface, selections were organized into blocks of eight trials. Participants first performed one practice session which was consisted of two commands and ten blocks (data discarded) to ensure that they could use the interfaces successfully. They then carried out 17 blocks of eight selections each. Targets were presented in random order (sampling without replacement) for each block. After each interface, participants were allowed to rest, and filled out a questionnaire based on the NASA-TLX survey [24]. At the end of each pair of techniques, participants gave their preferences between two the systems. Order of the interfaces in each part, along with the order of study parts was counterbalanced using a Latin square design.

Participants and Apparatus

Fourteen participants were recruited from a local university; one person's data could not be used due to technical difficulties, leaving 13 participants (6 female; mean age 24 years). The study was conducted on a Dell multitouch monitor (24-inch screen, 1920x1080 resolution) and a Windows 7 PC. The interfaces were written in JavaFx, and the study software recorded all experimental data including selection times, errors, and incorrect set selections.

Design and Hypotheses

The study used 2×17 within-participants RM-ANOVAs; with factors *Interface* (HandMark-Finger vs. Tabs-2; and HandMark-Multi vs. Tabs-8), and *Block* (1-17). Dependent measures were selection time per command, errors per block, and incorrect tabs per block. Interfaces and sets were counterbalanced. Hypotheses were:

H1. Selection will be faster for *HandMark* than for *Tabs.*

H2. *HandMark* will be faster both for novices and experts.

H3. There will be no evidence of a difference in error rates between *HandMark* and *Tabs*.

H4. There will be no evidence of a difference in selecting the wrong set between *HandMark* and *Tabs*.

H5. There will be no evidence of a difference in perception of effort for *HandMark* and *Tabs*.

H6. Users will prefer *HandMark* over *Tabs*.

Results: HandMark-Finger vs. Tabs-2

Selection Time per Command

We calculated mean selection time for each command by dividing the total trial time by the number of commands in that block. Mean selection times were 0.62 seconds faster per command with *HandMark-Finger* (2.32, s.d. 0.79s) than with *Tabs-2* (2.94s, s.d. 0.95s), see Figure 6.

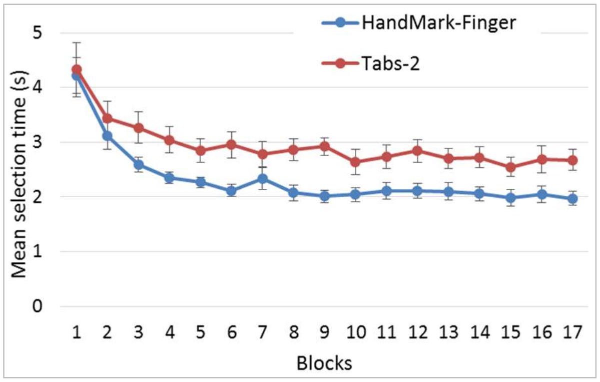

Figure 6. Mean selection time by Interface and Block.

RM-ANOVA showed a significant main effect of *Interface* ($F_{1,12}$=37.59, *p*<.0001). For the small menus, we therefore accept **H1** – HandMark-Finger was 21% faster than Tabs-2. As shown in Figure 6, selection times decreased across trial blocks for both interfaces; RM-ANOVA showed a significant effect of *Block* ($F_{16,192}$=18.04, *p*<.0001). There was no interaction between *Interface* and *Block* ($F_{16,192}$=1.00, *p*=.456) as HandMark-Finger was faster than Tabs-2 throughout. For HandMark-Finger, we therefore accept **H2**.

Errors

We also analyzed errors per command (counted as any incorrect selection). RM-ANOVA showed no effect of *Interface* on errors, with HandMark-Finger at 0.04 errors/command, s.d. 0.09, and *Tabs-2* at 0.04 errors/command, s.d. 0.08 ($F_{1,12}$=0.01, *p*=.924). We therefore accept **H3** (errors are considered further below). There were no effects of *Block* ($F_{16,192}$=1.07, *p*=.388) on errors, and no *Interface* x *Block* interaction ($F_{16,192}$=1.42, *p*=.138).

Incorrect Set Selection

We also recorded the number of times participants selected the wrong command set (e.g., the wrong tab or the wrong hand/finger combination). RM-ANOVA showed no effect of *Interface* ($F_{1,12}$=0.26, *p*=.623) on errors, with 0.05 incorrect sets/command, s.d. 0.09 for both *HandMark-Finger* and *Tabs-2*. There was also no effect of *Block* ($F_{16,192}$=0.6, *p*=.88). Therefore we accept **H4** for HandMark-Finger.

Subjective Responses: Effort and Preferences

Participants' responses were positive for both interfaces, but there were no strong differences in NASA-TLX scores (Friedman test, Table 1) for HandMark-Finger and Tabs-2. There were no significant differences on any question, and the mean scores were similar. Therefore, we accept **H5**.

	HandMark-Finger	Tabs-2	χ_r^2	p
Mental	5.54(2.73)	5.46(2.37)	0.08	0.78
Physical	5.38(2.79)	6.62(2.47)	1.23	0.27
Temporal	5.00(2.89)	4.77(1.74)	0.08	0.78
Performance	8.69(0.95)	6.92(1.93)	7.69	0.01
Effort	5.31(2.75)	6.77(2.35)	0.31	0.58
Frustration	2.00(1.78)	4.69(3.01)	4.92	0.03

Table 1. Mean (s.d.) effort scores (0-10 scale, low to high).

We also asked participants about their preferred interface in terms of several qualities (Table 2). Counts were easily distinguishable, and overall, 92% of participants preferred HandMark-Finger. We therefore accept **H6.**

	HandMark-Finger	Neither	Tabs-2
Speed	12	0	1
Accuracy	9	2	2
Memorization	11	2	0
Comfort	12	0	1
Overall	12	0	1

Table 2. Counts of participant preferences.

Results: HandMark-Multi vs. Tabs-8

Selection Time per Command

Mean selection times were 0.62 sec/command slower with *HandMark-Multi* (3.84s, s.d. 2.1s) than with *Tabs-8* (3.22s, s.d. 1.43s), giving a main effect of *Interface* ($F_{1,12}$=4.86, p=.048). However, this result must be interpreted in light of the significant interaction between *Interface* and *Block* ($F_{16,192}$=4.96, p<.0001). In early blocks, Tabs-8 was faster than HM-Multi, but by the final four blocks, the two techniques were similar (RM-ANOVA for blocks 14-17 showed no significant effect of *Interface*, $F_{1,12}$=.008, p=.932). Hypotheses **H1** and **H2** therefore cannot be clearly rejected – HM-Multi was slower overall, but there was no difference in performance once users learned item locations.

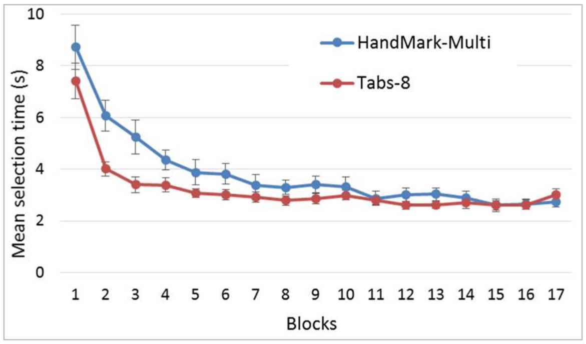

Figure 7. Mean selection time by Interface and Block.

Errors

RM-ANOVA showed similar trends to the smaller menus: *HandMark-Multi* had 0.06 errors/command, s.d. 0.1, and *Tabs-8* had 0.04 errors/command, s.d. 0.09, with no main effect ($F_{1,12}$=2.47, p=.142). We therefore accept **H3** (errors are considered further below). There was no effect of *Block* ($F_{16,192}$=1.6, p=.07), and no interaction ($F_{16,192}$=.75, p=.74).

Incorrect Set Selection

RM-ANOVA showed a different trend to the smaller menus: *HandMark-Multi* had more incorrect set selections (0.64 per command, s.d. 0.86) than *Tabs-8* (0.18 per command, s.d. 0.38) ($F_{1,12}$=9.78, p<.01). There was also a significant interaction between *Interface* and *Block* ($F_{16,192}$=6.51, p<.0001). We therefore reject **H4** for HandMark-Multi.

Subjective Responses: Effort and Preferences

Again, participants gave positive responses for both interfaces, with no significant differences (Friedman test, Table 3) between HandMark-Multi and Tabs-8. Mean scores were close in all cases; therefore, we accept **H5**.

Participant preference counts (Table 4) were again easily distinguishable, with a strong preference for HandMark-Multi (73% overall). We therefore accept **H6**.

	HandMark-Multi	Tabs-8	χ_r^2	*p*
Mental	7.62(2.36)	6.23(1.83)	2.77	0.10
Physical	6.00(2.71)	7.23(2.13)	1.23	0.27
Temporal	6.46(2.76)	6.31(2.06)	0.08	0.78
Performance	7.62(1.5)	7.54(2.26)	0.69	0.41
Effort	7.00(2.45)	7.54(1.61)	0.08	0.78
Frustration	3.69(2.02)	4.08(2.5)	1.23	0.27

Table 3. Mean (s.d.) effort scores (0-10 scale, low to high).

	HandMark-Multi	Neither	Tabs-8
Speed	10	0	3
Accuracy	6	2	5
Memorization	12	1	0
Comfort	10	1	2
Overall	10	1	2

Table 4. Counts of participant preferences.

Use of Hands as Landmarks

To consider whether participants made use of their hands as landmarks, we analyzed the number of selections made without any visual feedback (meaning that people used only their hands as a reference for selection) and the performance of different locations around the hand.

Selection with no visual feedback

We recorded the number of selections made without waiting the 300ms for visual feedback (i.e., "expert mode"). For both types of HandMark menu, selection without feedback started near zero in the early blocks, but increased to approximately 8% of selections in the final block.

In addition, experimenter's informal visual observations showed that all users moved their selection finger towards the correct region on the menu hand even before the menu hand was placed. That is, even when people did wait for system feedback, they were preparing for a correct selection by correctly positioning their finger before the menu was displayed. These preparatory actions suggest that people were developing proprioceptive memory and were remembering the mapping of commands to hand locations.

Performance of Selection by Target Location

We also analyzed selection time and expert mode use by target location. For HandMark-Finger, three areas were defined: finger-top, between-fingers and near-thumb. RM-ANOVA showed finger-top locations were better than others with 0.03 expert-selections/command, s.d. 0.12 ($F_{2,24}$=1.41, p=.26) and faster command selection (mean 2.3s, s.d. 0.1) ($F_{2,24}$=0.74, p=.5). For HandMark-Multi, two areas were defined: close and far from index and thumb. Here, the targets located close to index and thumb performed best – for these better-landmarked locations, selections were faster and expert mode was used more (0.08 selections/command, s.d. 0.16) with a significant main effect ($F_{1,12}$=6.67, p<.05).

Participant Comments

Participant comments followed the pattern of preference results. Participants made several comments on how spatial stability and quick activation using both hands helped the speed of HandMark menus: one participant said *"Really neat technique that allows you to browse through different tabs based on the number of fingers."* One person, however, remarked on the difficulty of remembering the different hands' sets: *"It was difficult to remember which hand has the right kind of tool."*

Other comments suggested that the HandMark interfaces helped participants to learn command locations: one said *"It was easier to remember where icons were relative to spaces on fingers instead of just on the tabs."* Another said "*[HandMark was] easier to use and faster to remember,*" and another stated "*[Tabs were] more difficult to memorize.*"

Some participants stated that they were initially concerned with slow memorization in HandMark-Multi, but eventually preferred it. One person stated "*Remembering was slightly slower at early stage but in a short amount of time it became quite strong and it became easier to answer.*" This participant also stated that the Tab interface *"was fast at the beginning but it could not build up strong memory and hence it became tough to apply them later."*

DISCUSSION

Our study of HandMark menus provides six main results:

- HandMark-Finger was significantly faster than Tabs-2 (0.62s/selection), and was faster in all blocks.
- HandMark-Multi was slower overall than Tabs-8 (0.62s per selection), but only in the early blocks.
- The only difference in errors between the two approaches was that HM-Multi had a slightly higher rate than Tabs-8.
- There were no significant differences for perceived effort between interfaces in either pair, but most participants preferred both HandMark menu types.

Explanation and Interpretation of Results

Performance analysis of HandMark menus

The study showed that HandMark-Finger was faster than a visually-guided tab menu (at all stages) and HandMark-Multi was slower (although only during early use). There are a few reasons for these results, based on the command selection steps for both novice and expert. For novices, there are three steps needed: invoking the correct command set, searching for the target command, and executing a selection action. Invoking the menu was different for both interfaces. For HandMark-Finger, invocation involves pressing with all five fingers anywhere on the touch surface, which was easy and fast. In contrast, the Tabs-2 required touching at a specific position (the tab buttons at the top of the screen). Participants had to reach further for the tab interface, and had to be more precise in their selections because of the lower angle of the display. These factors likely led to additional time to invoke the tab interface.

For HandMark-Multi, displaying a command set involves different combinations of fingers. With eight sets, there were eight different combinations of fingers. Novices spent a large amount of time determining which finger combination belonged to which set; in contrast, Tabs-8 showed names and specific positions for each tab. Even though novices also had to spend time searching through the different tabs, the visual presentation allowed people to better organize their task.

Searching for a specific command within a set required a similar strategy for all interfaces. The visual search needed for HandMark-Finger could take longer initially, since the commands are shown at different places around the hand compared to the grid presentation of Tabs-2. In particular, in some cases people's hands could partially occlude the menu items. However, as people became more experienced, the additional landmarks in HandMark-Finger appeared to help people with retrieval [23]. HandMark-Multi also supports development of spatial memory, although the grid of items does not contain as many landmarks as HandMark-Finger, and so there was not as large an advantage in spatial memory as was seen in the smaller menu.

The final step – executing the selection action – was similar for all interfaces, although accurate touches appeared to be more difficult with Tabs due to the screen's oblique angle.

For experts, selection in HandMark menus requires only two steps: retrieval of the command's set and location from memory, and execution of the selection action at any position on the touch surface. The lack of a spatial reference frame for Tabs, however, means that users must still perform some degree of visual search, even when they are familiar with the location. The performance advantage for expert use of HandMark menus arises in the speed of execution, which can be achieved by chording the menu choice with the selection.

In addition, the amount of time taken for reaching to the tabs (at the top edge of the display) was a substantial component of the overall performance of the Tabs technique. This is not the case for HandMark menus, since these are always invoked close to the user. Therefore, the performance difference between a tab interface and HandMarks will to some degree depend on where the tabs are located – as tabs become further away, the performance advantage for HandMarks will increase. In future work we will also compare against palette-based techniques like Toolglasses [8]: these methods bring tools close to the work area, but often require that the user execute additional actions (e.g., to grab the palette and drag it near to the selection hand).

Error rates with HandMark

Error rates per command were high in all the techniques: 4% for both HandMark-Finger and Tabs-2; 6% and 4% for HandMark-Multi and Tabs-8. This high error rate might be an artifact of our experimental protocol, which instructed participants to select commands quickly, and noted that errors could be corrected afterwards. There are other possible explanations for the error rates, however. First, the quick execution of a selection in all the interfaces may have encouraged participants to view errors as amenable to rapid correction, thereby encouraging users towards a 'guess and correct' mode of operation [23].

Second, it is possible that people's memory of a command's spatial location was imperfect, and so participants may have experienced 'near misses' more often with larger sets (HandMark-Multi and Tabs-8). Third, it is very easy to touch down with fingers on touch surface but, it is somewhat more difficult to change the combination of fingers very quickly, which sometimes caused unintentional touches for HandMark-Multi. Last, for both Tabs-2 and Tabs-8, the oblique viewing angle may also have increased errors. Further work is needed to explore these sources of error, and to determine whether the high error rates for both techniques occur in real-world use.

Command-set browsing with HandMark

Incorrect command-set selections were also relatively high for all the interfaces: 0.05 sets/command for both HandMark-Finger and Tabs-2; 0.64 and 0.18 tabs/command for HandMark-Multi and Tabs-8. As the number of tabs was smaller for the first pair, participants made fewer errors; in the larger menus, the rate was considerably higher. This can be explained by the larger number of items, and the increased need for visual search overall. As described above, the visual representation of the tabs in the standard interface may have allowed participants to better organize their visual search, whereas people's search in HandMark-Multi was often poorly organized. This indicates one disadvantage of hand-centric interfaces – information such as the name of the set cannot be shown on the reference frame (i.e., the hand).

An additional reason for differences in set-selection errors is the physical position of fingers in HandMark-Multi. As fingers are very close to one another, people sometimes touched the wrong finger onto the surface. More work is needed to evaluate the ergonomic and effort characteristics of different hand and finger combinations – it may be that a smaller number of menus (using only the easy-to-produce finger combinations) will improve browsing performance.

Real-world use of HandMarks

The two HandMark prototypes represent a tradeoff between landmarks and command capacity – HM-Finger makes more extensive use of the hands and is faster overall, but is limited in size; HM-Multi can accommodate more commands, but overloads one region (between thumb and index finger).

Although further studies with the techniques are needed, we speculate that people will be more successful at learning locations with HM-Finger, due to its richer landmarking, and therefore more likely to use the expert selection mode in real-world use. We are also interested in how continued use will change the ways that people carry out preparatory actions – for example, once the locations around the hand are learned, it is possible to use tactile feedback alone to prepare for a selection (e.g., register the pointing finger between two fingers of the menu hand, and then touch the surface).

Although HandMark-Multi does not have the same rich landmarks, the consistent presentation of command sets between thumb and index finger is still likely to be valuable in real-world use. HM-Multi can be considered as a version of earlier Palette techniques, but with the tool items always presented using a consistent spatial reference frame.

In future work we will also compare HandMark menus to other recent designs, such as Marking Menus [32], Flower Menus [4], and Arpège [18]. One area of particular interest is how the different approaches support transitions from novice to expert use – both HandMarks and Marking Menus, for example, are based on the principle of rehearsing expert actions in the novice mode; in other gesture-based and chord-based systems such as Arpège, learning the commands requires explicit training.

Limitations and design possibilities

Our investigation of HandMark menus considers only a portion of the likely issues present in real-world use, and there are several ways in which the HandMark techniques can be extended in future.

Hand postures and ergonomics. A few participants reported difficulties with the finger combinations required to choose different command sets with HandMark-Multi. Participants noted that changing quickly between sets was initially difficult as it required good finger dexterity (even though all of our hand postures are "relaxed" [18]). These initial problems were quickly overcome, and it was seen as helpful that the menus move and adapt as the hands are moved, allowing users to choose comfortable hand positions. Finally, two participants had longer fingernails in our study, but they did not have difficulty using HandMark menus.

Mapping of commands to menus and locations. In our study, we arbitrarily assigned commands to locations, and command sets to different hands – in real use, performance and learning could likely be substantially improved with a more thoughtful mapping. For example, with HandMark-Multi, both the left and right hands "held" four different command sets, and some participants initially had difficulty remembering which hand contained their desired set. The study suggested that choosing the wrong hand was a costly error, as participants tended to check each of the sets on that hand before trying the other hand. Therefore, further work is needed to determine how menu contents are best mapped to different hands. In addition, since some locations around the hand appeared to be faster and easier to learn, frequent or important commands could be assigned to these locations.

Occlusion of the menu. HM-Finger shows items between fingers, so it is possible for the hand itself to occlude the menu, particularly if the hand is not directly in front of the user. In the study, some participants with smaller fingers occasionally experienced this problem. In future work we will explore solutions to this problem, such as better determination of the actual shape of the hand, automatically scaling the icons for different hand sizes, or moving the icons upwards (while still maintaining relative spatial positioning) if there is not enough space between fingers. Previous work by Vogel and colleagues [44] has shown that unrestricted models of hand occlusion can be inferred from touch points, so we are confident that our technique can be extended to the general usage case. The problem of occlusion primarily affects the learning stages, however, when users are still using the visual guidance of the displayed menu; once item locations are known, users can position their selection finger using the hand rather than the display.

Multiple users and different orientations. The prototype system supported only one person at a fixed location, and further work is needed to determine how the hand detection technique will perform with multiple hands and with hands at any orientation. We believe that our hand-posture algorithms can handle these additional demands with additional sensing of the environment, such as finger-contact areas and shapes, or a depth camera that can track each person's approximate location around the table [17].

Increasing the number of commands. Our prototypes explored two command-set sizes (42 for HandMark-Finger, and 160 for HandMark-Multi). It is possible to increase these numbers (e.g., by having a larger grid between thumb and index finger [11, 38], by stacking additional layers above the fingers in HM-Finger, or by using the different positions in HM-Finger as triggers for second-level sets). However, further work is needed to determine whether larger command sets are beneficial – for example, our study showed that initial learning was more difficult for the multiple sets in HM-Multi, and it may be advantageous to restrict the number of commands to improve learnability.

Indicating hand menu contents. One problem identified in the study was that HandMark-Multi does not provide any visual indication of the mapping between finger combinations and command sets in order to assist users who are in the novice stages of learning. A visual "map legend" could be shown on the display as a reminder, but it would also be possible to use augmented-reality techniques to show menu contents (e.g., project symbols on the display near the hands above the table, or on the hands themselves).

Device orientation and size. In our experiment we used a relatively small tabletop touch surface (24-inch diagonal). We believe that this setup reasonably approximates the actions that will be needed on a larger table, but we will confirm this in future studies with larger surfaces. In addition, we are interested in how HandMarks will work on smaller surfaces such as tablets. Many devices are now large enough to accommodate a whole hand, and a technique such as HandMark-Finger could be successful on smaller devices (although comparisons will be needed to other techniques such as FastTap and bezel-based interaction).

Advanced interactions. Most interfaces include widgets that are more advanced than buttons – for example, sliders or color pickers can be used to provide a finer degree of control over application parameters. We will explore how these kinds of widgets can be converted to work with HandMark menus – for example, HandMark could be adapted to use a Toolglass-style interaction in which users click through the command and start their manipulation at the same time. It may also be possible to combine HandMarks with other gesture-based techniques such as marking menus [29]. For example, people could activate different command modes with one hand and perform gestures with the other.

Single-handed use. It is not currently possible to use HandMark menus with one hand – both hands are required for novice and expert mode. The two-handed nature of the technique increases interaction bandwidth, and is one of the reasons why it performed well. However, it would be possible to modify the technique to be usable with one hand. Although chorded operation would not be possible, it would be relatively easy to have the menu stay present for a short time after being invoked – this would allow people to invoke the menu and then select with the same hand (still being able to take advantage of spatial memory).

CONCLUSIONS

Command selection on large multi-touch surfaces can be difficult, because techniques are not well suited to the display setting, and because the lack of landmarks makes it harder for users to build up familiarity with spatial locations. People's hands are always present in the workspace, however, and can be used as a reference frame for designing touch-based selection techniques. A few techniques take advantage of hands, but often these methods are limited in the number of items they can accommodate.

We designed two hand-centric techniques for multitouch displays – one allowing 42 commands, and one allowing 160 – and tested them in an empirical comparison against standard tab widgets. We found that the small version (HandMark-Finger) was significantly faster at all stages of use, and that the large version (HandMark-Multi) was slower at the start but equivalent to tabs after people gained experience with the technique. Participants strongly preferred both of the HandMark menus over tabs.

Our work shows that the hands, and people's intimate knowledge of them, are an under-used resource for interaction. We demonstrate that hand-centric interfaces are feasible, can be faster than standard techniques, and are preferred by users. Techniques using hands as landmarks can improve the performance and usability of interfaces for tables and other multi-touch systems.

ACKNOWLEDGMENTS

This work was supported by NSERC and the Surfnet research network. Our thanks to Andy Cockburn and to the anonymous referees for valuable comments and suggestions.

REFERENCES

1. Jason Alexander, Andy Cockburn, Stephen Fitchett, Carl Gutwin, and Saul Greenberg. 2009. Revisiting read wear. In *Proceedings of the 27th international conference on Human factors in computing systems* (CHI '09), 1665–1674. http://dx.doi.org/10.1145/1518701.1518957
2. Jussi Ängeslevä, Ian Oakley, Stephen Hughes and Sile O'Modhrain. 2003. Body Mnemonics - Portable device interaction design concept. In *Proceedings of the 16th annual ACM symposium on User interface software and technology* (UIST '03).
3. Oscar Kin-Chung Au and Chiew-Lan Tai. 2010. Multitouch finger registration and its applications. In *Proceedings of the 22nd Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction* (OZCHI '10), 41-48. http://dx.doi.org/10.1145/1952222.1952233
4. Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2008. Flower menus. In *Proceedings of the working conference on Advanced visual interfaces* (AVI '08), 15-22. http://dx.doi.org/10.1145/1385569.1385575
5. Gilles Bailly, Jörg Müller, and Eric Lecolinet. 2012. Design and evaluation of finger-count interaction: Combining multitouch gestures and menus. *Int. J. Hum.-Comput. Stud.* 70, 10 (October 2012), 673-689. http://dx.doi.org/10.1016/j.ijhcs.2012.05.006
6. Olivier Bau and Wendy E. Mackay. 2008. OctoPocus. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (UIST '08), 37-46. http://dx.doi.org/10.1145/1449715.1449724
7. Hrvoje Benko, T. Scott Saponas, Dan Morris, and Desney Tan. 2009. Enhancing input on and above the interactive surface with muscle sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (ITS '09), 93-100. http://dx.doi.org/10.1145/1731903.1731924
8. Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, Tony D. DeRose. 1993. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (SIGGRAPH '93), 73-80.
9. Volkert Buchmann, Stephen Violich, Mark Billinghurst, and Andy Cockburn. 2004. FingARtips. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Austalasia and Southe East Asia* (GRAPHITE '04), 212-221. http://dx.doi.org/10.1145/988834.988871
10. Andy Cockburn, Philip Quinn, Carl Gutwin, Gonzalo Ramos, and Julian Looser. 2011. Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback. *Int. J. Hum.-Comput. Stud.* 69, 6 (June 2011), 401-414. http://dx.doi.org/10.1016/j.ijhcs.2011.02.005
11. Kiran Dandekar, Balasundar I. Raju, and Mandayam A. Srinivasan. 2003. 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. http://dx.doi.org/10.1115/1.1613673
12. Chi Tai Dang, Martin Straub, and Elisabeth André. 2009. Hand distinction for multi-touch tabletop interaction. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (ITS '09), 101-108. http://doi.acm.org/10.1145/1731903.1731925
13. Rudolph P. Darken and John L. Sibert. 1996. Wayfinding strategies and behaviors in large virtual worlds. In *Proceedings of the SIGCHI conference on Human factors in computing systems common ground* (CHI '96), 142-150. http://dx.doi.org/10.1145/238386.238459
14. Paul Dietz and Darren Leigh. 2001. DiamondTouch. In *Proceedings of the 14th annual ACM symposium on User interface software and technology* (UIST '01), 219-226. http://dx.doi.org/10.1145/502348.502389
15. Brian D. Ehret. 2002. Learning where to look. In *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves* (CHI '02), 211-218. http://dx.doi.org/10.1145/503376.503414
16. Philipp Ewerling, Alexander Kulik, and Bernd Froehlich. 2012. Finger and hand detection for multi-touch interfaces based on maximally stable extremal regions. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces* (ITS '12, 173-182. http://dx.doi.org/10.1145/2396636.2396663
17. Aaron M. Genest, Carl Gutwin, Anthony Tang, Michael Kalyn, and Zenja Ivkovic. 2013. KinectArms: a toolkit for capturing and displaying arm embodiments in distributed tabletop groupware. In *Proceedings of the 2013 conference on Computer supported cooperative work* (CSCW '13, 157-166. http://dx.doi.org/10.1145/2441776.2441796
18. Emilien Ghomi, Stéphane Huot, Olivier Bau, Michel Beaudouin-Lafon, and Wendy E. Mackay. 2013. Arpège: learning multitouch chord gestures vocabularies. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces* (ITS '13), 209-218. http://dx.doi.org/10.1145/2512349.2512795
19. Yves Guiard. 1987. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behaviour* 1987, Vo1. 19, No. 4, 486-517.
20. François Guimbretiére and Terry Winograd. 2000. FlowMenu: combining command, text, and data entry.

In *Proceedings of the 13th annual ACM symposium on User interface software and technology* (UIST '00), 213-216. http://dx.doi.org/10.1145/354401.354778

21. Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. 2010. Imaginary interfaces: spatial interaction with empty hands and without visual feedback. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology* (UIST '10), 3-12. http://dx.doi.org/10.1145/1866029.1866033
22. Sean Gustafson, Christian Holz, and Patrick Baudisch. 2011. Imaginary phone: learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11), 283-292. http://dx.doi.org/10.1145/2047196.2047233
23. Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. 2014. Faster command selection on tablets with FastTap. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (CHI '14), 2617-2626. http://dx.doi.org/10.1145/2556288.2557136
24. Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *P. A. Hancock and N. Meshkati (Eds.)* Human Mental Workload. Amsterdam: North Holland Press.
25. Christian Holz and Patrick Baudisch. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the 28th international conference on Human factors in computing systems* (CHI '10), 581-590. http://dx.doi.org/10.1145/1753326.1753413
26. Paul Kabbash, William Buxton, and Abigail Sellen. 1994. Two-handed input in a compound task. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '94), 417-423. http://dx.doi.org/10.1145/191666.191808
27. Kenrick Kin, Björn Hartmann, and Maneesh Agrawala. 2011. Two-handed marking menus for multitouch devices. *ACM Transactions on Computer-Human Interaction* 18, 3, Article 16 (August 2011), 23 pages. http://dx.doi.org/10.1145/1993060.1993066
28. Gordon Paul Kurtenbach. 1993. *The design and evaluation of marking menus*. PhD Thesis, University of Toronto.
29. Gordon Kurtenbach and William Buxton. 1994. User learning and performance with marking menus. In *Conference companion on Human factors in computing systems* (CHI '94), 258-264. http://dx.doi.org/10.1145/259963.260376
30. Gordon Kurtenbach, George W. Fitzmaurice, Russell N. Owen, and Thomas Baudel. 1999. The Hotbox: efficient access to a large number of menu-items. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (CHI '99), 231-237. http://dx.doi.org/10.1145/302979.303047
31. Gordon Kurtenbach, Thomas P. Moran, and William Buxton. 1994. Contextual animation of gestural commands. *Computer Graphics Forum*. Vol. 13. No. 5. Blackwell Science Ltd.
32. Gordon Kurtenbach, Abigail Sellen, and William Buxton. 1993. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Human-Comp. Interaction* 8, 1 (mar 1993), 1-23. http://dx.doi.org/10.1207/s15327051hci0801_1
33. G. Julian Lepinski, Tovi Grossman, and George Fitzmaurice. 2010. The design and evaluation of multitouch marking menus. In *Proceedings of the 28th international conference on Human factors in computing systems* (CHI '10), 2233-2242. http://dx.doi.org/10.1145/1753326.1753663
34. Julien Letessier and François Bérard. 2004. Visual tracking of bare fingers for interactive surfaces. In *Proceedings of the 17th annual ACM symposium on User interface software and technology* (UIST '04), 119-122. http://doi.acm.org/10.1145/1029632.1029652
35. Frank Chun Yat Li, David Dearman, and Khai N. Truong. 2009. Virtual shelves: interactions with orientation aware devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (UIST '09), 125-128. http://dx.doi.org/10.1145/1622176.1622200
36. Hao Lü and Yang Li. 2011. Gesture avatar: a technique for operating mobile user interfaces using gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11, 207-216. http://dx.doi.org/10.1145/1978942.1978972
37. Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. 2013. Promoting Hotkey use through rehearsal with ExposeHK. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13, 573-582. http://dx.doi.org/10.1145/2470654.2470735
38. Pekka Parhi, Amy K. Karlson, and Benjamin B. Bederson. 2006. Target size study for one-handed thumb use on small touchscreen devices. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services* (MobileHCI '06), 203-210. http://dx.doi.org/10.1145/1152215.1152260
39. George Robertson, Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel, and Maarten van Dantzich. 1998. Data mountain. In *Proceedings of the 11th annual ACM symposium on User interface software and technology* (UIST '98), 153-162. http://dx.doi.org/10.1145/288392.288596

40. Quentin Roy, Sylvain Malacria, Yves Guiard, Eric Lecolinet, and James Eagan. 2013. Augmented letters: mnemonic gesture-based shortcuts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13, 2325-2328. http://dx.doi.org/10.1145/2470654.2481321

41. Joey Scarr, Andy Cockburn, Carl Gutwin, and Andrea Bunt. 2012. Improving command selection with CommandMaps. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems* (CHI '12), 257-266. http://dx.doi.org/10.1145/2207676.2207713

42. Dominik Schmidt, Ming Ki Chong, and Hans Gellersen. 2010. HandsDown. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction Extending Boundaries* (NordiCHI '10), 432–441. http://dx.doi.org/10.1145/1868914.1868964

43. Dominik Schmidt, Ming Ki Chong, and Hans Gellersen. 2010. IdLenses. In *ACM International Conference on Interactive Tabletops and Surfaces* (ITS '10), 131–134. http://dx.doi.org/10.1145/1936652.1936678

44. Daniel Vogel. 2010. *Direct Pen Input and Hand Occlusion.* Ph.D Dissertation. University of Toronto. Retrieved from http://tspace.library.utoronto.ca/handle/1807/24905

45. Daniel Vogel and Géry Casiez. 2012. Hand occlusion on a multi-touch tabletop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12). 2307-2316. http://dx.doi.org/10.1145/2207676.2208390

46. Julie Wagner, Eric Lecolinet, and Ted Selker. 2014. Multi-finger chords for hand-held tablets: recognizable and memorable. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14), 2883-2892. http://dx.doi.org/10.1145/2556288.2556958

47. Robert Y. Wang and Jovan Popović. 2009. Real-time hand-tracking with a color glove. *ACM Trans. Graph*. 28, 3, Article 63 (July 2009), 8 pages. http://doi.acm.org/10.1145/1531326.1531369

48. Mike Wu and Ravin Balakrishnan. 2003. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proceedings of the 16th annual ACM symposium on User interface software and technology* (UIST '03), 193-202. http://dx.doi.org/10.1145/964696.964718

49. Shengdong Zhao, Maneesh Agrawala, and Ken Hinckley. 2006. Zone and polygon menus. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (CHI '06), 1077-1086. http://dx.doi.org/10.1145/1124772.1124933