Evaluation of an Eyes-Free Cursorless Numeric Entry System for Wearable Computers

Gábor Blaskó Steven Feiner

Department of Computer Science
Columbia University
New York, NY, 10027
[gblasko, feiner]@cs.columbia.edu

Abstract

We report on the results of a user study to investigate the utility of passive haptics for eyes-free numeric entry. This work targets cursorless user interfaces designed for use with a watch-sized wrist-worn computer. Our study compared three approaches for selecting one of a set of eight numeric parameters and entering its value, both with and without visual feedback. The three selection methods utilized physical buttons alone, buttons with a touch-sensor utilizing passive haptics, and the touch sensor with passive haptics alone. The results show that passive haptics allowed users to perform parameter selection and number entry tasks, with statistically insignificant differences in accuracy and speed when used with and without visual feedback. Furthermore, there was no statistically significant difference in accuracy and speed between the button-based methods and the purely touch-sensor–based approaches.

1. Introduction

Wearable computing systems need interfaces that allow the user to focus their visual attention on their primary task. For example, maintaining eye contact in social interactions is often crucial; it is impolite to interrupt a conversation to glance at the display of a hand-held or wrist-worn device, or break eye contact by looking at content shown to the side on a head-worn display.

Interfaces that require the navigation and continuous visual tracking of an on-screen cursor are inappropriate for wearable computing because of the demands they place on the visual system. Cursor-operated on-screen widgets take up screen real-estate, which is problematic for see-through head-worn displays, because on-screen widgets obscure real world content: while annoying in some situations, this may be quite dangerous in others.

As an alternative to cursor-operated graphical user interfaces, we have been exploring the utility of passive haptics for eyes-free wearable user interfaces. By employing an input device with tangible physical features that can be felt by the user's fingers, it is possible to develop interfaces that can be operated covertly and inconspicuously, while the user interacts with others.

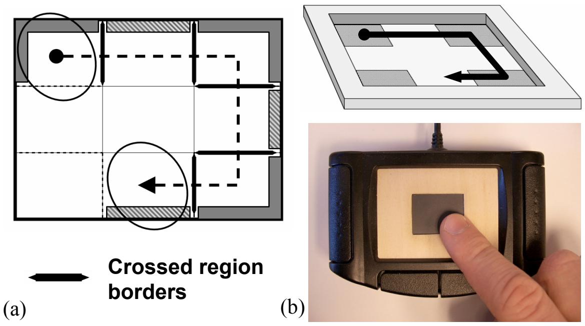

Figure 1. (a) Using tactile landmarks to assist finger-dragging motions on a touch-sensitive device. Example shows entry of number five by passing a combination of five edge and corner regions. (b) Touch sensor input device used in tests.

In prior work, we reported on the creation of a menu navigation system based on the use of a touch sensor framed by tangible physical features. These tangible features make it possible for the user to home their finger to the device and operate it with dragging motions, or gestural shortcuts, without looking [2]. We have utilized this cursorless wearable user interface framework to build testbed applications for a projector-equipped wrist-worn computer simulator [1] and for a wearable password repository system [3].

Menu navigation and command activation is one component of an interface; another is the availability of parameter adjustment facilities. Users should not only be able to directly invoke system functions with an interface, but should also be able to access and modify application parameters (e.g., numeric values) quickly, accurately, and, if possible, eyes-free. In Edgewrite [7], a text entry system for mobile devices and assistive systems, the physical features of a rectangular overlay guide text entry; however, the role of visual feedback in the use of Edgewrite has only been discussed, but not evaluated. Isokoski and Käki compared a clock-face-metaphor–based unistroke system for single-digit numeric entry with a finger on a touchpad [5]. Their system did not take advantage of passive haptics offered by a physical frame, and they did not evaluate their system without visual feedback.

In this paper, we present the results of a comparative user study, designed to evaluate the utility of tangible physical features for direct parameter entry. We compare user performance for the manipulation of eight numeric parameters using different pre-selection techniques that could be implemented on a small handheld or wrist-worn computer system, such as the IBM WatchPad [6]. We also investigate how user performance is influenced by the availability of visual feedback.

We begin with an overview of a cursorless parameter adjustment pre-selection method called the MultiWidget technique, and then describe our user study design. Next, we report the quantitative results of our experiments and participants' responses to a post-trial questionnaire. We then conclude with a discussion of our observations of input device use and our findings on user strategies and practices.

2. Dial-wheels and MultiWidget technique

Small devices, which may be either wrist-worn (e.g., WatchPad) or handheld (e.g., a mobile phone), may be equipped with a small rectangular touch sensor or touchscreen that can be surrounded by a tangible physical frame. The corner and edge sections of this frame can be felt by the fingertips, and eight regions along the periphery of the rectangular touch-sensitive surface may be discriminated and individually identified by touch alone: four corners, which we call *tactile landmarks*, and four edges, which we call *inter-landmark regions*. For example, the user can home their finger to the device and position it in the top-left corner eyes-free, based on just touch alone, since the fingertip is able to feel the left and top edges of the frame simultaneously.

It is possible to operate a dial-wheel widget implemented with this touch-sensitive device, eyes-free, as follows. Since the user can discriminate corner and edge segments of the rectangular region by touch, we may increment a discrete parameter's value when the fingertip moves in the clockwise (CW) direction from corner to edge or from edge to corner, and decrement it if the motion occurs in a counter-clockwise (CCW) direction. The user only has to remember that moving from a corner to a corner (across an edge) changes a value by two, since two region borders are crossed, while moving from a corner to an adjacent edge, or an edge to an adjacent corner, changes a value by one. For example, if the user wishes to increment a variable by five, then, as shown in Figure 1(a), she only needs to start a CW dragging motion (e.g., from the top-left corner region) and move the fingertip through two edges and stop halfway along the third (in this case, passing through the top edge, top-right corner, right edge, and bottom-right corner, and ending in the middle of the bottom edge).

To allow the user to directly manipulate multiple parameter variables with the same small rectangular touch-sensitive device, we introduced the MultiWidget input technique [2].

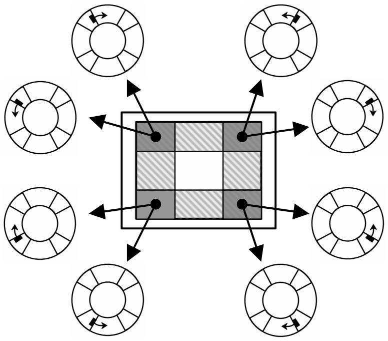

Figure 2. Illustration of MultiWidget input technique: Initially contacted landmark region and initial motion directionality determine which of eight dial-wheel widgets is operated.

Since the user may differentiate which corner (tactile landmark) of the device they initially contact through their sense of touch alone, that landmark may be used to select an associated dial-wheel widget. This makes it possible to directly access up to four separate dial-wheels without requiring the user's visual attention.

We can double the number of directly accessible dial-wheels to eight by taking into account the initial direction in which a widget is operated, as illustrated in Figure 2. This approach sacrifices the ability to immediately control a selected dial-wheel in either the CW or CCW direction during the initial finger motion. Monitoring the direction of the first corner-to-edge region crossing can thus be used to associate two different dial-wheels with the same landmark. Following the selection based on the landmark first touched and the direction of the first region crossing, a subsequent change in direction can then be used to increment a dial-wheel activated with an initial CCW motion or decrement a dial-wheel activated with an initial CW motion. The parameter's value can then be arbitrarily increased or decreased until the finger is removed from the sensor surface.

Since our interface is cursorless, viewing the screen may be necessary only to confirm the result of the manipulation, and not needed at all if output is instead provided through a haptic or audio channel.

3. User study

We conducted a user study to formally compare and evaluate variants of the MultiWidget input technique and dial-wheel–based number entry system. We wanted to compare user performance for the manipulation of eight numeric parameters using three different pre-selection techniques that could be implemented on a watch-sized wrist-worn device. Additionally, we wanted to find out how user performance with our input device and input techniques is influenced by whether visual feedback is available.

To control more than one parameter with a single touch sensor implementing a dial-wheel widget, it is necessary to have a *pre-selection step* to specify the particular parameter that is to be adjusted, which we call a *slot*. On a watch-sized device, the possibilities are limited by physical space constraints: if an array of eight discrete physical buttons were used for pre-selection, each button could be at most a few millimeters in size, making the buttons quite challenging to operate.

We decided to test three pre-selection methods that could be implemented on a watch-sized device:

- MB: A pair of buttons is used to circularly select among the eight slots. One of the buttons iterates a selector through the slots in one direction, while the other button iterates through the slots in the opposite direction. After pre-selecting a slot, a single dial-wheel can be used to manipulate the numeric parameter.
- ML: Using the initial-landmark–based MultiWidget technique alone, the user can choose one of a set of four dial-wheels. To control eight parameters, two sets of four dial-wheels are needed; therefore, during pre-selection, a pair of buttons determines the set to use. The user presses a button on the left to select the 'left' set of four dial-wheels or a button on the right to select the 'right' set of four dial-wheels. (We use two buttons, rather than a single button that toggles between sets, to avoid giving preference to one of the sets.)
- MLD: Using the initial-landmark-and-direction–based MultiWidget technique alone, the user can directly choose one of eight dial-wheel widgets.

We also wished to examine how user performance is affected by the amount of visual feedback provided during interaction. Therefore, we tested two levels of visual feedback, provided on a desktop computer screen:

- V1: Users were given constant on-screen visual feedback, as would be the case on a device where the touch sensor and display are physically not overlaid on top of one another (e.g., by using a separate wrist-worn or head-worn display).
- V2: Users were not given any on-screen visual feedback, simulating worst-case interaction with a small touchscreen (e.g., as on WatchPad) that would be occluded by the user's finger during interaction. However, users were able to glance at the input device and their finger during interaction. Only the task cue was visible on the screen; the portion of the screen that provided visual feedback in V1 was physically occluded in V2.

3.1. Design

We employed a within-subject study design for the factors: 3 (Pre-Selection Method) × 2 (Visual Feedback). The trials were counterbalanced and each user was tested for all test conditions during a single session.

We recruited 12 (11 men, 1 woman) right-handed participants by email and flyers posted on our campus. The average age was 24 (min: 19, max: 33). Participants were compensated $10 for taking part in the session, which lasted approximately 60–75 minutes. All participants tested all input modes; they were randomly assigned to one of three groups (blocks), determining the input mode (MB, ML, or MLD) with which they would start their trials, in a Latin square design. For each input mode, users were first tested with V1 and then V2. Only after users had completed their first trial, were they informed that they would be using the same input mode again without visual feedback. For each trial, we gave users the chance to try out the test application and do a practice run, asking them to use the system until they felt comfortable. Participants usually completed approximately 10 to 20 tasks during practice, and then notified the study supervisor that they were ready for the timed tests.

Our test application recorded timing information for the following events: task cueing (task appearing on the screen), first button press (only in MB and ML), and initial contact with and release of the touch sensor.

We wished to find out how reliably users could manipulate the dial-wheel widgets for number entry with and without visual feedback on their first try. We also wanted to avoid users wasting time making corrections. Therefore, in all conditions, we counted lifting the finger off the touch sensor, after manipulating the dial-wheel, as terminating the task. At this point, timing information was saved and users were cued with the next task.

3.2. Equipment

Because we did not want the slow processing speed of the WatchPad hardware prototype—equipped with a ~17MHz processor—to influence our timing results, we used a desktop computer (1.6 GHz P4) with an external Synaptics TouchPad, shown in Figure 1(b). The TouchPad is equipped with buttons to the left and right of the touch-sensitive area. To simulate the small rectangular touch sensor of a wrist-worn computer and its tangible frame, we attached an overlay on the surface of the touch sensor with a cutout equivalent in size to the WatchPad touchscreen (2.5cm × 2cm). The center of each side button was located at a horizontal distance of 4cms from the right (or left) edge of the cutout.

3.3. Tasks

Participants were presented with a screen in which the upper part of the display was dedicated to cueing the user. As shown in Figure 3(a–c), this cueing area contained eight dark gray rectangles, one of which had a highlighted border. The task was to use the input device to enter the number shown in the middle of the upper screen section into the corresponding location in the lower screen section. The screen layout varied slightly for each of the different input scenarios.

Each timed trial for all six test conditions consisted of 60 tasks, a total of 360 tasks for the entire testing session. When creating the test dataset of numbers ranging from –12 to 12 for cueing, we made sure that participants would be cued with an equal number of positive

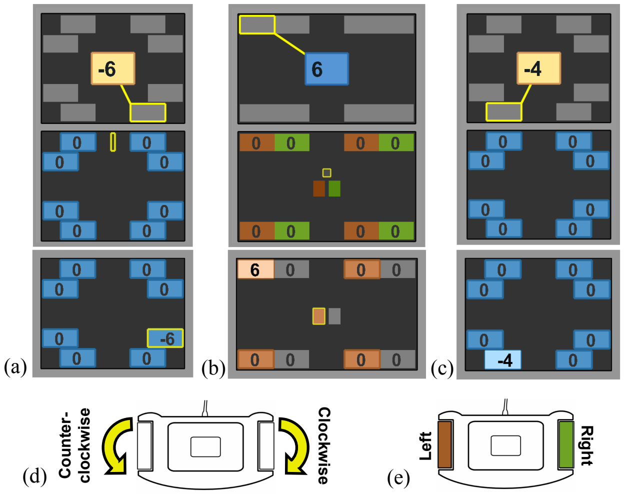

Figure 3. Screenshots of testing application: (a) MB mode (b) ML mode. (c) MLD mode. Each figure shows (top) cueing screen and number entry screen in base state, and (bottom) number entry screen during number value adjustment. Illustration of button interactions on device: (d) MB mode: buttons move slot selector highlight. (e) ML mode: buttons select between left or right set of four dial-wheels.

and negative numbers, requiring equivalent CW and CCW motions. We also made sure that there were the same number of odd and even numbers in the dataset, requiring participants to stop their dragging motions an equal number of times on edge sections and corner sections. Additionally, we made sure that there were no subsequent repetitions of cued locations or cued values.

In MB (Figure 3a), upon starting a new task a small yellow rectangle marker was placed in a “neutral” position in the middle upper part of the circle of numbers. The user needed to move this marker to the desired parameter slot location either CW or CCW, by using the left or right button, respectively. The user could freely go back and forth in both directions, automatically skipping the neutral position. Once one of the eight slots was highlighted with the yellow marker, the user could operate the dial-wheel, starting at any location along the sensor edge to increment or decrement the value with a CW or CCW motion, respectively.

In ML (Figure 3b), the eight parameter slots were arranged in the four corners of the rectangle in pairs, with the left element of each pair selected by the left button and the right element by the right button. We color-coded the left set dark brick and the right set dark lime. Once one of the two sets was selected, the other set was grayed out. The buttons could be pressed any number of times until the selection was finalized when the user contacted the touch sensor. The landmark that was touched first selected the corresponding slot and the dial-wheel within the set.

In MLD (Figure 3c), the touch sensor alone was used for interaction. The user could directly pre-select the parameter to adjust using the MultiWidget technique. In all three modes, during the operation of the dial-wheel widget, the colored rectangle behind the actively changed number was displayed brighter than the other seven parameters, as shown at the bottom of Figure 3.

4. Results

We calculated the mean values for our 12 participants for our main dependent variables (Figure 4a–c): (a) ‘Cue-Int’: the time from cueing to the start of interaction (for MB/ML: cue time to button press, for MLD: cue time to touch sensor contact); (b) ‘Int-Rel’: the time of ‘interactivity’ (for MB/ML: time from first button press to release of touch sensor, for MLD: time of touch sensor use); and (c) ‘Cue-Rel’: total task completion time from the presentation of the cue location and number until the release of the touch sensor. We performed a two-factor repeated measure ANOVA (analysis of variance) to analyze the data for the 3 (Pre-Selection Method) × 2 (Visual Feedback) conditions, with our subjects as the random variable. A significance level of 0.05 was used for all tests.

4.1. Interaction preparation time

Mean values for Cue-Int for the 6 trials are illustrated in Figure 4(a). Both Pre-Selection Method [F(2,22) = 16.10, p<< .001] and Visual Feedback [F(1,11) = 17.10, p<< .001] had effects on the Cue-Int times. We expected this result, since it is realistic to assume that the different input modes require the user to make decisions about how they would begin the interaction: for MB/ML, which button to press; and for MLD, which of the four corner landmarks to contact first. We believe that Cue-Int is a good indicator of the cognitive load involved with the input methods. A post-hoc t-test showed that the Cue-Int times for MB-V1 were significantly shorter than those of ML-V1 [t(11) = 7.04] and MLD-V1 [t(11) = 4.13]. The difference between ML and MLD was statistically insignificant in V1 and V2.

When we analyze the Cue-Int times for V1 and V2 for all pre-selection modes, we see that only the differences between MB-V1 and MB-V2 [t(11)=4.15], and between MLD-V1 and MLD-V2 [t(11) = 2.76] are statistically significant.

4.2. Input device interaction time

When we analyzed touch-sensor usage times from initial contact to release, we found that only Pre-Selection Mode had a significant effect [F(2,22) = 28.90, p<< .001]. That operation of the touch sensor takes significantly longer in MLD can be attributed to the touch sensor being responsible for both parameter slot selection and number entry, whereas in MB and ML the touch sensor is only used for number entry. For MB and ML, for both V1 and V2, touch-sensor operation times do not differ significantly. Furthermore, since visual feedback had no effect on touch-sensor operation time, we conclude that with the help of passive haptics, users can enter numbers with similar speeds in all Pre-Selection Modes, with or without visual feedback.

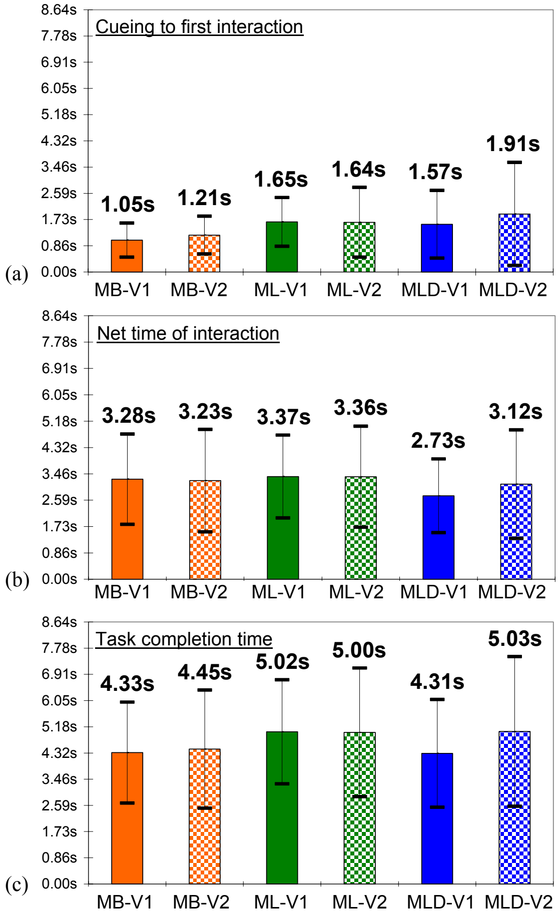

Figure 4. Graphs of mean times for trials: (a) Cueing to first interaction step (MB and ML: first button press, MLD: Initial touch sensor contact). (b) Net time of interaction (MB and ML: initial button press to release of touch sensor, MLD: Touch sensor contact to release). (c) Task completion time, from cueing to release of touch sensor.

A good comparison can be gained for the net time users spent manipulating the input device by analyzing the Int-Rel time, for which the mean values are charted in Figure 4(b). The ANOVA indicates that neither Pre-Selection Mode nor Visual Feedback had a statistically significant effect on Int-Rel times. While not statistically significant, we note that the Int-Rel mean times for MLD-V1 and MLD-V2 are smaller than the mean times for the other input modes. This is a very positive result, as it indicates that once interaction was begun, all Pre-Selection Modes had comparable speeds. Because MLD requires only a touch sensor and no physical buttons, its use could be a significant benefit on a wearable platform by simplifying the input device hardware.

4.3. Task completion time

Task completion time (Cue-Rel) was measured starting at the moment when the subject was presented with cueing information until the moment the touch sensor was released. The ANOVA revealed that only Pre-Selection Method had a significant effect [F(2,22)=4.82, p< 0.05]. For V1, ML-V1 took significantly more time than MB-V1 [t(11)=3.42] and MLD-V1 [t(11)=4.12]; however, the difference between MB and MLD was insignificant. For V2, only MB-V2 and MLD-V2 differ significantly [t(11)=2.96, p<0.05]. We note that even though this was the first time that users had tried initial-landmark–based and initial-landmark-and-direction–based pre-selection, they achieved approximately the same speed for both MB and MLD in V1.

Our results show that input hardware may be simplified to only a single touch sensor implementing the MultiWidget technique to provide direct access to the manipulation of eight input parameters, without greatly sacrificing task completion time. Furthermore, as visual feedback did not have a statistically significant effect on our measured timing results, we conclude that users were able to achieve similar results with regards to time, with and without visual feedback, utilizing physical landmarks and passive haptic feedback to assist number entry.

4.4. Error rates

In addition to timing data, another dependent variable was the task completion success rate (both slot location and number had to be correct). Task completion success rates are charted on Figure 5. The ANOVA indicates that Visual Feedback had an effect [F(1,11)=17.90, p<0.001]; however, the Pre-Selection Method had no effect on task completion success rates. Among all the post-hoc pairwise comparisons of the conditions, only the success rate decrease between MB-V1 (88.76%) and MB-V2 (73.34%) was significant [t(11)= 4.08 p<0.01], while the differences between V1 and V2 for ML and MLD were both insignificant. This indicates, as we had expected, that the input system with tactile landmarks allowed users to perform the tasks with comparable accuracy even without visual feedback.

We note that User 5 improved his success rate from ML-V1 (92.31%) to ML-V2 (94.92%) and commented: “The ‘lack of eyes’ made it more difficult but helped me concentrate on counting.” Two other users made slight improvements in MLD: User 9 (V1: 82.14% → V2: 83.93%), User 10 (V1: 80.36% → V2: 85.45%).

Overall, this is quite a positive result, as it indicates that users were able to achieve similar success rates for our input techniques utilizing ML and MLD pre-

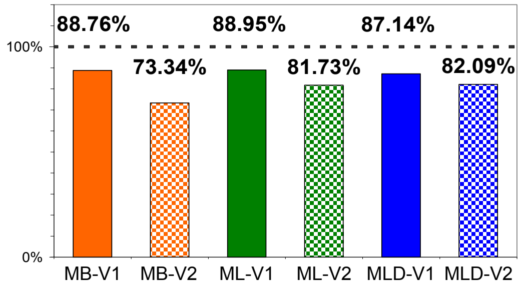

Figure 5. Graph of task completion success rates

selection. It seems that in V2, when users needed to select a specific landmark corner of the touch sensor, they entered the number more cautiously and accurately.

It is also interesting to observe that the differences in the mean success rates decreased slightly between V1 and V2, as more of the pre-selection process was 'transferred' from physical buttons, to initial-landmark–based selection, to initial-landmark-and-direction–based selection (MB to ML to MLD). The insignificant differences between V1 and V2 in ML and MLD indicate that the use of initial-landmark–based pre-selection helped users achieve similar success rates whether (V1) or not (V2) they had visual feedback. User 1, who started the session with MB-V1, commented on MB-V2:

> *"For this test, it finally dawned on me how easy it was to count to the correct number using multiples of two. That actually made it easier, in a way, than being able to look at the screen. It was like learning how to touch type."*

Another positive result is that neither the pre-selection method nor the availability of visual feedback had an effect on the slot selection success rates. This indicates that both novel methods of parameter slot selection—ML and MLD—were as effective as MB.

4.5. Follow up use

Two days after the first session, five of our participants returned for a second session. For each user, trials were conducted in the same order as in the first session. We performed a three-factor ANOVA analysis on our results, for 3 (Pre-Selection Mode) × 2 (Visual Feedback) × 2 (Session), with the users as random variables.

Mean task completion success rates for the five users increased from the first to the second session, as illustrated in Figure 6; however, none of the factors had a statistically significant effect on success rates. Accuracy rate differences for MB-V1 (92.80%) and MLD-V2 (89.17%) were less than four percentage points, indicating that accuracy was almost equivalent when using the MultiWidget technique on simpler hardware and without visual feedback, as with MB with visual feedback.

The time between cueing and starting the interaction was affected by Input Mode [F(2,8)=7.13, p<0.05] and by Session [F(1,4)=8.55, p<0.05]. Of the 6 trial conditions only the speedup of MB-V1 [$t_{1st=}$1.107s, t_{2nd}=0.862s,t(4)=2.9, p<0.05] and ML-V1 [t_{1st}=1.759s, t_{2nd}=1.484s, t(4)=6.58, p<0.05] were statistically significant.

The touch sensor operation time (from initial contact to release) was affected by Input Mode [F(2,6)=13.00, p<0.01] and by Session [F(1,4)=23.60, p<0.01]. Here there was a significant improvement for ML-V1 [t_{1st}=2.170s, t_{2nd}=1.598s, t(4)=3.33, p< 0.05] and both MLD modes: MLD-V1 [t_{1st}=2.689s, t_{2nd}=2.252, t(4)=3.87, p<0.05] and MLD-V2 [t_{1st}=3.160s, t_{2nd}=2.277s, t(4)=3.62, p<0.05]. This indicates that second session users became quicker in operating the touch sensor when pre-selection was done with the initial-landmark–based and initial-landmark-and-direction–based techniques.

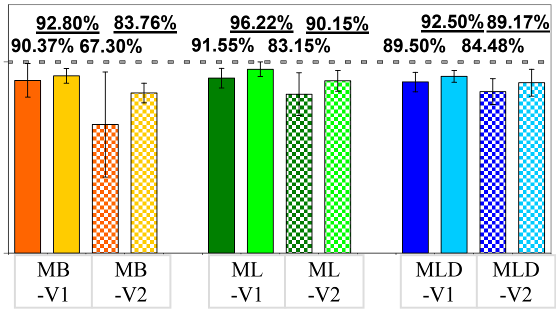

Figure 6. Task completion success rates for follow-up session. (follow-up values shown with lighter colored columns and underlined)

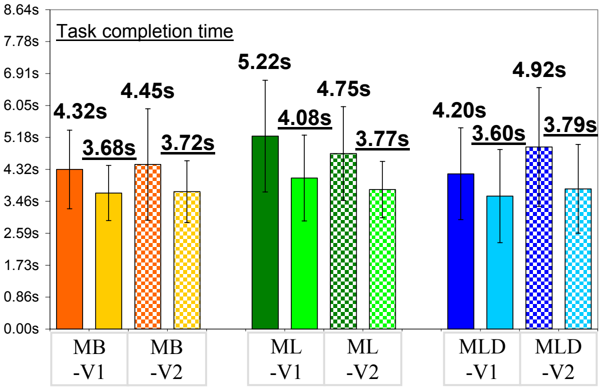

Figure 7. Task completion times from cueing to release of sensor. (follow-up values shown with lighter colored columns and underlined)

Finally, we analyzed task completion times, comparing results for both sessions, shown in Figure 7. Although Pre-Selection Method and Visual Feedback independently did not have an effect on the task completion times, interaction between these factors was significant [F(2.8)=7.23, p<0.05]. Session had a significant effect [F(1,4)=22.10 p<0.01]. The means for the five participants showed significant improvements for MB-V1 [t(4)=3.45, p<0.05], ML-V1 [t(4)=4.57 p< 0.05], ML-V2 [t(4)=2.88, p<0.05], and MLD-V2 [t(4)=5.88, p<0.01]. We also note that the differences between the mean completion times for MB and MLD (in V1 and V2) were less than 0.2 seconds apart.

5. Observations and user strategies

Users were asked to fill out a post-experiment questionnaire, make comments about the system, and describe strategies they employed. Using this feedback, we examined our log data to uncover user practices.

5.1. Number Parity

In the study questionnaire, two users explicitly mentioned that they felt that their performance in the number entry task depended on the parity of the number they needed to enter. In ML, all users had to begin number entry in a particular landmark region, and would finish their dragging gesture in either a non-landmark region along the edge if the number was odd or a corner landmark region if the number was even. Figure 8 shows a point plot of mean sensor touch-to-release times for all participants in ML-V1 and ML-V2.

For both V1 and V2, there are multiple neighboring even and odd numbers where this usage pattern can be observed. The utility of corner landmarks in the entry of even numbers is especially noticeable in V2 (–7 took longer than –6 or –8; –11 longer than –10 or –12; +5 and +7 longer than +4, +6, or +8; and +11 longer than +10 or +12). As expected, entering larger numbers required the user to drag their finger longer, as illustrated by the symmetrical V-pattern of the times and the regression lines shown for the 0 to ±12 ranges.

5.2. Starting corner preference

In MB, even though users could initially place their finger anywhere on the device, we noticed that users often had a preferred starting corner. For some users, this corner depended on whether the number was positive or negative, determining the direction in which they would move their finger. Some users started exhibiting this approach in MB-V2, when they needed to devise a strategy that did not depend on visual feedback. Some chose to consistently start in specific corners they could use as an "origin." We calculated the corner ratios for both negative and positive numbers, giving us the percentage of positive (or negative) numbers that were entered by starting the dragging motion in a specific corner. In the following discussion, '→' represents the change in this percentage value from V1 to V2, and we use acronyms for corner locations (Top-Right = TR, Top-Left=TL, Bottom-Right=BR, and Bottom-Left=BL).

User 4, 7 and 10 were from the block who started with MB, and thus did not know that in the forthcoming ML and MLD tests they would need to be mindful of the starting corner. User 4 consistently started from TR for positive numbers (51.72% → 100%) and for negative numbers (51.72% → 96.30%). User 7 settled on the strategy of starting the entry of negative numbers from TL (0% → 100%) and positive numbers from BL (60% → 89.66%), indicating a preference for starting number entry with a downward or upward stroke along the left. User 10 consistently used TR for negative numbers (55.17% → 96.97%), but switched the preferred corner for positive numbers: TL (76.67% → 0%) for V1 and TR (6.67% → 100%) for V2.

We noticed a similar tendency to prefer TR for negative and TL for positive number entry from all four users in the block that initially started with MLD, indicating a preference to start the initial dragging motion with a horizontal drag across the top of the touch sensor. For negative number entry using TR: U3: 3.7% → 48.28%, U6: 51.72% → 55.56%, U9: 71.43% → 89.66%, U12: 3.45% → 28.57%. For positive numbers using TL: U3: 0% →44.44%, U6: 76.67% → 81.48%, U9: 96.55% → 100%, U12: 40% →42.86%. User 8 from the ML block used the same corners but differently, negative numbers from TL (44.83% → 100%), and positive numbers from TR (30% → 93.10%), indicating a preference to start all circular dragging motions with a downward dragging motion along the left edge for negative numbers and right edge for positive numbers.

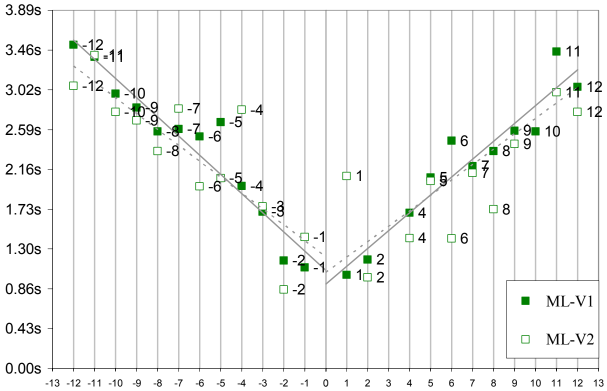

Figure 8. Mean net touch sensor operation times (contact to release), graphed by cued number.

5.3. User comments

While a 'preferred starting corner' strategy could not be utilized in ML or MLD, we observed another strategy in MLD. Seven users explicitly mentioned using a 'back and forth' technique for pre-selection and number entry in MLD. They divided the touch-sensor task into two distinct phases: first, for slot specification, they started in the appropriate corner and then moved out of the landmark region in the appropriate direction—thereby completing pre-selection. Five of the seven moved only to the neighboring inter-landmark section, while the other two moved all the way to the nearby corner landmark region. Then, they completed pre-selection by returning to the original corner, starting the number entry phase from this 'base' corner. User 9 stated:

> *"I almost always treated the slot/direction selection and number entry as two separate phases. I would move in the direction, reset to the corner, then enter the number, rather than exploiting if I'd already moved in the appropriate direction for the sign [+, –] of the number. Once I got the hang of it, the strategy of choose-slot, then enter-number I developed for this version felt faster than the 'feedback' strategy, where I just relied on [visual] feedback for whether I'd gotten the number 'moving in the right direction or not."*

For MLD, Users 2 and 12 described a slightly different understanding of the 'back-and-forth' approach and even described a 'tweak':

> *"I used the first move to indicate the slot, then moved back to the corner, then produced the number… Later learned a 'trick' not to move back to the corner if the direction of the slot and the sign of the number agreed."(User 12)*

> *"This one [MLD] was perplexing until I realized that for 'opposite polarity', what I needed to do was to first intentionally make a 'mistake', and then correct it."(User 2)*

Unlike gestural input systems with alphabets that need to be memorized, such as Unistrokes [4] or Edgewrite [7], in our numeric entry system the user may transition from ‘feature counting’ to gestures, which may be executed quicker. We found it very interesting that User 11 explicitly described his number entry approach as a system of gestures, rather than a procedure of counting corner and edge sections passed during dragging. He stated:

“I made use of the fact that 8 is a full circle, later learned other patterns: 3 → knight movement in chess, 4 → half circle, 6 → letter “C”, 7 → almost a circle but don’t close it.”

6. Questionnaire results

The questionnaire also asked participants to rate each of the six test conditions for ease of use and intuitiveness on a five-point Likert scale. The results are shown in Figure 9. Additionally, they were asked for free-form comments. Users were not informed about timing results or success rates before filling out the questionnaire to avoid bias. Multiple users stated that MLD was intuitive to use and easier than MB or ML, since they could do slot selection and number entry in one motion. Users commented about MLD:

“This section [MLD] I found more intellectually confusing than the others, but somehow easier. I did better, the less I attempted to concentrate.”(User5)

“[MLD was] the most intuitive: the spatial directional relationship between my movements and the slot selection was natural” (User9)

“[In MLD-V1, it was] a bit difficult to master positive/negative. More intuitive than using side buttons… I liked not having to press buttons on the last one [MLD]. There seems to definitely be a learning curve in mastery of the system and I might have messed up positive + negative direction on this test, but I think that once the system is mastered, errors would go way down.”(User 10)

Because we immediately initiated the next task upon the release of the touch sensor in our tests, some users commented that in MLD-V1, they felt frustrated when they were not able to select the appropriate slot; however, in V2 this did not play a role since they could not see their slot selection. User 12 commented:

“The blindfold tests [V2] were less frustrating; seeing the irreversible mistakes I made on the non-blindfold tests was jarring and disruptive.”

Finally, we found it surprising that half the participants mentioned that they had difficulty remembering the mapping of CW and CCW motion to increase and decrease the value, respectively.

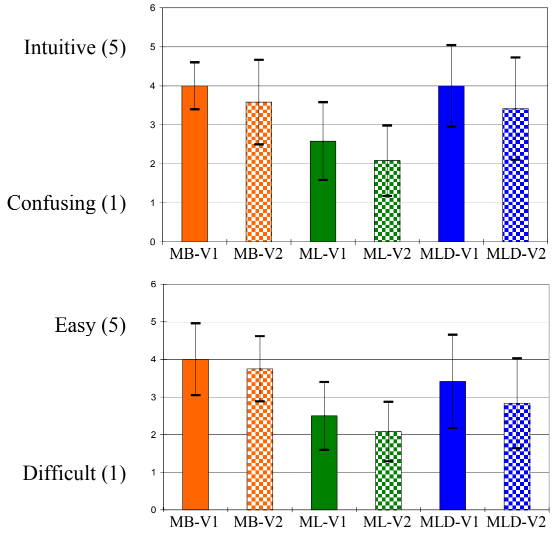

Figure 9. Results of user study questionnaire.

7. Conclusions

In this paper, we evaluated the utility of passive haptics in the form of corner and edge sections of a frame around a small (2.5cm × 2cm) touch-sensitive region. Furthermore, we performed a comparative user study to evaluate different pre-selection techniques, allowing the user to manipulate the value of eight numeric parameters.

As can be seen from the results of the follow up session in Figures 6 and 7, with some practice, users were able to achieve similar, and sometimes more accurate results with the use of the MultiWidget technique, which can be implemented with simpler hardware.

Users rated the MultiWidget technique as intuitive, but a bit more difficult to use than the approach that utilized just buttons for pre-selection. Our results indicate that with the presented input mechanisms—utilizing tangible physical features to aid eyes-free interaction—users were able to perform pre-selection and number entry with the input device with comparable speed and accuracy (Figures 4 and 5), with and without visual feedback.

8. Acknowledgements

This research is funded in part by DAFAFRL Contract FA8650-05-2-6647, NSF Grant 0099184, ONR Contract N00014-04-1-0005, and gifts from IBM, Microsoft, and Synaptics.

References

[1] Blaskó, G., Coriand, F., and Feiner, S., Exploring Interaction with a Simulated Wrist-Worn Projection Display, *In Proc. ISWC '05,* 2005, pp.2-9.

[2] Blaskó, G. and Feiner, S., An Interaction System for Watch Computers Using Tactile Guidance and Bidirectional Segmented Strokes, *In Proc. ISWC '04,* 2004, pp.120-123.

[3] Blaskó, G., Narayanaswami, C., and Raghunath, M. A wristwatch-computer based password-vault. IBM Technical Report RC23616 (W0503-066), 2005

[4] Goldberg, D. and Richardson, C., Touch typing with a stylus, *In Proc. INTERCHI '93*, 1993, pp.80-87.

[5] Isokoski, P. and Käki, M., Comparison of Two Touchpad-Based Methods for Numeric Entry, *In Proc. CHI '02,* 2002, pp.25-32.

[6] Narayanaswami, C. and Raghunath, M., "Designing a new form factor for wearable computing," *IEEE Pervasive Computing*, vol. 1, no. 4, pp. 42-48, 2002.

[7] Wobbrock, J.O., Myers, B.A., and Kembel, J.A., EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion, *In Proc. UIST '03,* 2003, pp.61-70.