Harpoon Selection: Efficient Selections for Ungrouped Content on Large Pen-based Surfaces

Jakob Leitner, Michael Haller
Media Interaction Lab, University of Applied Sciences Upper Austria
Hagenberg, Austria
mi-lab@fh-hagenberg.at

ABSTRACT

In this paper, we present the *Harpoon* selection tool, a novel selection technique specifically designed for interactive whiteboards. The tool combines area cursors and crossing to perform complex selections amongst a large number of unsorted, ungrouped items. It is optimized for large-scale pen-based surfaces and works well in both dense and sparse surroundings. We describe a list of key features relevant to the design of the tool and provide a detailed description of both the mechanics as well as the feedback of the tool. The results of a user study are described and analyzed to confirm our design. The study shows that the Harpoon tool performs significantly faster than *Tapping* and *Lassoing*.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors.

Keywords: Selection Tool, Lasso, Crossing, Pen Input

INTRODUCTION

The interaction design for pen-based interactive whiteboards is challenging due to various reasons. In this paper, we specifically look at challenges concerning interaction techniques for selecting a large number of ungrouped items (e.g., separate ink strokes in handwritten notes or sketches). Interactive whiteboards often lack interaction possibilities (e.g., no hover state, low tracking resolution, lack of modifier keys), which might be required to use selection techniques to their full extend. Tracking data might be less reliable, causing unintended results like premature cancelation of existing selections. Selection techniques optimized for desktop environments do not provide means to easily compensate for such errors. Additional challenges arise due to the size of such surfaces. Frequent mode changes might require users to take round trips to access menus which can be time consuming. Existing selections might be hard to see and modify on large screens. Also selection techniques (e.g., Tapping, traditional Rubber-Band selections) often rely on keyboard-shortcuts for quickly accessing certain functionality (e.g., for adding or removing items). These options are inaccessible on whiteboards as they often do not have supplementary buttons. Even techniques like *Lassoing* that are designed for pen input [9, 13, 15, 16] may rely on supplementary buttons and may also require difficult steering tasks that are highly constrained and error prone [1, 14].

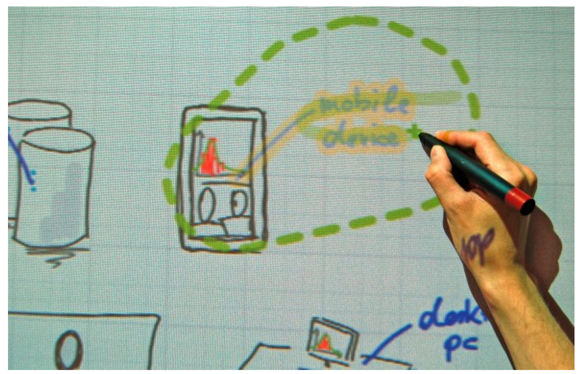

Figure 1: The Harpoon selection tool in action on an interactive whiteboard.

We analyzed these challenges and present several key features that should be considered in the design process of new selection tools. We also present the *Harpoon* selection tool that addresses the aforementioned challenges. We validate our design through a user study and present further improvements based on the results.

Summarizing, the main contributions of this paper are:

- A list of key features that should be considered when designing a selection tool for interactive whiteboards.
- A detailed description of our selection technique and visualization that is based on the aforementioned key features.
- A novel way to implicitly change selection mode, which facilitates fluid modifications of existing selections.
- Description and discussion of study results that confirm our design.

RELATED WORK

The interaction techniques on pen-based surfaces differ greatly from traditional desktop environments. Ren et al. [19] state that *"...current target selection strategies for pen-based systems are mostly only imitations of selection techniques for mouse and touch-screen devices."* In their paper, the authors present detailed design recommendations

UIST'11, October 16–19, 2011, Santa Barbara, CA, USA.

for selection strategies for pen-based systems, stating that crossing outperforms other tested selection methods in speed, error-rate, and user-preference. More recently, Apitz et al. [3] formulate design recommendations for user interfaces based on the crossing paradigm. They also provide an extensive list of related work, which we recommend for further reading. In contrast, our work focuses on selection tasks of large groups of randomly arranged items that can have different sizes.

Crossing has been explored for selection of single [1, 19] as well as multiple objects [22]. However, it can be difficult to cross every single stroke in a sketch or a handwritten word. This is especially the case for small targets (e.g., dots, commas, or quotation marks). Area cursors [12] help selecting single targets. DynaSpot [6] extends the idea of area cursors by dynamically resizing the area cursor based on the movement speed but it is also designed for only selecting single targets. Land et al. [14] analyze input and dynamically change selection properties for lasso selections. Some problems with lasso selections like overlapping items however are not addressed with this approach.

We take the idea of dynamic area cursors and crossing and extend it to facilitate efficient selections of very large numbers of items in various conditions. Based on our experience with pen-based interactive surfaces [10] as well as related work, we identified the several key factors that have been considered when designing the Harpoon tool. In the following paragraphs, we present these key factors together with related publications and present implications for the *Harpoon* tool.

Button and hover free: Most interactive whiteboard systems (e.g., Smartboard) do not offer barrel buttons or hover states on pens. If available, buttons located at the pen still might require from users to shift their grip, interrupting the users' input flow [11]. They can also be hit accidently, which may result in unexpected behavior. *Harpooning* does not rely on hardware-dependent tracking data like button-presses, hover information or pressure values.

Support discontinued drag operations: It can be difficult to perform long drag operations with a pen [18]. Bezels in multiscreen-setups prohibit a smooth and continuous dragging across the entire workspace. To support pen usage on large, multi-screen surfaces and to minimize unintended premature selection cancelation, *Harpoon* selections can be comprised of several short drag operations.

Compatibility with established techniques: To alleviate operation for novice users, common interactions should be supported. The *Harpoon* selection tool integrates established selection techniques like tapping [15, 21] and *Lasso* selections [5, 13] to exploit each technique's benefit while compensating for possible weaknesses.

Localized: If required, user interface elements should be placed close to the users' current input position to minimize long, interruptive round trips [7, 10]. The positioning should be dynamic yet predictable to enable fluid interaction. In our application an explicit mode-change is required to switch between different tools. For example, users explicitly need to switch between *Inking* and *Selecting*. We offer two different ways to perform this mode change. Each user can open a localized, digital menu at any location on the drawing canvas. Alternatively a physical tool palette can be used. This initial mode change, while crucial for a pleasurable experience, will not be further discussed in this paper as the focus of this paper is the selection process itself. Further information on our digital menu as well as the physical tool palette can be found in [10]. An overview of tool-switching techniques is given in [8]. Implicit mode changes can be used to avoid explicit mode changes altogether and effectively removing UI placement problems and visual clutter [20]. Once engaged, *Harpooning* does not require any more explicit mode changes during operation.

Support modifications of selections: To minimize interruptions, users should be able to quickly specify whether items should be added or removed from the current selection without explicitly pressing a button or changing mode [16]. *Harpooning* uses implicit mode changes to allow users to quickly modify existing selections.

Large numbers of ungrouped items: Automatic parsing to select high-level structures in ink-based documents [4, 17] is used in several systems [8] to enable a quick selection of pre-defined groups. However, as robust automatic parsing cannot be assumed in all applications and for all domains, the *Harpoon* selection tool supports fast selection of large numbers of ungrouped items.

Support selection in dense surroundings: Lassoing within a dense surround of notes is tedious, because steering [1] along a winding path is slow and error prone for *Lassos* with accuracy constraints [14]. The *Harpoon* selection tool is designed to improve selection speed and accuracy for item-clusters as well as overlapping items.

HARPOON SELECTION

The *Harpoon* selection tool is designed to combine and extend several selection methods for an easy and fast selection of single targets as well as collection of ungrouped items. It is based on the "Slide Touch" strategy as proposed by Ren et al. [19]. Targets are selected when the pen-tip touches the target for the first time after landing on the screen surface. In contrast to the strategy described in [19], a selection area (which we term *spot* from now on, cf. [6]) instead of a single point is used for hit-testing.

Basic Selection

A new selection is started by dragging the pen anywhere on the surface. A yellow dotted line shows the path covered by the stylus tip (see Figure 2). The current spot size is indicated with a circle in the same color. The target is selected as soon as the spot touches it. A small "+"-sign appears indicating that all hit items are added to the current selection. The spot, the dotted selection path as well as the "+"-sign change to the same color (*green*) to provide additional feedback. All selected objects are outlined with a border and they turn semitransparent. A convex hull enclosing all selected objects appears. Continuing the movement in a straight line selects the whole phrase. Note that all strokes (even the small once like the dot on the “i”) are added to the selected group. Once the pen is lifted from the surface, the spot, the selection path, and the cursor fade out. Finally, the color of the convex hull changes back to default (see Figure 2).

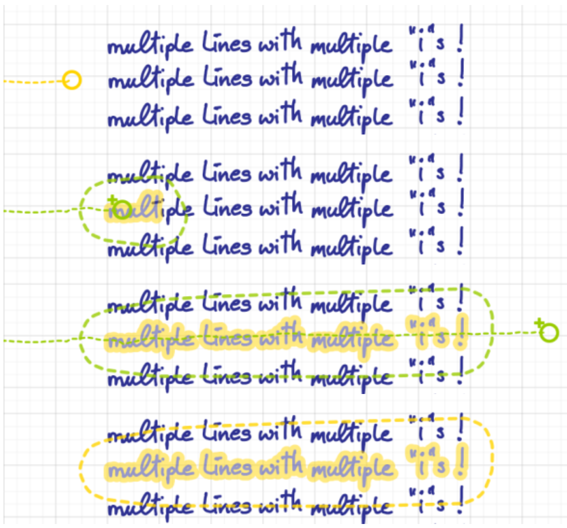

Figure 2: Performing a new selection with the Harpoon tool.

Speed-dependent spot size

It is impossible to predefine a spot size that works well for all purposes. On the one hand, a small diameter is well suited for precise selection operations, giving users fine-grain control over which individual objects to select. However, a small diameter makes selection of larger groups of objects cumbersome since all targets have to be crossed separately. Figure 3 (*top*) for example, shows that small objects (e.g., quotation marks or the dots on the “i”) are easily missed. On the other hand, choosing a large diameter prohibits fine-grain operations and might result in the selection of unwanted objects as depicted in Figure 3 (*bottom*).

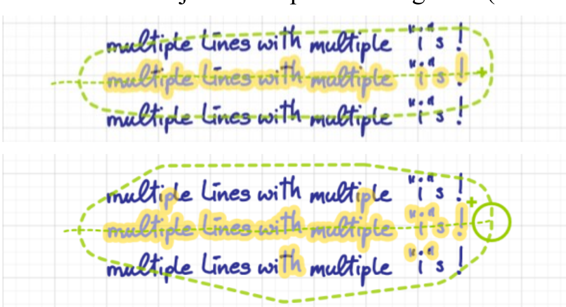

Figure 3: If the spot is too small, small strokes in the phrase might not be selected (*top*). If the cursor size is too large, close by strokes might get selected unintentionally (*bottom*).

Offering multiple different diameters in a menu would allow users to choose an appropriate diameter. However, this mode switch requires additional time and interrupts interaction flow. Also different diameters might be beneficial during a single drag operation.

Hence our *Harpoon* selection tool *dynamically resizes* the spot size based on the current movement speed. We assume that coarse selection operations result in higher input speed. In this case, we chose a large spot size. Precise selection operations require precise pen steering, resulting in slow pen movement. In this case, we choose a small spot diameter. We constantly update the spot size during movement. This results in a fluent transition between coarse and fine-grain actions without the introduction of artificial delays.

We tested several speed to spot-size transfer-functions and noticed that selecting targets in dense environments often requires the users to maintain a constant spot size. Erratic size changes lead to the selection of unwanted targets and should hence be avoided. We also noticed that it is impossible to predefine an input speed at which the system should be most sensitive as selection target sizes vary. The same amount of control is required over a large range of input speeds. Finally, to prevent selection of targets outside the user’s attention area, the spot size has to be limited. To avoid erratic size-changes, which can result from noisy input values, we smooth the values using a running average. While this introduces a certain amount of delay, it greatly reduces the number of unwanted selection operations. We chose a linear function instead of an exponential one [6], as this results in the same degree of control for the whole range of movement velocities. Initial empirical tests showed that this worked well, but further experiments might be needed to validate this decision. A maximal threshold for the average speed is used. Values above the threshold are simply clamped. This upper threshold largely depends on the dimensions of the physical setup as well as the display resolution and can be configured in our system.

Figure 4 shows both raw input values as well as smoothed output values. In contrast to [6], we also allow the spot to shrink while moving.

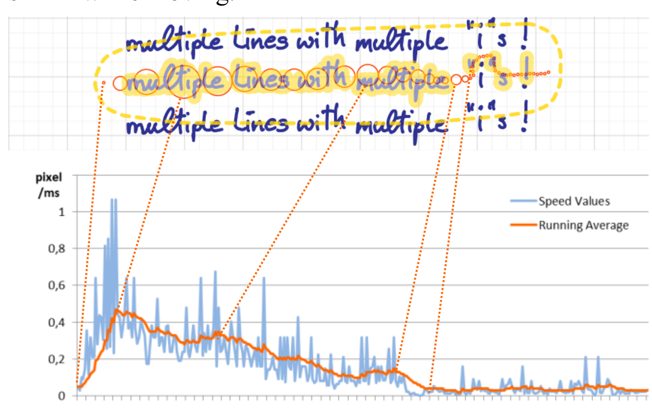

Figure 4: The orange circles represent the spot size during the selection operation (top). The graph (bottom) shows the raw input values (blue), and the resulting smoothed values (orange) used to calculate the spot size.

Selection modifications

Starting a new selection in the *Harpoon* tool leaves already existing selections unchanged, even if no items are selected in the new selection operation. For most other selection techniques (e.g., *Lasso*) starting a new selection automatically resets the last selection. For the *Harpoon* tool, a selection can only be reset when a single tap on the drawing canvas is performed. This minimizes the risk of accidently cancelling the current selection.

Adding items to an existing selection is usually achieved by switching to a dedicated mode (e.g., by pressing the "shift"-key on a keyboard). Removing objects from a current selection is achieved in a similar way. With the *Harpoon* tool, existing selections can be modified without an explicit switch to a different mode.

Single tap

A single tap on any object (e.g., ink) toggles its selection status. Figure 5 shows the result after two separate *single-taps*. Tapping the previously unselected *"h"* in the line above adds it to the current selection, clicking the *"m"* of the current selection removes it.

Figure 5: Adding and removing objects can be achieved with a single tap.

If a user clicks on overlapping targets, only the topmost object is selected or deselected. This allows users to achieve very precise modifications and quickly add or remove only a single item to/from the current group.

Dragging

Tapping is well suited for single targets [15]. However, it is cumbersome for large groups. Hence, a dragging operation can also be used to modify the current selection. We explored different methods how to add and remove items. Initially both operations could be performed in a single drag operation. Each items status was simply toggled upon hit. Already selected items were deselected and vice versa. However, early user tests quickly showed that this required a very precise steering during the whole operation and often resulted in unwanted *select* or *deselect* operations. For the *Harpoon* tool, the type of selection operation is determined upon touching the first item. We call the implicit mode-change *"first hit"*.

Figure 6: Starting a new selection operation does not reset the previous selection.

Figure 6 shows an existing selection as well as the start of a new selection. At this point the type of operation is not yet determined, the selection path, the spot and the convex hull are rendered in the default color (*yellow*). If at this point the pen would be lifted, the current selection will remain unchanged. The mode-change happens as soon as an item is hit (cf. Figure 7). If the *first hit* item was already part of the selection, the *Harpoon* selection tool enters the *deselect*-mode. A small "-"-icon appears above the cursor and the tools visuals change their color to provide additional feedback, cf. Figure 7 (*top*).

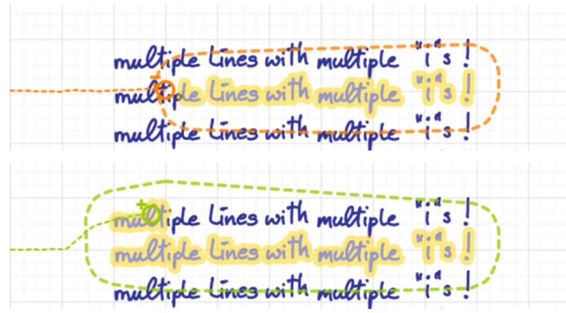

Figure 7: Larger modifications can be achieved through dragging. The first hit object determines selection mode.

Any hit item that was already part of the selection is now removed from it. Objects that are not part of the current selection remain unaffected and cannot be added accidently. If instead the first hit item is not part of the current selection, *select*-mode is activated, see Figure 7 (*bottom*). In this mode, all elements that are already selected remain unaffected and cannot accidently be removed.

After lifting the pen, the *Harpoon* tool automatically switches back to default mode. For each new operation the mode is again decided upon "first hit". This way, it is possible to quickly achieve complex selections by chaining together several separate *select* and *deselect* operations.

Overlapping selection targets

Lasso selections often require a certain percentage of the target (e.g. 50%) to be contained within the *Lasso*-region. This can result in problems when users are trying to select overlapping targets. In contrast to *Lasso* selections it is easy to select several overlapping selection targets with the *Harpoon* selection tool. Any target has to be hit only once with the selection spot. This results in a less constrained and a less complex selection path. In Figure 8 only a single vertical stroke is required to select all three overlapping targets. If errors are made, the previously described options to quickly modify existing selections further improve selection of overlapping strokes.

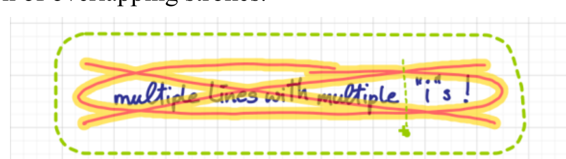

Figure 8: Multiple overlapping strokes (red) can be selected efficiently without selecting any underlying strokes (blue).

EXPERIMENT

To validate our design we conducted a user study comparing our design to other selection techniques. The techniques are described below. Visual feedback on which items were selected was the same for all described techniques.

Selection Techniques

Four different techniques were tested to compare the performance of our proposed selection technique: *Tapping*, *Lassoing*, *Crossing*, and *Harpooning*.

Tapping

Targets are selected by tapping on individual targets. Selected targets can be deselected with a second click. Clicking on the canvas resets the current selection.

Lassoing

For the *Lasso* tool, targets had to be contained 50% within the *Lasso* region to be selected. To allow users to correct selection errors, new *Lasso*-selections did not reset the current selection, but modify the existing selection by toggling the selection state of each object in the new *Lasso* selection. Items that were not part of the existing selection were added to the selection and vice versa. As with all other techniques a single click on the canvas could be used to reset the current selection.

Crossing

Targets are selected by crossing them with the tip of the pen. Several targets can be selected in a single dragging operation. In contrast to the harpoon selection technique for this technique hit testing was only performed for a single point instead of a larger cursor area. Selections could be modified the same way as described for the *Harpoon* tool. Again a single click on the canvas reset the current selection.

Harpooning

The *Harpoon* tool in the experiment worked as described in the previous section. Targets are selected by touching (crossing) them with the spot during a dragging operation. Cursor size was defined by movement-speed and ranged from 1 pixel to 50 pixels in diameter.

HYPOTHESIS

Before conducting the experiment the following experimental hypotheses were developed:

- *Tapping* will be the slowest technique overall.
- *Harpooning* will be the fastest technique overall.
- *Harpooning* will be best for objects within dense surroundings.
- *Lassoing* will be best for objects within sparse surroundings.
- *Crossing* will be best for single objects.

The corresponding null hypotheses to be tested were that none of the differences predicted in the experimental hypotheses would in fact exist.

EXPERIMENT METHOD

Participants

12 participants (2 females) were recruited from the local university as well as from a local company. Participants' age ranged from 20 to 31 years. All had experience with image editing software; 7 had used interactive whiteboards before. All participants (one left-handed) controlled the stylus with their dominant hand.

Apparatus

The experiment was conducted in a quiet room equipped with an 80" Polyvision Eno 2610 interactive whiteboard. A Toshiba ex20 short throw projector with a resolution of 1024×768 was used for projection. The board was calibrated at the beginning of user testing.

Task

Our experiment task is based on the experiment of Grossman et al. [8]. Our task environment consists of a red start-button, a 6×6 grid of targets, as well as a green finish button. Similar to [8], only the 16 internal targets are candidates for the selection. The border targets are present to ensure the same distance constraints for all 16 internal targets. At the beginning of each trial, all targets are inactive (*black*). The finish button is deactivated and the start button is activated. Each trial begins by pressing the red start button, which turns a set of targets red. Participants now have to select all red targets as quickly as possible using the currently active selection technique.

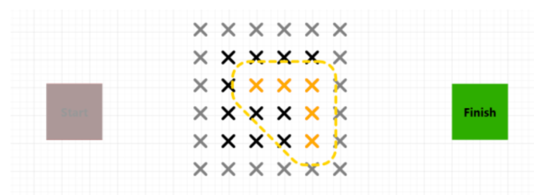

Figure 9: The experimental environment after successfully selecting all targets.

A semitransparent yellow overlay is used to highlight all selected targets. Once all active targets have been selected successfully, the finish button turns green. This happens only after the selection operation is finished and the pen is lifted from the board. If the wrong targets had been selected, the finish button remains inactive. Participants have to correct their selection before being able to press the finish button. Sets of targets varied in terms of *complexity, difficulty,* and *distance*. Three different levels of complexity, two difficulty levels, and two distances were tested.

Figure 10 shows the different target sets for a distance of 27-pixel. The distance is measured between the centers of the crosses. The complexity levels were *Single*, *Line,* and *Corner*. In the first difficulty setting, crosses were used as targets. Both target's width and height were 25 pixels, which results in a target-size of 40mm on the whiteboard. In the second difficulty setting, four circles with five pixels (8mm) diameter each where placed around the center of the cross. All circles had to be selected. The second difficulty setting was added to simulate selection of ungrouped items like unparsed, handwritten text, which often consists of multiple separate strokes. Participants were told to complete each trial as fast as possible. Trial times were measured between pressing the start and the end button.

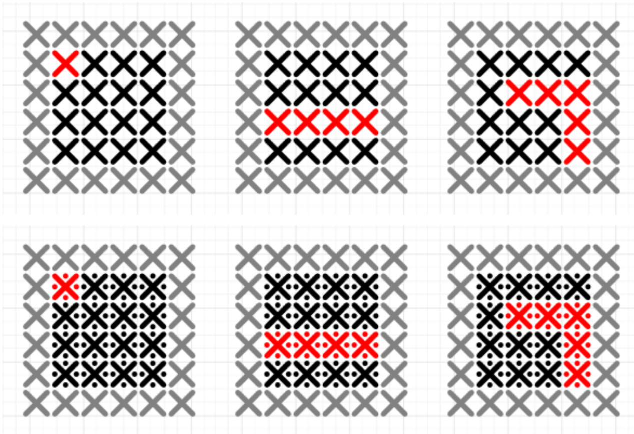

Figure 10: In the experiment, we implemented three different complexity levels. All red-colored items had to be selected. In the scenarios at the bottom, participants also had to select the four dots.

All trials could only be completed if all active targets were selected. Consequently trial times included the time necessary to correct errors. This approach to collect quantitative data was taken as the design of the *Harpoon* tool offers many different strategies (sequences of select and deselect operations) to get a correct selection. As a result it is impossible to quantitatively count errors. Instead, we used the qualitative measurements as an indicator for how often the different techniques produced unwanted or unexpected selection results that required the participant to modify the selection.

Experimental Design

A repeated measures within-subject design was used. Technique (*Tapping, Lassoing, Crossing, Harpooning*), Complexity (*Single, Line, Corner),* Difficulty (*cross without dots, cross with dots*), and Distance (*27px, 50px)* were used as independent variables. Presentation order for the techniques was counterbalanced using a 4×4 Latin Square. Before starting the actual trials for each technique the technique was explained to the participants. They were then allowed to familiarize themselves with each technique for an unlimited training period. For each technique, a total of 36 trials had to be completed. Summarizing, each participant completed a total of 144 trials (4 techniques × 2 distances × 3 complexities × 2 difficulties × 3 trials).

In addition, qualitative feedback was collected after each technique using a questionnaire. Participants had to rate each technique in regards to *ease of learning, speed, error rate, error correction* and *overall usability* on a 7-point Likert scale. After completing the whole test, participants were also asked to choose the best and worst techniques for each category. The whole test including training sessions and questionnaires lasted for approximately 30 minutes per person.

QUANTITATIVE RESULTS

Due to our task design, all trials could be used for evaluation. For all tests an alpha level of 0.05 was used. The Greenhouse-Geisser correction was used if the assumption of sphericity was violated. A repeated measures analysis of variance showed main effects for *Technique* ($F_{3,44}$ = 35.046, $p < .0001$), *Distance* ($F_{1,22}$ = 22.355, $p < .001$), *Difficulty* ($F_{1,22}$ = 548.549, $p < .0001$), *Complexity* ($F_{2,33}$ = 384.323, $p < .0001$), and *Trials* ($F_{2,33}$ = 22.132, $p < .0001$)

Hypothesis

Post-hoc analyses on the main effects were conducted in order to confirm/reject the formulated hypotheses. These consisted of paired-samples *t*-tests with family wise error rate controlled across the test using Holm's sequential Bonferroni approach. For all bar charts, the error bars indicate the range of two standard errors of the mean (above and below the mean).

Hypothesis 1: Tapping will be the slowest technique overall

Post-hoc analysis showed that Tapping was significantly slower than all other techniques ($p < .0001$). Hence the first Hypothesis can be confirmed. Tapping was also voted the slowest technique by all 12 participants. Figure 11 shows the overall mean selection times by technique. Overall selection times for all techniques were 5.40s (*SD* = 0.81) for *Tapping*, 4.10s (*SD* = 0.71) for *Lassoing*, 3.80s (*SD* = 0.63) for *Crossing* and 3.48s (*SD* = 0.44) for *Harpooning*. On average targets took 1.6s longer to select all targets using *Tapping*.

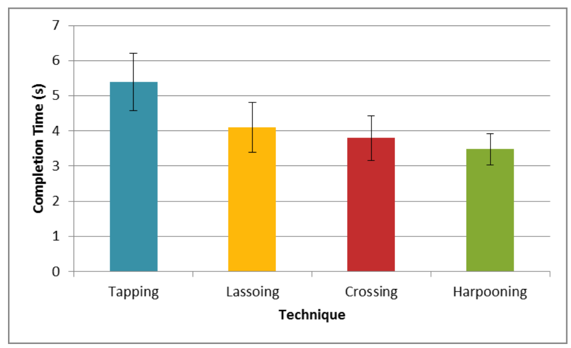

Figure 11: Overall completion times by technique.

Hypothesis 2: Harpooning will be the fastest technique overall

Overall *Harpooning* was significantly faster than *Tapping* and *Lassoing* ($p < .01$). *Harpooning* in our experiment also was on average 0.32s faster than *Crossing*, however pairwise comparison showed no statistical significance (p = .121). Hence the hypothesis has to be rejected.

Hypothesis 3: Harpooning will be best for objects within dense surroundings

Harpooning did not perform best for selections within dense surroundings, the hypothesis has to be rejected. Average completion times were 5.04s (*SD* = 0.72) for *Tapping*, 5.01s (*SD* = 1.02) for *Lassoing*, 3.89s (*SD* = 0.55) for *Crossing,* and 4.06s (*SD* = 0.65) for *Harpooning*.

Crossing performed significantly better than *Tapping* and *Lassoing* ($p < .01$), *Harpooning* performed significantly better than *Tapping* ($p < .01$). There was no significant difference in performance between *Crossing* and *Harpooning*. Figure 12 shows completion times for different distances.

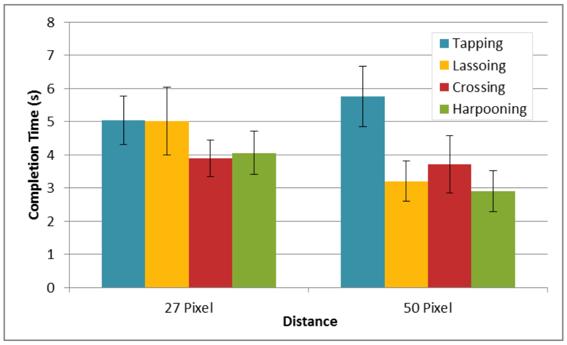

Figure 12: Completion times by technique and distance.

Hypothesis 4: Lassoing will be best for objects within sparse surroundings

Lassoing resulted only in the second fastest completion time. Hence the hypothesis has to be rejected. Instead *Harpooning* was significantly faster than all other techniques ($p < .01$) for the 50 pixel distance trials. Completion times were 5.76s ($SD = 0.91$) for *Tapping*, 3.20s ($SD = 0.61$) for *Lassoing*, 3.70s ($SD = 0.86$) for *Crossing,* and 2.90s ($SD = 0.62$) for *Harpooning*.

Hypothesis 5: Crossing will be best for single objects

Crossing was fastest for single objects however this did not reach significance. Completion times for single objects were 1.86s ($SD = 0.27$) for *Tapping*, 2.55s ($SD = 0.47$) for *Lassoing*, 1.80s ($SD = 0.37$) for *Crossing,* and 1.81s ($SD = 0.22$) for *Harpooning*, cf., Figure 13.

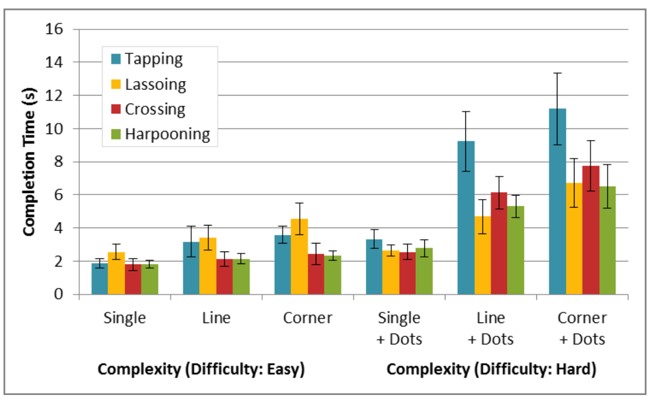

Figure 13: Completion times for both difficulty settings by technique and complexity

Lassoing was significantly slower than all other techniques ($p < .001$) for selecting single objects. No other significant differences were found. Figure 13 shows completion times by technique and complexity separated by difficulty setting.

Discussion

Looking at the quantitative data, the *Harpoon* selection tool shows very promising results with a significant overall speed improvement over the more traditional methods of *Tapping* and *Lassoing*. Instead of excelling at one specific category, *Harpooning* has proven to result in consistently fast completion times for very diverse selection tasks. Even in cases where other techniques might have performed better, the *Harpoon* tool never was significantly slower. This characteristic is especially beneficial for environments, where tool changes are time-consuming (such as interactive whiteboards).

Performance gains might be even larger in real life situations, as the following two examples will show. In a first pilot test using the same minimal distance of 25 pixels as in [8] (compared to 27 pixels used in the actual experiment), trials took very long to complete for *Lassoing*. Participants got really frustrated, even claiming to quit the test if it would stay this difficult. No such problems were noticed with the other techniques. While keeping the 25-pixel scenario might have resulted in much greater completion time differences we did not want to risk annoying participants and as a result distorting trial results. For the same reason, trials using overlapping targets where omitted. Consequently, in the final study design a two-pixel gap remained between the targets in the 27-pixel distance trials. However, in everyday use overlapping items are very well possible, which would suggest an even greater benefit for alternative tools like the *Harpoon* tool over the lasso tool. We are especially pleased that *Harpooning* even offers a significant speed improvement over *Lassoing* in the 50-pixel scenarios. These results are achieved even with qualitative results indicating that the *Harpoon* tool is the most difficult to learn. We believe that as participants become more familiar with the speed-dependent spot size, selection times in sparse environments can be reduced even further.

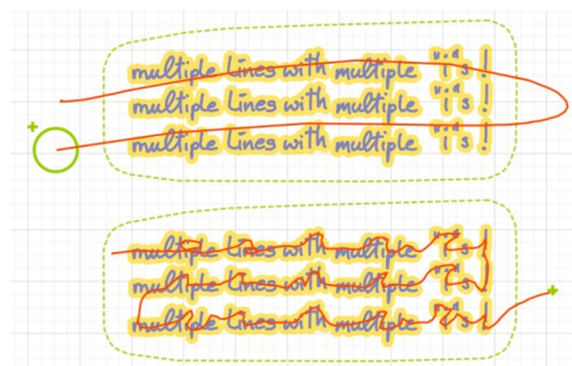

Figure 14: *Harpooning* (*top*) takes considerable less time and effort than *Crossing* (*bottom*). For better visibility the selection path is displayed in red.

Surprisingly, the overall completion time results show only a small difference between *Harpooning* and *Crossing*. We strongly believe that this difference will get larger in real world use-cases especially for sparse ink strokes. Figure 12 shows that already in the 50-pixel scenarios *Harpooning* offers a significant performance improvement over *Crossing*. With larger numbers of (potentially smaller) targets, this gap would get even larger.

The three handwritten phrases "*multiple lines with multiple "i" s!*" as seen in Figure 14 already consist of 3×36 = 108 separate strokes, many of which have only half the size of the dots used in the study. Also the layout of the individual objects is not fixed like in the study, requiring users to constantly plan their selection path and check not to forget any objects. *Crossing* each individual object consequently results in a very complex steering task that requires constant user attention. (cf., Figure 14 *bottom*). In contrast one swift movement with the *Harpoon* tool selects all objects (cf., Figure 14 *top*). Also *Tapping* all 108 separate strokes would result in a much more tedious and time consuming experiment. Then again looking at a different selection scenario, selecting only a single, overlapping stroke could be virtually impossible using the *Lasso*-tool.

Consequently using similar selection targets as depicted in Figure 14 in the study would have biased the study towards *Harpooning* and would also most likely required exclusion of *Lassoing* and *Tapping*. While both *Tapping* and *Lassoing* are well studied techniques they still represent the standard selection tool in many applications to date and hence were included as base-line techniques. However based on the insights of this study a follow up experiment comparing *Crossing* and *Harpooning* using free from ink selection targets might help show the benefits of *Harpooning* even more clearly. Such a comparison as well as follow up studies concerning learning- or fatigue effects, error rates and error correction is seen as future work.

Summarizing we believe that based on the insights gained through the study as well as additional user observations as described earlier, *Harpooning* successfully combines properties that allow users to quickly select ungrouped objects in various different situations. While further testing will show how the tool will perform in real-world scenarios, first indicators already look very promising.

QUALITATIVE RESULTS

After each set of trials participants had to rate each technique concerning *ease of learning, speed, error rate,* and *error correction.* Additionally participants were asked to rate overall usability. Each questions was answered using a Likert scale from 1 to 7 (1 = very good and 7 = not very good). The results for the first four categories are summarized in Figure 15. Overall results are shown in Figure 16.

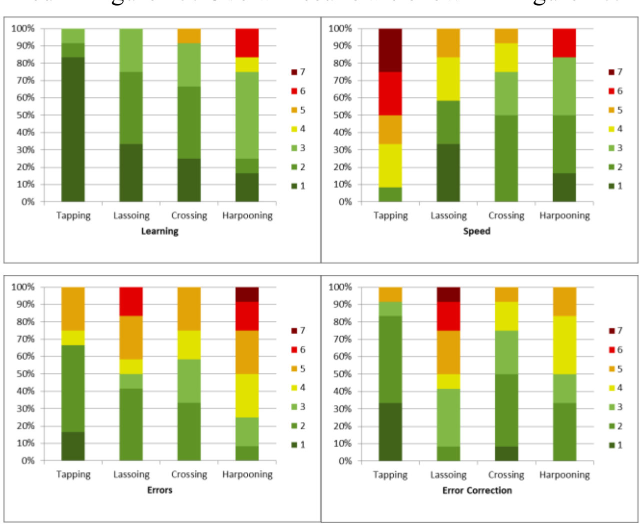

Figure 15: Ratings on *ease of learning, speed, error rate,* and *error correction* based on a 7-point Likert scale (1 = very good).

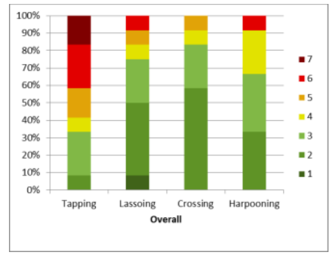

Figure 16: Overall usability ratings of the different techniques (1 = very good).

After completing all trials, participants were also asked to select their favorite and least favorite techniques for each of the aforementioned categories. These results are presented in Figure 17. The qualitative data suggests that *speed* was most influential factor on the participants' *overall* rating. Looking at the raw data shows that for 10 out of 12 participants the fastest technique was also their most favorite overall. For the slowest technique, we get similar results. 8 out of 12 participants selected the same technique as the slowest technique as well as least favorite overall. This can also be seen in Figure 17 (relevant bars are highlighted). At first glance other factors like ease of learning, error rate, or error correction seem to play only a secondary role. The majority of participants said that *Tapping* performed best on ease of learning, error rate, and error correction. Nevertheless, no participant chose *Tapping* as his/her favorite.

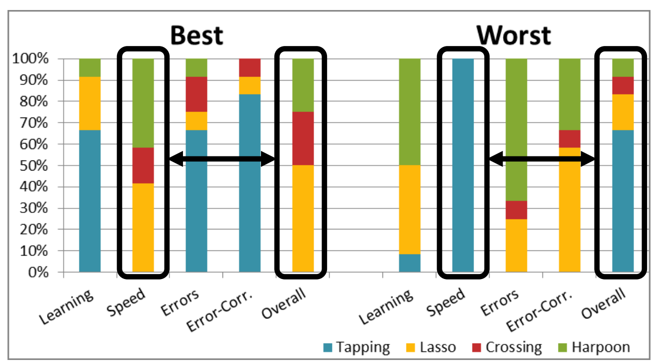

Figure 17: After completing all trials participants were asked to choose the best and worst technique. Overall rating was most influenced by the participants' speed rating.

The quantitative results as well as qualitative speed ratings are in favor of *Harpooning.* Still *Lassoing* was selected as the overall favorite tool by most participants. This is because both error rate and error correction performance indirectly influenced speed ratings and consequently overall ratings. This can be explained with the chosen experimental design: participants could only press the finish button after all targets have been selected correctly: Therefore, both errors and the ability to quickly correct them influenced the overall time. Consequently, techniques resulting in fewer errors required less additional time to complete the trial. Looking again at the raw data, we noticed that participants who rated a technique's error rate poorly also gave poor speed ratings. By reducing the number of errors, not only the measured completion time would be reduced, but most likely also subjective speed assessments and overall usability. The qualitative results show that *Harpooning* resulted in a higher error rate. We have identified several problems that contributed to this high error-rate:

- The current spot-size for the *Harpoon* selection was very hard to see, especially for faster movement. As a result choosing the correct movement speed to achieve an optimal spot size was very difficult.
- It was impossible to notice exactly, which areas have already been visited. Thus, it was hard to quickly spot objects that have been missed.
- Smoothing the input values caused spot size to grow too slowly.

Based on these results, we made several changes to the *Harpoon* selection tool as described in the next session.

IMPROVEMENTS

Several participants commented that the current feedback of the *Harpoon* tool made it very difficult to see which areas have been selected. This made it more difficult to quickly see whether any objects had been missed. As a result, we changed the feedback of the selection path from a dotted line to a semitransparent stroke with its thickness defined by the spot diameter, cf., Figure 19.

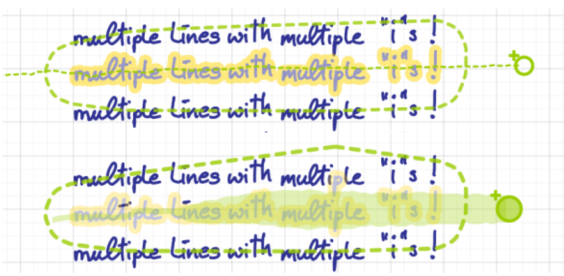

Figure 18: Instead of a dotted line (*top*), a semi-transparent stroke with varying thickness is drawn (*bottom*). This makes it a lot easier to notice which areas have been visited and helps spotting objects that might have been missed.

In addition, the spot is now filled to improve visibility during fast movement. During the selection operation, all selected objects turn even more transparent to highlight missed objects, cf. Figure 18 (*bottom*).

In the *deselect*-mode, selected items turn opaque to stand out more against the background (see Figure 19). We also render a semitransparent white area between the current selection and the background to further improve this contrast.

Figure 19: When deselect-mode is engaged, all selected items turn completely opaque to stand out more against the background.

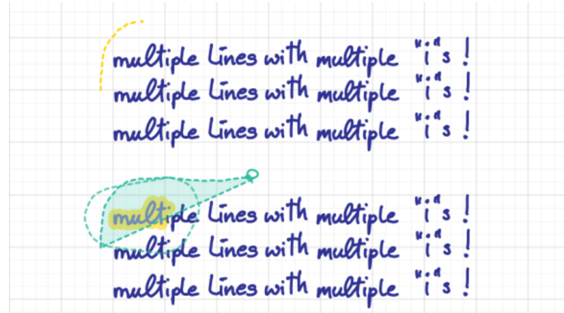

Figure 20: If a user starts circling around objects, the tool automatically switches to the Lasso mode. If a stroke is crossed directly, the Harpoon mode is engaged.

First informal user tests show that these improvements help users finding missed objects more quickly. The changes also help to find overlapped strokes.

In certain cases, participants would lasso objects even in a different selection technique (e.g. *Harpooning*). We automatically detect this and switch mode accordingly. When a new selection is started, the *Harpoon* selection tool also simultaneously performs *Lasso* hit-tests during movement. If users are starting to circle an object without directly crossing it, the harpoon tool automatically switches to *Lasso* mode as soon as the first object is contained (e.g. 50%). However, if the first stroke is hit directly with the spot, *Lasso* hit-testing is disabled and the *Harpoon* tool acts as described before (see Figure 20).

We also noticed that smoothing the input values sometimes caused the spot size to grow too slowly. As a result, objects often were missed at the beginning of a selection operation. Therefore, we perform a second row of hit-tests for the first input points with the current spot diameter after receiving 10 input points. Figure 21 shows a debugging view of this approach. This allows us to compensate for this initial delay without increasing the update rate and introducing the risk of erratic spot-size changes.

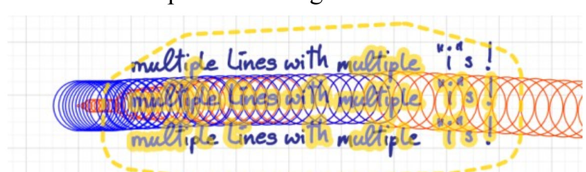

Figure 21: Smoothing the input values sometimes causes the spot size to grow too slowly (orange circles). This can be corrected after a sufficient number of values have been received (blue circles).

CONCLUSION & FUTURE WORK

In this paper, we have presented a new kind of selection technique, called *Harpooning*. *Harpooning* is designed especially for interactive whiteboard surfaces and is based on crossing objects using a speed dependent area cursor. A first user test showed that *Harpooning* resulted in significantly reduced selection times compared to *Tapping* and *Lassoing*. User-feedback showed that *Harpooning* results in high error rates that influence overall usability of the tool. During the study, we have discovered several error sources and we have presented solutions for each problem. We have started prolonged tests outside the lab. Gathered feedback will be collected to further improve of the tool.

For future work, we are also interested in how the tool can integrate with systems that facilitates fast selection of grouped objects [8]. Such a combination would enable users to select grouped objects quickly without sacrificing easy access to fine grain control of selection.

ACKNOWLEDGMENTS

This project is part of the Research Studio Austria NiCE, funded by the FFG, 818621. We would like to thank Florian Perteneder, Christian Rendl and Thomas Seifried for their help with the project. We also would like to thank our reviewers for their useful comments.

REFERENCES

1. Accot, J. and Zhai, S. 1997. Beyond Fitts' law: models for trajectory-based HCI tasks. In CHI '97, Steven Pemberton (Ed.). ACM, New York, NY, USA, 295-302.
2. Accot, J. and Zhai, S. 2002. More than dotting the i's - foundations for crossing-based interfaces. In CHI '02. ACM, New York, NY, USA, 73-80.
3. Apitz, G., Guimbretière, F. and Zhai, S. 2008. Foundations for designing and evaluating user interfaces based on the crossing paradigm. ACM Trans. Comput.-Hum. Interact. 17, 2, Article 9 (May 2008), 42 pages.
4. Brewster, S.A. 1998. Using Earcons to Improve the Usability of a Graphics Package. In HCI '98, Hilary Johnson, Laurence Nigay, and Chris Roast (Eds.). Springer-Verlag, London, UK, 287-302.
5. Buxton, W., Fiume, E., Hill, R., Lee, A. and Woo, C. 1983. Continuous Hand-Gesture Driven Input. Graphic Interface. 191-195.
6. Chapuis, O., Labrune, JB. and Pietriga E. 2009. DynaSpot: speed-dependent area cursor. In CHI '09. ACM, New York, NY, USA, 1391-1400.
7. Fitzmaurice, G., Khan, A., Piek R., Buxton, W. and Kurtenbach, G. 2003. Tracking menus. In UIST '03. ACM, New York, NY, USA, 71-79.
8. Grossman, T., Baudisch, P., and Hinckley, K. 2009. Handle Flags: efficient and flexible selections for inking applications. In GI '09. Canada, 167-174.
9. Guimbretière, F., Stone, M., and Winograd, T. 2001. Fluid interaction with high-resolution wall-size displays. In UIST '01. ACM, New York, NY, USA, 21-30.
10. Haller, M., Leitner, J., Seifried, T., Wallace, J. R., Scott, S. D., Richter, C., Brandl, P., Gokcezade, A., Hunter, S., 2010. The NiCE Discussion Room: Integrating Paper and Digital Media to Support Co-Located Group Meetings. In CHI '10, ACM, New York, NY, USA, 609-618.
11. Hinckley, K., Baudisch, P., Ramos, G., and Guimbretiere, F. 2005. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In CHI '05. ACM, New York, NY, USA, 451-460.
12. Kabbash, P. and Buxton, W. 1995. The "prince" technique: Fitts' law and selection using area cursors. In CHI '95, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 273-279.
13. Kurtenbach, G. and Buxton, W. 1991. Issues in combining marking and direct manipulation techniques. In UIST '91. ACM, New York, NY, USA, 137-144.
14. Lank, E. and Saund, E. 2005. Sloppy Selection: Providing an Accurate Interpretation of Imprecise Stylus Selection Gestures. Computers and Graphics. 29(4): 490-500.
15. Mizobuchi, S. and Yasumura, M. 2004. Tapping vs. circling selections on pen-based devices: evidence for different performance-shaping factors. In CHI '04. ACM, New York, NY, USA, 607-614.
16. Moran, T.P., Chiu, P. and Melle, W. 1997. Pen-based interaction techniques for organizing material on an electronic whiteboard. In UIST '97. ACM, New York, NY, USA, 45-54.
17. Plamondon, R. and Srihari, S. 2000. On-line and Offline Handwriting Recognition: A Comprehensive Survey. IEEE Pattern Analysis. 22(1), 63-84.
18. Rekimoto, J. 1997. Pick-and-drop: a direct manipulation technique for multiple computer environments. In UIST '97. ACM, New York, NY, USA, 31-39.
19. Ren, X., and Moriya, S. 2000. Improving selection performance on pen-based systems: a study of pen-based interaction for selection tasks. ACM Trans. Comput.-Hum. Interact. 7, 3 ,384-416.
20. Saund, E. and Lank, E. 2003. Stylus input and editing without prior selection of mode. In UIST '03. ACM, New York, NY, USA, 213-216.
21. Saund, E., Mahoney, J., Fleet, D., Larner, D. and Lank, E. 2002. Perceptual Organization as a Foundation for Intelligent Sketch Editing. AAAI Symposium on Sketch Understanding. 118-125.
22. Yin, J. and Ren, X. 2007. Investigation to line-based techniques for multi-target selection. In INTERACT'07, Springer-Verlag, Berlin, Heidelberg, 507-510.