Off-Limits: Interacting Beyond the Boundaries of Large Displays

Anders Markussen, Sebastian Boring, Mikkel R. Jakobsen and Kasper Hornbæk
Department of Computer Science, University of Copenhagen, Copenhagen, Denmark
{amark, sebastian.boring, mikkelrj, kash}@di.ku.dk

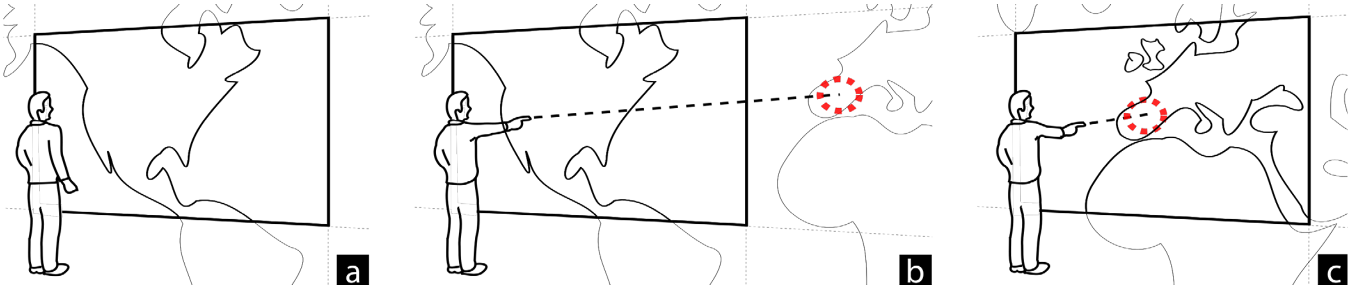

Figure 1. An illustration of the *Off-Limits* concept: (a) a user looks at a view of North America on a display; (b) the user points at the location of Europe in off-screen space; (c) the user drags Europe onto the screen from its off-screen location.

ABSTRACT
The size of information spaces often exceeds the limits of even the largest displays. This makes navigating such spaces through on-screen interactions demanding. However, if users imagine the information space extending in a plane beyond the display's boundaries, they might be able to use the space beyond the display for input. This paper investigates *Off-Limits*, an interaction concept extending the input space of a large display into the space beyond the screen through the use of mid-air pointing. We develop and evaluate the concept through three empirical studies in one-dimensional space: First, we explore benefits and limitations of off-screen pointing compared to touch interaction and mid-air on-screen pointing; next, we assess users' accuracy in off-screen pointing to model the distance-to-screen vs. accuracy trade-off; and finally, we show how *Off-Limits* is further improved by applying that model to the naïve approach. Overall, we found that the final *Off-Limits* concept provides significant performance benefits over on-screen and touch pointing conditions.

Author Keywords
Mid-air pointing; Off-screen pointing; In-air pointing; Spatial user interfaces; Freehand pointing.

ACM Classification Keywords
I.3.6 Methodology and Techniques: Interaction techniques

CHI'16, May 07-12, 2016, San Jose, CA, USA
© 2016 ACM. ISBN 978-1-4503-3362-7/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858083

INTRODUCTION
Many information spaces (e.g., detailed maps) exceed the limits of even wall-sized displays. The display is thus a viewport into these spaces and information residing *off-screen* can be brought *on-screen* (i.e., inside the viewport) through navigation. Users commonly do this by moving the information space (and its digital representation respectively) through either mouse and keyboard [17], personal devices (e.g., tablets or phones [25]), touch input, or mid-air interaction to interact from afar on larger displays [29].

Although users may be able to imagine the information extending into *off-screen* space, the physical dimensions of a display often determine the available input space: touch is performed directly *on-screen*; and mid-air pointing commonly uses ray casting onto the display's surface. More traditional input methods (e.g., a mouse pointer) are also bound to the screen, as users have to keep track of cursors. As a consequence, the input space is usually tied to the display containing the visual output and is thus much smaller than the actual information space that users interact with.

Display size constraints on mobile devices have forced researchers to think *outside the box*, allowing for interactions to occur *around* the device rather than on its display (e.g., [14,15]). In this paper, we adapt the general idea of using off-screen space to large displays. Unlike previous approaches for mobile devices, however, we *extend* the input space beyond a large display's boundaries – thus allowing people to seamlessly use both *on-screen* and *off-screen* space throughout an interaction.

OFF-LIMITS
As illustrated by Figure 1, *Off-Limits* allows the use of mid-air interaction within an input space that extends beyond display boundaries, thereby enabling a much larger input space than previous techniques (e.g., [22,34]). In doing so, we match the input space to a much larger part of the in-

formation space instead of tying it to the space within the display's boundaries. With *Off-Limits*, people can use their knowledge of spatial relations in the presented information. For example, locations on a map have certain distances and orientations relative to each other. Despite being out of view, points of interest can be addressed in off-screen space directly, using the parts of the information space that are visible on the display as reference.

Figure 1 illustrates this idea using a map shown on a large display: (a) in the beginning the display shows North America, but the user wants to display the map of Europe; (b) the well-defined relation between the two continents allows the user to point at Europe's location in off-screen space; (c) Europe can now be dragged into the viewport in a single dragging operation, where that drag started in off-screen space and ends in on-screen space (in case the user underestimated the distance to Europe, the drag may even continue into off-screen space on the opposite side of the display).

The main benefit of *Off-Limits* is that it frees the input space from the physical limitations of a display. It extends two common operations: (1) it allows for addressing a point of interest (of which users know the spatial location) directly in *off-screen* space, without using repeated on-screen dragging operations (i.e., clutching); and (2) it allows for starting and/or continuing dragging operations in *off-screen* space (i.e., beyond the display's border), without interrupting interaction when the display's borders are reached. Further, *Off-Limits* can be implemented to allow for bi-manual operation (similar to bi-manual *Multipoint* [2], yet in off-screen space), where users may address two off-screen areas simultaneously (e.g., to perform on-screen comparison).

In this paper, we contribute three experiments that help develop and evaluate this concept on large displays: the first experiment demonstrates that off-screen space is suitable (and complementary) for interacting with large displays. In the second study, we assess users' accuracy in pointing to locations in off-screen space, leading to a model for estimating the perceived location of a point in off-screen space based on the points' distance from the display's center. With this model we refine the naïve adaptation of *Off-Limits*. In the third experiment, we demonstrate *Off-Limit*'s superior performance compared to the naïve implementation regarding interaction time, number of interactions and user satisfaction. Our improvements make *Off-Limits* a compelling candidate for future large display interactions.

RELATED WORK

The work presented in this paper builds on research in interaction techniques for large displays, techniques that allow for distant interaction, and off-screen interaction.

Interaction Techniques for Large Displays

Large, high-resolution displays have introduced new challenges for designers of touch- or pen-based interaction techniques: (1) due to the display's size, content may be out of the users' reach; and (2) content is not always readable to users even though it is shown on the display, due to its distance to the user. Previous work, discussed below, has focused on these challenges and presented a number of techniques to improve the interaction with distant content, when being close to the display.

Approaches either virtually increase the user's reach or bring distant objects or areas closer to the user for selection. With *HybridPointing* [10], users can switch back-and-forth between absolute and relative pointing (using a direct input device) to increase their reach. *Drag-and-throw* and *Push-and-throw* [8] allow users to bridge large distances while dragging an object. The user only has to move a short distance, which is then amplified. *Press-and-flick* [26] works similarly for moving objects across large distances, but it does not allow for controlling the object's movement once it is released (i.e., open-loop). *Drag-and-pop* [31] brings remote objects of interest closer (e.g., dragging a document brings potential targets closer to the user). *Frisbee* [21], on the other hand, creates a portal to a distant area of the display, allowing for fast interchange of objects between both the local and remote area.

Each of these techniques enables access to distant (and potentially off-screen) locations on large displays. At the same time, however, they discourage using the inherent benefits of large displays – namely the users' ability to move freely (e.g., to get an overview of the information space [1]).

Interaction at a Distance

To free users from having to interact up close to the display, researchers have explored how to support interacting while further away from the display. One such way is to use dedicated devices, such as keyboards or gyroscopic mice. A user's personal device (e.g., a phone or a tablet) can also be used. *Touch Projector* [4] is one such system that allows interacting with distant displays using a phone's camera. Jansen et al. [18] present tangible remote controllers that can be used in concert with a tablet to allow for rich input to a large display from afar. *ARC-Pad* [25] uses a world-in-miniature view on a handheld device in combination with absolute and relative pointing. Nancel et al. [27] increase pointing accuracy, despite the mismatch in size between a mobile device and a large display.

To avoid having secondary and dedicated interaction devices, research turned to mid-air pointing techniques, where people point using their arms and fingers (e.g., [7,20,28]). In addition to pointing at targets, previous work also focused on target selection using mid-air techniques. Vogel & Balakrishnan [34] created two techniques for triggering a selection: *AirTap* and *ThumbTrigger*. They further evaluated techniques for distant freehand pointing, and found absolute pointing to be beneficial when tasks required bridging larger *on-screen* distances. Banerjee et al. [2] presented techniques for bi-manual on-screen manipulation of objects and found them superior to laser pointer based techniques. Nancel et al. [29] extended such interactions to also allow for zooming (besides panning) using mid-air gestures. Unlike *Off-Limits*, these interactions operate on-screen: they improve on-screen selection performance or bring off-screen content into the display through a series of on-screen pan and zoom interactions.

Kopper et al. [22] explored distal pointing tasks on large displays. They intended to model such techniques using Fitts' Law, and found that it is possible as a function of angular amplitude (i.e., the distance of the target in degrees) and angular target size. In contrast to Fitts' Law, however, the task difficulty grew as a quadratic function, indicating that pointing difficulty grows rapidly with increasing angular amplitudes. Further, a target's angular size had an even larger impact than angular amplitude. This is highly relevant for *Off-Limits*, as angular amplitudes further increase when going beyond the display.

Off-Screen Interaction

When the display acts as a viewport onto a larger information space, parts of the information space will reside *off-screen*. Users can access off-screen content through pan-and-zoom interfaces, where they must integrate overview and detail over time [6] or by using world-in-miniature representations of the entire information space [32]. Proxy-based interfaces further allow for accessing off-screen content: *Halo* [3], *Wedge* [12], and *City Lights* [36] are examples of such interfaces, which provide on-screen visual cues of an object's distance and orientation. Irani et al.'s work on off-screen content validates that proxies are useful for quickly accessing off-screen objects [15,16]. However, proxy-based systems require that the system knows the objects of interest before the interaction takes place.

Each of these techniques still uses the display as input space. Instead of changing the information space's representation through on-screen operations, a device can (if possible) also be moved to change its viewport into that space. *Peephole Displays* [35] is one such approach, where the information space remains fixed and a mobile device instead is moved within that space. *Virtual Shelves* [23] similarly makes use of a larger information space (here: applications sorted in a grid), which users can select by pointing at their invisible location in space. Like *Off-Limits*, these systems enable users to move to an off-screen location, rather than moving digital information on-screen.

Finally, and most related to our work, is research on free-hand off-screen pointing and interaction. Ens et al.'s work [9] and *AD-Binning* [13] allows placing and retrieving objects or locations in the proximity of small handheld devices. Hasan et al. [14] found that using off-screen pointing does outperform on-screen navigation at the expense of accuracy. Jones et al. [19] further used the off-screen space of a mobile device to allow for a greater interaction volume, as well as removing the apparent screen occlusion. Takashima et al. [33] explored how users can start dragging operations on-screen and continue them outside the device after crossing its boundary. Despite being designed for small-screen devices, these techniques were inspiring for our work on *Off-Limits*. Due to the different display sizes (and thus motor spaces), we do not expect that off-screen interaction performance on mobile devices can be generalized directly to large displays. Hence, directly porting techniques from small devices to large displays is not straightforward, as the different devices afford substantially different motions when interacting with them.

USER STUDIES

We contribute three user studies, each of which investigates the users' perception and use of off-screen space. Our goal is to successfully bring off-screen interaction to large displays. To do so, we first evaluate the performance of naïvely applying the concept of off-screen pointing. Then, we investigate how the information space extends beyond the display. Given the display as the frame of reference, we explore whether users can extrapolate locations in the information space in a linear fashion beyond the display's boundaries, or whether the information is scaled based on how far users point into off-screen space. Finally, we create a model based on the findings, improve the concept, and evaluate it against the naïve approach. We chose to not give on-screen visual guidance regarding the current pointing location, although such guidance may increase users' pointing accuracy. Instead, our focus is to assess users' perception of off-screen space with the display being the only frame of reference. We leave an exploration of the effects of visual guidance for future work.

In all experiments, we used an OptiTrack motion capturing system with eight Flex 13 cameras (with 56° field of view). Each camera runs at 120 fps with a resolution of 1280 × 1024 pixels. Thus, we are able to track a volume of 3 × 3 meters at sub-millimeter accuracy. To track pointing interactions, participants wore a glove on their right hand with reflective markers attached to it as follows: one rigid body to find the hand, one marker on the dorsal side of the distal phalanx and one marker on the dorsal side of the proximal phalanx (see inlay in Figure 2). We used the ray emanating through those markers onto the planar extension of the display for interaction. As display, we used an 84" SMART Board 8084i with an active display area of 1.86 × 1.04 meters (3840 × 2160 pixels). Figure 2 shows our setup.

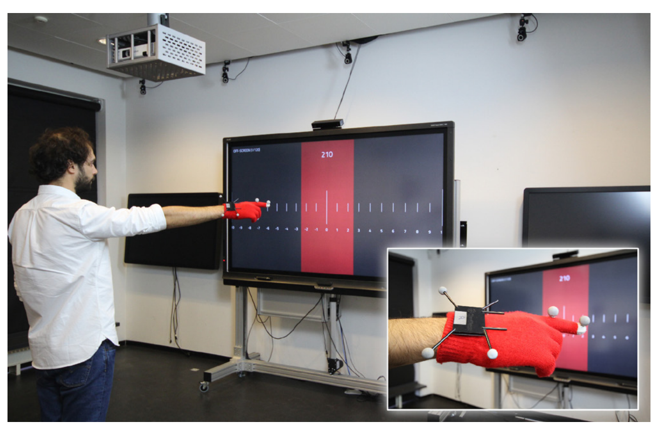

Figure 2. The experimental setup used in all three studies.

STUDY 1: OFF-SCREEN INTERACTION PERFORMANCE

The first experiment compares the naïve implementation of off-screen interaction to an on-screen mid-air condition and a touch condition. Participants performed 1D-docking operations by panning a horizontal line of numbers.

Task

Participants performed a 1D-docking task, where they had to drag a horizontal line of numbers until a given number is within the target area in the center of the display (which covered a quarter of the display's width). At the beginning of each trial, the display showed the initial range of -10 to 10 (both inclusive), with 0 in the center. At the top of the display, the number participants had to dock was shown (see Figure 2). Task values ranged from ±10 to ±210 (i.e., most numbers were initially off-screen). Participants had to drag the horizontal line, so that the desired number appeared on-screen. The correct number was indicated with a thick red line. Once the number entered the target area, the area changed its color to green. A trial ended when the target was stationary within the target area for 500ms. We recorded time from the first interaction until the docking was completed (with the 500ms stationary dwell time excluded), as well as the number of dragging operations.

Interfaces

Participants used three interfaces: two baseline conditions, Touch and On-screen mid-air pointing, and the experimental condition, Off-screen mid-air pointing. We included touch as a condition, because it is widespread even on large displays. In detail, the interfaces were:

Touch. This baseline interface is implemented to allow for dragging operations using single-touch. We added inertia (which is common in touch interfaces), and implemented it as follows: while dragging, we compute the velocity of the drag. Once a finger is lifted (end of a drag), we multiply that velocity with a factor 0.985 for every 10ms passed since the release (i.e., the start of an inertia scroll). Velocity was capped to a maximum value determined through a pilot study with four participants.

On-screen Mid-air Pointing (*On-Screen*)**.** Like *Touch*, this technique is bound to the display (i.e., an interaction can only occur while a user is pointing at the display). To determine the on-screen location users point at, we used an approach similar to Vogel and Balakrishnan [34], where an imagined ray was emanating from the tip of the extended index finger. As our task was a 1D-docking task, however, we only used the *x*-coordinate of the intersection to give feedback to participants regarding the pointing location.

Unlike other mid-air techniques (e.g., [5,34]), we chose to have participants click a mouse in their non-dominant hand. Note that this suggests bi-manual operation where one hand points at an off-screen location while the other hand is used for selecting that location. We did so to ensure robust operation, and to minimize the effects of clicking on pointing performance. The mouse's left button acted as selection trigger (dragging was enabled while the button was pressed). Once participants pointed off-screen, an automatic *release* event was triggered to tie the interaction to the display. If the button remained pressed and the cursor reentered the display, we trigged a *press* event. This technique used the same inertia implementation as *Touch*.

Off-screen Mid-air Pointing (*Off-Screen*)**.** The off-screen pointing technique was implemented in the same way as its on-screen counterpart. The important difference, however, was that there were no automatic *press* and *release* events when the cursor left the display. Thus, participants could freely make use of off-screen space. That is, they could point at locations not residing on-screen, and perform dragging operations beyond the display's boundaries (i.e., either start or end in off-screen space). This technique allowed for the same inertia behavior as the other two.

Study Design & Procedure

The experiment used a within-subjects design with *Interface* (*Touch*, *On-Screen*, and *Off-Screen*) and *Value* (*±10*, *±50*, *±90*, *±130*, *±170*, and *±210*) as independent variables. The values represent equally distributed numbers up to ten times the display width to either side of the display. The order of *Interfaces* was systematically varied between participants using a balanced Latin Square. The order of *Value* was randomized. In total, we collected *12 Participants × 3 Interfaces × 12 Values × 5 Repetitions = 2160 Trials* (180 data points per participant).

First, participants were asked to complete a demographic questionnaire, and were introduced to the experiment. Before performing tasks with an *Interface*, participants received a short introduction and then practiced until they felt comfortable using the interface. All participants used less than 10 practice trials for each *Interface*. For *Touch*, participants were standing in front of the center of the display; for the two mid-air conditions, participants stood 2 meters away from the display. Once they completed all trials with an *Interface*, they were administered a device assessment questionnaire (ISO-9241-9). At the end of the study, participants ranked the three *Interfaces* (1 was best). Participants completed the study on average in 60 minutes.

Participants

We recruited 12 paid participants (7 female, 5 male) ranging in age from 18 to 36 (M=25.3, SD=5.0). All were right-handed or ambidextrous. Furthermore, participants reported normal mobility in both arms as well as hands.

Hypotheses

We hypothesized that *Off-Screen* would outperform the other two techniques. In particular, we (H1) expected *Off-Screen* to have shorter interaction times for targets further in off-screen space, due to (H2) fewer dragging operations.

Results

We performed separate Repeated Measures Analysis Of Variance (RM-ANOVA) tests on task completion time and number of operations. Outliers calculated for each interac-

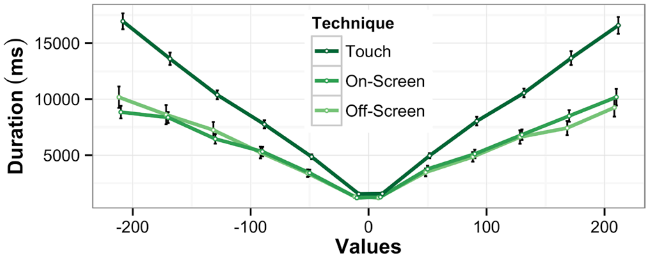

Figure 3. Task durations across values. Error bars denote 95% confidence intervals.

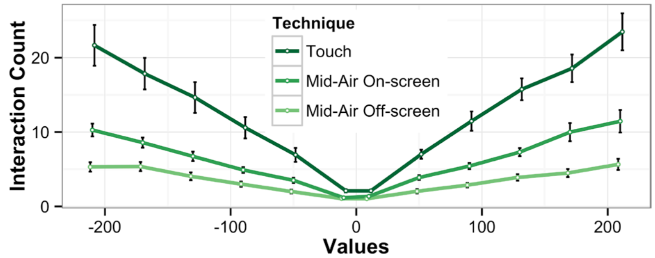

Figure 4. Interaction count (drags) across values. Error bars denote 95% confidence intervals.

tion technique and value combination were removed. A trial was regarded as an outlier if task duration was more than three interquartile ranges above the upper quartile (Q3) or below the lower quartile (Q1). In total, we removed 32 trials (*Touch*: 4, *On-Screen*: 3, and *Off-Screen*: 25). In cases where sphericity was violated, we corrected the degrees of freedom using Greenhouse-Geisser correction. For pairwise post-hoc comparisons, we used Bonferroni correction. Unstated *p*-values were non-significant ($p > 0.05$).

Completion time. We found significant main effects for *Interface* ($F_{2,22} = 60.663$, $p < .001$) and *Value* ($F_{3.352,36.869} = 501.713$, $p < .001$). Pairwise comparisons revealed that *Touch* was slower than both *On-Screen* and *Off-Screen*. The effect of *Value* was unsurprising, as larger values naturally require longer time to reach them. Pairs of *values* of the same absolute value (e.g., +10 versus -10) did not differ significantly, but all others did (all $p < .05$).

We also found an *Interface* × *Value* interaction ($F_{22,242} = 28.111$, $p < .001$). As shown in Figure 3, *Touch* was consistently slower than the other two *Techniques* for all *Values* but *-10* (all $p < .023$). For *-10*, *Touch* was only slower than *On-Screen* ($p < .007$), but no difference was found between *Touch* and *Off-Screen*. Across all *Values*, there was no significant difference between the two mid-air conditions. Overall, *Touch* (*M*=9211ms, *SD*=5391) was slower than both *On-Screen* (*M*=5774ms, *SD*=3286), and *Off-Screen* (*M*=5687ms, *SD*=3631).

Number of operations. Since no difference in performance was found between *On-Screen* and *Off-Screen*, we investigated the number of dragging operations used to see if participants used *Off-Screen* similar to *On-Screen* (i.e., not making use of off-screen space). Note that a lower number indicates larger dragging distances, and thus the use of off-screen space. We again found significant main effects for both *Interface* ($F_{1.337,14.709} = 476.461$, $p < .001$) and *Value* ($F_{3.212,35.332} = 515.353$, $p < .001$). The main effect of *Values* was unsurprising, as larger ranges require more dragging operations. Again, *Values* of the same absolute value (e.g., +10 versus -10) did not differ significantly, but all others except for the two largest *Values* did (all $p < .05$).

An *Interface* × *Value* interaction ($F_{22,242} = 160.583$, $p < .001$) was also observed. Figure 4 reveals the source of the detected interaction, where – as the absolute *Values* increase – *Off-Screen* had fewer and fewer operations than both *On-Screen* and *Touch* (all $p < .001$). Even for the smallest *Value* (±10), *Off-Screen* had the lowest number of operations (all $p < .001$). Overall, *Touch* (*M*=12.69, *SD*=9.36) required most operations, followed by *On-Screen* (*M*=6.21, *SD*=4.31) and *Off-Screen* (*M*=3.37, *SD*=2.32). Figure 4 summarizes these results.

To further support the differences in interaction strategies, we observed large differences in the percentage of viewport movement being caused by inertia per participant and *Interface*. A higher percentage indicates that participants more strongly relied on inertia. In total, *Touch* had 59.8% (*SD*=2.72), *On-Screen* 36.3% (*SD*=4.55), and *Off-Screen* 10.6% (*SD*=9.19) of movement caused by inertia. Fewer interactions paired with less inertia-based movement supports that off-screen interaction was used actively.

Subjective feedback. One participant did not answer all the questions of the ISO-9241-9 questionnaire, and we removed that participant's answers before further analysis. A MANOVA analysis using Wilks' lambda revealed significant differences between the *Interfaces* (Λ=.165, $F_{26,36} = 2.022$, $p < .05$). Post-hoc comparisons showed no significant differences between the two mid-air conditions. *Touch*, however, was regarded as requiring more physical effort, being slower, being less comfortable, and to cause higher fatigue in the finger, the arm and shoulder (all $p < .05$) than *On-Screen*. In addition, *Touch* was regarded as being slower, but as being more accurate than *Off-Screen* (all $p < .05$).

We further asked participants to rank the three *Interfaces*. We found a significant difference in ranks ($\chi^2(2) = 12.667$, $p < .002$). A post-hoc analysis using Wilcoxon Signed-Rank Tests then revealed that *On-Screen* was the preferred *Interface* (all $p < .005$), whilst *Touch* and *Off-Screen* showed no significant difference. Overall, *On-Screen* was ranked first ten times, the other two only once each.

Discussion

We hypothesized that *Off-Screen* would outperform both *Touch* and *Off-Screen* in terms of completion time by reducing the number of operations required. However, we found that both mid-air *Interfaces* exhibited similar performance regarding completion time, which means that we reject H1. We further found that *Interfaces* were used differently, which is outlined by the number of operations using during a trial. Here, *Off-Screen* required significantly fewer dragging operations than the other two techniques, which we believe stems from the larger range available for interaction. This supports H2. During the experiment, we made important observations (which were supported by participants' comments during and after the study):

First, for all *Interfaces*, participants often adjusted their speed of panning to a level that allowed reading the numbers on the display. One participant specifically commented that this was the case and that he had to go slower in *Touch* than with the mid-air interfaces to see the numbers (because he was very close to the display).

Second, for *Touch* and *On-Screen*, all participants heavily relied on inertia. For *Off-Screen*, participants fell into two groups: either they relied on a similar inertia strategy as for *Touch* and *On-Screen*, or they attempted to accurately point in off-screen space to directly drag the target on-screen. We observed that the latter group often undershot the target and continued the dragging operation across the display to end in the off-screen space opposite to where they had started. This resulted in increased effects of hand-tremor.

Third, when using *Off-Screen*, participants attempted to correct for the undershooting they experienced. This often resulted in severe overshooting when trying to point at large values due to the planar and linear extension of the plane. After experiencing this a few times, participants adopted a similar usage pattern as observed for *On-Screen*, i.e., mostly using the display instead of off-screen space.

In summary, *Off-Screen* did not outperform the two baseline conditions *Touch* and *On-Screen,* as we had anticipated, but performed on par. We also identified issues with the off-screen pointing *Interface*, namely severe undershooting and overshooting effects that encouraged further work.

STUDY 2: UNDERSTANDING OFF-SCREEN SPACE

The aforementioned challenges may stem from the users' perception of off-screen space and their ability to point accurately within that space. In the first experiment, it became clear that users' perception of off-screen space is not necessarily a linear extension of the display. Instead, there may be a different function that relies on a target's distance to the display. In this experiment, we explore the users' perception of locations in off-screen space.

To assess participants' perception of off-screen pointing, we adopted a method from *Psychophysics* and applied the so-called *Magnitude Production* using a predefined scale [11]. Despite its design for physical stimuli (e.g., force, or sound pressure), we found the experimental setup to be a good fit for this experiment. We chose a predefined scale, because we foresee off-screen use cases that are based on scales defined by on-screen content, rather than scales defined by a user's mental model.

Interface

As we were interested in assessing off-screen pointing accuracy, we only used one interface – namely *Off-Screen*. Compared to the first experiment, the visual representation was simplified: a horizontal line was the basis of the scale. The number 0 was in the center, and ±10 were located at the edges of the display. No additional numbers were on the scale. Note, that ±10 here was not exactly at the edge (as in experiment 1), but slightly moved inward, so that the numbers were better readable.

Task

At the beginning of each trial, participants were presented with a number (visually on-screen through a textbox at the bottom of the display), which they had to acquire using *Off-Screen* pointing. That is, they had to point at the location where they thought the number should be. While pointing at the number's location, they clicked the mouse button. No visual feedback was given to indicate their accuracy.

We instructed participants as follows: "(1) Imagine that the straight line shown on the display extends horizontally to both sides (into off-screen space) without any defined limits. (2) A linear scale ranging from -10 to 10 is defined on the display. 0 is at the center of the display. (3) You will be given a set of numbers. For each number, we ask you to point to the number on the extended line. (4) Please be aware that both negative and positive number will occur". For each trial, we measured where participants actually pointed at in off-screen space.

Experimental Design & Setup

The first experiment revealed that pointing accuracy in off-screen space seemingly degrades with an increased angular distance (where an angle of 0° is equal to pointing directly at the display's center). Unlike in experiment 1, we chose to distribute *Values* based on an equal spread across *pointing angles* to the display. As the largest numbers in the first experiment (i.e., ±210) correspond to pointing angles of ±84.9°, we chose to use ±84° as largest values. In between, we chose six degree intervals (rounded to the closest integer), leading to the following 29 *Values*: 0, ±2, ±5, ±7, ±10, ±12, ±16, ±20, ±24, ±30, ±38, ±49, ±67, ±102, and ±207. *Values* were presented in random order. Note that nine *Values* reside on-screen (with only 0, and ±10 being indicated on the horizontal line as ticks). In total, we recorded *29 Values × 12 Repetitions = 384 trials* per participant. *Values* were presented in random order.

Environmental features (e.g., room size, display size, and objects) might affect users' ability to point accurately. We attempted to minimize such effects by suspending black fabric from the ceiling, covering the walls and any objects that could be used as landmarks.

Procedure

Participants were first asked to complete a demographic questionnaire, and they were introduced to the experiment. Before beginning the pointing tasks, participants completed a VZ-2 paper-folding test for spatial visualization [30]. The test provides an assessment of the participants' ability to imagine/visualize spatial relationships. Results range between 0 and 20 (higher is better) based on 20 imagined paper-folding/whole-punching tasks. During the pointing task,

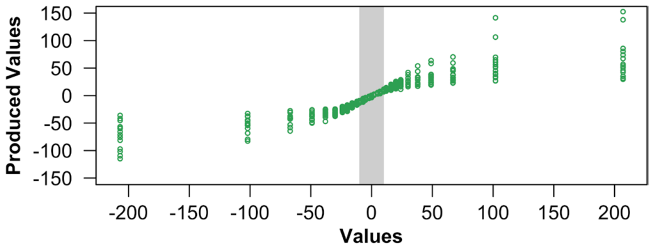

Figure 5. Scatter plot of participants' mean pointing performance. The grey area indicates the display area.

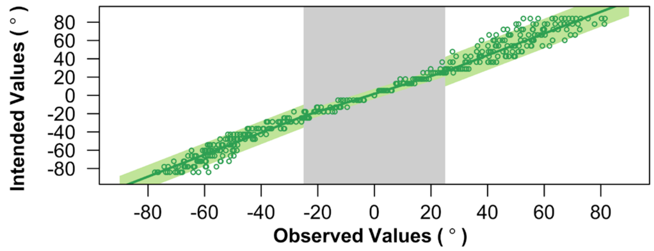

Figure 6. Scatter plot and regressions of our model. Grey indicates the display; green areas show 95% prediction intervals.

participants stood 2 meters away from the center of the display (as in experiment 1). When needed, users were allowed to take short breaks. Participants completed the experiment on average in 30 minutes.

Participants

We recruited 15 paid participants (6 female, 9 male) ranging in age from 21 to 64 (M=36.1, SD=11.2). All were right handed and all reported normal mobility in both arms and hands. Participants scored between 8 and 20 (M=13.1) on the VZ-2 paper-folding test.

Results

Before beginning the analysis, we removed 149 trials, where participants misread the number's sign and hence pointed in the wrong direction; and 107 outliers (i.e., more than three interquartile ranges above the upper quartile or below the lower quartile). In total, 265 trials were removed.

Next, we computed the *Value* error, which is the absolute difference between *Value* (the cued value) and the *Produced Value* (the value participants actually pointed at). For each participant, we compared this error to the VZ-2 test scores using linear regression. The associated analysis of variance did not show any significant predictive power of a participant's test score on their pointing accuracy ($F_{1,13}$ = .087, p = .38), which suggests that individual spatial visualization abilities cannot predict errors in *Value*.

To minimize the effects of within user variance, we calculated each participant's average per *Value*. As expected, participants' accuracy degraded rapidly as *Values* (and thus angles) increased (see Figure 5). A closer examination of outliers supports this: the four largest *Values* (i.e., ±102 and ±207) accounted for 71% of all outliers. As we suspected, Figure 5 further reveals a non-linear correlation between a trial's *Value* and the *Produced Value*. Participants consistently tended to undershoot the *Value*, which supports our observation from study 1.

Modeling Off-Screen Space

The presented data can be used to improve the performance of off-screen pointing in experiment 1. To do so, we created a model that predicts a user's *intended value* (*iv*) based on an *observed value* (*ov*). The intended value is the location the user *intends* to point at, whereas the observed value is the location the user *actually* points at (in the system's coordinate system, based on the display's pixel grid).

We experimented with different regression models (i.e., *linear*, *logarithmic*, and *exponential*) on values in pixel space, but were unable to find a model that explained all data well enough. For that reason, we transformed the values from pixel-space into the corresponding angular values, and found that linear regression worked more promising with angles. However, because accuracy seems to differ between on- and off-screen pointing, we decided to divide the data into on- and off-screen values: *Negative Off-Screen* (i.e., left of the display), *On-Screen* (i.e., within the display), and *Positive Off-Screen* (i.e., right of the display). *On-Screen* data was modeled well using linear regression ($iv = 0.974 \times ov + 1.656, R^2 = .971$).

We then fitted similar models to both off-screen areas. We decided to include on-screen points (i.e., the *Positive On-Screen* model would include on-screen points on the right/positive part of the display). *Negative Off-Screen* led to $iv = 1.159 \times ov + 4.262$ with $R^2 = .947$, and *Positive Off-Screen* led to $iv = 1.121 \times ov + 1.317$ with $R^2 = .909$. However, this led to unwanted, non-seamless transitions when crossing the display's edges (i.e., discontinuous predictions). To achieve a continuous connection between the models, we adjusted the model slightly to allow for seamless transitions (by forcing the off-screen linear regressions to go through the endpoints of the on-screen regression): *Negative Off-Screen* was then $iv = 1.202 \times ov + 7.356$ with $R^2 = .934$, and *Positive Off-Screen* changed to $iv = 1.188 \times ov - 3.694$ with $R^2 = .896$. Equation 1 shows the final combination of the three models (note that predictions larger that or equal to ±90 degrees is not applicable).

$$iv = \begin{cases} 1.202 \times ov + 7.356 & \text{if } -80.995 < ov < -25 \\ 0.974 \times ov + 1.656 & \text{if } -25 \leq ov \leq 25 \\ 1.188 \times ov - 3.694 & \text{if } 25 < ov < 78.867 \\ N/A & \text{if } ov \leq -80.995 \vee ov \geq 78.867 \end{cases} \quad (1)$$

Figure 6 shows a scatter plot of the pointing data together with the final model (the shaded area represents the 95% prediction intervals). Our model gives an estimate of the user's intended pointing position based on an observation, allowing for correcting previously observed undershooting effects. The prediction interval reveals in which interval a user intended to point in case of a new observation, with 95% confidence. Predicting a user's intended position and identifying the limits of off-screen pointing may be useful for future techniques, such as zooming automatically based on the uncertainty of an observation.

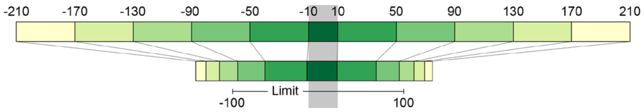

Figure 7. Relationship between *values* in original input space (top) and corrected input space (bottom).

OFF-LIMITS: MAKING OFF-SCREEN POINTING WORK

Based on the results and observations of the two experiments, we refined the naïve implementation and created the *Off-Limits* technique, which improves off-screen pointing in two important aspects: (1) we correct for the intended value using our defined model; and (2) we limit the interaction when users point too far away from the display. Figure 7 outlines the model and the limits used.

Model-based Correction. We found that participants systematically undershot their intended target. The model defined in the second experiment corrects for this – particularly in off-screen space. To ensure seamless transitions across the display's edges, we make use of the adjusted model explained in the previous section (see Equation 1).

Constrained Interaction Space. Despite the original intention of allowing for limitless off-screen interaction, results from the first two experiments suggest that the uncertainty associated with off-screen pointing quickly reaches levels where this becomes a hindering factor. More precisely, for locations far away from the display's center, slight angular changes result in large positional changes. The results of the second experiment suggested a maximum meaningful limit of 5 times the display width from the display's center. The limit is enforced on the *model-corrected* input.

STUDY 3: OFF-LIMITS' PERFORMANCE

In the first experiment, off-screen pointing showed competitive, yet not superior, performance to touch and mid-air on-screen pointing. In this experiment, we evaluate *Off-Limits* and compare it to off-screen pointing.

Interfaces

We compared two interfaces: (1) *Off-Screen* was implemented as in the first experiment; and (2) *Off-Limits* was implemented using the model and constraints as described in the previous section. We did not include the *Touch* interface, which was shown inferior in experiment 1, and neither included *On-Screen*, as it was comparable to *Off-Screen*.

Task, Study Design & Procedure

The task was modeled in the same way as in the first experiment. That is, participants performed a 1D-docking task using each of the two interfaces. We recorded the same data, namely time (from the first interaction until the docking was completed) and the number of dragging operations.

Like the first experiment, this study used a within-subjects design with the two independent variables *Interface* and *Value*. To allow for informal comparisons to the original experiment, we kept the *Values* identical (i.e., ranging from -210 to +210). Note, that the enforced limit for *Off-Limits* therefore is ±100. In contrast to the first study, we did increase the number of repetitions from 5 to 10 per *Interface* and *Value* combination. In total, each participant completed *2 Techniques × 12 Values × 10 Repetitions = 3120 trials.*

The procedure was also adapted from the first experiment. On average, participants spent approximately 60 minutes.

Participants

We recruited 13 paid participants (9 female, 4 male) ranging in age from 20 to 33 years (M=23.8, SD=3.4). Two participants had previously participated in experiment 1. All of our participants were right-handed and reported normal mobility in both arms as well as hands.

Hypotheses

We hypothesized, that (H1) *Off-Limits* is faster than *Off-Screen* with increasing *Values* (i.e., larger distances to the display's center), and (H2) requires fewer operations.

Results

We performed RM-ANOVA tests on task completion time and number of operations. We removed outliers before beginning our analysis (as described for experiment 1). We removed 87 outliers (*Off-Screen*: 59; *Off-Limits*: 28). We corrected degrees of freedom using Greenhouse-Geisser when sphericity was violated. For pair-wise post-hoc comparisons, we used Bonferroni correction.

Completion time. We found significant main effects for *Interface* ($F_{1,12}$ = 7.093, p < .021) and *Value* ($F_{2.666,31.997}$ = 104.643, p < .001) and an *Interface* × *Value* interaction ($F_{3.621,43.448}$ = 6.973, p < .001). As for the first experiment, the *Value* effect is explained by requiring more time to reach targets further away. The results are in line with the previous studies, in that neither values of the same absolute value nor values with an absolute value of 170 or larger differed significantly.

Figure 8 reveals the source of the interesting interaction. While *Off-Screen* seems to perform equally well compared to *Off-Limits* when *Values* are on-screen or only slightly beyond the display, *Off-Limits* outperforms *Off-Screen* for *Values* further away from the display. In particular, we found that *Off-Limits* was faster than *Off-Screen* for the following *Values*: +130, +210, -170, and -210 (all p < .015). Although there are no significant differences between the two *Interfaces* for -130 and +170, Figure 8 indicates a trend in that *Off-Limits* may work better in the off-screen space further away. On the other hand, for *Values* -10 (p < .007)

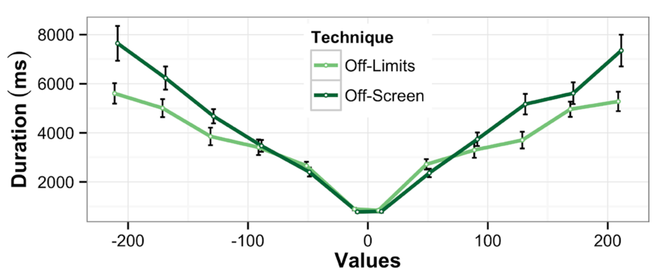

Figure 8. Task durations across values. Error bars denote 95% confidence intervals.

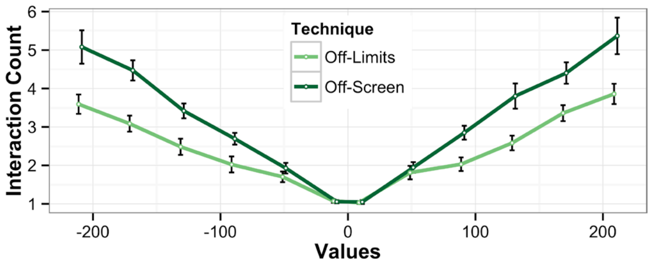

Figure 9. Interaction count (drags) across values. Error bars denote 95% confidence intervals.

and +50 ($p < .049$), *Off-Screen* was slightly faster, suggesting that the off-screen correction may have had a slight influence. Overall, however, *Off-Limits* (*M*=3526.81, *SD*=216.94) was faster than the original *Off-Screen* interface (*M*=4209.7, *SD*=285.42). These results support H1.

Number of operations. We found significant main effects for *Interface* ($F_{1,12}$ = 23.238, p < .001) and *Value* ($F_{3.925,47.095}$ = 142.108, p < .001) and an *Interface* × *Value* interaction ($F_{3.769,45.234}$ = 8.277, p < .001). As for completion time, *Values* further away require more operations. Like in the other experiments, post-hoc pair-wise multiple means comparisons revealed that positive/negative *Values* of the same absolute value did not differ, and neither did *Values* from ±170 onward (all p < .05).

The source of the *Interface* × *Value* interaction is shown in Figure 9. Both *Interfaces* did not differ significantly for the lowest two *Values* (i.e., ±10 and ±50). For all other *Values*, *Off-Limits* required fewer operations than pure *Off-Screen* pointing (all p < .013). This is in line with completion times and further supports that *Off-Limits* works well for *Values* further away from the display's center. This supports our hypothesis H2. Overall, *Off-Limits* (*M*=2.384, *SD*=0.095) required fewer operations than *Off-Screen* (*M*=3.165, *SD*=0.145). Figure 9 summarizes the results.

Subjective feedback. As in study 1, we ran a MANOVA using Wilks' lambda on the ISO-9241-9 questionnaire answers. Overall, the MANOVA did not find the *Interfaces* to be different (Λ=.354, $F_{12,13}$ = 1.688, p > .05). Post-hoc comparisons of the individual questions, however, showed significant differences on the accuracy of pointing and the overall ease of use (both p < .05). *Off-Limits* was perceived more accurate and easier to use. All 13 participants preferred the use of *Off-Limits*. One stated that "*I feel that I hit what I'm aiming at more accurately, using this technique*", another that "*I don't need to slow down to see the numbers because I know what I'm holding onto*". Others had similar comments. Only one stated that she "*had to slow down to see the numbers in both techniques*".

All participants preferred *Off-Limits*. Interestingly, one participant who was part of the first experiment commented that *Off-Limits* was "***by far** the best technique I have tried*". These results and statements indicate that our improvements to the naïve off-screen pointing implementation had the desired effect.

DISCUSSION

We set out to explore off-screen pointing as an interaction technique for navigating large information spaces on large displays. Our main hypothesis was that off-screen pointing provides significant advantages over state-of-the-art interaction techniques (e.g., touch and mid-air), whose input is bound to the display.

The first experiment found *Touch* to perform slower than the mid-air techniques. The study did not show a performance increase when using off-screen space, as participants consistently undershot the target when they tried to directly point at it. Undershooting greatly decreased performance, and brought it overall closer to the mid-air on-screen technique. One reason for this is that users do not perceive off-screen space as linearly extending the on-screen space.

The second study, which collected data to understand and model users' behavior, showed that users' pointing accuracy degraded rapidly when pointing further away from the display. We also observed that participants systematically undershot the target. We derived a model that predicts the value a user intends to point at, which can be used to correct for undershooting. The model limits output to only the range in off-screen space where users can accurately point (i.e., to avoid angles close to 90 degrees that are problematic). The results led to the final design of *Off-Limits*.

The third experiment compared *Off-Limits* to the naïve *Off-Screen* implementation of the first experiment. The results show significant improvements: *Off-Limits* is significantly faster, requires fewer (clutching) operations, and is preferred by participants. Figure 10 highlights that participants used off-screen space (and particularly the limit of ±100) more extensively with *Off-Limits* compared to the naïve approach (where participants pointed closer to the display).

Angular vs. Orthogonal Pointing

Our results show that naïvely extending the plane linearly did not lower task completion time significantly, when working on large displays. However, earlier work on mobile devices successfully used this style of off-screen pointing [13,14]. We believe that this stems from the way people point: on mobile devices, people would point orthogonally into the extended space with off-screen locations reasonably close to the display. On large displays, pointing orthogonally in off-screen space would require people to move parallel to the display. Instead, people point in an angular fashion by turning their forearm. During the process of deriving our model, we found that angular pointing was used.

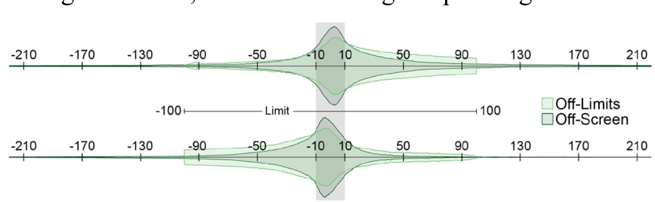

Figure 10. Aggregation of drag movement across positive (top) and negative (bottom) values. The shaded areas show the proportion of interactions in different parts of the motor space.

Figure 11. A user interacting with the Kinect prototype: (a) a user looks at a view of North America on a display; (b) the user points the location of Europe in off-screen space; (c) the user drags Europe onto the screen from its off-screen location.

A second, yet unconfirmed, reason might be that users have a different perception of off-screen space. Although we had instructed them to imagine a *planar* extension of the display, it is not clear that they actually did. Also, imagining positions further off-screen may be linked to the motor space. That is, when we point using our forearm (or potentially the entire arm), the extension may become *spherical* instead of *planar*. While our angular model reflects this, there is potential for future work.

Limitations of the Results

We acknowledge that there are limitations to the generalizability of our results. First, most information spaces are two-dimensional, and not one-dimensional as we used in our experiments. It is not yet clear, whether the model would hold for 2D off-screen interaction. However, our approach to defining a model may inform future work on deriving similar models for 2D spaces.

Second, participants were standing at a fixed distance (2m) from the fixed-size (84"). However, distance and size may influence off-screen pointing: there is a possibility that our model might need to be adapted to other distances and displays sizes. We believe, however, that this affects the cutoff of *Off-Limits*, whereas the angular model itself should translate well to other distances and displays. Tracking the user's position continuously can update the model dynamically. However, we leave the exploration for future work.

Finally, our results are affected by the chosen ray-casting and selection mechanism (as previously found by Mayer et al. [24]). Yet, we believe that the dominant factor is the uncertainty in users' perception of off-screen space. Thus, we expect similar results with other ray-casting and selection mechanisms. Nonetheless, this is left for future work.

Possible Applications using Off-Limits

To further illustrate potential use cases for *Off-Limits*, we developed an example application that allowed for visualizing high-resolution tiled images (we chose: maps, satellite images, high resolution deep space image, and electronic microscope samples). Our application makes use of three interaction techniques, each of which takes advantage of off-screen pointing: (1) *Off-Screen Panning*, which is similar to our experiments; (2) *Off-Screen Jumping* allows for jumping to a selected off-screen location using a zooming transition; and (3) *Uncertainty Feedback*, which shows an on-screen popup when a user interacts off-screen. The popup provides a real-time view into off-screen space, with automatic zooming based on pointing uncertainty.

Moreover, we implemented *Off-Limits* using an off-the-shelf Kinect v2 depth sensor mounted on top of the display to control the application, to demonstrate that *Off-Limits* is useful with less sophisticated tracking systems than the one used in the studies (see Figure 11). The Kinect was calibrated to the OptiTrack's coordinate system by capturing a person's wrist with both tracking systems while that person moved around. Naturally, the tracking accuracy is lower than using OptiTrack, yet the Kinect did provide sufficient accuracy to allow for effective off-screen pointing. The accompanying video figure illustrates the interaction techniques using OptiTrack and Kinect in operation.

CONCLUSIONS & FUTURE WORK

In this paper, we presented our findings from an exploration of off-screen pointing as an interaction technique for large displays. In three studies, we found that off-screen pointing provides significant benefits over state-of-the-art interaction techniques (*Touch* and *Mid-Air On-Screen* pointing). We systematically explored users' ability to point to off-screen locations in a horizontal extension of the on-screen information space. The results help better understand off-screen pointing, and have informed the design of *Off-Limits*, a novel off-screen interaction technique for large displays.

Our focus has been on quantitatively assessing users' abilities and to quantify performance gains of *Off-Limits*. We also gathered some positive subjective feedback on the use of *Off-Limits*, but future work could further investigate user's experience with *Off-Limit*. An obvious next step is to apply our work to two-dimensional information spaces; the applications we presented here that use *Off-Limits* already hint at the potential of off-screen pointing in 2D. A third direction for future work is to explore the effect of a person's distance to the display, as well as varying display sizes. Overall, we conclude that off-screen pointing (on large displays) has shown potential and will receive future improvements to create even better opportunities for interacting with large information spaces.

ACKNOWLEDGMENTS

This work was supported by the European Research Council under Grant Agreement n. 648785.

REFERENCES

1. Robert Ball, Chris North, and Doug A. Bowman. 2007. Move to Improve: Promoting Physical Navigation to Increase User Performance with Large Displays. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 191–200. http://doi.org/10.1145/1240624.1240656
2. Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. 2012. MultiPoint: Comparing laser and manual pointing as remote input in large display interactions. *International Journal of Human-Computer Studies* 70, 10: 690–702. http://doi.org/10.1016/j.ijhcs.2012.05.009
3. Patrick Baudisch and Ruth Rosenholtz. 2003. Halo: A Technique for Visualizing Off-screen Objects. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 481–488. http://doi.org/10.1145/642611.642695
4. Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. 2010. Touch Projector: Mobile Interaction Through Video. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2287–2296. http://doi.org/10.1145/1753326.1753671
5. Florian van de Camp, Alexander Schick, and Rainer Stiefelhagen. 2013. How to Click in Mid-Air. In *Distributed, Ambient, and Pervasive Interactions*, Norbert Streitz and Constantine Stephanidis (eds.). Springer Berlin Heidelberg, 78–86. http://link.springer.com/chapter/10.1007/978-3-642-39351-8_9
6. Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. 2009. A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Comput. Surv.* 41, 1: 2:1–2:31. http://doi.org/10.1145/1456650.1456652
7. 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: 401–414. http://doi.org/10.1016/j.ijhcs.2011.02.005
8. Maxime Collomb, Mountaz Hascoët, Patrick Baudisch, and Brian Lee. 2005. Improving Drag-and-drop on Wall-size Displays. *Proceedings of Graphics Interface 2005*, Canadian Human-Computer Communications Society, 25–32. http://dl.acm.org/citation.cfm?id=1089508.1089514
9. Barrett Ens, David Ahlström, Andy Cockburn, and Pourang Irani. 2011. Characterizing User Performance with Assisted Direct Off-screen Pointing. *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, ACM, 485–494. http://doi.org/10.1145/2037373.2037445
10. Clifton Forlines, Daniel Vogel, and Ravin Balakrishnan. 2006. HybridPointing: Fluid Switching Between Absolute and Relative Pointing with a Direct Input Device. *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, ACM, 211–220. http://doi.org/10.1145/1166253.1166286
11. George A. Gescheider. *Psychophysics: The Fundamentals*. Lawrence Erlbaum Associates, Publishers.
12. Sean Gustafson, Patrick Baudisch, Carl Gutwin, and Pourang Irani. 2008. Wedge: Clutter-free Visualization of Off-screen Locations. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 787–796. http://doi.org/10.1145/1357054.1357179
13. Khalad Hasan, David Ahlström, and Pourang Irani. 2013. Ad-binning: Leveraging Around Device Space for Storing, Browsing and Retrieving Mobile Device Content. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 899–908. http://doi.org/10.1145/2470654.2466115
14. Khalad Hasan, David Ahlström, and Pourang P. Irani. 2015. Comparing Direct Off-Screen Pointing, Peephole, and Flick & Pinch Interaction for Map Navigation. *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, ACM, 99–102. http://doi.org/10.1145/2788940.2788957
15. Pourang Irani, Carl Gutwin, Grant Partridge, and Mahtab Nezhadasl. 2007. Techniques for Interacting with Off-Screen Content. In *Human-Computer Interaction – INTERACT 2007*, Cécilia Baranauskas, Philippe Palanque, Julio Abascal and Simone Diniz Junqueira Barbosa (eds.). Springer Berlin Heidelberg, 234–249. http://link.springer.com/chapter/10.1007/978-3-540-74800-7_19
16. Pourang Irani, Carl Gutwin, and Xing Dong Yang. 2006. Improving Selection of Off-screen Targets with Hopping. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 299–308. http://doi.org/10.1145/1124772.1124818
17. Mikkel R. Jakobsen and Kasper Hornb\a ek. 2015. Is Moving Improving?: Some Effects of Locomotion in Wall-Display Interaction. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 4169–4178. http://doi.org/10.1145/2702123.2702312
18. Yvonne Jansen, Pierre Dragicevic, and Jean-Daniel Fekete. 2012. Tangible Remote Controllers for Wall-size Displays. *Proceedings of the SIGCHI Conference*

on Human Factors in Computing Systems, ACM, 2865–2874. http://doi.org/10.1145/2207676.2208691

19. Brett Jones, Rajinder Sodhi, David Forsyth, Brian Bailey, and Giuliano Maciocci. 2012. Around Device Interaction for Multiscale Navigation. *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, ACM, 83–92. http://doi.org/10.1145/2371574.2371589
20. Ricardo Jota, Miguel A. Nacenta, Joaquim A. Jorge, Sheelagh Carpendale, and Saul Greenberg. 2010. A comparison of ray pointing techniques for very large displays. *Proceedings of Graphics Interface 2010*, Canadian Information Processing Society, 269–276. http://dl.acm.org/citation.cfm?id=1839214.1839261
21. Azam Khan, George Fitzmaurice, Don Almeida, Nicolas Burtnyk, and Gordon Kurtenbach. 2004. A Remote Control Interface for Large Displays. *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, ACM, 127–136. http://doi.org/10.1145/1029632.1029655
22. Regis Kopper, Doug A. Bowman, Mara G. Silva, and Ryan P. McMahan. 2010. A Human Motor Behavior Model for Distal Pointing Tasks. *Int. J. Hum.-Comput. Stud.* 68, 10: 603–615. http://doi.org/10.1016/j.ijhcs.2010.05.001
23. Frank Chun Yat Li, David Dearman, and Khai N. Truong. 2009. Virtual Shelves: Interactions with Orientation Aware Devices. *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, ACM, 125–128. http://doi.org/10.1145/1622176.1622200
24. Sven Mayer, Katrin Wolf, Stefan Schneegass, and Niels Henze. 2015. Modeling Distant Pointing for Compensating Systematic Displacements. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 4165–4168. http://doi.org/10.1145/2702123.2702332
25. David C. McCallum and Pourang Irani. 2009. ARC-Pad: Absolute+Relative Cursor Positioning for Large Displays with a Mobile Touchscreen. *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, ACM, 153–156. http://doi.org/10.1145/1622176.1622205
26. Miguel A. Nacenta, Dzmitry Aliakseyeu, Sriram Subramanian, and Carl Gutwin. 2005. A Comparison of Techniques for Multi-display Reaching. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 371–380. http://doi.org/10.1145/1054972.1055024
27. Mathieu Nancel, Olivier Chapuis, Emmanuel Pietriga, Xing-Dong Yang, Pourang P. Irani, and Michel Beaudouin-Lafon. 2013. High-precision Pointing on Large Wall Displays Using Small Handheld Devices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 831–840. http://doi.org/10.1145/2470654.2470773
28. Mathieu Nancel, Emmanuel Pietriga, Olivier Chapuis, and Michel Beaudouin-Lafon. 2015. Mid-Air Pointing on Ultra-Walls. *ACM Trans. Comput.-Hum. Interact.* 22, 5: 21:1–21:62. http://doi.org/10.1145/2766448
29. Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. 2011. Mid-air Pan-and-zoom on Wall-sized Displays. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 177–186. http://doi.org/10.1145/1978942.1978969
30. J. W. French R. B. Ekstrom. 1976. Kit of Factor-Referenced Cognitive Tests.
31. George Robertson, Mary Czerwinski, Patrick Baudisch, et al. 2005. The Large-Display User Experience. *IEEE Comput. Graph. Appl.* 25, 4: 44–51. http://doi.org/10.1109/MCG.2005.88
32. Richard Stoakley, Matthew J. Conway, and Randy Pausch. 1995. Virtual Reality on a WIM: Interactive Worlds in Miniature. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Press/Addison-Wesley Publishing Co., 265–272. http://doi.org/10.1145/223904.223938
33. Kazuki Takashima, Nana Shinshi, and Yoshifumi Kitamura. 2015. Exploring Boundless Scroll by Extending Motor Space. *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ACM, 557–566. http://doi.org/10.1145/2785830.2785884
34. Daniel Vogel and Ravin Balakrishnan. 2005. Distant freehand pointing and clicking on very large, high resolution displays. *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ACM, 33–42. http://doi.org/10.1145/1095034.1095041
35. Ka-Ping Yee. 2003. Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1–8. http://doi.org/10.1145/642611.642613
36. Polle T. Zellweger, Jock D. Mackinlay, Lance Good, Mark Stefik, and Patrick Baudisch. 2003. City Lights: Contextual Views in Minimal Space. *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, ACM, 838–839. http://doi.org/10.1145/765891.766022