Combining Direct and Indirect Touch Input for Interactive Desktop Workspaces using Gaze Input

Simon Voelker[1], Andrii Matviienko[1], Johannes Schöning[2], Jan Borchers[1]

[1] RWTH Aachen University
Aachen, Germany
{voelker, matviienko, borchers}@cs.rwth-aachen.de

[2] Expertise Centre for Digital Media
Hasselt University – tUL – iMinds, Diepenbeek, Belgium
johannes.schoening@uhasselt.be

ABSTRACT

Interactive workspaces combine horizontal and vertical touch surfaces into a single digital workspace. During an exploration of these systems, it was shown that direct interaction on the vertical surface is cumbersome and more inaccurate than on the horizontal one. To overcome these problems, indirect touch systems turn the horizontal touch surface into an input devices that allows manipulation of objects on the vertical display. If the horizontal touch surface also acts as a display, however, it becomes necessary to distinguish which screen is currently in use by providing a switching mode. We investigate the use of gaze tracking to perform these mode switches. In three user studies we compare absolute and relative gaze augmented selection techniques with the traditional direct-touch approach. Our results show that our relative gaze augmented selection technique outperforms the other techniques for simple tapping tasks alternating between horizontal and vertical surfaces, and for dragging on the vertical surface. However, when tasks involve dragging across surfaces, the findings are more complex. We provide a detailed description of the proposed interaction techniques, a statistical analysis of these interaction techniques, and how they can be applied to systems that involve a combination of multiple horizontal and vertical touch surfaces.

Keywords

Interactive Surfaces and Tabletops; Workspaces; Touch; Indirect Touch; Gaze-based Interaction; Tabletop interaction

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces Input Devices and Strategies

1. INTRODUCTION & MOTIVATION

Direct interaction with vertical displays is as old as the history of graphical user interfaces. In 1963, Sutherland's Sketchpad [21] already used a light pen to draw on the system's vertical screen. However, this style of interaction is inherently difficult for the human physique, since holding one's arm in mid-air while interacting with a vertical display leads to fatigue, a phenomenon known as the *Gorilla Arm effect* [18, 27]. Desktop user interfaces quickly moved input to the horizontal surface in front of the screen, leading to indirect graphical input devices. This allows users to rest their arms on the surface while interacting. Mouse and trackpad are relative, graphic tablets are absolute indirect input devices.

Figure 1: With help of the users gaze one can interact on the horizontal surface using direct touch and on the vertical surface via indirect touch.

Projects such as Tognazzini's Starfire [22] envisioned future desktop workplaces by including both vertical and horizontal touch surfaces. In the last few years we have seen a rise of interactive surfaces and tabletops [14]. Multi-touch tabletop research projects such as Curve [28], BendDesk [27], and Magic desk [2] have made such setups increasingly feasible, but they also showed that direct interaction with a vertical surface is difficult and less accurate than interacting with the horizontal surface [27]. The first commercial products now exist that make use of a vertical and horizontal touch surface such as Sprout (sprout.hp.com).

Indirect multi-touch allows users to interact with the vertical surface in a comfortable way [13, 26]. Most recently with the new version of iOS 9 for iPad, Apples touch-sensitive QuickType keyboard now also features a indirect touch mode, which transform the keyboard into a trackpad whenever one set down two fingers on the keyboard portion of the screen. Indirect touch also allows the users of interactive workspace, to comfortably interact with the vertical surface using multi-touch input on the horizontal surface. Instead of touching the vertical surface directly, the touch points are transferred using an absolute mapping from the horizontal to the vertical display.

SUI 2015, August 8–9, 2015, Los Angeles, California, USA.

ACM 978-1-4503-3703-8/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2788940.2788949

However a drawback of indirect touch is that the horizontal surface is only used as an input area. Should the horizontal input surface also become a output surface, a mode switching problem arises: users need somehow to specify if a touch was meant for a direct control of the horizontal display they physically touched, or if it should provide an indirect control over the vertical display in front of them. We present two different gaze-based mode switching techniques and compare them to the direct touch system with horizontal and vertical touch screens. The key idea of our approaches is to use the user's gaze at the moment of the initial touch to select the right surface. In the technique called Indirect Touch Surface Selection (ITSS), the gaze is used to select a screen to which the touch is mapped absolutely. In the other technique called Indirect Touch Object Selection (ITOS), the gaze is used to map the touch directly onto an object displayed on the vertical surface.

In our experiments, we investigate a system which combines one horizontal touch surface that is also an output surface (multi-touch desk), and one vertical input/output touch surface (vertical screen) in front of the user. Our interaction techniques can be a first step to make future interactive workspaces, as for example envisioned by Tognazzini et al. [22], more user friendly. The techniques could overcome important limitations of current interactive workspaces, which hinder users to use them on a day to day basis.

To summarise, our paper addresses the following three important aspects to improve future interactive workspaces:

1. We present two gaze-based interaction techniques that allow users to switch between direct touch interaction on a horizontal and indirect touch on a vertical surface.
2. We provide an empirical evaluation of the users' performance with these techniques compared to the direct touch system with both horizontal and vertical surfaces.
3. We outline how these results could be used to study multi-screen environments.

2. RELATED WORK

In the following subsections we discuss the related work in the fields of direct and indirect touch input and gaze supported touch interaction.

2.1 Direct and Indirect Touch Input

Weiss et al. [27] and Wimmer et al. [28] show how direct touch can enrich interactions for interactive workspaces, although it is cumbersome and tiring when interacting with the vertical screen. It is also shown that operating on the horizontal surface is faster and less exhausting than on the vertical [26]. To improve the users performance on vertical touch surfaces, Schmidt et al. [13] propose an approach that allow users to comfortably interact with a vertical screen of the interactive workspace utilizing indirect multi-touch. In such a setup, the neck pain and strain can be reduced as users do not have to bend over the horizontal surface. The problem of the proposed system is that users are not able to directly click or select a target object. A hover state is needed to indicate where the touch point is mapped onto the vertical surface. Schmidt et al. address this problem by projecting the shadow of the arm hovering above the horizontal input surface onto the vertical display. This require the users to constantly hover their arms above the horizontal surface, which is, according to his users, again cumbersome and uncomfortable. The problem of the missing hover state was later picked-up by Voelker et al. [26] by extending Buxton's [4] common two state interaction model of touch interfaces with a *tracking* state that allow users to touch the surface without directly manipulating the object below the touch point. This allow users to rest their arms on the horizontal display without unintentionally manipulating an object displayed on the vertical surface. Gilliot et al. [7] analyzed in impact of size and aspect ratio differences between the input and the output surfaces on indirect pointing tasks. Their studies showed that especially a different aspect ratio between input and output surface has a strong negative effect on the users performance. In contrast to the related work, we show how to switch between the two modes (direct and indirect touch)and we overcome the problem of indirect touch systems where the horizontal touch surface is degraded to merely an input surface.

2.2 Touch and Gaze Input

Results by Stellmach et al. [20] indicate that gaze input may be used as a natural input channel as long as certain design considerations are taken into account. Other researchers conclude that due to the inaccuracy, the *double role of eye gaze*, and the *Midas Touch* problem [19], it is ineffective to use eye gaze to directly manipulate digital content or control cursors. The study by Turner et al. [23] show that manual input conditions outperform gaze in transferring objects from a personal device to a public display and vise versa. Also, it is shown that a dwell time method is slower in comparison to techniques that allow users to confirm their actions using touch [24]. Eye focus selection as an independent channel of input is used in several research projects due to its high speed, familiarity and naturalness [17, 18, 25]. The eyes typically acquire a target well before manual pointing is initiated, following the principle "what you look at is what you get" [29]. Users tend to look at a target before issuing a command, starting an interaction, and look at the screen of interest, which makes gaze tracking a good interaction technique for window targeting [15, 25]. Using gaze as an additional input modality was also studied with a variety of user modalities other than touch. A study by Ashdown et al. [1] shows that combining head tracking with mouse input for a multi-monitor system is preferred by the users due to reduced mouse movement. Head motion is more stable, less accurately indicates the user's focus of attention, because eyes can move independently of the head.

The MAGIC (Manual And Gaze Input Cascade) technique proposed by Zhai et al. [30] is a combination of mouse and gaze input for fast target selection. Fono and Vertegaal [6] present an attentive windowing technique that uses eye tracking for focus selection, evaluated four focus selection techniques, and conclude that eye-controlled zooming windows with key activation provides efficient and effective alternative to current focus windows selection techniques. Eye tracking with key activation is, on average, about as twice as fast as mouse or hotkeys. Fono and Vertegaal results also show that despite the difference in speed between automatic activation and key activation for eye input, the eye input with key activation is a more effective method overall for focus window selection (about 72% faster than manual conditions), and was also preferred by most of the participants. Nancel et al. [11] investigated high precision pointing techniques for remotely acquiring targets and concluded that using head orientation for coarse control of the cursor and touch for precise selection was the most favorable and successful technique.

There are several approaches to combine gaze with manual interaction. Turner et al. [23, 24] combine gaze with mobile input modalities in order to transfer data between public and close proximity

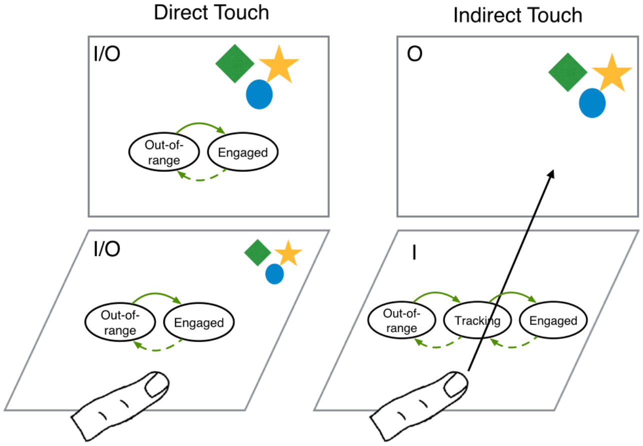

Figure 2: Direct and indirect touch interaction models for interactive workspaces

personal displays. The techniques for interaction in such environments are already outlined [15, 24]. Turner et al.'s study shows that manual input conditions outperform gaze positioning, which gives an advantage to the usage of manual input and leaves gaze a supporting role as a switching technique between the screens of interest. As recently shown by Pfeuffer et al. [12], the user's gaze can be used to perform this mode switch. They use gaze in combination with a single tabletop to place the user's touch points at the point of the display where the user is looking at by following the principle "gaze selects & touch manipulates". Their qualitative user study confirms the benefits of combining gaze and touch such as reachability, no occlusion, speed, less fatigue and less physical movement, and they provide a design space analysis of the properties of combining gaze and touch versus direct touch, and present several applications that explore how gaze-touch can be used alongside direct touch.

In contrast to Pfeuffer et al. and other related work, we make use of the user's gaze in interactive workspaces and extend the existing interaction models by two new variants. To our knowledge gaze supported touch input for interactive workspaces has not been investigated or researched before. Both modalities (touch and gaze) have been researched before separately or in combination, but in different contexts and especially not in the context of interactive workbenches as described in this paper. We also provide quantifiable results with a series of controlled experiments. Although gaze in combination with touch has been researched in recent years, we think that our results are important to get a better understanding for how gaze and touch can complement each other in other contexts, such as an interactive workbench that we investigate in detail in this paper.

3. INTERACTION TECHNIQUES

Currently, the default way to interact with interactive desktop workspaces is to use Buxton's [4] two-state touch model for both surfaces, as shown in Figure 2 (left). This allows the users to interact with both surfaces in the same way (namely direct touch) without having to change between different interaction techniques. Furthermore, it allows the users to interact with both surfaces using both hands and multiple finger at the same time. In our evaluation we use direct touch (DT) as a baseline condition to compare against the two new techniques we introduce in the paper. For DT a strong predictor for the time to point to a certain target is Fitts' law [5]. Recently, it has been adapted and extended to predict the pointing performance on modern touch screens [3]. As both of our new interaction techniques use a combination of touch and gaze, Fitts' law cannot be directly applied. Fitts' law was designed to predict the time needed for rapid aimed pointing tasks only. Therefore it cannot be used to predict the output of our studies. Nevertheless, similar to what is predicted by Fitts' law for simple target acquisitions, we also observed that both new interaction techniques (ITTS and ITOS) follow a similar, but more complex, model, that could be formalised in future research. As mentioned earlier, interacting with the vertical surface using DT can be cumbersome [27] and users prefer to provide input mainly on the horizontal surface. To overcome this problem, users can make use of indirect touch (IT) as illustrated in Figure 2 (right). The touch input on the horizontal surface is mapped to the vertical surface and allows users to remain in a comfortable seating positing, while interacting with the vertical surface. Again the users can use both hands and multiple fingers at the same time. One could assume the following situation: If users selects a target with our gaze supported technique, there could be a case where they simply need to confirm their gaze supported selection without even moving their hand. In this case, the link between the time needed to select a target and the distance to this target in the real world is broken. In addition, we also evaluate our interaction techniques with different dragging operations. The main difference to direct touch is that the users is not directly touching the objects anymore. The user rather has to estimate the area of the horizontal surface which maps to the object they want to touch.

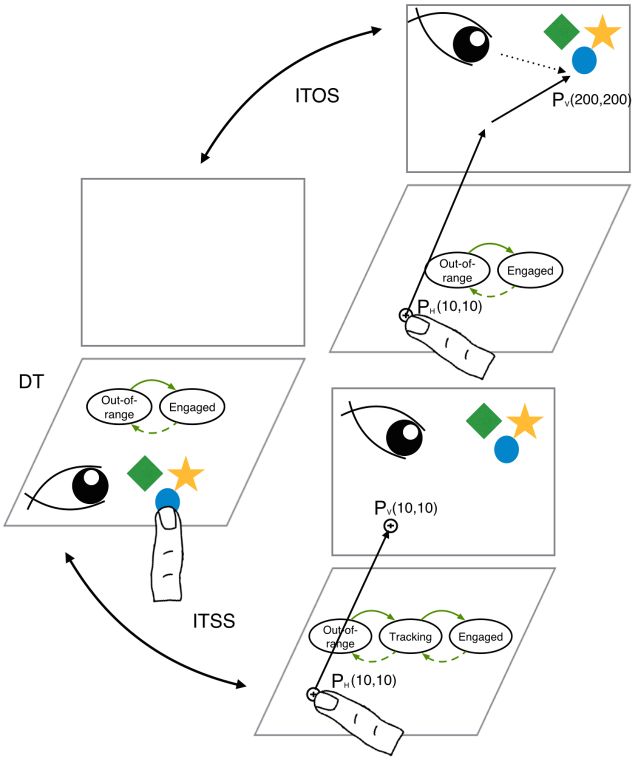

Figure 3: DT– traditional direct touch interaction. While using ITSS, user's touch from the horizontal screen is absolutely mapped to the vertical screen, if the user is looking at it. While using ITOS, user's touch is directly mapped to the object on the vertical screen that is in the user's focus.

This and the fact that in the common touch interaction model each touch directly manipulates the object that was hit by the touch lead to the problem that the user could unintentionally manipulate other objects while trying to hit one specific target using indirect touch. Therefore, a *tracking* [26] state is needed that allows users to aim for an object without manipulating the other objects (Fig. 2 right).

Furthermore, in an IT system the horizontal surface is only used as an input and not as an output device. In this paper, we present two novel interaction techniques, which combine both direct touch on the horizontal surface and indirect touch on a vertical surface. Since both interaction techniques could use the horizontal surface for input, the system needs to decide if the touch point created by the user should be used to directly interact with the horizontal surface (DT) or used to indirectly interact with a vertical surface (IT). We make use of the user's gaze to do this, as gaze input can be easily used as an additional input channel [20]. Furthermore, it was also shown by several research project that especially the combination of touch and gaze input works very well [12, 18, 19]. The problem of using the user's gaze is inconsistency of the eye due to its constant movement [10] and the fact that it can be easily distracted by unexpected events. For those reasons, our interaction techniques are using the user's gaze only for the *non-critical tasks* such as selecting the surface, area, or object which the user is currently looking at and to translate the initial position of the touch point. Every other operation, such as translating or manipulating the object, is done by the user's touch to prevent errors and reduce the impact of the gaze on the usability of the system.

In the following, we present two novel interaction concepts that use a gaze supported switching mechanism. Both are also illustrated in the attached video.

3.1 Indirect Touch Surface Selection (ITSS)

The first interaction technique, named ITSS, combines absolute DT and absolute IT: If the gaze is directed towards the horizontal touch surface, the system maps the touch point to the horizontal screen, allowing the user to interact with the object using the two-state DT interaction model (Fig. 3 left). If the user is looking at the vertical surface, the touch is translated using an absolute mapping to this particular vertical surface. For example, if both surfaces have the same size and resolution, when a user touches at point $P_H(10, 10)$ on the horizontal screen, the touch point is mapped to point $P_V(10, 10)$ on the currently looked at surface. Now, instead of using the two state interaction model, the three state indirect touch model [26] is used. Each new touch point that is mapped to a vertical surface is in a *tracking* state, which allows the user to move the finger over the surfaces without manipulating any object. To change the touch to the *engaged* state, which allows object manipulation, the user has to execute a *lift-and-tap* gesture as suggested by Voelker et al. [26]. This process is illustrated in Figure 3. In order to provide feedback, a cursor represents the touch point, displayed on the surface to which it is mapped. In both cases, if the touch is mapped to the horizontal or to a vertical surface, the touch stays on the surface until the user releases the finger from the input surface.. However, this is not the focus of this paper.

For the ITSS interaction technique we used an absolute mapping to prevent confusion when multiple cursors are present on the screen. Especially usage scenarios which involve multi-touch and bi-manual multi-touch input, cognitive load from mapping multiple touches and multiple cursors would be overwhelming.

3.2 Indirect Touch Object Selection (ITOS)

The first step of the ITOS interaction technique is the same as ITSS. Again, if the gaze is directed towards the horizontal touch surface, the system maps the touch point to the horizontal screen, allowing the user to interact with the object using the two-state DT interaction model.

For the second step, we initially planed to use the user's gaze not only to select the surface as in ITSS, but instead transfer the initial touch point to the position of the surface where the user is currently looking at. However, to the constant movement of the eyes, it is complicated to determine the exact position where the user is looking at as shown by Stellmach et al. [18].

Therefore, we choose to use a similar snapping mechanism for the ITOS technique as proposed by Pfeuffer et al. [12]. If the user is looking at the vertical surface or outside the horizontal screen, the touch is now translated to the object, on which the users gaze is concentrated. In this case, the user's gaze selects an object by highlighting it and a touch confirms this selection. To determine which object the user is currently looking at, we use an approach similar to the Bubble Cursor technique introduced by Grossman and Balakrishnan [8]. If a user touches the surface while he is looking at the vertical screen, the system calculates the area at which the user look at in the last 50 ms. If this area contains only one object this object is selected. If multiple objects are located in this area, the system calculates the center of the area and then selects the object which is closest to this center. For example, if a user touches point $P_H(10, 10)$ on the horizontal screen, and looks at the object on the vertical screen that is located at $P_V(200, 200)$, the touch point $P_H(10, 10)$ is mapped to point $P_V(200, 200)$. In contrast to the ITSS technique, using ITOS requires no *tracking* state, since the system highlights an object to which a touch is mapped before the user touches the screen. This allows the user to be sure to interact with one specific object without manipulating other objects.

4. USER STUDIES

We designed three different experiments to compare users interacting with different interaction techniques. We compared ITSS and ITOS against a DT baseline condition for a *tapping*, *dragging* (dragging an object on the vertical or the horizontal surface) and *cross dragging* (dragging an object from the vertical to the horizontal surface and vice versa) task. Even so it is known that interacting with a vertical display using DT leads to fatigue, as described in the introduction, we believe that users of future interactive workspaces will prefer direction manipulation [16] using touch instead of still relying on mouse and keyboard input typically used for GUIs, because the use of mouse and keyboard would require switches between different input concepts.

In this paper, we focus only on single-touch tasks to first understand how users perform using our two proposed interaction technique in basic tasks. In the future we plan to also evaluate more complex tasks that involve multiple fingers and both hands. Our experiments aims to answer the following questions:

1. Which technique is preferred in the indirect touch setup?
2. Which technique allows users to complete tasks faster and more accurate?

All three experiments were within subject experiments and we used the same setup and general procedure.

4.1 Participants

We recruited 14 participants (five female and nine male) aged between 23 and 36 (mean age 27.0). Twelve of the participants were right handed and two were left handed. All three experiments were conducted with the dominant hand of the user. On average, it took the participants about 1.2 h to complete all three experiments.

4.2 Apparatus

Participants sat at a desk with two touch displays, as shown in Figure 1. As a horizontal, screen we used a capacitive touch-sensing 27” Acer Touch display embedded in a custom made table at a height of 72 cm following ISO9241-5. For the vertical screen, we used a 27” Perceptive Pixel display, which was placed 55 cm from the edge of the table. Both displays had the same resolution of 2560 x 1440 pixels and the size of 597 x 336 mm. Both displays were connected to a Mac Pro running the software for the experiments. The effective touch frame rate for both displays was set to 60 Hz.

To determine the gaze of the users, we used the *Dikablis Glasses* by Ergoneers[1]. The *Dikablis Glasses* are a head-mounted eye tracking system that is able to detect the position of the user’s gaze in a visual marker coordinate system. Two markers were placed around the vertical display, as shown in Figure 1. By doing so, we can convert the gaze coordinates into the the pixel coordinate system of the vertical screen with an accuracy of about 1.5 cm (63 px). The effective frame rate of the eye tracker was also set to 60 Hz. The eye tracker was calibrated with a standard routine that comes with the eye tracker for each user before conducting the user study. This calibration process took about 30 seconds.

4.3 General Procedure

The participants conducted each experiment with all three interactions techniques (DT, ITSS and ITOS). Each participant conducted the experiments in a random order. No learning effects were observed by the experimenter or appeared in the data. Before the experiments, the users could run a ten minute test trials to familiarize themselves with the new interaction techniques and the different techniques. It was emphasized to solve a task as fast and as accurately as possible.

4.4 Experiment 1: Tapping

In the first experiment, we investigated the effect of the three different interaction techniques on the users’ performance by running the tapping task on both the horizontal and vertical surface.

4.4.1 Task

Participants were asked to touch blue circles, which were displayed alternating on the horizontal and vertical surface. As soon as the user touched the circle he had to hold his finger for 0.5 seconds on the circle before the circle disappeared and a new target circle on the other surface appeared. The task time was measured from the moment the target circle was visible till the moment it was successfully touched by the user. In order to complete one trial, users had to repeat this task for 50 targets, 25 on the vertical and 25 on the horizontal surface. The exact position of these targets was predefined and was the same for all the users. During a trial the circle size was fixed. The users had to conduct one trial for three different circle radii—63 px (1.5 cm), 126 px (3 cm), 252 px (6 cm). The 1.5 cm circle represents the smallest touchable button on a mobile device such as the Apple iPhone, the 3 cm circle a control element, and the 6 cm circle a picture or a document. The experimental design was a 3 (interaction technique) × 3 (target size) × 2 (target surface) mixed design with repeated measurements, which summarizes to a total of 450 tapping tasks per user. Since the required arm movement in the ITSS condition is expected to be smaller compared to ITTS and DT, we hypothesized the following outcome:

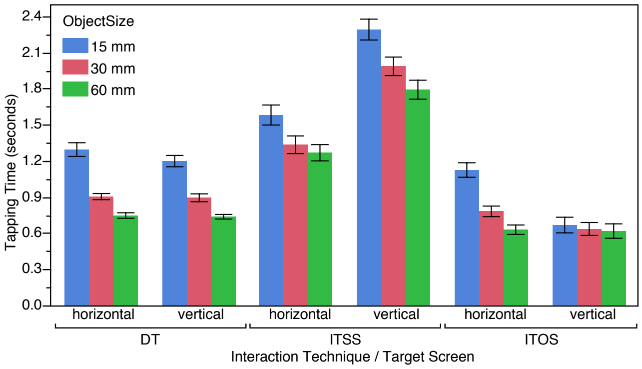

Figure 4: Users tapping times using all three interaction techniques in the Tapping experiment. Whiskers denote 95% confidence interval.

H1: Touching a target displayed on the vertical surface using indirect touch object selection is faster than using indirect touch surface selection and direct touch.

4.4.2 Results

The measured values were logarithmically transformed, according to the logarithmic distribution of the data. The data was analyzed for all dependent variables *interaction technique*, *target size* and *target surface* using a repeated measures ANOVA. We saw a significant main effect for the factor *interaction technique* in the ANOVA results $(F(2, 221) = 438.8255; p = 0.0001)$. The post-hoc Tukey HSD test comparison showed that overall tapping durations using ITOS (mean 0.61 sec) were 32% shorter than while using DT (mean 0.9 sec) and 60% shorter than ITSS (mean 1.54 sec). The ANOVA showed a significant main effect of the factor *target size* $(F(2, 221) = 78.7119; p = 0.0001)$. The post-hoc Tukey HSD comparison showed that the tapping time on the objects with a size of 63 px (1.5 cm) (mean 1.14 sec) was 19% longer than on the objects with a size of 126 px (3 cm) (mean 0.92 sec) and 29% longer than on the objects of size 252 px (6 cm) (mean 0.8 sec) for all three techniques. The main effect of the factor *target surface* was not significant. The ANOVA showed a significant interaction effect between the factors *interaction technique, target size* and *target surface* $(F(4, 221) = 4.301; p = 0.0001)$. The post-hoc Tukey HSD comparison revealed among other results the following: Using ITSS on the vertical screen the tapping times for all three target sizes was significantly slow compared to both other interaction techniques. On average the users need 2.15 sec to tap the small circles, 1.87 sec to tap the medium circles, and 1.68 sec to tap the large circles. Compared to both other interaction techniques, the users were faster using ITOS (H1). On average the users need 0.52 sec to tap the small circles, 0.5 sec to tap the medium circles, and 0.49 sec to tap the large circles.

4.4.3 Discussion

As expected, the ITOS technique was overall the fastest tapping technique in comparison to DT and ITSS (Fig. 4). This can be

---

[1] `http://www.ergoneers.com/`

explained by observing how users executed these tapping tasks. At the moment the new target was displayed, the users already touched the horizontal surface, since they previously touched a target object on the horizontal surface. So they only had to find and look at the new target to trigger the selection process. As soon as the target was highlighted, they only had to lift and tap anywhere on the horizontal surface again. Both of these actions can be executed very fast, especially finding and looking at a object on a nearly empty display. But also executing a lift and tap gesture on the horizontal surface can be done very fast, the users did not have to hit the same position on the display which they touched before releasing the finger. In comparison to other interaction techniques, ITSS required a longer interaction sequence. In the direct touch condition, the users had to move their entire arm to touch an object on the vertical surface, which required more time, since not only the arm muscles are involved in the movement, but also the shoulder. In the ITSS condition, the users also had to find and look at the new target, but instead of lifting and tapping anywhere on the surface, the users had to execute a more complex sequence of actions. First, the users had to estimate to which of the surfaces their touch is currently mapped. Secondly, the users had to move their arms to the estimated area and touch the horizontal surface with their fingers. Next the users had to identify whether the cursor on the vertical surface was actually on the target. If so, the user had to execute a lift-and-tap gesture, in order to successfully hit the target. If not, the users had to move their fingers until the cursor was on the target and then execute the gesture. In contrast to the lift and tap gesture in the ITOS condition, the users had to make sure that they tapped on the target.

Considering the object size, the objects with a bigger size were selected faster than smaller ones. However, this is not true in the ITOS condition for targets that where displayed on the vertical surface. For these targets, no significant differences were observed. This can also be explained by the fact that finding and looking at an object on an otherwise empty screen is very fast and is in this experiment not influenced by the size of the object. In other use cases, where in a small area of the surface a lot of object are displayed (e.g. menu with multiple buttons), the user would require more time to find the desired object. Also, due to the constant eye movement, the system would have taken longer time to decide at which of the object the user is currently looking.

4.5 Experiment 2: Parallel Dragging

After we analyzed how three interaction techniques influenced the users performance on tapping the objects, we wanted to explore how the users' performance is influenced by our interaction techniques while dragging the objects on the horizontal and vertical surfaces. Furthermore, for the direct touch condition, we wanted to check whether the vertical dragging introduces a fatigue effect that influences the users' performance.

4.5.1 Task

Users were asked to drag blue circles (160 px) to yellow rings (160 px) within the same display on the horizontal and vertical screens one after another. The initial scene displayed two circle ring pairs with a fixed distance of 1300 px (30 cm). Users were instructed to first start an the horizontal screen. The object is accounted as being at the destination if the position of the circle matches the destination ring within a range of 20 px. When circle and ring match and the user releases the hand, both objects disappear from the scene. This task is then repeated on the vertical surface. The next trial starts from the screen where the previous one was finished. To complete this task, users had to drag 25 objects into its targets on each screen. As depended variables we measured the dragging times on vertical and horizontal screens. Time was measured from the moment the circle was touched by the user until it was successfully released in its target ring on the same surface. Furthermore, the length of dragging trajectories was recorded.

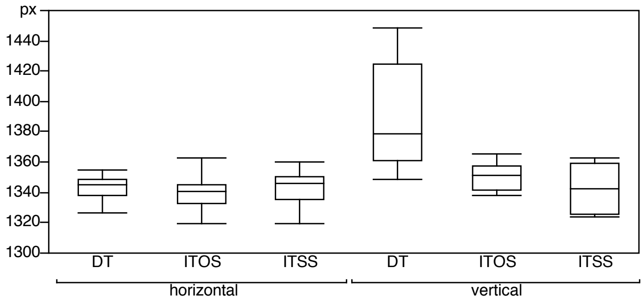

Figure 5: User's dragging trajectory length using all three interaction techniques subtask in the Dragging experiment. Scale starts at 1300 px which was the minimal distance the user had to drag. Whiskers denote 95% confidence intervals.

The experimental design was a 3 (interaction technique) × 2 (surface) mixed design with repeated measurements, which summarizes to a total of 300 dragging tasks per user. Based on previous research [27], that investigated the use of DT for an interactive workspace, and based on the fact that the user cannot rest their arms while interacting directly with the vertical surface, we hypothesized the following outcomes for the second experiment:

H2: Direct dragging is faster than indirect.

H3: Direct dragging an object on the vertical surface is less accurate than direct dragging on the horizontal and indirect dragging on the vertical surface.

H4: The dragging trajectory length increases over time while dragging objects on the vertical surface using DT.

4.5.2 Results

Due to the logarithmic distribution of the measured values for both depended variables, dragging time and trajectory length were logarithmically transformed. We analyzed both depended variables using a repeated measures ANOVA. For the *trajectory length*, the ANOVA reported a significant main effect of the factors *interaction technique* ($F(2, 65) = 13.4972; p = 0.0001$) and *surface* ($F(2, 65) = 18.5804; p = 0.0001$). The interaction also showed a significant effect ($F(2, 65) = 12.5804; p = 0.0001$). The post-hoc Tukey HSD showed that the dragging trajectory for the DT (mean 1368 px) was significant longer then the dragging trajectories for ITSS (mean 1347 px) and ITOS (mean 1346 px). It also showed that the trajectory length on the vertical surface (mean 1362 px) was significant longer than the trajectory length on the horizontal surface (mean 1345 px). The post-hoc Tukey HSD for the interaction showed that the trajectory length for the DT condition on the vertical surface (mean 1391 px) was significant longer than the other conditions (mean 1344–1350 px), as shown Figure 5. For the variable *time*, the ANOVA reported a significant main effect of the factors *interaction technique* ($F(2, 65) = 27.6531; p = 0.0001$) and *surface* ($F(2, 65) = 19.2332; p = 0.0001$). The interaction

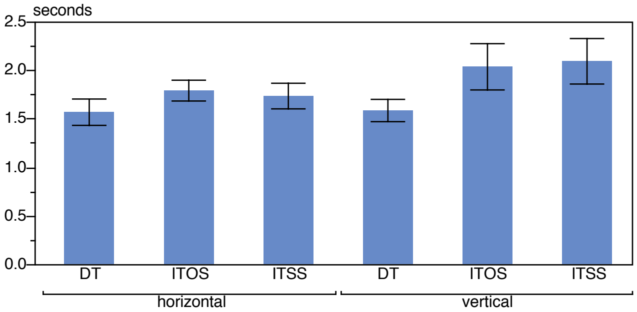

Figure 6: Users dragging times using all three interaction techniques subtask in the Dragging experiment. Whiskers denote 95% confidence interval.

also showed a significant effect $(F(2,65) = 6.9140; p = 0.0001)$. The post-hoc Tukey HSD showed that the dragging time for the DT (mean 1.583 sec) was significant shorter than the dragging trajectories for ITSS (mean 1.901 sec) and ITOS (mean 1.8729 sec). It also showed that the dragging time on the horizontal surface (mean 1.869 sec) was significantly shorter than the dragging time on the vertical surface (mean 1.695 sec). The post-hoc Tukey HSD for the interaction (Fig. 6) showed that the dragging time for the ITOS and the ITSS condition on the vertical surface (mean 1.994 sec; 2.08 sec) was significantly longer than the other conditions (mean 1.571–1.75 sec).

4.5.3 Discussion

As shown in Figure 5, the user's dragging trajectory is longer while dragging an object directly on the vertical surface in comparison to dragging it directly on the horizontal or indirectly on the vertical surface. As this could be expected, this can be explained by the understanding of the user's dragging operation execution. Movement of the fingers on the horizontal surface involves mostly the movement of the forearm and the wrist. However, users are able to rest their hands on the table during the horizontal dragging operation. When users are directly touching the vertical surface, the dragging movement mostly involve the upper arm and shoulder joints, which is more inaccurate as shown by Hammerton et al. [9]. Interestingly, as shown in Figure 6, the DT technique was overall the fastest dragging technique on the vertical screen in comparison to the other two. The shorter task-completion time in the DT condition might be caused by a fatigue users experienced after some time of interaction. Therefore, physical exhaustion decreases the time users want to spend holding their hands in the air. Furthermore, this could also indicated that dragging an object using indirect touch is cognitively more challenging than dragging it directly. Both points need to be taken into consideration, when designing interaction workspace for all-day use.

Considering the distance an object traveled on the vertical screen, it was longer for the DT technique than for the other two techniques, which could be explained by the loss of accuracy after a long-term DT interaction on the vertical screen. Physical movement of the hand while using ITSS and ITOS was always performed on the horizontal surface, which was causing less fatigue over time, because users were resting their hands on the surface while interacting. This shows an interesting interplay between the vertical and the horizontal surface. After this dragging experiment, mostly all users (twelfth) stated that especially dragging objects in the DT condition was extremely exhaustive. However, H2 was rejected since our recorded data did not show that this had any effect on the dragging time or trajectory length. But since the experiment took only about 3–5 minutes maybe it was too short to show a fatigue effect using direct touch on the vertical surface.

	df		*F*	*p*
Overall time				
Interaction technique	2	276	57.1986	<.0001
Dragging direction	2	276	12.7149	<.0001
Vertical time				
Interaction technique	2	276	10.9089	<.0001
Dragging direction	2	276	146.1459	<.0001
Switching time				
Interaction technique	2	276	33.1058	<.0001
Dragging direction	2	276	15.3929	<.0001
Overall trajectory				
Interaction technique	2	276	65.3526	<.0001
Dragging angle	2	276	29.1385	<.0001
Vertical trajectory length				
Interaction technique	2	276	3.4964	.0316
Dragging angle	2	276	9.5764	<.0001
Dragging direction	2	276	26.3364	<.0001
Horizontal trajectory length				
Interaction technique	2	276	6.7045	.0014
Dragging angle	2	276	9.5764	<.0001
Dragging direction	2	276	12.0905	<.0001

Figure 7: Significant main effects and interaction for the dependent variables in the CrossDragging experiment.

4.6 Experiment 3: Cross Dragging

After analyzing dragging operations that were only involving one of the surfaces, we wanted to explore how the interaction techniques affect the user performance in dragging tasks that involve switching from one to the other surface. Furthermore, we also wanted to explore if the effect that Weiss et al. [27] found in their cross surface dragging experiment can be observed using our interaction techniques. They showed that in diagonal dragging operations that involved a horizontal and a vertical surface the user dragging trajectories are significant longer than in dragging operations that go straight up or downwards.

4.6.1 Task

The task setup is similar to the cross dragging experiment conducted by Weiss et al. [27]. Users were asked to drag a blue circle (160 px) placed on the one surface into a white ring (160 px) placed on the other surface. To execute this task, users had to drag the blue circle to the edge of the surface such that it is visible on the other surface. Then they had to switch to the other surface to continue dragging the circle. The initial scene displays a circle ring pair on the fixed distance of 1631 px (37 cm). Trials appeared in seven different movement angles: 45°, 30°, 15° to the left, 0° (which is straight up or downwards) and 15°, 30°, 45° to the right.

Dragging had to start either on the horizontal (upwards) or vertical (downwards) display. The object is accounted as being at the destination if the position of the circle matches the destination ring within a range of 20 px. When the circle and ring match and the users releases their hand, both objects disappear and a new pair appears. Participants worked through 35 upwards and 35 downwards trials for each of the three interaction techniques, which results in a total of 210 dragging operations per user. The system automatically stores horizontal/vertical distance, and vertical, horizontal, and switch time.

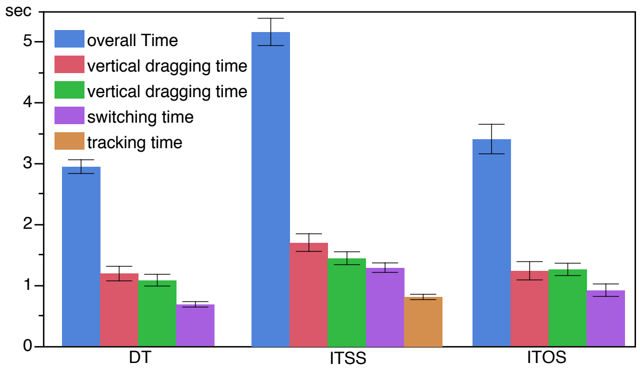

Figure 8: Users dragging times for the different subtask in the CrossDragging experiment. Whiskers denote 95% confidence interval.

The experimental design was a 3 (interaction technique) × 2 (vertical direction) × 7 (dragging angle) mixed design with repeated measurements. With five repetitions per target, each user had to perform 210 cross surface dragging operations. Again, by extrapolating the results of the study conducted by Weiss et al. [27] (H7) and based on the assumption that users can glance at an object faster than touching the object (H5, H6), we hypothesized the following outcomes:

H5: Overall, users complete the dragging operations faster using ITOS than using the other interaction techniques.

H6: Using ITOS, the time in which the user switches from interacting with one surface to interacting with the other surface is the shortest.

H7: The overall dragging trajectory is longer for larger dragging angles.

4.6.2 Results

Due to the logarithmic distribution of the measured values for all dependent variables, such as *overall time* (overall task completion time), *vertical time* (time needed to move an object on the vertical screen), *switching time* (time needed to switch from the horizontal to the vertical screen and vice versa), *overall trajectory* (overall physical distance user's finger traveled on both screens), *vertical trajectory length* (physical distance user's finger traveled on the vertical screen), *horizontal trajectory length* (the physical distance user's finger traveled on the horizontal screen), all of them were logarithmically transformed. A repeated measures ANOVA was conducted to compare the effect of interaction technique, dragging direction, and dragging angle as well as their interactions on the overall, horizontal and vertical dragging trajectory length, time, and switching time. The significant results are shown in Figure 7. For the post-hoc test the Student's t-test was used for the dragging direction variable. For the other variables we used the Tukey HSD test.

The post-hoc test for the *interaction technique* showed that the overall time using ITSS (4.99 sec) was significantly shorter than ITOS (3.183 sec) and DT (2.87 sec). Furthermore, upwards dragging (3.87 sec) was significantly slower than the downwards dragging (3.29 sec). The post-hoc test for the *vertical time* revealed the same results as for the overall time. Using ITSS (1.54 sec) was significantly slower than ITOS (1.06 sec) and DT (0.98 sec). Also, dragging upwards on the vertical surface (1.78 sec) took significantly longer than dragging downwards (0.77 sec). Similarly, for the horizontal dragging time, the post-hoc test revealed for the *interaction technique* the same results as for the overall time. Using ITSS (1.29 sec) was significantly slower than ITOS (1.08 sec) and DT (0.95 sec). Dragging upwards (0.94 sec) on the horizontal surface was significantly faster than dragging downwards (1.53 sec).

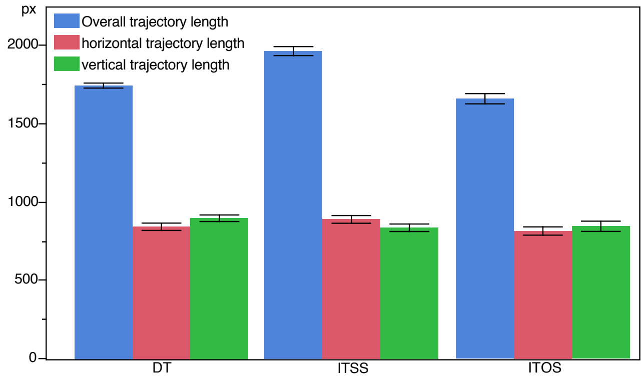

Figure 9: Users dragging trajectory length in the CrossDragging experiment. Whiskers denote 95% confidence interval.

Switching using DT (0.59 sec) was significantly faster than ITOS (0.83 sec), which was significantly faster than ITSS (1.32 sec). Switching form the vertical to the horizontal surface (0.74 sec) was faster than switching from horizontal to the vertical (1.01 sec).

The post-hoc test for the *interaction technique* showed that the *overall length* using ITOS (1568 px) was significantly shorter than for DT (1689 px) and ITSS (1879 px). For the *deltaAngle* factor the post-hoc test showed that *overall length* for 0° angle (1707 px) was significantly shorter than for 15° (1756 px), 30° (1809 px) and 45° (1851 px). The same tendency was shown for the factor *horizontal length*: for 0° angle *horizontal length* (802 px) was significantly shorter than for 15° (831 px), 30° (857 px) and 45° (879 px).

4.6.3 Discussion

As expected, the overall time and time on the vertical surface were longer for the ITSS technique than for the other two (Fig. 8). The primary reason is the existence of an additional *tracking* state in the interaction model. Users spend a lot of time on the moving the cursor to the object they want to select, while for DT, they could directly physically reach the target or for ITOS just look at the object of interest, which requires a much smaller amount of time. However, since no difference between the users performance between using ITOS and DT was found, H5 was rejected. A not obvious factor is the physical distance the user's arm had to travel in the air while switching between the vertical and horizontal surface. For DT, this distance is fixed and the smallest, and equals to the distance between the lower edge of the vertical screen and the upper edge of the horizontal. For ITOS, this distance is not fixed and depends on the strategy the user has chosen to use. As far as the user is not restricted by the touch area, after reaching the border between the horizontal and vertical screen, the distance depends on where the user touched the horizontal surface. Therefore, it equals the distance between the lower edge of the vertical screen and the point on the horizontal surface the user touched, which lays between the higher and lower edges of the horizontal screen. In the case of ITSS, traveling distance is always equal to the maximum—

Figure 10: Concept how our gaze supported interaction techniques can be applied to multi-screen environments.

the distance between the lower edge of the vertical screen and the lower edge of the horizontal. Moreover, for both the upward and downward moving direction those distances were the same. For this reason, the switch time as shown in Figure 8 is the longest for ITSS and comparably shorter for ITOS and DT. Furthermore, since in this case DT outperformed ITOS in terms of switching time H6 also does not hold.

Considering the overall time duration on the horizontal screen, it was the longest for ITSS in comparison to the other two techniques. The horizontal time for the three techniques for the upward direction is the same; because they repeat the same sequence of actions, the most influential part lays on the downward direction. As mentioned above, for the DT technique the physical movement distance in the air is static for both upwards and downwards directions. However, for the ITOS technique, users could overcome the border between the screens without re-grabbing an object and move it for some time on the horizontal screen without reaching the target. Therefore, the time needed for the movement on the horizontal screen was comparably lower than for ITSS, where users always had to move an arm from the lower edge to the upper edge of the horizontal screen. H7 was confirmed by our results and it seems that the dragging trajectories are longer for more diagonal dragging operations. These results show, that this effect is not only true on curved surfaces [27], but also for systems that combine horizontal and vertical surfaces that are not connected by a curved surface. Interestingly is that this effect is also true if the users are not directly interacting with the vertical surface. If we look at the results for the horizontal and vertical dragging trajectories, this effect is only visible for the horizontal surface. This leads to the assumption that users try to minimize their movement on the vertical surface even when they are not directly interacting with it. In general these results indicate that even if interaction using indirect touch and direct touch are execute on the same surface user tend to prefer direct touch over interact touch in basic operations.

5. MULTI-SCREEN WORKSPACES

Both interaction techniques, ITSS and ITOS, can be extended to system setups that include multiple vertical surfaces (Fig. 10). Instead of switching only between horizontal and vertical screens for ITSS, the system allows switching among a number of vertical surfaces and activating the screen that is currently in the user's focus. Using ITOS, it does not matter for the system how many displays are presented in the setup, since the user selects an object using its gaze. Selection can be done for objects on one horizontal surface or n vertical surfaces as long as the user is able to see an object on one of the surfaces. However, in a setup with multiple vertical screens that are far away from the user using ITSS seems to be a better solution. The reason is the easiness of determining whether a user is looking at a large screen that is far away or identifying one small object on that screen. Moreover, for ITSS low-cost head- or eye-trackers can be used. We also envision head- or eye-tracking technology, that is embedded in the displays, that could replace the head-mounted head- or eye-tracker and could open up further interaction contexts. Using the head position as a focus 'pointer' is more robust and precise, and usage of external eye-gaze trackers is more comfortable, especially for a long time interaction.

6. CONCLUSION & FUTURE WORK

In this paper, we propose two novel gaze-based interaction techniques, namely ITSS and ITOS, for easy touch interaction for interactive workspaces. With the help of these gaze supported interaction techniques it is possible to enrich the interaction with interactive workspaces as first envisioned by Tognazzini's Starfire [22] concept. By introducing gaze as a additional modalities we are able to reduce the time that is needed to reach targets on the vertical screen as well as reduce effort that is needed to interact with the system. This enables users to comfortably interact with interactive workspaces for a longer time (e.g. a full working day). Nevertheless further studies are needed to investigate these long-term effects. We evaluated the performance of these interaction techniques compared to DT for different standard tasks, such as tapping and dragging an object within or between the screens. Our results indicate that ITOS outperforms the DT and ITSS in terms of tapping speed. ITOS was about 32% faster than DT. We believe that these techniques can make interactive workspaces more user friendly in the future.

With the help of our studies we analyzed the characteristic of these interaction techniques compared to DT. In the case of the dragging an object on a screen, DT outperformed ITOS and ITSS in speed on the vertical screen, but was least accurate, reporting the longest traveled distances on the vertical screen. Results of cross dragging between the screens indicate that moving an object downwards over all techniques was faster than upwards. The traveled distance of the finger on the screen was increasing with the angle between the object and the target. For this task, and in terms of speed, direct touch outperformed ITSS by 42% and ITOS by 9%. We conclude that ITOS provides an efficient and effective alternative in some situations for manipulation with indirect touch systems. We only explored basic single finger tasks in which only few objects were displayed on the surfaces. In use cases where multiple small objects are displayed in the same area of a vertical surface, using ITOS could become more difficult. In these cases, it is more complicated for the system to decide at which object the user is looking at, which could lead to a wrong object selection.

We believe that our new techniques can be used in different interaction contexts, even so both require the users to wear a eye-tracker. These interaction contexts include future office spaces or traffic control rooms. Gaze-based interaction offers a new direction of using systems that combine one or more horizontal and vertical touch screens and is not only limited to interactive workspaces. In the future, we want to investigate the usage of our proposed interaction techniques for setups with one horizontal, but multiple vertical touch screens. We also believe that advantages in technology will make setups with continuous surfaces possible as originally envisioned by Curve [28] and BendDesk [27]. It would also be interesting to extend or adapt Fitts' law to predict the time needed to select a certain target with ITTS or ITOS. Furthermore, we also want to explore mobile settings, where the horizontal surface is, for example, an iPad, and the users interacts with multiple screens situated in their environment. This will lead to new application designs, where eye gaze can be used complementary to direct touch interaction. Additionally, we intend to specify the limitations of such systems, use cases.

7. ACKNOWLEDGEMENTS

This work was partially funded by the German B-IT Foundation. We would like to thank the anonymous reviewers for their valuable comments. We are grateful to Davy Vanacken for proof-reading the paper.

8. REFERENCES

[1] Ashdown, M., Oka, K., and Sato, Y. Combining Head Tracking and Mouse Input for a GUI on Multiple Monitors. In *CHI EA*, ACM (New York, USA, 2005), 1188–1191.

[2] Bi, X., Grossman, T., Matejka, J., and Fitzmaurice, G. Magic Desk: Bringing Multitouch Surfaces into Desktop Work. In *Proc. CHI*, ACM (New York, USA, 2011), 2511–2520.

[3] Bi, X., Li, Y., and Zhai, S. FFitts Law: Modeling Finger Touch with Fitts' Law. In *Proc. CHI*, ACM (New York, USA, 2013), 1363–1372.

[4] Buxton, W. A Three-State Model of Graphical Input. In *Proc. INTERACT*, North-Holland Publishing Co. (Amsterdam, The Netherlands, 1990), 449–456.

[5] Fitts, P. M. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of experimental psychology 47*, 6 (1954), 381.

[6] Fono, D., and Vertegaal, R. EyeWindows: Evaluation of Eye-controlled Zooming Windows for Focus Selection. In *Proc. CHI*, ACM (New York, USA, 2005), 151–160.

[7] Gilliot, J., Casiez, G., and Roussel, N. Impact of Form Factors and Input Conditions on Absolute Indirect-touch Pointing Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (New York, USA, 2014), 723–732.

[8] Grossman, T., and Balakrishnan, R. The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area. In *Proc. CHI*, ACM (New York, USA, 2005), 281–290.

[9] Hammerton, M., and Tickner, A. H. An Investigation into the Comparative Suitability of Forearm, Hand and Thumb Controls in Acquisition Tasks. *Ergonomics 9* (1966), 125–130.

[10] Kern, D., Marshall, P., and Schmidt, A. Gazemarks: Gaze-based Visual Placeholders to Ease Attention Switching. In *Proc. CHI*, ACM (New York, USA, 2010), 2093–2102.

[11] Nancel, M., Chapuis, O., Pietriga, E., Yang, X.-D., Irani, P. P., and Beaudouin-Lafon, M. High-precision Pointing on Large Wall Displays Using Small Handheld Devices. In *Proc. CHI '13*, ACM (New York, USA, 2013), 831–840.

[12] Pfeuffer, K., Alexander, J., Chong, M. K., and Gellersen, H. Gaze-touch: Combining Gaze with Multi-touch for Interaction on the Same Surface. In *Proc. UIST*, ACM (New York, USA, 2014), 509–518.

[13] Schmidt, D., Block, F., and Gellersen, H. A Comparison of Direct and Indirect Multi-touch Input for Large Surfaces. In *Proc.INTERACT*, Springer-Verlag (Berlin, Heidelberg, 2009), 582–594.

[14] Schöning, J. Touching the Future: The Rise of Multitouch Interfaces. *PerAda Magazine* (2010), 5531.

[15] Shell, J. S., Vertegaal, R., Cheng, D., Skaburskis, A. W., Sohn, C., Stewart, A. J., Aoudeh, O., and Dickie, C. ECSGlasses and EyePliances: Using Attention to Open Sociable Windows of Interaction. In *Proc. ETRA*, ACM (New York, USA, 2004), 93–100.

[16] Shneiderman, B. Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces. In *Proceedings of the 2Nd International Conference on Intelligent User Interfaces*, IUI '97, ACM (New York, USA, 1997), 33–39.

[17] Smith, J. D., and Graham, T. C. N. Use of Eye Movements for Video Game Control. In *Proc. ACE*, ACM (New York, USA, 2006), Article 20.

[18] Stellmach, S., and Dachselt, R. Look & touch: Gaze-supported target acquisition. In *Proc. CHI*, ACM (New York, USA, 2012), 2981–2990.

[19] Stellmach, S., and Dachselt, R. Still Looking: Investigating Seamless Gaze-supported Selection, Positioning, and Manipulation of Distant Targets. In *Proc. CHI*, ACM (New York, USA, 2013), 285–294.

[20] Stellmach, S., Stober, S., Nürnberger, A., and Dachselt, R. Designing Gaze-supported Multimodal Interactions for the Exploration of Large Image Collections. In *Proc. NGCA*, ACM (New York, USA, 2011), 1:1–1:8.

[21] Sutherland, I. E. Sketch Pad a Man-machine Graphical Communication System. In *Proc. DAC*, ACM (New York, USA, 1964), 6.329–6.346.

[22] Tognazzini, B. The "Starfire" Video Prototype Project: A Case History. In *Proc. CHI*, ACM (New York, USA, 1994), 99–105.

[23] Turner, J. Cross-device Eye-based Interaction. In *Proc. UIST Adjunct*, ACM (New York, USA, 2013), 37–40.

[24] Turner, J., Bulling, A., and Gellersen, H. Combining Gaze with Manual Interaction to Extend Physical Reach. In *Proc. PETMEI*, ACM (New York, USA, 2011), 33–36.

[25] Vertegaal, R., Mamuji, A., Sohn, C., and Cheng, D. Media Eyepliances: Using Eye Tracking for Remote Control Focus Selection of Appliances. In *CHI EA*, ACM (New York, USA, 2005), 1861–1864.

[26] Voelker, S., Wacharamanotham, C., and Borchers, J. An Evaluation of State Switching Methods for Indirect Touch Systems. In *Proc. CHI*, ACM (New York, USA, 2013), 745–754.

[27] Weiss, M., Voelker, S., Sutter, C., and Borchers, J. BendDesk: Dragging Across the Curve. In *Proc. ITS*, ACM (New York, USA, 2010), 1–10.

[28] Wimmer, R., Hennecke, F., Schulz, F., Boring, S., Butz, A., and Hussmann, H. Curve: Revisiting the Digital Desk. In *Proc. NordiCHI*, ACM (New York, USA, 2010), 561–570.

[29] Zhai, S. What's in the Eyes for Attentive Input. *Commun. ACM 46*, 3 (Mar. 2003), 34–39.

[30] Zhai, S., Morimoto, C., and Ihde, S. Manual and Gaze Input Cascaded (MAGIC) Pointing. In *Proc. CHI*, ACM (New York, USA, 1999), 246–253.