Multi-touch 3D Positioning with the Pantograph Technique

Thomas Butkiewicz
University of New Hampshire
Durham, New Hampshire, USA
tbutkiewicz@ccom.unh.edu

Andrew H. Stevens
University of New Hampshire
Durham, New Hampshire, USA
astevens@ccom.unh.edu

Colin Ware
University of New Hampshire
Durham, New Hampshire, USA
cware@ccom.unh.edu

ABSTRACT

One advantage of touch interaction is the sense of direct manipulation; there is perhaps no more-intuitive interface than just reaching out and touching virtual entities. However, direct manipulation is generally limited to objects located on the 2D display surface. For 3D spaces extending behind or in front of a touchscreen, the direct manipulation metaphor quickly falls apart. In these cases, gestures are needed to convert 2D finger positions into 3D cursor positions. This paper presents the pantograph technique, a simple two-finger interaction method for positioning a 3D cursor within mono and stereoscopic applications. The pantograph's pseudomechanical linkage between fingers and cursor provides helpful depth cues and maintains the sense of direct manipulation. Extensions to the technique, which integrate selection and other advanced actions, are explored within the context of real-world visual analysis applications. A series of human factors experiments showed that, while the pantograph technique outperformed other similar multitouch 3D positioning techniques, multi-touch was still inferior to other traditional, non-touch-based interfaces for sustained 3D positioning tasks.

CCS CONCEPTS

• Human-centered computing → Interaction techniques

KEYWORDS

Multitouch, interaction, technique, evaluation, touchscreen, HCI.

ACM Reference format:

Thomas Butkiewicz, Andrew H. Stevens, and Colin Ware. 2019. Multi-touch 3D Positioning with the Pantograph Technique. In *Symposium on Interactive 3D Graphics and Games (I3D '19), May 21–23,2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3306131.3317014

1 Introduction

Multi-touch interfaces are now ubiquitous, and stereoscopic displays have matured from specialty equipment to mass-produced electronics. Combinations of the two are an exciting area of research, with one obvious question: How do you use a 2D input device to interact in 3D?

Virtual reality researchers have investigated this, extending the standard mouse into virtual environments, but found actual 3+ degrees of freedom (DOF) interaction devices were superior [Ware and Jessome 1988]. Why then, might another 2D input device, the multi-touch surface, be any more successful than a mouse for 3D interaction?

Multi-touch interfaces have many promising advantages: While each touch provides only two DOF, multiple touches can provide supplemental DOFs, either directly in the form of additional (x,y) coordinates, or through relative metrics, such as distances or angles between touches.

Touchscreens also do not physically encumber or inconvenience users with wired devices or worn markers. This avoids decoupling users from their normal workflow, making touchscreens more suited for real-world work.

Finally, there is a significant steadying effect obtained by physical contact with a touchscreen. An outstretched hand is difficult to hold steady, causing errors when using 3DOF devices [Ware and Balakrishnan 1994]. A finger, however, can cast a ray into a 3D world while being steadied by the physical contact.

Many multi-touch interaction methods focus on selection and manipulation of virtual objects, often sparsely distributed in a simple world. That most methods apply to pre-existing objects is not surprising, as direct selection is a classic "selling point" of touchscreens. However, specifying arbitrary locations in 3D space is more challenging than merely selecting from a set of objects within the same space. This is the distinction between the more specialized task of selection and the general task of positioning.

This paper presents the pantograph technique for multi-touch 3D positioning. This simple one-handed technique uses the thumb and forefinger in a pinching gesture, with the spread of the fingers controlling the depth of the cursor. Visual feedback, in the form of a pseudo-mechanical linkage between fingers and the cursor, provides helpful depth cues and preserves the feeling of direct manipulation. Extensions enabling additional actions, such as selection, are explored within the context of real-world scientific visualization applications.

Finally, a series of experiments compare the performance of the pantograph technique to other multi-touch 3D positioning techniques and explore the broader question of how effective multi-touch 3D positioning is in comparison to more traditional, non-touch-based positioning methods.

I3D '19, May 21–23, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6310-5/19/05. . . $15.00
https://doi.org/10.1145/3306131.3317014

2 Related Work

There are a number of examples in the literature of multi-touch interaction within 3D and stereoscopic applications. However, Martinet et al. [Martinet et al. 2010a] note that, while many multi-touch techniques have been proposed for selecting and manipulating virtual objects in 3D, there is a distinct lack of techniques allowing "free 3D positioning." This is largely due to focusing on the direct manipulation aspect of touch interaction.

A significant contrast between the aims of a generalized 3D positioning technique and many existing techniques for virtual object manipulation is differences in the virtual environments. Most stereoscopic multi-touch systems in the literature present their techniques in sparse virtual environments [Benko and Feiner 2007; Hachet et al. 2011; Strothoff et al. 2011]. This is in contrast to the denser, space filling environments encountered in volumetric medical, atmospheric, and geospatial visualizations. In these types of applications, the entire screen can be filled with data, and the user needs to be able to interact with any particular point, whether it is part of a data item or in the spaces between.

Steinicke et al. [Steinicke et al. 2007] introduce the concept of interscopic interaction within graphical interfaces of fish tank VR [Ware et al. 1993] systems. They present ways to merge monoscopic and stereoscopic elements to avoid disturbing the stereoscopic illusion. These include techniques such as changing the depth of a 2D cursor between areas of different stereo depth.

Schöning et al. [Schöning et al. 2009] expanded upon this by incorporating multi-touch interaction within "Windows on the World", a 2D control interface which the user can directly touch and manipulate to perform navigation within the deeper 3D virtual environment. The control window is presented at the depth of the touch surface to preserve the sense that the user is actually touching and manipulating an object directly.

A longstanding problem in VR is that when objects in a 3D environment are at a distance, they can be out of reach for direct selection or manipulation. On a 2D touch surface, objects at any depth (other than directly on the screen) will be out of reach. A common solution is ray casting, with a variety of techniques to differentiate multiple intersected targets. However, for positioning, there must be a method for controlling the depth of a cursor constrained along the ray. Most multi-touch 3D positioning techniques use this cursor-along-a-ray strategy, but differ in how they position the ray and control the cursor position along it.

Benko and Feiner's balloon selection technique [Benko and Feiner 2007] positions a cursor above a touch surface based on the metaphor of holding a floating balloon at two points on its string. This bimanual 3DOF positioning technique uses one primary finger to determine (x,y) cursor location, and distance to a second finger (on the other hand) to determine cursor height. Selection is indicated by placing a third finger directly adjacent the stretching finger. They found selection speeds similar to a 3DOF wand, but with less error. They noted that, even for users who preferred the wand selection, the balloon selection technique was reported as less fatiguing. These two results support the predicted steadying effect of physical contact with a touch screen over the use of a freely outstretched arm holding a 3DOF positioning device.

When manipulating/repositioning virtual objects, Toucheo [Hachet et al. 2011] provides a vertical ray linking the 3D object to a 2D widget on the touchscreen below. Virtual objects cannot be repositioned in all three dimensions simultaneously. Instead, users must iteratively adjust the (x,y) location via dragging and adjust the (z) depth using a stretching widget.

Another way to deal with vergence-accommodation conflicts that arise from stereoscopic viewing [Hoffman et al. 2008] is to decouple the multi-touch input from the display. Evaluations by Simeone [Simeone 2016; Simeone and Gellerseny 2015] showed that multi-touch interactions on a separate tablet are comparable in speed and accuracy to input techniques directly coupled with the display. Lopez et al. [López et al. 2016] also employ a tablet for multi-touch input and to show a monoscopic 'snapshot' of a dataset rendered on a larger stereoscopic display. Users reposition themselves around the stereoscopic display to change the tablet's frame of reference, and interact solely on the tablet.

Martinet et al. [Martinet et al. 2010a] present the z-technique for multi-touch 3D selection/positioning. The first finger touching the screen extends a ray orthogonally into the screen, selecting the first intersecting object. Finger movement repositions the object parallel to the screen. Vertical movement of a second finger moves objects further away from, or closer to the screen. This is essentially a multi-touch recreation of a simple mouse technique [Venolia 1993] in which the mouse moves a cursor in 2D as usual, while the mouse's scroll wheel moves the cursor in depth into/out of the screen. Z-technique and balloon selection are very similar; the main difference being what secondary finger movements are measured relative to (primary finger vs. screen vertical).

Pierce et al. [Pierce et al. 1997] use a thumb-forefinger pinching action to interact at a distance in their "head crusher" 3D object selection technique. They consider the positions of the pinching fingers relative to the image plane as a pinching gesture encloses the projection of a virtual object. A selection ray is then cast from the user's eye through the midpoint of the fingers.

Hancock et al. [Hancock et al. 2007] present a single-touch method for 5DOF movement of 3D objects in shallow 3D environments by making a finger "stick" to the touched part of an object. As the finger is dragged, the object moves/rotates in an attempt to make the touched point as close to the surface as possible. They extend this to a bi-manual, two-touch version, in which a second finger induces pitch/roll; it is briefly mentioned that z-movement can be induced by changing the distance between the two points.

Hancock et al. later expand upon this z-movement behavior in their sticky fingers technique [Hancock et al. 2009]. When the user touches the projected image of a 3D object with two fingers, the points of "contact" on the object stick to the fingers, with any movement of the fingers inducing movement of the object to maintain those points of contact. Thus, by spreading the fingers apart, objects are lifted towards the screen (a pinch lowers them away from the screen). While suited for moving existing objects, sticky fingers does not translate well to moving a 3D point cursor.

An earlier, less refined version of the pantograph technique was first presented for multi-touch 3D positioning within a fish tank VR application [Butkiewicz and Ware 2011]. It was used to insert dye particles into a volumetric ocean flow model. The ability to quickly reposition the cursor simultaneously in all three dimensions was integral to support exploratory dye release into moving water masses. While the basic concept of controlling a 3D cursor's position via thumb and forefinger persists, the particulars of the technique, and especially the onscreen widget have evolved to add new functionality and improve stereoscopic depth cues.

Shortly thereafter, Strothoff et al. [Strothoff et al. 2011] presented triangle cursor, a very similar multi-touch positioning technique for stereoscopic 3D. Like the pantograph technique, it positions a 3D cursor at the midpoint of the thumb and forefinger. In contrast to the pantograph technique, it was intended for use in above-tabletop virtual environments, so finger spread instead controls the height of the cursor above the surface. It also does not take advantage of the benefits that a cursor offset has been shown to provide [Benko et al. 2006; Potter et al. 1988]. A fourth DOF was integrated by using inter-finger angle to rotate selected objects about the z-axis. Participants in an informal evaluation against balloon selection for moving along a 3D pathline indicated that triangle cursor was faster and more appropriate for positioning.

Triangle cursor was also formally evaluated against a balloon selection technique modified to support z-axis rotations for moving cubes into goal volumes with different locations, heights, and orientations. Triangle cursor was generally found to be significantly faster than balloon selection in all cases, and for cases that involved orientation changes, it was significantly more accurate. They note that the gain in speed is partially attributable to the ability to begin triangle cursor positioning with the fingers already at a distance approximating the desired depth value, lowering the amount of adjustment required. They also attribute the precision to the dual finger midpoint cursor positioning.

Our pantograph technique functions analogously to a combination of the (x,y) positioning of Benko et al.'s dual finger midpoint technique [Benko et al. 2006], with its associated increased precision and decreased occlusion, and the (z) depth positioning of Hancock et al.'s sticky fingers [Hancock et al. 2009] technique, without the requirements of contacting an object.

The pantograph technique has two significant advantages over balloon selection: Balloon selection requires an initialization gesture to set initial string length (an overhead cost each time it is used.) In contrast, upon placing two fingers on the screen, the pantograph technique instantly sets the cursor's (z) position based on inter-finger distance, avoiding this overhead. Furthermore, positioning can start near the target depth by choosing finger-spread before the hand even contacts the touchscreen. Strothoff et al. [Strothoff et al. 2011] confirms that overcoming the string-setting overhead and the "start at a particular depth" strategy was key to triangle cursor's speed advantage over balloon selection.

Benko and Feiner [Benko and Feiner 2007] reported users had difficulty under-standing the string stretching states, which did not map well to a physical metaphor. They alleviated this with sound effects and changing colors. In contrast, the pantograph has a strict 'physical' linkage metaphor between the fingertips, their connection, and the cursor. Explicit lines show the relationships between these elements and provide strong visual feedback to reinforce the pantograph metaphor, making it easily understood.

Another important distinction from balloon selection and z-technique is the change in (x,y) cursor positioning from the primary fingertip to a midpoint between two fingers, which alleviates occlusion from the hand and doubles (x,y) precision. Strothoff et al. [Strothoff et al. 2011] observed the same benefits with triangle cursor. The pantograph technique goes a step beyond triangle cursor by offsetting the midpoint away from the hand to avoid occlusion from pinching fingers.

While balloon selection adjusts volumetric cursor size with an additional tertiary finger, the 4DOF pantograph technique permits this adjustment using only the original two fingers and without interrupting the positioning task.

The along-axis pantograph mode does not appear in any existing multi-touch 3D positioning techniques. This feature is critical for data with dimensions that are meaningful to constrain movement along, especially as viewing angles change. It can increase depth adjustment precision by mapping to a relatively narrow range in data bounds, instead of deeper screen-depth ranges resulting from viewing at an angle or at a distance.

3 Design & Implementation

This section presents a 3D positioning technique based on the physical behavior of a pantograph. As illustrated in Figure 1, the pantograph technique uses the distance between two fingers to control the linear motion of a cursor along the third dimension not directly accessible on a 2D touchscreen.

Specifically, as shown in Figure 2, the pantograph technique determines the (x,y) position of the cursor based on the orthogonally offset midpoint between the thumb and forefinger. The (z) depth position of the cursor is mapped to the distance between the thumb and forefinger.

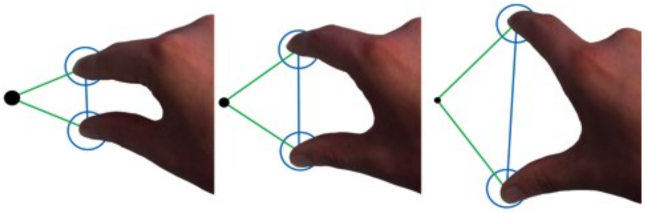

Figure 1: The pantograph technique maps inter-finger distance to cursor depth. As the fingers spread apart, the cursor (black dot) moves deeper into the screen.

Figure 2: The cursor (black) is offset orthogonally (red) from the midpoint (orange) of the touch-points. Connector lines (green) visually tie the 2D touch-points and widget to the 3D cursor, providing important depth cues.

Because the cursor's location is calculated using both finger locations, the granularity of cursor movements is twice as fine as for a single finger. This has been shown to be an effective means of increasing touchscreen precision [Benko et al. 2006].

The cursor's (x,y) location is offset to avoid occlusion by the hand/fingers, especially when pinching close together, which would obscure the midpoint. Handedness, and thus offset direction, can be automatically detected by hand shadows in DI systems [Echtler et al. 2008], shape/orientation of touch points on FTIR surfaces [Dang et al. 2009], or external RGB or depth cameras [Butkiewicz 2012]. An offset of ~3cm is generally sufficient.

Offsetting the cursor from the midpoint also increases the obliqueness of the lines connecting it to the fingertips at the depth of the screen. Without this offset, these lines would be shorter and viewed more directly from straight above, reducing depth cues.

Martinet et al.'s taxonomy [Martinet et al. 2010b] of multi-touch 3D manipulation techniques makes distinctions between direct and indirect fingers, based on physical distance between finger and target object. By this definition, the pantograph technique would be considered indirect. However, because the pantograph widget visually enforces the illusion of a virtual "mechanical linkage", which acts as an extension of the user's fingers into the screen, it could be considered direct, as the cursor's movement/location is strictly linked to the movement/location of the fingers. Strothoff et al. report the observation of this same perceptual illusion of direct interaction in their triangle cursor technique [Strothoff et al. 2011].

The pantograph technique distributes physical demand to both a pinching action of the fingers, and movement of the hand as a whole, using the robust musculature of the arm and shoulder. In Card et al.'s [Card et al. 1991] characterization of input device effectiveness, it is noted that consideration should be given to the bandwidth of human muscle groups involved. By collapsing positioning tasks into complex multi-finger movements, there is a risk of saturating the muscle control bandwidth that exists for the fingers. By offloading a portion of the movement to muscle groups controlled by other regions of the motor cortex, the burden is distributed across multiple pathways, with an overall greater bandwidth. Moscovich and Hughes [Moscovich and Hughes 2008] confirmed the ease of simultaneously adjusting the spread and position of the thumb and forefinger on touchscreens.

The pantograph technique's distribution of physical movements is ideal for configurations in which a large format touch display contains a relatively shallow 3D virtual environment, as movements in (x,y) are often of significant physical distance, requiring articulations of the entire arm; whereas depth adjustments are made across a smaller scale, to which the limited range of thumb-forefinger distance is able to satisfactorily accommodate. Techniques such as Lopez et al.'s [López et al. 2016] are more appropriate for smaller displays and deeper 3D worlds.

3.1 Along-Axis Depth Adjustments

In the basic version described so far, depth adjustments are screen relative; moving the cursor orthogonally into/out of the display. However, along-axis depth adjustments are another option that can be useful for datasets which are expansive in two dimensions but relatively shallow in the third dimension. This is especially common with geospatial datasets, which often cover wide regions, but have relatively small height/depth ranges.

For along-axis adjustments, the screen coordinates of the offset midpoint are projected onto the surface of the model to get the (x,y) location of the cursor in model coordinates.

An advantage of the along-axis method is that it keeps the dimensional mapping of physical actions consistent between different camera angles. For example, in geospatial datasets, moving the hand around the screen always adjusts latitude/longitude, while changing the inter-finger distance always adjust the height of the cursor above the terrain.

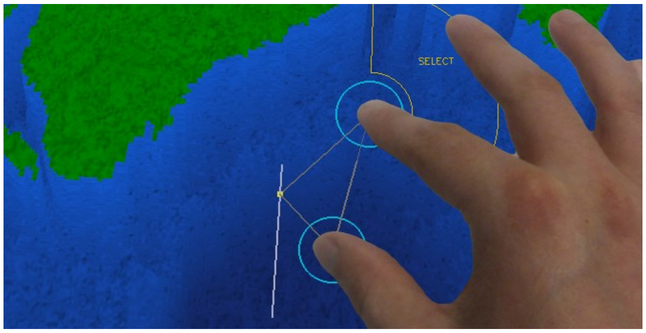

Figure 3: Along-axis cursor in an ocean model. Offset finger midpoint controls lat/long coordinates of cursor pole, and inter-finger distance determines the cursor's water depth.

3.2 Visual Feedback

The touch feedback ambiguity problem is when users experience unexpected behavior, but there's not enough feedback to determine the cause (e.g. system unresponsive, touch surface failed to detect, improper usage). Wigdor et al. [Wigdor et al. 2009] provide design recommendations to overcome touch ambiguity problems by providing effective feedback. The design of the pantograph widget incorporates sufficient visual feedback to ensure proper usage is reinforced with subtle indicators, while misuse and/or errors are disambiguated with attention-getting visual feedback.

Positive visual feedback is provided by thin circles around the fingertips. These circles clarify which fingers are the main positional controls, and indicate finger tracking functionality.

A line connects the fingertips to indicate their behavioral connection and inter-finger distance constraints: If fingers get too close together (which some touchscreens may erroneously report as a single touch), the line turns red and does not shorten further. If fingers get too far apart, the portion exceeding the maximum distance is colored red. In both cases, while the interactive elements of the widget are still tethered to the fingers, the lines connecting to the cursor are strictly attached to only the valid portion of the connecting line, enforcing the strict linkage metaphor and illustrating the constraints.

3.3 Stereoscopic Considerations

While the pantograph technique can be used monoscopically, it was specifically designed for stereoscopic 3D environments.

The lines connecting fingertips on the 2D surface to the 3D cursor provide strong stereoscopic depth cues in a smooth transition from the fingers to the virtual depth of the cursor, which helps perception of relative depth between them. Near the cursor, these lines add more visual details to what is otherwise a single point, ensuring better cursor depth perception.

Similarly, drawing axis-aligned poles adds disparity-based depth cues, and relates the cursor's location to surrounding features and boundaries. Additional lines can connect the cursor to nearby objects too, providing more relative depth information.

Unlike point cursors, volumetric cursors have enough screen real estate to add significant extra details. Making a volumetric cursor partially transparent has been shown to be effective at enhancing depth cues [Zhai et al. 1994]. Opaque details can cause occlusion, which can be counter-acted by animation. For example, detail can be added to a spherical cursor by animating a number of rings spinning on its surface. Figure 4 shows a volumetric cursor utilizing a combination of these techniques.

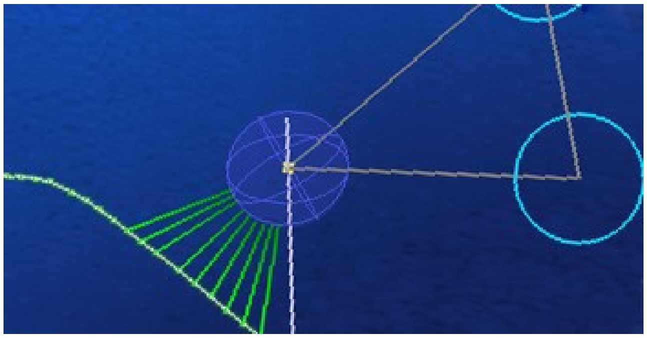

Figure 4: A transparent volumetric cursor. Green lines connect nearby waypoints, providing additional depth cues.

3.4 Selection

After the cursor is positioned satisfactorily, users need a way to signal they wish to make a selection or perform an action at the cursor's location. The onscreen widget (at zero-disparity) provides buttons for these actions, located conveniently around the user's forefinger. The buttons can be pressed with either the middle finger or the forefinger.

The middle finger is optimal for actions performed multiple times throughout a positioning task, such as adding waypoints to define a route, as it does not interrupt the positioning gesture. However, using the middle finger could lead to involuntary forefinger movements, and thus unwanted cursor movement.

Therefore, for tasks requiring precision, or for less dexterous users, the forefinger can be used instead. Lifting the forefinger automatically locks the cursor in place (as long as the thumb is held down). The forefinger can then be used without disrupting the cursor's position. Positioning is resumed by placing the forefinger back into the circle from which it was initially lifted.

The widget follows the forefinger, and automatically rotates to match the pose of the user's hand. As shown in Figure 5, this keeps the primary action button conveniently under the middle finger, and prevents elements from being obscured.

3.5 Four DOF Extensions

The angle between thumb and forefinger can independently and simultaneously control a fourth degree of freedom. While this may seem an overly complex movement, Moscovich and Hughes [Moscovich and Hughes 2008] observed that for one handed interaction, users were indeed capable of simultaneously adjusting the position, spread, and rotation of two fingers.

Fu et al. [Fu et al. 2010] used this angle to switch between navigational modes in an astrophysics simulation. Strothoff et al. [Strothoff et al. 2011] utilized it to rotate selected objects about the z-axis (yaw).

Beyond categorical mode selection and rotation, this angle can also control a continuous variable. However, because the range of inter-finger angles that can be comfortably made across the screen is fairly small (<120°), this can only be practically used to control variables with a limited range.

Figure 6 shows this technique as used to plot waypoints for underwater survey missions. A middle-finger button inserts waypoints as the cursor is moving, with the angle of the hand setting the desired speed the vehicle takes to the next waypoint. The current angle/speed value is indicated by a tick mark on an arc-shaped scale drawn above the widget.

Because it can be fatiguing to maintain a particular angle while adjusting (x,y) position and/or finger spread, a 'lock speed' button is provided to maintain the currently selected speed regardless of angular changes. Similarly a 'lock depth' button temporarily ignores changes in finger spread.

This rotational control was applied to a volumetric cursor widget for editing survey mission pathlines. As shown in Figure 7, the angle adjusts the size of the spherical cursor. This makes it easy to quickly re-adjust the cursor size as needed to select individual waypoints or entire pathline sections, without having to interrupt the positioning task.

This 4DOF mode has also been used in an astrophysics visualization [Aygar et al. 2018], where the 1st 3DOF control a 3D cursor, and the 4th DOF controls an axis through the point cloud data, around which the cloud constantly rotates to produce strong motion parallax visual cues. This allows the axis to be moved as needed to reduce occlusion and reveal more features.

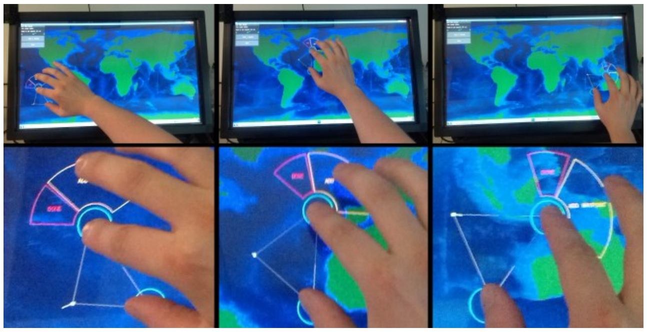

Figure 5: Hand pose changes naturally across the screen. The widget matches this rotation, keeping the primary action button under the middle finger.

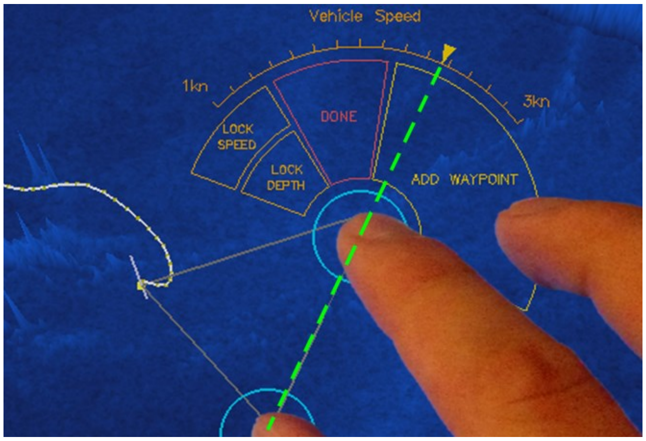

Figure 6: 4DOF widget for setting 3D navigation waypoints with variable speeds. The green line shows the inter-finger vector’s extension to a speed value on the radial scale. Lock buttons allow the 3rd and 4th DOF to be temporarily fixed.

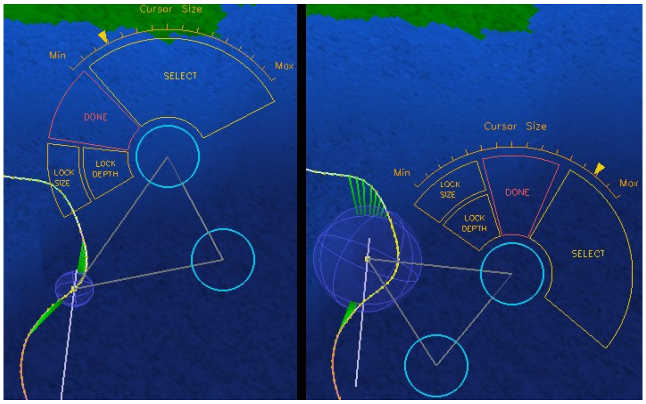

Figure 7: Selecting waypoints using a volumetric cursor with a radius controlled by hand pose / inter-finger angle.

4 Limitations

The pantograph technique has two inherent limitations: It is difficult to position the cursor near the edges of the screen, and the range of depth adjustments is relatively narrow.

Using the offset midpoint of two fingers makes it difficult or impossible to position the cursor within the offset distance of some screen edges. Corners can be particularly troublesome, as the fingers cannot encompass them.

This limitation can be addressed by dynamic offset adjustment. However, for applications where users can control the camera/viewpoint, most interactions will naturally tend to occur towards the center of the display as opposed to the periphery.

The percentage of the screen that is inaccessible due to offsets is inversely proportional to screen size. Thus, the pantograph technique is best suited for medium to large format displays, such as desktop monitors and tabletops. It is not optimal or intended for use on smaller displays, such as smartphones or tablets.

The second limitation arises from directly mapping thumb-forefinger distance to available depth range. While a narrow depth range is desirable for fish tank VR [Ware et al. 1993], this can limit depth adjustment granularity in other, relatively deeper virtual environments. For this reason, the technique is best suited for relatively shallow 3D environments and datasets with narrow height ranges (e.g. geospatial data).

There are many ways to overcome this limitation: Martinet et al. [Martinet et al. 2010a] use the speed of finger movements to dynamically adjust depth movement rates. Strothoff et al. [Strothoff et al. 2011] applied quadratic scaling to the inter-finger distance-to-height mapping, giving higher precision near the surface, but lower precision and increased reach away from it.

Another strategy is clutching (as in balloon selection [Benko and Feiner 2007] and z-technique [Martinet et al. 2010a]). Additional drag buttons could appear when inter-finger distance approaches min/max values, for adjustment beyond usual limits.

Finally, a new strategy is to narrow the range of inter-finger distances that directly map to depth adjustment (so the min/max is easily exceeded), and applying this mapping to only a fraction of the overall depth range. To move the cursor beyond the active portion of the depth range, the user would temporarily exceed the min/max inter-finger distance, moving the cursor and the active portion of the depth range in the corresponding direction.

5 Experiment 1

To determine any benefit to adjusting (x,y) and (z) coordinates simultaneously, a human factors study was conducted to evaluate the pantograph technique’s performance compared with two variations of the popular z-technique.

Fifteen subjects were shown a shallow (15cm virtual depth) 3D environment on a 1920x1080 24” stereoscopic multi-touch display. A Kinect provided head tracking for correct stereoscopic rendering. Subjects had to select spherical objects and move them into desired goal volumes. Objects and goal volumes varied in size. To ensure consistent starting poses, hands had to simultaneously touch the bottom corners of the display to start each trial.

Each participant did 60 trials, 20 with each of three gestures: pantograph, absolute z-technique (ZTA), and relative z-technique (ZTR). Both ZTA and ZTR directly map one finger’s (x,y) position to cursor (x,y), while the other finger adjusts depth: ZTA directly maps the second finger’s vertical position on the screen to cursor (z) depth, while ZTR allows clutching, i.e. the second finger can be moved vertically anywhere on screen, with cursor (z) movement matching finger movement relative to finger starting position (similar to balloon selection).

The most revealing measurement of the experiment was target acquisition time (TA), i.e. elapsed time between the trial’s start and when the cursor first touches the target. 4.8% of trials were discarded due to a TA > 10s (more than 3σ above the mean TA of 3.8s), indicating subjects had not correctly performed the trial.

The three positioning techniques each had significantly different [$F(2,752.1)=36.87$, $p<0.0001$] mean target acquisition times, with pantograph performing best, ZTA worst, and ZTR in between. Predictably, target size also had a significant impact [$F(2,752.1)=13.39$, $p<0.0001$].

An interesting observation, visible in Figure 8, is that while pantograph acquisition times decreased as target sizes increased (following Fitts' Law [Fitts 1954]), z-technique acquisition times did not continue to decrease for targets beyond the medium size condition. This implies that the pantograph gesture allows users to make faster movements when targets are larger and there is more room for error; whereas the z-techniques has a performance ceiling (due to having to adjust depth independently of (x,y) position) that restricts speed increases in cases where less-precise positioning is "good enough". This suggests that the pantograph gesture may be more apt for exploratory usage.

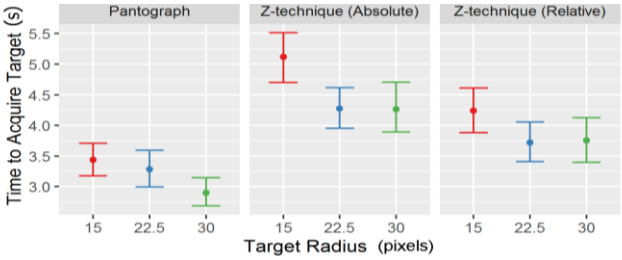

Figure 8: Target acquisition time versus interaction technique and target radius.

Subjects completed post-experiment questionnaires, in which they rated the pantograph highest for ease of use, speed, and accuracy, and lowest when asked how fatiguing each gesture was. This is not surprising, as previous research on user preferences for touch surface interaction have shown simpler is always better, with users preferring single finger and single hand gestures over whole hand and two-handed gestures [Morris et al. 2010].

The success here of the pantograph over z-technique variants is partially predictable from previous research into the matching of interaction techniques to tasks at hand. Jacob et al. [Jacob et al. 1994] present strong evidence that "performance improves when the structure of the perceptual space of a graphical interaction task mirrors that of the control space of the input device."

Jacob et al. evaluated performance in speed and accuracy, as well as by plotting trajectories of subjects' movements. Replicating this analysis, participants' movements were plotted, aligning them start to end, with one axis as distance from goal in (x,y) and distance from goal in (z) on the other axis. Tracks from the dimensionally-integrated pantograph tended to cross diagonally towards the goal, while the z-technique tracks moved in a city-block pattern: adjustments first in x,y, then in depth. This confirms findings that for tasks with 3DOF perceptual structure, integral controls will result in the three attributes being "manipulated as a unit," while a separable control will be "manipulated along each attribute in turn" [Jacob et al. 1994].

It is important to note that this relationship works both ways, as it is also possible to decrease efficiency by presenting the user with input methods having too many degrees of freedom for a particular task. Thus the need for the "lock" buttons on the 4DOF pantograph interface; to temporarily reduce the degrees of freedom when the desired task demands it (e.g. plotting a portion of a vehicle's course with constant depth).

6 Experiment 2

This experiment evaluated the relative performance of 3D positioning via multi-touch versus more-traditional positioning interfaces. The pantograph, having been shown in the previous experiment to be an effective multi-touch 3D positioning gesture, was tested against a traditional desktop mouse, with its scroll wheel adjusting cursor (z) depth, and a true-3DOF electro-magnetically tracked Polhemus 3Space Fastrak joystick.

The study environment, seen in Figure 9, presented point clouds from an astrophysics simulation. Each trial had subjects use a volumetric cursor to select sets of points along random spline structures meandering through the point cloud. This tested how quickly and efficiently subjects could move along curving pathlines in 3D space. Half of the trials were rendered monoscopically, and half stereoscopically.

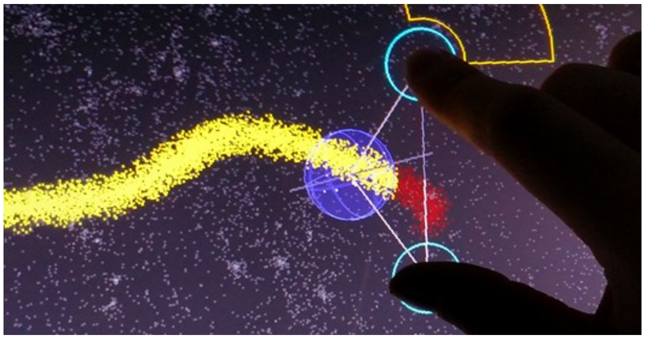

Figure 9: Selecting points with the pantograph, within the point cloud environment of Experiment 2.

The experiment was 3x2 full factorial within-subjects with repeated measures. There were three interaction conditions (mouse, pantograph, Polhemus) and two viewing conditions (monoscopic, stereoscopic), yielding six combinations. Each combination was seen 30 times, split over 3 blocks of 60 trials and randomized to mitigate any learning effects. 15 subjects (8 female, 7 male) from the authors' university completed 2700 total trials.

Before analysis, 15.7% of trials were removed due to subjects not performing the task correctly (did not start at the end of a spline, took inordinately long, etc.).

For elapsed time spent tracing the spline from end to end, interaction method had a significant main effect [$F(2,174.3)=11.25$, $p<0.0001$]. As can be seen in Figure 10, Tukey's HSD found the Polhemus and mouse performed significantly faster than the multi-touch pantograph technique.

There was also a significant effect of spline depth range [$F(1,2083)=70.99$, $p<0.0001$] and interaction between interface method and spline depth range [$F(1,2082)=4.01$, $p=0.0183$]. This is predictable, as changing cursor depth via the mouse scroll wheel is far slower and very repetitive when compared to the pinching/spreading motion of the pantograph or moving one's hand with the Polhemus.

Precision and efficiency of tracing movements was analyzed using the average minimum distance to the spline and total cursor

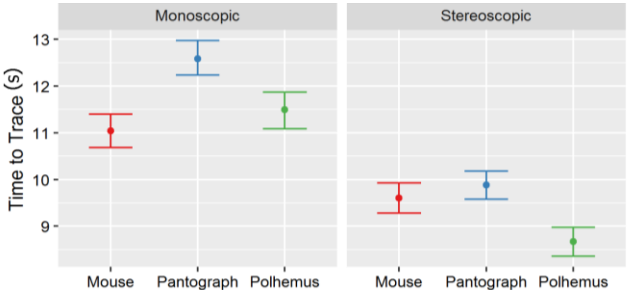

Figure 10: Time to trace vs interface and stereo.

travel. Interaction method had a significant main effect on following distance [F(2,174.7)=116.70, p<0.0001] and overall cursor travel [F(2,173.2)=487.76, p<0.0001]. In both cases, Tukey's HSD showed the mouse cursor traveled less distance and followed the spline significantly closer than the Polhemus, which was itself significantly more efficient and closer than the pantograph.

The use of stereoscopy consistently lead to significantly better performance over monoscopic 3D across all metrics: time to acquire spline, [t(177.4)=21.77, p<0.0001], time to trace spline [F(1,174)=87.4, p<0.0001], closer spline following [F(1,174.6)=83.1, p<0.0001], and shorter cursor travel [F(1,173.1)=54.40, p<0.0001]. This confirms previous research showing the benefits of stereoscopic rendering for 3D tasks, even with 2D input devices.

The results of this experiment suggest that multi-touch is not a particularly appropriate interface for sustained 3D positioning.

Despite the often frustratingly repetitive scrolling of the wheel, the standard desktop mouse performed the spline tracing task faster, more precisely, and more efficiently. This is likely due to a number of factors including user familiarity, not having to hold one's arm out, and the steadying effect of resting on the desktop. The true-3DOF Polhemus device also performed well, which is not surprising, as its dimensionality matches the task.

The only metric in which the pantograph excelled was time to first acquire the spline with the cursor. As shown in Figure 11, interaction method had a significant main effect [F(2,178.6)=8.13, p=0.0004], with Tukey's HSD finding the pantograph to be significantly faster than mouse or Polhemus at p<0.05. This indicates that while the pantograph technique may not be apt for sustained positioning, as encountered in a tracing task, it may be more suited for quickly selecting single locations, as it appears to be better at starting with the cursor near the desired depth.

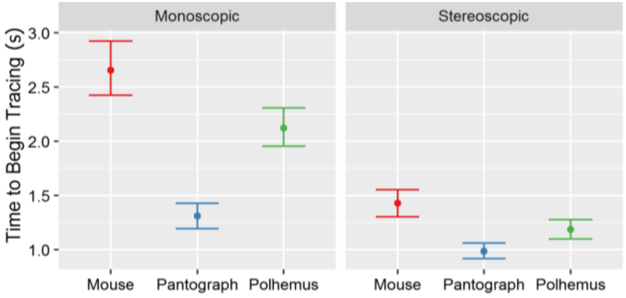

Figure 11: Time to begin tracing vs interface and stereo.

7 Conclusion

Multi-touch displays have been integrated into a variety of interactive systems, with the ability to directly manipulate onscreen elements being a major selling point. However, when attempting to extend touch interaction into 3D applications, all but the shallowest virtual environments move out of reach. Stereoscopic 3D systems in particular pose even more issues, as accurately touching the region on a touchscreen corresponding to an object at a different virtual depth is challenging, and parallax issues can further render existing strategies futile.

Making multi-touch work in 3D applications requires techniques which interface with the user at the depth of the touch-surface, while extending their actions out into the depths of the virtual environment. This approach can harness the direct manipulation concept, and maintain the user's sense of being "in contact" with the interface, while performing actions at a distance in a manner that is still perceived as being direct.

Accomplishing this successfully requires significant attention to the issues surrounding depth perception as well as the complex human factors and ergonomic issues involved with touch interaction. A survey of existing techniques related to multi-touch3D positioning confirms the importance of addressing these issues, and the performance and usability gains that can be realized by doing so.

The pantograph technique allows for fast and precise positioning in mono- and stereoscopic 3D environments. The first experiment presented here showed that the technique's ability to simultaneous adjust a cursor's (x,y) and (z) position in a single movement permits better performance than existing techniques which discretely split (x,y) and (z) adjustments to different movements and/or fingers, reinforcing existing research findings [Benko and Feiner 2007].

The second experiment presented here demonstrated that, although the pantograph technique is an effective method for 3D positioning via multi-touch, a multi-touch interface is still not a great input device for 3D applications; as a specialized 3DOF device and surprisingly even a standard desktop mouse performed faster with more precision for most metrics. It also confirmed the well-known benefits of adding stereoscopy to 3D applications.

Continuing to develop and improve upon similar general purpose multi-touch 3D interaction techniques has the potential to help 3D multi-touch systems transition from an awkward combination found only in research environments to a powerful and possibly ubiquitous interactive technology.

ACKNOWLEDGMENTS

This work was supported by NOAA grants NA10NOS4000073 and NA15NOS4000200.

REFERENCES

Erol Aygar, Colin Ware, and David Rogers. 2018. The Contribution of Stereoscopic and Motion Depth Cues to the Perception of Structures in 3D Point Clouds. *ACM Transactions on Applied Perception (TAP)* 15, 2 (2018), 9.

Hrvoje Benko and Steven Feiner. 2007. Balloon Selection: A Multi-finger Technique for Accurate Low-Fatigue 3D Selection. In *3D User Interfaces, 2007. 3DUI'07. IEEE Symposium on*. IEEE.

Hrvoje Benko, Andrew D Wilson, and Patrick Baudisch. 2006. Precise Selection Techniques for Multi-Touch Screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1263–1272.

Thomas Butkiewicz. 2012. A More Flexible Approach to Utilizing Depth Cameras for Hand & Touch Interaction. *International Journal of Virtual Reality (IJVR)* 11, 3 (2012), 53–57.

Thomas Butkiewicz and Colin Ware. 2011. Multi-Touch 3D Exploratory Analysis of Ocean Flow Models. In *OCEANS 2011*. IEEE, 1–10.

Stuart K Card, Jock D Mackinlay, and George G Robertson. 1991. A Morphological Analysis of the Design Space of Input Devices. *ACM Transactions on Information Systems (TOIS)* 9, 2 (1991), 99–122.

Chi Tai Dang, Martin Straub, and Elisabeth André. 2009. Hand Distinction for MultiTouch Tabletop Interaction. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ACM, 101–108.

Florian Echtler, Manuel Huber, and Gudrun Klinker. 2008. Shadow Tracking on MultiTouch Tables. In *Proceedings of the Working Conference on Advanced Visual Interfaces*. ACM, 388–391.

Paul M Fitts. 1954. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology* 47, 6 (1954), 381.

Chi-Wing Fu, Wooi-Boon Goh, and Junxiang Allen Ng. 2010. Multi-Touch Techniques for Exploring Large-Scale 3D Astrophysical Simulations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2213–2222.

Martin Hachet, Benoit Bossavit, Aurélie Cohé, and Jean-Baptiste de la Rivière. 2011. Toucheo: Multitouch and Stereo Combined in a Seamless Workspace. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. ACM, 587–592.

Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. 2007. Shallow-Depth 3D Interaction: Design and Evaluation of One-, Two- and Three-Touch Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1147–1156.

Mark Hancock, Thomas Ten Cate, and Sheelagh Carpendale. 2009. Sticky Tools: Full 6DOF Force-Based Interaction for Multi-Touch Tables. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ACM, 133–140.

David M Hoffman, Ahna R Girshick, Kurt Akeley, and Martin S Banks. 2008. Vergence–Accommodation Conflicts Hinder Visual Performance and Cause Visual Fatigue. *Journal of Vision* 8, 3 (2008), 33.

Robert JK Jacob, Linda E Sibert, Daniel C McFarlane, and M Preston Mullen Jr. 1994. Integrality and Separability of Input Devices. *ACM Transactions on Computer-Human Interaction (TOCHI)* 1, 1 (1994), 3–26.

David López, Lora Oehlberg, Candemir Doger, and Tobias Isenberg. 2016. Towards an Understanding of Mobile Touch Navigation in a Stereoscopic Viewing Environment for 3D Data Exploration. *IEEE Transactions on Visualization and Computer Graphics* 22, 5 (2016), 1616–1629.

Anthony Martinet, Gery Casiez, and Laurent Grisoni. 2010a. The design and evaluation of 3D positioning techniques for multi-touch displays. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*. IEEE, 115–118.

Anthony Martinet, Géry Casiez, and Laurent Grisoni. 2010b. The Effect of DOF Separation in 3D Manipulation Tasks with Multi-Touch Displays. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*. ACM, 111–118.

Meredith Ringel Morris, Jacob O Wobbrock, and Andrew D Wilson. 2010. Understanding Users' Preferences for Surface Gestures. In *Proceedings of Graphics Interface 2010*. Canadian Information Processing Society, 261–268.

Tomer Moscovich and John F Hughes. 2008. Indirect Mappings of Multi-Touch Input Using One and Two Hands. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1275–1284.

Jeffrey S Pierce, Andrew S Forsberg, Matthew J Conway, Seung Hong, Robert C Zeleznik, and Mark R Mine. 1997. Image Plane Interaction Techniques in 3D Immersive Environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*. ACM, 39–ff.

Richard L Potter, Linda J Weldon, and Ben Shneiderman. 1988. Improving the Accuracy of Touch Screens: An Experimental Evaluation of Three Strategies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 27–32.

Johannes Schöning, Frank Steinicke, Antonio Krüger, Klaus Hinrichs, and Dimitar Valkov. 2009. Bimanual Interaction with Interscopic Multi-Touch Surfaces. In *IFIP Conference on Human-Computer Interaction*. Springer, 40–53.

Adalberto L Simeone. 2016. Indirect Touch Manipulation for Interaction with Stereoscopic Displays. In *3D User Interfaces (3DUI), 2016 IEEE Symposium on*. IEEE, 13–22.

Adalberto L Simeone and Hans Gellerseny. 2015. Comparing Indirect and Direct Touch in a Stereoscopic Interaction Task. In *3D User Interfaces (3DUI), 2015 IEEE Symposium on*. IEEE, 105–108.

Frank Stenicke, Timo Ropinski, Gerd Bruder, and Klaus Hinrichs. 2007. Interscopic User Interface Concepts for Fish Tank Virtual Reality Systems. In *Virtual Reality Conference, 2007*. VR'07. IEEE, 27–34.

Sven Strothoff, Dimitar Valkov, and Klaus Hinrichs. 2011. Triangle Cursor: Interactions with Objects Above the Tabletop. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ACM, 111–119.

Dan Venolia. 1993. Facile 3D Direct Manipulation. In *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*. ACM, 31–36.

Colin Ware, Kevin Arthur, and Kellogg S Booth. 1993. Fish Tank Virtual Reality. In *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*. ACM, 37–42.

Colin Ware and Ravin Balakrishnan. 1994. Reaching for Objects in VR Displays: Lag and Frame Rate. *ACM Transactions on Computer-Human Interaction (TOCHI)* 1, 4 (1994), 331–356.

Colin Ware and Danny R Jessome. 1988. Using the Bat: A Six-Dimensional Mouse for Object Placement. *IEEE Computer Graphics and Applications 8, 6 (1988)*, 65–70.

Daniel Wigdor, Sarah Williams, Michael Cronin, Robert Levy, Katie White, Maxim Mazeev, and Hrvoje Benko. 2009. Ripples: Utilizing Per-Contact Visualizations to Improve User Interaction with Touch Displays. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. ACM, 3–12.

Shumin Zhai, William Buxton, and Paul Milgram. 1994. The "Silk Cursor": Investigating Transparency for 3D Target Acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 459–464.