FingerGlass: Efficient Multiscale Interaction on Multitouch Screens

Dominik Käser[1,2,4]
dpk@pixar.com

Maneesh Agrawala[1]
maneesh@eecs.berkeley.edu

Mark Pauly[3]
mark.pauly@epfl.ch

[1] University of California
Berkeley, CA 94720
United States

[2] ETH
8006 Zürich
Switzerland

[3] EPFL
1015 Lausanne
Switzerland

[4] Pixar Animation Studios
Emeryville, CA 94608
United States

ABSTRACT

Many tasks in graphical user interfaces require users to interact with elements at various levels of precision. We present *FingerGlass*, a bimanual technique designed to improve the precision of graphical tasks on multitouch screens. It enables users to quickly navigate to different locations and across multiple scales of a scene using a single hand. The other hand can simultaneously interact with objects in the scene. Unlike traditional pan-zoom interfaces, FingerGlass retains contextual information during the interaction. We evaluated our technique in the context of precise object selection and translation and found that FingerGlass significantly outperforms three state-of-the-art baseline techniques in both objective and subjective measurements: users acquired and translated targets more than 50% faster than with the second-best technique in our experiment.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces (D.2.2, H.1.2, I.3.6)—*Interaction Styles*

General Terms

Design, Performance

INTRODUCTION

We interact with our environment in many different scales. For example, creating a painting requires us to work on its global composition as well as its finest details. The physical world provides a natural way of transitioning between different scales by allowing us to move our viewpoint towards or away from our objects of interest.

Some computer applications operate on virtual scenes or artboards. Examples are graphical content creation systems such as Adobe Illustrator or map browsing tools like Google Maps. In contrast to the physical world, their user interfaces are limited in size and resolution and therefore encompass a small range of scales. Such systems typically overcome this limitation by providing zoom controls that redefine the current viewport. However, when a user performs a zoom, there is a loss of contextual information. In addition, the input of such tools has usually been limited to single-point input devices such as the mouse, and repeatedly switching back and forth between navigation and interaction with the same input device is time-consuming.

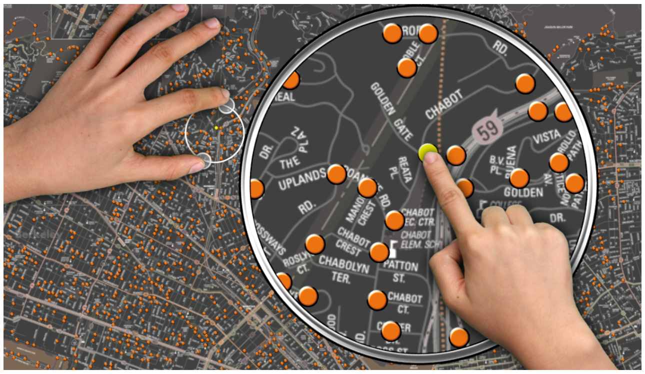

Figure 1: Interaction with *FingerGlass*: The user specifies an area of interest with one hand and interacts with the magnified objects with the other hand. During the interaction, a new area of interest can be defined. Releasing all fingers makes the tool vanish.

Multitouch workstations provide more degrees of freedom than single-point input devices. They also reduce the mental effort required for interacting with virtual objects by removing the indirection of an external pointing device. These advantages come at the cost of screen occlusion and reduced precision. Nonetheless, Forlines et al. [13] showed that touch-based devices can achieve faster task completion times and comparable error rates to a mouse, given sufficiently large targets.

The precise selection of small on-screen targets has been well-studied. However, with the recent advent of multitouch-based content creation applications such as Eden [22], we require tools for more complex interactions than just selection. In this work, we focus on developing a more general technique enabling users to quickly navigate through the space of potential viewports while selecting and translating targets.

We propose *FingerGlass*, a technique that lets the user define a viewport using one hand. The other hand can simultaneously interact with objects in the scene. At all times, the contents of this viewport are shown twice on the screen: once in a global zoomed-out view stretched out across the entire screen, retaining contextual information, and once as

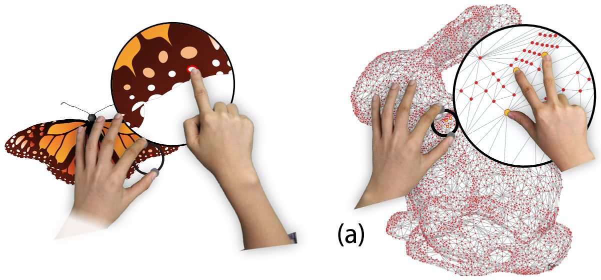

Figure 2: Two fictional example applications employing *FingerGlass*: (a) Translation of a 2D shape in a vector graphics application. (b) Translation of vertices in a multitouch 3D modeling system.

a magnified copy on top of the zoomed-out view. We call the latter the *magnified view*. Any interaction with objects in the scene takes place in the magnified view. This way, fingertips do not occlude the area of interest in the zoomed-out view. Figures 1 and 2 show sample applications of FingerGlass.

We evaluated FingerGlass and its variant, *FingerMap*, in the context of precise object selection and translation. Our formal user study shows that FingerGlass significantly outperforms three state-of-the-art techniques: with FingerGlass, users acquired targets more than 50% faster than with PrecisionGlass, the second-fastest technique in our experiment. Translation times were between 40% and 100% faster than the second-fastest technique for distances between 2.7mm and 350mm. Users also subjectively preferred using FingerGlass than the other techniques.

Participants also responded positively to the ergonomics of our tool: FingerGlass requires only individual taps to perform precise selection. For translations, the amount of physical dragging is limited to local corrections.

RELATED WORK

Two distinct areas of research are relevant for our work: multiscale navigation and interaction with virtual scenes, and precise target interaction on (multi-)touch screens.

Multiscale Navigation and Interaction

For the exploration of two-dimensional scenes, researchers developed zoomable user interfaces (ZUIs) such as *Pad++* [4] or *Speed-Dependent Zooming* [20]. However, ZUIs inherently suffer from context loss and are hard to navigate once zoomed in. Researchers have addressed this in a variety of ways: Pook et al. [27] and Hornbaek et al. [19] provide global context information in small overviews or extra layers. Other focus+context approaches include *Fisheye Views* [14], *Perspective Wall* [23], *Document Lens* [30], *Melange* [8] or *High-Precision Magnification Lenses* [2]. *DTLens* [12] is a similar technique for multitouch screens. Furnas and Bederson introduced the *Space-Scale Diagram* [15], an analytical framework for analyzing multiscale navigation interfaces such as the ones discussed here. None of these systems consider manipulations of the scene but are limited to navigation. *Magic Lenses* [6] allow users to place virtual lenses using one hand and interact with the scene using the other hand. However, they are not designed to be fast and there are no time measurements provided.

Precise Touch-Screen Interaction

Precision and occlusion issues associated with touch-based input devices are an active topic of research. Researchers have evaluated their solutions on the task of target selection. *Offset Cursor* [28] was the first technique addressing occlusion by remapping the physical touch location to a non-occluded screen location. *ThumbSpace* [21] and *Dual Finger Offset* [5] extend this concept. Instead of remapping the touch location, *Shift* [34] displays an offset copy of the occluded area. While these techniques address occlusion, they do not increase precision.

To address precision, *PrecisionHandle* [1] enhances the control-display ratio by giving the user a lever-like widget at hand. Correctly placing this widget requires additional time and taps. A simpler technique doubling the control-display ratio is *Dual Finger Midpoint* [5]. Researchers have proposed a number of content-sensitive techniques to solve input precision problems. *Bubble cursor* [16], *MagStick* [32] and *Semantic pointing* [7] increase the target size in motor space. *Escape* [35] assigns different directional vectors to targets lying next to each other. To select a target, the user swipes over it in a target-specific direction. *Sliding Widgets* [25] provide a generalization of this technique to a wider scope of widgets. *Enhanced Area Cursors* [10] let the user specify a coarse area in the first phase. Then, in a second phase, they invoke an angular menu containing all available targets in this area.

All these techniques are designed for selection tasks: after the initial touch, the user performs corrective dragging movements or disambiguation gestures. Selection is then usually triggered by releasing the finger. This makes extensions to translations or complex multi-finger gestures nontrivial. A few techniques remedy this problem for single-point dragging operations by offering both a tracking and a dragging state: *DTMouse* [9] works similarly to Dual Finger Midpoint [5], and *Pointing Lenses* [29] are pen-based precision enhancement techniques. In addition to being limited to single-point, they still require corrective dragging movements which are relatively slow to perform and can be uncomfortable. Furthermore, none of the techniques above serves for visual exploration.

Visual exploration is addressed by *Zoom Pointing* [1], *TapTap* [32] and *Rubbing and Tapping* [26] which let the user redefine their viewport by zooming. *Dual Finger Stretch* by Benko et al. [5] gives the user the option to perform unaided target selection, and to summon an in-place magnifying glass using a second finger to facilitate corrective movements. The technique by Mankoff et al. [24] displays an in-place magnification if and only if the first tap was ambiguous. All these techniques suffer from a loss of context after zooming in since the original area of interest is occluded.

Researchers evaluated the above techniques only on target selection, but not on translation tasks which comprise a large part of our interaction with GUIs. Context-sensitive techniques to facilitate dragging operations over long distances are *Drag-and-Pop* and *Drag-and-Pick* [3]. These techniques distort geometrical relationships and assume that there is a small discrete set of possible destinations for a scene object.

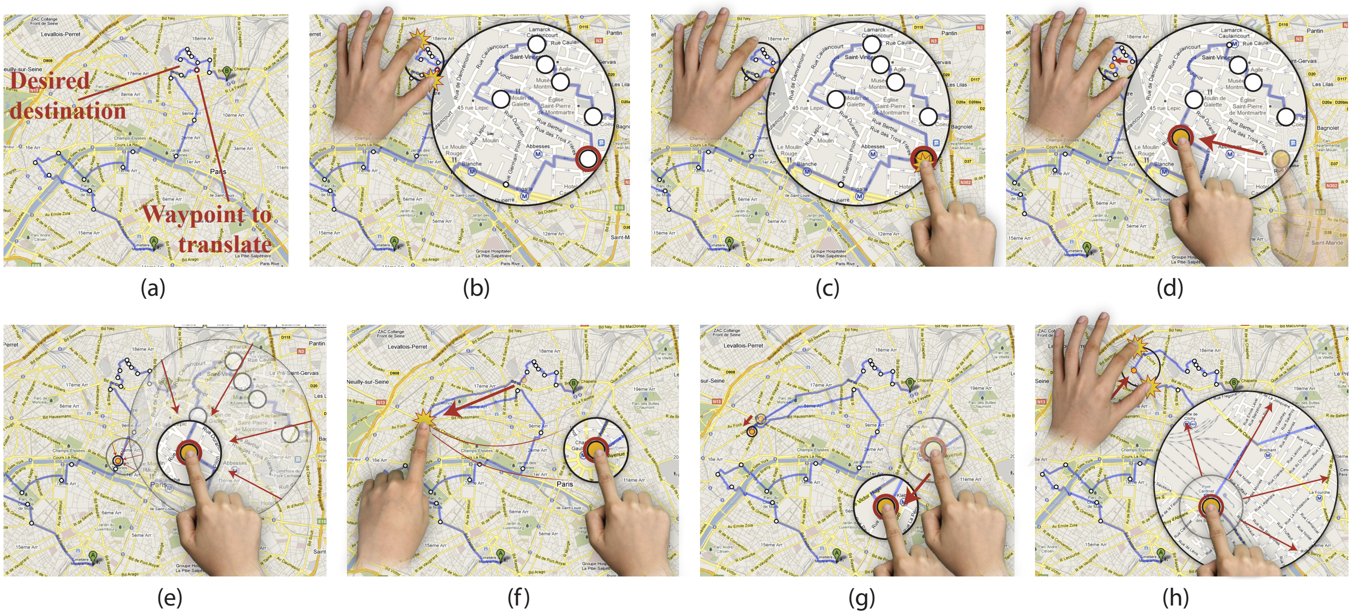

Figure 3: Interaction with Google Maps using *FingerGlass*: (a) Task description. (b) Once the user defines the area of interest, the magnified view appears. (c) The user selects the waypoint in the magnified view and (d) translates it. (e) After releasing the coarse hand, the magnified view shrinks. (f) If the user taps at a location in the original viewport, FingerGlass translates the waypoint to this location. (g) FingerGlass applies any translation of the fine hand to the waypoint on a smaller scale. (h) If the user specifies a new area of interest, the magnified view grows again.

FINGERGLASS

Figure 3 shows a walkthrough of *FingerGlass* in a trip planning application: the user would like to move the marked waypoint (Figure 3a) to a different street intersection in the same neighborhood. At the initial scale the waypoint is too small to be selected by a finger, and street names are not visible. To get a close-up view of the waypoint and its surroundings, the user touches the screen with two fingers. Their tips span a circle which we call the *area of interest*. FingerGlass immediately displays an enlarged copy of this area which we call the *magnified view* (Figure 3b). We call the fingers spanning the area of interest *defining fingers*, their hand *coarse hand* and the other hand *fine hand*. Based on Guiard's studies on human bimanual action [17], we suggest that users use their non-dominant hand as the coarse hand.

The magnification ratio of the magnified view is prescribed by the application. Developers are advised to use a ratio that enlarges the smallest pickable targets to about the size of a fingertip. If the magnification ratio is too large, small translations require too much physical movement by the fine hand. If it is too small, selection of small targets can be difficult. For our street map application, ratios between 4x and 6x have worked well. The magnified view is always tangent to the area of interest. A placement algorithm determines its position such that the prescribed zoom ratio can be achieved as closely as possible and that the fine hand can comfortably interact with its contents. As the user moves his coarse hand, the magnified view follows. Once the user releases his coarse hand, the magnified view vanishes.

To translate the waypoint, the user touches it inside the magnified view (Figure 3c) with a finger on his fine hand. He then drags it to the desired destination (Figure 3d). FingerGlass translates the waypoint in the zoomed-out view accordingly in real-time. This behavior allows users to focus on the original viewport during dragging operations in order to judge their effect in a global context. While the fine hand is translating objects in the magnified view, the area of interest is locked and any movement of the coarse hand has no effect.

The destination of the waypoint might lie outside the current area of interest. The user can release his defining fingers while retaining the finger of the fine hand on the screen. The magnified view then centers around the selected waypoint and shrinks down to a size encompassing just the object's immediate surroundings. The area of interest shrinks accordingly to maintain the zoom ratio (Figure 3e).

A finger of the coarse hand then taps on the desired destination in the original viewport. The selected waypoint immediately moves to the location of the tap (Figure 3f). The fine hand can then refine its position: FingerGlass applies any movement of the fine hand to the waypoint, scaled down by the magnification ratio. The magnified view follows the finger and its content continues to display the current neighborhood of the waypoint (Figure 3g). If the desired destination in the original viewport is occluded by the magnified view, the user can first move the magnified view by dragging his fine hand, and then tap at the desired destination.

The user might want to explore the neighborhood of the desired destination before finishing the translation. To do so, he defines a new area of interest by pressing and holding two fingers of his coarse hand. The magnified view then grows again to accomodate the size of the new area of interest while maintaining the zoom ratio. FingerGlass translates

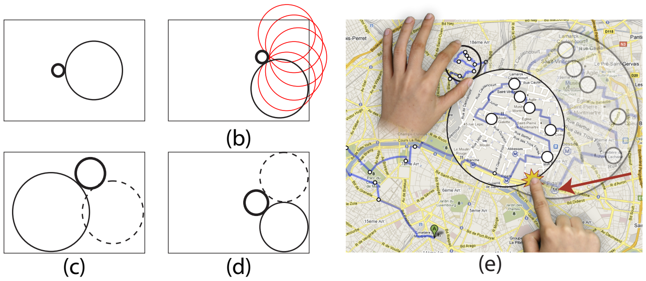

Figure 4: (a)-(d) Placement of the magnified view, assuming a right-handed user: (a) Optimal placement. (b) FingerGlass searches for compatible configurations along the border of the area of interest. (c) Tradeoff between shrinking the magnified view or moving it further away. (d) Ambiguous configuration. (e) By touching the screen outside of the magnified view, the magnified view can be manually defined.

the waypoint to the new area of interest. To determine the exact location of the waypoint, we use its relative position at the time before the magnified view shrunk (Figure 3h). The interaction then continues as in Figure 3d, except that the area of interest and the magnified view are detached. The translation operation ends once both hands are released.

Automatic Placement of the Magnified View

In this section, we assume a right-handed user. We noted in the previous section that FingerGlass computes a suitable size and location for the magnified view. We developed an optimization algorithm with the following three goals.

- **Minimal eye movement**: In the course of an interaction, the eye has to travel from the area of interest to the magnified view and back. Thus, FingerGlass only considers configurations in which the magnified view is adjacent to the area of interest.
- **Comfort for fine hand**: In our setup, we configured FingerGlass to place the magnified view as far to the right as possible. The right hand can then comfortably interact with objects in the magnified view while the left hand specifies the area of interest.
- **Usage of prescribed magnification ratio**: Developers provide a recommended magnification ratio with their application. FingerGlass will try to use this ratio if possible, but resort to smaller ratios if necessary.

Figures 4a-d show areas of interest in different scenarios, and the magnified view as determined for a right-handed user by our algorithm. Without any boundary restrictions, FingerGlass places the magnified view to the right of the area of interest (Figure 4a). If the magnified view would thereby extend beyond the screen boundaries, we call this location *incompatible*. FingerGlass then searches for compatible locations along the boundary of the area of interest (Figure 4b). Our system also tries to shrink the magnified view to obtain more compatible locations. There can be a tradeoff between reducing the magnification ratio and shifting the magnified view further to the left (Figure 4c). We control this tradeoff with a parameter in our code. If there are multiple locations with very similar qualities, a term for temporal coherence prevents the magnified view from jumping back and forth (Figure 4d).

Manual Adjustment of the Magnified View

In some cases, the user may not be satisfied with the placement of the magnified view. For example, the placement algorithm is unaware of the current position of the fine hand as it hovers over the screen. Hence, there may be cases in which the magnified view opens up in a place that would require the fine hand to travel a large distance or to make uncomfortable movements. In other cases, the user may want to employ a different magnification ratio than the prescribed one. For example, the scene may contain objects with a wide variety of sizes, and small objects need a higher magnification ratio than large ones.

FingerGlass allows users to redefine the size and location of the magnified view: once the user touches the screen outside of the current magnified view, FingerGlass will reposition the magnified view such that it lies exactly between the area of interest and the touching finger (see Figure 4e).

FINGERMAP

FingerGlass occludes a significant portion of the screen and requires users to shift their attention to the magnified view to interact with scene objects. Although we minimized the required eye movement with careful placement of the magnified view, performing many interactions could still lead to fatigue. *FingerMap* is an alternate design without magnified views which follows the interaction model underlying FingerGlass as closely as possible. FingerMap is optimized for situations in which the user wants to maintain his focus on the area of interest at all times.

Figure 5 shows an abridged walkthrough of FingerMap in the same trip planning task we used in Figure 3. As with FingerGlass, the user specifies the area of interest with his coarse hand. Then he touches the screen with a finger of his fine hand anywhere outside of the area of interest. We call this finger *tracking finger*. FingerMap then displays a selection cursor at the center of the area of interest (Figure 5a). The shape of the cursor is a cross surrounded by a circle, and its radius is the radius of an average fingertip (e.g. 10mm), scaled down by the magnification ratio. As an extension, we made the magnification ratio dependent on the distance from the initial touch of the tracking finger to the area of interest: a larger distance leads to a bigger magnification ratio.

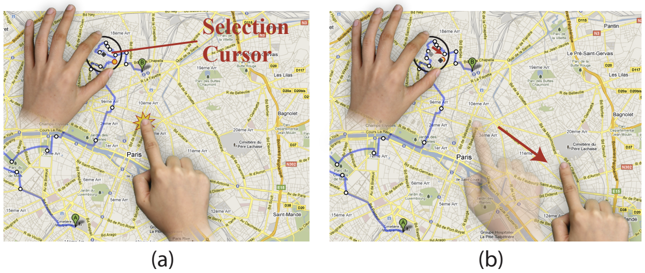

Figure 5: Interaction with *FingerMap*: The coarse hand specifies an area of interest. (a) The user touches the zoomed-out view with the tracking finger, and a selection cursor appears in the center of the area of interest. (b) FingerMap applies any translation of the fine hand to the cursor on a smaller scale. Once the cursor overlaps with the desired waypoint, the user selects it by releasing the coarse hand.

As long as the tracking finger remains pressed, it operates as an indirect control for the selection cursor: FingerMap applies any movement of the tracking finger to the selection cursor, scaled down by the magnification ratio (Figure 5b). Once the cursor overlaps with the desired waypoint, the user selects it by releasing the coarse hand. To translate the waypoint, the user keeps the tracking finger pressed. This finger then indirectly controls the position of the selected waypoint, scaled down by the magnification ratio. The remaining interaction works analogously to FingerGlass in Figures 3f-h.

The initial world space position of the tracking finger is always at the center of the area of interest. To minimize corrective dragging movements, the user should choose the area of interest such that its center is as close as possible to the desired target. This makes target selection using FingerMap somewhat similar to *Dual Finger Midpoint* [5], but adds small-scale corrective movements. Pilot testers of our system stated that acquiring targets using FingerMap feels similar to capturing them with a lasso.

DESIGN PRINCIPLES

Based on the techniques presented above, we establish design principles for bimanual interaction techniques on touch screens that enable users to efficiently navigate to objects and manipulate them across different scales. Our principles should be general enough for various types of multiscale interaction scenarios. Most touch devices report both the time stamp and the two-dimensional location of touch events. These events comprise, at a minimum, *press*, *move* and *release*. In order to design for a wide range of multitouch systems, we will not make use of any other input information. In particular, we do not use pressure [31] or touch area [5] information to implement a tracking state.

Bimanual Interaction

(P1) *The nondominant hand should set the reference frame for subsequent actions performed by the dominant hand.*

In his study on the division of labor in human bimanual action, Guiard [17] noted that the two hands assume very different roles and strongly depend on each other in the vast majority of everyday tasks. The nondominant hand defines a frame of reference and performs coarse granularity actions. Subsequently, the dominant hand performs fine grain interactions within this frame. Hinkley et al. [18] showed that this concept is applicable to human interaction with digital user interfaces as well. More recently, Schmidt et al. [33] used this insight to let the non-dominant hand define user-specific private areas on multi-user touch screens for dexterous interactions by the dominant hand.

Redefining Viewports

(P2) *When operating on a small scale, the user should be able to quickly redefine the viewport to any other location.*

Many target selection tools in the literature provide an increased zoom ratio to enhance precision and to facilitate selection of small targets. However, this approach is not sufficient for tasks that go beyond target selection. A large zoom ratio implies that the tool maps fingertip positions from a large screen area (domain) to a much smaller area (range). The choice of this mapping is crucial for translation tasks: if the range does not contain both target and destination, a translation can not be completed without altering the mapping. Such a change is slow and the user needs to reorient himself. Yet, if target and destination are far apart, this change may be necessary to keep the zoom ratio large and should be well supported by the tool. For example, FingerGlass lets the user specify the range and determines a well-suited domain that does not intersect with the range.

From Selection To Translation

(P3) *Once an object is acquired, no further events should be necessary to start transforming it.*

Translation operations with objects in graphial user interfaces are a two-stage process. In the first stage, the user specifies the object of interest. In the second stage, he continuously translates this object. The transition into this second stage can be seamless: some mouse-based interfaces allow users to hit an object and to immediately transform it by translating the mouse. Other interfaces may require the user to first release the input controller in order to complete the selection process before the transformation can begin. For an efficient interaction technique, we suggest a seamless transition from the selection to the transformation phase.

Ambiguities

(P4) *Contact area interaction should be used for acquisition tasks. No target should be acquired in ambiguous cases.*

Moscovich [25] pointed out that direct screen touches by a fingertip should not be interpreted as if only one single pixel was touched. Doing so would ignore ambiguities and inappropriately resolve them in an unpredictable way. Rather than selecting one single point in a somewhat arbitrary fashion when multiple points are touched at once, the system should perform no selection and indicate all points under the finger to the user. Thus, the user can navigate to a smaller scale and retry. This strategy addresses our goal of designing a tool with minimal error rates. Another advantage of contact area interaction is that the effective width of a target is increased, making selection easier in scenes with a sparse distribution of small objects.

EXPERIMENTAL EVALUATION

Our design process aimed for an efficient tool to navigate through virtual scenes and to select and translate objects. Navigation is a task which is difficult to quantify and to formally evaluate. However, the feedback from our pilot testers using FingerGlass in a map browsing application showed that our technique is a very promising alternative to the existing baseline tools for navigation.

For the selection and translation of targets, we conducted a lab experiment in which we measured the performance of participants for every technique.

Participants also answered a questionnaire in which they subjectively ranked the techniques according to different criteria such as personal preference or ease of learning. In addition, they were asked to compare a multitouch screen employing their favorite technique to a pen- and a mouse-based interface. The questionnaire also contained some space for written comments and suggestions.

Release-Tapping

We compared FingerGlass and FingerMap to three existing techniques from the literature. However, while they all facilitate precise target selection, none of them support subsequent translation: they let users first approach the desired target by corrective dragging movements and then complete the selection task by releasing their finger. This behavior is not extensible to subsequent translation operations without violating principle (P3). However, since our device could not sense pressure or touch area to introduce a tracking state, we had to implement the baseline techniques in a way that employs an additional discrete event to start the translation.

To this end, we created a technique we call *ReleaseTapping* (RT) that works as follows: once the user releases his finger, the system displays a semi-transparent solid circle around the selected target. This circle remains on-screen for a given time before it disappears. In order to translate the target, the user can touch this circle, keep his finger pressed and translate the target by dragging. For the radius of the circle, we used the radius of the target object plus 20mm. Regardless of whether or not the user hits the circle, tapping on the screen makes the current circle vanish immediately. We measured the time users spent for performing RT.

Baseline Techniques

In this section, we discuss the three techniques to which we compared our tools in the study. For fair comparison, we extended these techniques as follows:

- *Dual Finger Stretch* [5]: The user specifies an initial anchor location by pressing and holding his primary finger. Then he uses a secondary finger to scale a screen portion around the anchor location. The zoom ratio is proportional to the distance between the two fingers. We added Release-Tapping: performing RT with the primary finger starts the translation. Releasing the secondary finger makes the scaled portion vanish, and the user can enlarge new screen portions. The translation ends once the user releases the primary finger.

- *Shift* [34]: Once a finger touches the screen, the system displays a call-out copy of the occluded region with a cursor denoting the current finger position. The user can then refine this position by dragging. We added Release-Tapping: performing RT will start translation. The authors of Shift discuss CD gain as an extension, hence we added this functionality in the spirit of Dual Finger Slider [5]: by touching the screen with a second finger and dragging towards or away from the primary finger, Shift will magnify or de-magnify the offset view and modify the CD ratio accordingly.

- *PrecisionGlass*, a variation of PrecisionHandle [1]: The original technique enables the user to deploy a virtual handle on the screen. Any translation performed by the finger at the end of the handle will be applied on a smaller scale at the tip, thus increasing precision. Since the other two techniques offer visual magnification, we altered the technique to display a magnifying glass instead of a handle. After deploying the magnifying glass on the screen, it will remain on the screen for one second. During this second, the user can press and hold a target to start the translation. Our pilot studies showed that PrecisionGlass performed better than the original PrecisionHandle. As with our version of Shift, the user can change the zoom and CD ratio using a secondary finger.

Task and Stimuli

We asked participants to complete a series of target translation tasks with all 5 techniques – *FingerGlass*, *FingerMap*, *Dual Finger Stretch*, *Shift*, and *PrecisionGlass*. Depending on the current technique, each task consisted of two or three phases. Initially, the system presented two circular targets of width 7 pixels (2.0 mm) on the screen, separated by a given distance. The first touch event then started the *acquisition phase*, during which participants had to acquire the yellow *source target* as quickly and accurately as possible. For Dual Finger Stretch and Shift, this was followed by the *Release-Tapping phase*. Finally, during the *translation phase*, participants had to translate the selected yellow target onto the blue *destination target*.

We considered the acquisition phase successful if the user acquired the correct target. In the case of failure, the system did not proceed to subsequent phases. Instead, it presented a new pair of targets and the acquisition phase was repeated with the same parameters until successfully completed. The translation phase was considered successful if the source target and the destination target overlapped after releasing. In the case of failure, the entire task was repeated with the same parameters until successful completion.

At the beginning of each phase, a shrinking circle visually highlighted the corresponding target. The system displayed targets in front of a high-resolution street map in order to facilitate orientation in magnified views. In addition to the two targets in the task definition, 1500 red distractor targets were distributed uniformly at random across the screen. These distractors made it impossible to reliably acquire targets without assistive tools.

Apparatus

The experimental apparatus was *M2256PW*, a prototype 22” LCD touch screen manufactured by 3M. Its active display area is 476 x 295 mm, running at 1680 x 1050 pixels resolution with a pixel size of 0.28 x 0.28 mm. Our experiment used an area of 1371 x 914 pixels (388 x 257 mm) for the scene interaction, the remaining space was reserved for feedback about the completed tasks (see Figure 6). The refresh rate of the screen was set to 59 Hz. Participants were allowed to choose a tilt angle and desk height that was comfortable for them.

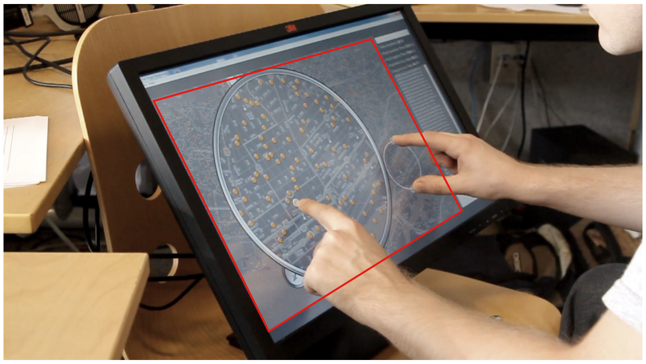

Figure 6: Our user study took place on a commercially available and freely tiltable 22” multitouch screen. The red area marks the portion of the screen we used for the interaction.

We implemented the techniques *FingerGlass*, *FingerMap*, *Dual Finger Stretch*, *Shift* and *PrecisionGlass*. We assumed a fingertip diameter of 10mm for contact area interaction. For the techniques FingerGlass and PrecisionGlass, the prescribed zoom ratio was 6x, thus leveraging the effective target size from 2mm to 12mm.

Our implementation was written in C++ using the Qt and OpenGL APIs. We chose to make use of graphics hardware acceleration in order to ensure maximal framerates in our test scene with thousands of targets displayed on top of a 4096×4096 texture. Running on a single-core CPU with an ATI Radeon HD 2600 card, we obtained frame rates consistently above 30fps.

Dependent Variables

In accordance to our design goals, the basic dependent measures for all phases of the task were completion time and error rate. The completion times of the individual phases are denoted *acquisition time*, *Release-Tapping time* and *translation time*. Their sum is denoted *total time*. For timing measurements, we took only successful attempts into account. In a similar fashion, we define the error rate for each subtask: the *acquisition error rate* is defined as the number of failed acquisitions divided by the number of total acquisitions. The *translation error rate* is obtained by dividing the number of failed translations by the number of total translations.

Independent Variables

We used a repeated measures within-subject factorial design for the study. The independent variables were *Technique* and *Distance*. We chose 8 values for *Distance* on a logarithmic scale. The longest chosen distance was 350 mm. To obtain the other distances, we successively divided by a factor of two. Hence, the other values were 175 mm, 87.5 mm, 43.75 mm, 21.87 mm, 10.94 mm, 5.47 mm and 2.73 mm. For the translation subtask, the combination of our target size with the chosen distance results in a range of index of difficulty (ID) values in Fitts’ law terms [11], from 1.2 to 7.5 bits.

Techniques were presented to each participant in random order. For every technique, 12 blocks had to be completed. Each block contained a random permutation of the 8 distances. We collected a total of 5 (techniques) x 12 (blocks) x 8 (distances) = 480 successful trials from each participant.

Participants

10 volunteers (4 female) with a mean age of 22.9 years participated in the experiment. All of them had some experience with touch screens from automated teller machines. All participants have used multitouch based phone or PDAs before, 5 participants use them on a daily basis. Only one participant has ever operated a multitouch workstation before. All participants were right-handed. We gave a $10 gift card to every participant as a reward.

Hypotheses

Unlike *Dual Finger Stretch*, *Shift* and *PrecisionGlass*, *FingerGlass* supports high-precision selection without the cost of any dragging operations. Both *FingerGlass* and *FingerMap* support subsequent and fast multiscale translation operations according to our principles (P2) and (P3). Therefore, we hypothesize:

- **(H1) Acquisition Time** – *FingerGlass has significantly shorter task completion times than the three baseline techniques when acquiring small targets ($r = 2mm$).*
- **(H2) Translation Time** – *FingerGlass and FingerMap have significantly shorter task completion times than the three baseline techniques when translating targets.*

RESULTS

We performed repeated measures analysis of variance on both trial completion time and error rate for the tasks of acquisition and translation. We classified timing results outside of 3 standard deviations as outliers. This way, we removed 126 (2.62%) and 116 (2.42%) trials for the tasks of acquisition and translation, respectively. In this section, we summarize our data analysis. More detailed results and figures can be found on the project website[1].

To verify if we could aggregate across the independent variable *Block*, we investigated the effect of this variable on task completion time. Participants significantly improved their total task completion time over their 12 trial blocks ($F_{11,99} = 9.37$, $p < .0001$). Concerned that the learning effect could influence the results of our study, we removed the first two trial blocks after visually inspecting the data. Although *Block* had no significant main effect on task acquisition time anymore ($p = 0.182$), it still had some on translation time ($p < 0.01$). However, there was no interaction between *Block* and *Technique*, neither for acquisition ($p = 0.889$) nor for translation ($p = 0.552$). We are mainly interested in a quantitative comparison of the different techniques, rather than in their absolute measurement. Therefore, it is sufficient to know that no tool is at an unfair advantage due to learning effects.

Task Completion Time

Figure 7 shows the total times for acquiring and translating a 2mm target across the screen. For *Shift* and *Dual Finger Stretch*, the time for Release-Tapping is added as well.

[1] http://www.fingerglass.com

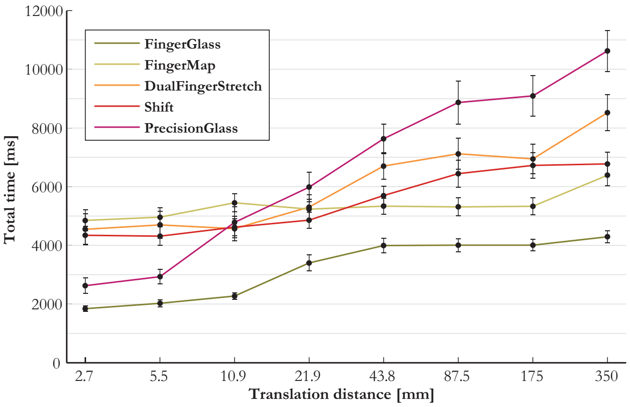

Figure 7: Total time (acquisition, Release-Tapping, translation) with respect to translation distance. Error bars represent the standard error of the mean.

Acquisition Time

Technique had a significant effect on acquisition time ($F_{4,36}$ = 7.6, $p < 0.001$). Paired samples t-tests show that *FingerGlass* was significantly faster than any other technique (all $p < 0.02$) for the selection of 2mm sized targets. This observation confirms our hypothesis H1. The second-fastest technique for this task was *PrecisionGlass*. The differences in acquisition time between *FingerMap* and all three baseline tools were insignificant (all $p > 0.1$). Table 1 lists the acquisition times for all techniques and comparison to *FingerGlass*. The mean completion time for the subtask Release-Tapping was 211ms (SD = 213ms). This time is not included in the acquisition time.

Translation Time

For translation, we performed a 5 × 8 (*Technique* × *Distance*) within subjects ANOVA aggregated across *Block*. We found significant main effects for both *Technique* ($F_{4,36} = 360.61$, $p < 0.001$) and *Distance* ($F_{7,63} = 473.94$, $p < 0.001$). However, most relevant is the significant *Technique* × *Distance* interaction ($F_{28,252} = 49.44$, $p < 0.001$). For the verification of our hypothesis H2, we were interested in post hoc multiple means comparisons. Paired samples t-tests showed that *FingerGlass* was significantly faster than the three baseline techniques for all distances equal to or greater than 10.9mm (all $p < 0.03$). For the smallest two distances, *FingerGlass* was significantly faster than *Dual Finger Stretch* and *Shift* (both $p < 0.03$), but not significantly different from *PrecisionGlass*.

These results for translation time confirm our hypothesis H2 for *FingerGlass* for all distances greater or equal to 10.9mm, but not for the two shortest ones. We reject the hypothesis for FingerMap: even at a distance of 350mm, the difference to Shift in translation time is insignificant ($p = 0.376$).

Paired samples t-tests on *Total Time* show that *FingerGlass* outperforms any other tool at any distance (all $p < 0.02$).

Effect of Translation Distance

For *Shift*, performance time increases smoothly over ascending distances. All other techniques operate differently for long distances than they do for short distances. This is reflected by a slope change in the curve at the threshold distance at which techniques change their behavior. With FingerGlass, some participants started redefining the area of interest during translation for distances 21.9mm and 43.8mm. For distances equal to or greater than 87.5mm, this was almost impossible to avoid: with an area of interest encompassing both the source and destination targets, the zoom ratio was often limited to 2x or less. Note that the threshold distance is about twice as large for *FingerGlass* than for *PrecisionGlass*: FingerGlass allows users to define the area of interest in a way that both the source and the destination of the dragging operation just barely fit in the magnified view, making full use of its space.

Total Acquisition Time [ms]				
Finger-Glass	*Finger-Map*	*Dual-Finger-Stretch*	*Shift*	*Precision-Glass*
1271 (×1.00)	**2571** (×2.02)	**2905** (×2.28)	**2194** (×1.73)	**2077** (×1.63)

Table 1: Time required for selecting a target with a diameter of 2mm. Factors in parentheses compare the respective tool to *FingerGlass*.

Manual Control of CD Ratio

Extending *PrecisionGlass* and *Shift* with a virtual slider for changing the CD ratio proved to be inefficient. With Shift, where users did not need to use the slider, this feature was hardly ever employed. With PrecisionGlass, changing the CD ratio was the only possible way to accomplish long-distance translation tasks. Thus, although PrecisionGlass performed very well for short distance translations, the timings were poor for medium- and long-distance ones. A closer analysis of the recorded performances showed that users often overshot or undershot the destination target after changing the CD ratio. The reason for this is that users chose a different CD ratio in every trial and thus could not predict the required distance in motor space. Thus, we conjecture that techniques addressing multiscale translation by varying the CD ratio should use a few (e.g. two) discrete levels so users can get accustomed to them quickly.

FingerMap

The performance of *FingerMap* did not meet our expectations: acquisition times were about twice as long as those of *FingerGlass*, and translation times were worse for all distances. For selecting targets, FingerMap sacrificed direct touch selection in order to minimize eye movement. Our result indicates that this does not lead to better task completion times. Whether or not it reduces fatigue would be subject to further research. For translation, the users might confuse the role of their hands without visual feedback, resulting in worse performance. Our performance logs show that participants often tried to move their fine hand, which only applies relative movements, towards the absolute position of the target. In addition, the design of FingerMap suffered from the same problem as our extension of PrecisionGlass: participants hardly made any strategic use of the controllable CD ratio. More often, they were overshooting or undershooting targets during both acquisition and translation.

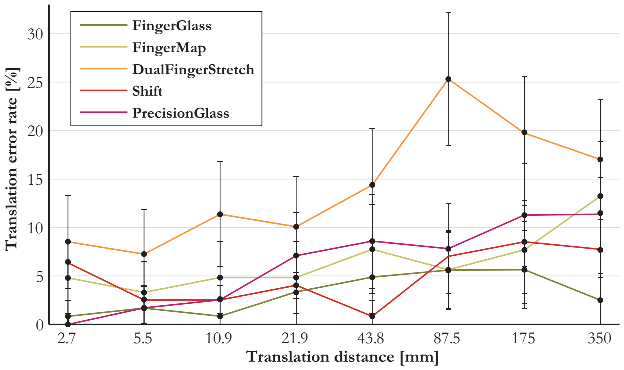

Figure 8: Error rate for the translation subtask with respect to distance. Error bars represent the standard error of the mean.

Effect of Target Location

We noticed that targets in the right half of the screen were somewhat harder to interact with than those in the left half. In some cases, the magnified view must be placed on the left side of the interest circle. Because all our subjects were right-handed, they had to either cross their arms or perform interactions in the magnified view with their non-dominant hand. To investigate this effect, we created a new grouping variable *XPos* indicating whether the source target was placed in the left, middle or right third of the screen. A 3×5 analysis of variance (*XPos*×*Technique*) on acquisition time and translation time revealed that there is a significant effect of the horizontal target position on translation time ($F_{2,18} = 17.15$, $p < 0.001$), but not on acquisition time ($F_{2,18} = 0.96$, $p = 0.38$). Similarly, the interaction between *XPos* and *Technique* was borderline significant for translation time ($F_{6,54} = 2.29$, $p = 0.032$), but not for acquisition time ($F_{6,54} = 0.39$, $p = 0.88$). FingerGlass had an average total time of 3214ms in the left, 2957ms in the middle, and 3561ms in the right third. As all other tools have a total time of over 5000ms in all thirds, this effect changes little about the relative performance of the tools.

Error Rate

To investigate selection and translation errors, we created variables *AcquisitionFail* and *TranslationFail* which measured the error rates for every condition (*Technique*, *Distance*, *Block*), aggregated across subjects. Note that there were more than $5 \times 8 \times 12$ trials for this analysis since subjects had to repeat erroneous attempts. The error rates are plotted in Figure 8.

Technique had no significant effect on *AcquisitionFail* ($F_{4,36} = 1.9$, $p = 0.128$), but had a significant effect on *TranslationFail* ($F_{4,36} = 10.81, p < 0.001$). *Distance* also had a significant effect on *TranslationFail*, but there was no interaction between *Technique* and *Distance*. Paired samples t-tests showed that *Dual Finger Stretch* had higher error rates than other tools with borderline significance. It had significantly higher error rates than both *FingerGlass* and *Shift* for distance 87.5mm, and than *Shift* for 43.8mm (both $p < 0.05$). By replaying the performances, we determined that many of the translation errors in Dual Finger Stretch happened in cases where users released the secondary finger before the first finger to end the translation. This reset the interface to the unmagnified state and moved the selected target to the absolute position of the primary finger. Subsequently releasing the first finger dropped the target in the wrong location.

We noticed that our touch device sometimes reported erroneous touch release events. This resulted in targets getting released early in translation operations and yielded false translation errors. To discard these cases, we used the event log to compute the velocity of a touch point immediately before its release event. We then removed trials with a higher release velocity than a threshold we determined by visual inspection of the data. To compute the velocity of a touch point, we averaged the pairwise distances of the last 5 touch move events. Using this method, we removed 201 translation attempts (3.16%).

	Dual-Finger-Stretch	Finger-Glass	Finger-Map	Shift	Prec.-Glass
acquisition	2.4	4.6	2.3	2.0	3.7
short drag	2.7	4.5	1.9	2.0	3.9
long drag	2.2	4.9	3.3	2.6	2.0
easy to learn	3.0	3.9	2.4	3.8	1.9
preference	2.5	4.5	3.4	2.3	2.3

Table 2: Mean subjective ranking, from 1 (worst) to 5 (best). Short dragging operations were defined to have a range of less than 5cm.

Subjective Evaluation

In the post-study questionnaire, participants were asked to rank the five techniques according to their preference for performing everyday tasks on their hypothetical personal multitouch workstations. Of the ten subjects, eight preferred *FingerGlass*. When asked to compare their preferred technique to a pen-based interface, users preferred the multitouch technique (3.3 out of 4 points, 4 being strong preference for multitouch). Comparison to a mouse yielded similar results (3.0 out of 4 points).

We also asked users which tool they find easiest to learn. The results show that *FingerGlass* was considered almost as easy to learn as *Shift*: 5 subjects would recommend *Shift* as the easiest technique to learn, 4 persons would recommend *FingerGlass*.

Finally, the participants ranked the tools by their subjective impression of the performance in object acquisition, short-distance dragging (5cm or less) and long-distance dragging (more than 5cm). The results confirmed our timing measurements. We assigned scores between 1 points (worst rank) and 5 points (best rank) to the votes and calculated the average scores as shown in Table 2.

CONCLUSIONS AND FUTURE WORK

We constructed two techniques enabling users to quickly navigate to different locations in different scales of a virtual scene, and to efficiently select and translate objects therein. Our experimental results show that one of these techniques, FingerGlass, significantly outperforms the current state-of-the-art techniques on touch screens for both precise selection and object translation.

FingerGlass does not require any information of the underlying scene and thus can be implemented independently on top of any existing application. In order to retain the gesture vocabulary of the underlying system, we suggest providing a modifier button in the fashion of the "Shift" or "Caps Lock" keys on computer keyboards to activate the tool temporarily or permanently.

In terms of limitations, our method requires at least three fingers to operate, and is designed for large multitouch workstations. We did not vary the screen size in our experiment. However, since three fingers occlude a significant area on small displays, it is likely that the advantage of FingerGlass decreases as the screens get smaller, compared to single-finger techniques such as Shift.

We believe that some of our findings are general enough to be applied to a wider range of applications. Therefore, we extracted a set of interaction principles for the efficient bimanual interaction with more general multiscale datasets. An example for such applications would be the modification of surfaces in 3D space. In such a system, the user could use his coarse hand to specify a small section of a surface. The view from a camera pointing along the surface's normal onto the surface would then be displayed in the magnified view. This technique would allow a user to temporarily look at scenes from a different point of view and perform operations like surface painting with his fine hand or moving small objects which are invisible from the original perspective.

The range of scales that can be explored using FingerGlass could be leveraged by allowing the user to recursively define areas of interest. By specifying a new area of interest in an existing magnified view, a new magnified view could appear, visualizing the scene at an even smaller scale.

ACKNOWLEDGEMENTS

We thank Tony DeRose, Björn Hartmann, Christine Chen and Kenrick Kin for their insightful comments and continuous support. This work was partially supported by NSF grant IIS-0812562.

REFERENCES

1. P.-A. Albinsson and S. Zhai. High precision touch screen interaction. In *Proceedings of CHI '03*, page 105, 2003.
2. C. Appert, O. Chapuis, and E. Pietriga. High-precision magnification lenses. *Proc. CHI '10*, page 273, 2010.
3. P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger. Drag-and-Pop and Drag-and-Pick. In *Proc. INTERACT 03*, 2003.
4. B. B. Bederson and J. D. Hollan. Pad++: A zooming graphical interface for exploring alternate interface physics. In *Proc. UIST '94*, pages 17–26, 1994.
5. H. Benko, A. Wilson, and P. Baudisch. Precise selection techniques for multi-touch screens. *Proceedings of CHI '06*, pages 1263–1272, 2006.
6. E. Bier, M. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of SIGGRAPH '93*, page 80, 1993.
7. R. Blanch, Y. Guiard, and M. Beaudouin-Lafon. Semantic pointing: improving target acquisition with control-display ratio adaptation. *Proc. CHI '04*, 2004.
8. N. Elmqvist, Y. Riche, N. Henry-Riche, and J.-D. Fekete. Mélange: Space folding for visual exploration. *IEEE transactions on visualization and computer graphics*, 16(3):468–83, 2010.
9. A. Esenther and K. Ryall. Fluid DTMouse: Better mouse support for touch-based interactions. In *Proc. AVI '06*, pages 112–115, 2006.
10. L. Findlater, A. Jansen, K. Shinohara, M. Dixon, P. Kamb, J. Rakita, and J. O. Wobbrock. Enhanced area cursors: Reducing fine pointing demands for people with motor impairments. In *Proc. UIST '10*, 2010.
11. P. M. Fitts. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology*, 47:381–391, 1954.
12. C. Forlines and C. Shen. DTLens: Multi-user tabletop spatial data exploration. In *Proc. UIST 2005*, pages 119–122, 2005.
13. C. Forlines, D. Wigdor, C. Shen, and R. Balakrishnan. Direct-touch vs. mouse input for tabletop displays. *Proc. CHI '07*, page 647, 2007.
14. G. W. Furnas. Generalized fisheye views. *Proc. CHI '86*, 1986.
15. G. W. Furnas and B. B. Bederson. Space-scale diagrams: understanding multiscale interfaces. *Proc. CHI '95*, 1995.
16. T. Grossman and R. Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *Proc. CHI '05*, 2005.
17. Y. Guiard. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behavior*, pages 486–517, 1987.
18. K. Hinckley, R. Pausch, D. Proffitt, and N. F. Kassell. Two-handed virtual manipulation. *ACM TOCHI*, 5(3):260–302, Sept. 1998.
19. K. Hornbaek, B. B. Bederson, and C. Plaisant. Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview. *ACM TOCHI*, 9(4):362–389, 2002.
20. T. Igarashi and K. Hinckley. Speed-dependent Automatic Zooming for Browsing Large Documents. In *Proc. UIST '00*, pages 139–148, 2000.
21. A. K. Karlson and B. B. Bederson. Direct Versus Indirect Input Methods for One-Handed Touchscreen Mobile Computing, 2007.
22. K. Kin, T. Miller, B. Bollensdorff, T. DeRose, B. Hartmann, and M. Agrawala. Eden: A professional multitouch tool for constructing virtual organic environments. In *Proc. CHI '11*, 2011.
23. J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: detail and context smoothly integrated. In *Proc. CHI '91*, 1991.
24. J. Mankoff, S. E. Hudson, and G. D. Abowd. Interaction techniques for ambiguity resolution in recognition-based interfaces. In *Proc. UIST '00*, pages 11–20, 2000.
25. T. Moscovich. Contact area interaction with sliding widgets. In *Proc. UIST '09*, 2009.
26. A. Olwal, S. Feiner, and S. Heyman. Rubbing and tapping for precise and rapid selection on touch-screen displays. In *Proc. CHI '08*, 2008.
27. S. Pook, E. Lecolinet, G. Vaysseix, and E. Barillot. Context and Interaction in Zoomable User Interfaces, 2000.
28. R. Potter, L. Weldon, and B. Shneiderman. Improving the accuracy of touch screens: an experimental evaluation of three strategies. *Proc. CHI '88*, 1988.
29. G. Ramos, A. Cockburn, R. Balakrishnan, and M. Beaudouin-Lafon. Pointing lenses: Facilitating stylus input through visual-and motor-space magnification. In *Proc. CHI '07*, pages 757–766, 2007.
30. G. G. Robertson and J. D. Mackinlay. The Document Lens. In *Proc. UIST '93*, 1993.
31. I. Rosenberg and K. Perlin. The UnMousePad: an interpolating multi-touch force-sensing input pad. In *Proc. SIGGRAPH '09*, 2009.
32. A. Roudaut, S. Huot, and E. Lecolinet. TapTap and MagStick: Improving one-handed target acquisition on small touch-screens. In *Proc. AVI '08*, 2008.
33. D. Schmidt, M. K. Chong, and G. Hans. IdLenses: Dynamic personal areas on shared surfaces. In *Proc. ITS '10*, 2010.
34. D. Vogel and P. Baudisch. Shift: a technique for operating pen-based interfaces using touch. In *Proc. CHI '07*, pages 657–666, 2007.
35. K. Yatani, K. Partridge, M. Bern, and M. W. Newman. Escape: a target selection technique using visually-cued gestures. *Proc. CHI '08*, 2008.