Touch Projector: Mobile Interaction through Video

Sebastian Boring, Dominikus Baur, Andreas Butz
University of Munich
Amalienstrasse 17, 80333 Munich, Germany
{sebastian.boring, dominikus.baur, andreas.butz}@ifi.lmu.de

Sean Gustafson, Patrick Baudisch
Hasso Plattner Institute
August-Bebel Str. 88, 14482 Potsdam, Germany
{sean.gustafson, patrick.baudisch}@hpi.uni-potsdam.de

ABSTRACT
In 1992, Tani et al. proposed remotely operating machines in a factory by manipulating a live video image on a computer screen. In this paper we revisit this metaphor and investigate its suitability for mobile use. We present Touch Projector, a system that enables users to interact with remote screens through a live video image on their mobile device. The handheld device tracks itself with respect to the surrounding displays. Touch on the video image is "projected" onto the target display in view, as if it had occurred there. This literal adaptation of Tani's idea, however, fails because handheld video does not offer enough stability and control to enable precise manipulation. We address this with a series of improvements, including zooming and freezing the video image. In a user study, participants selected targets and dragged targets between displays using the literal and three improved versions. We found that participants achieved highest performance with automatic zooming and temporary image freezing.

Author Keywords
Mobile device, input device, interaction techniques, multi-touch, augmented reality, multi-display environments.

ACM Classification Keywords
H5.2 [Information interfaces and presentation]: User Interfaces: Input Devices and Strategies, Interaction Styles.

General Terms
Design, Experimentation, Human Factors, Verification.

INTRODUCTION
In 1992, Tani et al. envisioned how users could interact with a real-world device located at a distance through live video [33]. Cameras observed industrial machinery and allowed users to manipulate mechanical switches and sliders over a distance by clicking and dragging within the live video image with a mouse. This was made possible by mapping portions of the video frame to the respective parts of the remote hardware. The system was revolutionary in that it established a particularly direct type of affordance — in many ways similar to the affordance of direct touch.

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.

While the metaphor is still interesting, the environments and usage scenarios have changed since that time. (1) The proliferation of displays on machines and computer systems has turned many spaces into *multi-display environments* [1]. (2) With the presence of portable computers such as laptops or tablet PCs, the displays within these environments may be *rearranged*. (3) In these flexible display setups, Tani's fixed camera setup is not necessarily appropriate anymore.

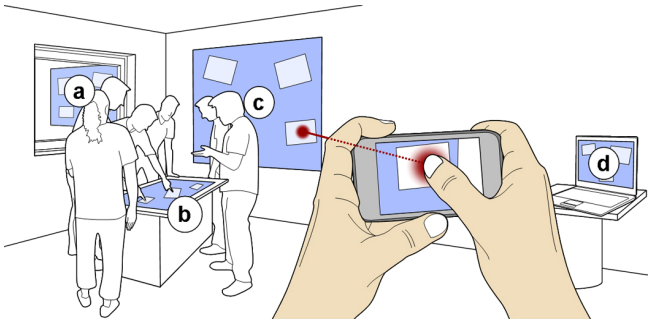

Figure 1. Touch Projector allows users to manipulate content on distant displays that are unreachable, such as (a) displays outside a window, or (b) a tabletop system crowded with people. It allows users to manipulate devices that are incapable of touch interaction, such as (c) a wall projection or (d) a laptop. Users point the device at the respective display and manipulate its content by touching and dragging objects in live video. The device "projects" the touch input onto the target display, which acts as if it had occurred on itself.

In this paper, we investigate how to apply "interaction through video" to these new scenarios and to what extent mobile devices can offer the required flexibility. We build on recent advances in mobile augmented reality (such as markerless tracking [25] and camera-based pose estimation [20]) combined with techniques for manipulating objects at a distance (such as distant pointing [36], input redirection [12] and local portholes from remote displays [32]).

TOUCH PROJECTOR
As illustrated by Figure 1, Touch Projector allows users to manipulate content on displays at a distance, including those that would otherwise be unreachable. It further allows users to manipulate devices that are be incapable of touch interaction, such as a wall projection, or a laptop computer. Users aim the device with one hand and then manipulate objects by touching and dragging it in the live video using the other hand. Touch input is "projected" onto the remote display, as if it had occurred on it.

With Touch Projector, users manipulate targets using both hands in concert. The non-dominant hand holds the device

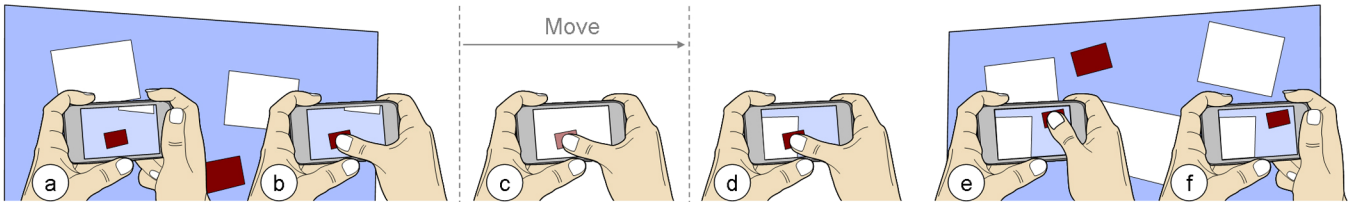

Figure 2. Walkthrough of the original metaphor: The user aims at a display (a) and touches the item of interest (b). When moving the device off-screen, a thumbnail of the dragged item is showing (c). After reaching the destination display (d), the item can be positioned precisely by moving the finger (e). When the finger is released, the item has been transferred successfully (f).

and coarsely orients it, while the dominant hand interacts within the reference frame established by the non-dominant hand (cf. *toolglass* interaction [5]). This combination allows interaction with large displays by moving the entire device (cf. *peephole* displays [37]) as well as interaction with small displays using touch input. Touch Projector preserves immediate feedback [29]: when content on the target display is changed, users immediately perceive these changes through the live video. This allows for a close connection of action and reaction as both occur on the mobile device.

Resulting interaction

Figure 2 shows how content is transferred using Touch Projector: (a) the user aims at the desired display. The content is seen in the live video on the mobile device. (b) The user touches the desired object and starts moving the Touch Projector device. As long as the device is pointed at the original display, the object keeps moving. It disappears as soon as its display leaves the device's viewing angle. (c) When dragging an item off-screen, a thumbnail is shown. (d) After reaching the destination display, (e) the object can be moved to its final location by moving the finger on the mobile device. (f) Releasing the finger ends the drag operation.

In order to allow mobile use, Touch Projector continuously tracks itself with respect to interactive displays in its surrounding using its built-in camera. It identifies displays around it and computes its spatial relationship between itself and any identified display. Knowledge about this spatial relationship is necessary for Touch Projector to transform the user's interaction on the mobile device into the target display's coordinate space.

System overview

The system consists of the Touch Projector device (here an Apple iPhone 3G), software on all display systems in the environment, and a server that controls the interaction (*environment manager*), all of which communicate over wireless LAN. On startup, all displays register with the environment manager and transfer their contents. The manager also receives updates if local changes are committed on a single display. The tracking system works for regular screen content (we used photos as examples).

To determine a Touch Projector's position and orientation, its camera stream is transferred to the environment manager and analyzed by deriving the relation to other displays from the perspective distortion of known elements (see IMPLEMENTATION section for details). All touch events received by a Touch Projector device are also routed through the environment manager (see Figure 3a). The manager handles touch events that require adjustment when performing drag operations across displays. If a screen leaves a Touch Projector's camera during a drag operation, both the touch point and item are removed from this screen (see Figure 3b). Similarly, when the mobile device reaches another display, the touch point is projected onto it and the item is added (see Figure 3c). If a touch ends while the device is not pointed towards a screen, the dragged item is returned to its original position.

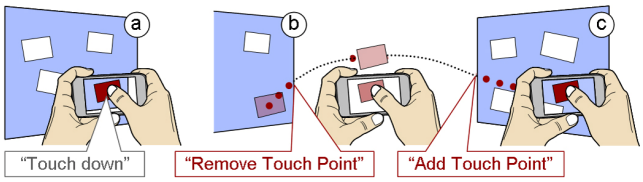

Figure 3. Touch event handling when interacting across displays. (a) A touch down event is projected onto the current screen. (b) When the item leaves the screen, the environment manager removes the touch point. (c) As soon as the device detects another screen, the manager adds a new touch point and the dragged item to the display.

Benefits

Compared to Tani's original system, Touch Projector offers three advancements:

1. The device tracks its spatial relationship to other displays, eliminating the need for modeling the environment. Since the tracking is purely based on the target display's visual content, the tracking can deal with new and rearranged display environments. This allows Touch Projector *impromptu* access of displays. Similar to *Stitching* [11], it can start and dismiss connections opportunistically.

2. Coarse/fine bimanual motion supports large target displays and dragging across distances (non-dominant hand) as well as precise local manipulations (dominant hand).

3. Touch Projector brings multi-touch input to single-touch, mouse-based, and even non-interactive displays. This means that multi-touch software can be used adequately on older and less interactive hardware. When performing cross-display operations (e.g., dragging an item from one screen to another), the unified interaction also solves compatibility problems, similar to *Pick-and-Drop* [23].

Limitations of naïve approach and resulting challenges

While the original "interaction through video" metaphor transfers well to mobile use, its implementation does not. Fixed cameras always produce steady images, but on mobile devices, instable images are created by minor hand movements (e.g., natural hand tremor). This influences the fine positioning of the dominant hand which in turn makes accurate interaction difficult. Fixed cameras further assume a constant distance between themselves and the device they are pointed at. These distances may greatly vary when the camera device is mobile. While zoom works with fixed cameras, the instability of the camera image increases due to the aforementioned hand movement. These limitations need to be addressed in order to successfully transfer interaction through video to mobile use. In the remainder of this paper, we present a series of modifications to the original metaphor for mobile devices.

RELATED WORK

Touch Projector builds on work in bimanual interaction, interaction at a distance (particularly across multiple displays), mobile augmented reality, world in miniature, and interaction through video.

Bimanual Interaction

Touch Projector is inspired by the specific type of bimanual interaction proposed by *Toolglasses* and *Magic Lenses* [5]. Both techniques position a (seemingly) transparent device with the non-dominant hand to enable the dominant hand to interact within it. Comparisons between pure bimanual techniques (both hands work independently) and dependent techniques (one hand influences the other one) show that the latter perform better [13]. These results were confirmed by Guimbretière et al. in a user study using a full factorial design [10]. They found merging command selection and direct manipulation to be the most important factor.

Bimanual interaction had previously been studied by Buxton et al. They found that bimanual input outperformed one-handed input for selection, positioning, and navigation tasks [7]. Latulipe et al. found performance benefits for bimanual input for the manipulation of multi-parameter functions, such as image corrections [15].

Interaction at a distance

Several at-a-distance techniques have been proposed to help when touch is unavailable. Relative pointing can be transferred to distant screens: *PointRight* allows mapping the mouse pointer to individual screens in multi-display environments [12]. *Perspective Cursor* accomplishes the same based on the user's perspective view [18]. A limitation of such systems is that users are required to locate/track their pointer among a potentially large number of other pointers.

Absolute pointing techniques address this [17]. *XWand* allows users to point with a virtual laser [36]. *Shadow Reaching* adds perspective as a further dimension allowing users to manipulate distant content by letting them cast a shadow [30]. In augmented and virtual reality, *Head Crusher* allows users to select objects by positioning thumb and forefinger around the object in their 2D projected image plane [21]. The *Go-Go* technique allows users to seamlessly reach both near and distant objects [22]. All these input strategies still require an indirect pointing device leading to similar effects seen in relative pointing (i.e., identifying the personal cursor among other ones on the screen). Accuracy of absolute pointing is limited by the user's fine motor skills. Motion errors are amplified with distance [4]. In the context of 2D touch screens, Sears et al. showed how to improve accuracy using local control display (CD) gain adjustments [27] which can also be used for interaction at a distance. Forlines et al. transferred the concept to wall displays, switching between absolute and relative pointing with a pen [9].

Nacenta et al. [19] surveyed interaction techniques that can be used for object movement in multi-display environments. The Touch Projector fits into their taxonomy as a *spatial* technique with a *perspective* display configuration which uses a *closed-loop* control method.

Camera phones and handheld augmented reality

Mobile display devices have been used to access remote content. *Sweep* uses optical flow analysis to enable continuous relative control of a remote pointer on a large screen [3]. Pears et al. use a camera phone for absolutely pointing on large screens [20]. Both (single touch) techniques use pointers on the remote screen.

Devices based on augmented reality add a local display into this model. The *Chameleon*, a spatially aware handheld computer, enabled users to browse information in 3D situated information spaces [8]. The *Boom Chameleon* is a spatially aware display mounted on a boom stand [34]. *Peephole Displays* simplify the metaphor to 2D [37]. Users of *Point & Shoot* take a photo of an object in order to interact with it [2]. Similarly, *Shoot & Copy* allows transferring content based on its photographic representation [6]. Interaction on these devices consists of two distinct phases: i.e., taking a photo and manipulating it.

Kato et al. utilized markers to extract the accurate position and orientation of a video camera [14]. The recognition of such markers is an established technology for the identification of augmented objects and the interaction with them.

World in miniature and "interaction through video"

Content can be brought to users to shorten the interaction distance: *Drag-and-Pop* shows content proxies in arm's reach [4]. *Tablecloth* extends the concept to screens with arbitrary content [24]. Other techniques create "portholes" that allow users to reach distant contents. *WinCuts* lets users use a mouse to "cut" regions of interest from a distant display in order to interact with them on a local screen [32]. Instead of transferring single elements, the *world in miniature* metaphor allows users to reach content by manipulating a scaled down complete version of it [31]. The perform-

ance of this technique varies with the magnification factor, which is dictated by the size of the manipulated world.

The aforementioned *Hyperplant* system allows users to control devices in a factory through a video image [33]. Liao et al. allowed users to annotate and drag-and-drop presentation slides between screens based on a video representation of the room [16]. Users could print slides by dragging them onto the video image of the printer. Users of *Sketch and Run* control vacuum cleaning robots through a video image shown on a pen-based tablet PC [26]. In *CRISTAL*, users collaboratively control a variety of digital devices in the living room through a virtually augmented video shown on a tabletop [28].

MAKING THE METAPHOR WORK

As discussed earlier, a literal adaptation of the original static metaphor does not work. In this section we present a series of improvements that make interaction through video work on mobile devices. We apply the following three improvements: (1) zoom, (2) temporarily freezing the preview, and (3) a virtual preview for optimized quality.

Step 1: Zooming allows reaching *distant* displays

Precise interaction requires the ratio between target size and viewing distance (referred to as a *target's apparent size*) to be reasonably high. While our tracking algorithm is comparably robust against small apparent sizes (i.e., the items are at least 20 pixels wide in the camera image), interacting with small targets is difficult due to the fat finger problem [35] and positioning jitter from the unstable video. We address the precision problem by adding variations of *zoom*.

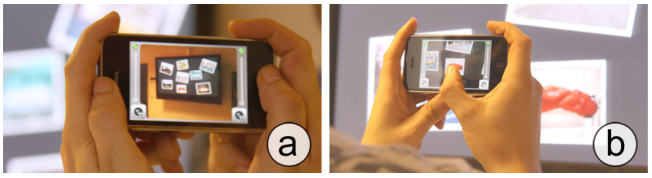

Figure 4. A 50" screen viewed at a distance of 1.5 m fills only 30% of the screen. (b) In order to see a 2" object large enough for manipulating it, the user has to go closer to the display.

Naturally, the user can "zoom" by moving closer to the display. However, as shown in Figure 4, the user has to get very close to obtain a reasonable object size. This is not appropriate for situations in which distant interaction is required. We therefore allow users to invoke a *zoom* feature. By adding a slider to the Touch Projector user interface we let users manually control the zoom. We decided against the commonly known pinch-gesture as all touch points are being projected onto the target display.

Optical zoom generally produces better image quality, but adds weight and cost to the device. In our implementation (similar to most of today's mobile devices), Touch Projector only offers *digital* zoom, which simply enlarges a subregion of the picture through image processing.

Automatic zooming saves user effort

We added two phases of automation to the zoom feature: (1) Touch Projector zooms out when it detects that it is no longer pointed at any objects. This is the case when the live camera image does not include any part of a remote display. (2) When the device is pointed towards a screen, it zooms in automatically (see Figure 5).

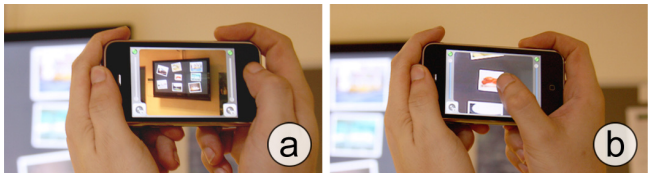

Figure 5. With *auto zoom* (a) small displays in its viewfinder cause Touch Projector to automatically zoom in until (b) the CD ratio has reached 1:2 (i.e., moving 1" on the Touch Projector causes a 2" movement on the target display.

The zoom factor is calculated by means of the distance between Touch Projector and the target display. The apparent size of any item (and the CD ratio) remains constant independently of the distance to the target screen. On However, zooming decreases the stability of the camera image, as a slight tilt of the mobile device is amplified to a large motion in the camera image. At increased zoom levels, the loose navigation of the non-dominant hand is too coarse for users to control it. While the bimanual navigation of Touch Projector partially counters this effect, we address it with a simplified type of image stabilization: the *freeze* feature.

Step 2: Freezing the camera image stabilizes zoom

Touch Projector allows users to temporarily freeze the live camera image by pressing a button (see Figure 6). The frozen image establishes a fixed reference frame within which the dominant hand can achieve higher precision. Freezing further eliminates the necessity to hold the device still or pointed at the screen, avoiding unnecessary fatigue. Live video is re-established by pressing the button again.

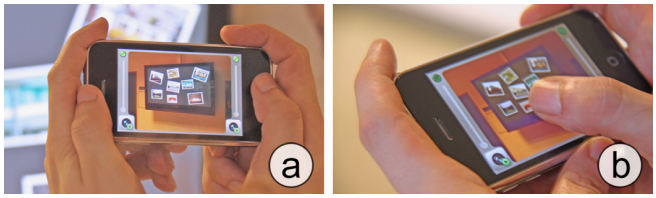

Figure 6. Our second improvement allows users to temporarily freeze the live video. (a) The user aims the camera at the desired region. (b) After pressing the "freeze" button, users can complete the interaction on a stable, non-changing image.

The freeze feature has two limitations, though. First, the limited sensitivity of the mobile device's camera makes it difficult to take a photo without motion blur. This is especially true in rooms that are dimly lit due to the use of projectors. Second, while the camera image is frozen, the device cannot show live visual feedback on its screen. To tackle these problems, we added computer-generated graphics to optimize the image quality and responsiveness.

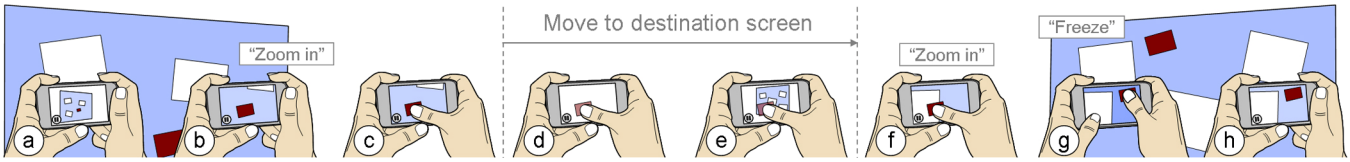

Figure 7. Dragging an object with the *updated* Touch Projector: (a) the user aims at a display (b) causing the device to automatically zoom in. (c) This allows the user to touch the red target object and (d) hold it while turning the device towards the other display. (e) Once the device detects the secondary display, (f) it zooms in again. (g) Pushing the freeze button with the thumb of the non-dominant hand causes the live camera image to pause for precise manipulation. (h) Lifting the finger releases the object.

Step 3: Virtual live preview optimizes "video" quality

When in freeze mode, we augment the live camera preview with computer-generated graphics. Touch Projector obtains the imagery wirelessly from the target screen. It uses the spatial relationship to the current target display in order to distort the computer-generated screen image accordingly.

The main advantage of the virtual live preview is that it gives immediate feedback on a temporarily frozen image. It combines the benefits of the live preview (i.e., direct manipulation) with the freeze feature (i.e., more comfortable postures). Thus, the virtual live preview preserves all properties of a physical preview; in particular, it also shows ongoing interactions by other users with the same screen.

The resulting Touch Projector works across distances

Together, zoom, freeze, and virtual live preview overcome the limitations discussed earlier.

Figure 7 shows a walkthrough of transferring content using the *updated* Touch Projector: (a) the user aiming at the desired display. The content is then seen through the live video. (b) Touch Projector recognizes the display and zooms in. This allows a CD ratio independently of the display's distance. (c) The user can now touch the desired object and start moving the Touch Projector device. As long as the device is pointed at the same remote display, the object moves on it. (d) When the user leaves the display, Touch Projector zooms out and a thumbnail is shown on the mobile device indicating which object is currently being dragged. (e) When the user reaches the destination display, Touch Projector zooms in again and the thumbnail is removed from the mobile device and dropped onto the destination screen for further manipulation. (f) The user can freeze the live image for fine-tuning the object's position on the remote display. (g) The object can now be moved to its final location with high precision by moving the finger on the mobile device. (h) Subsequently, the user can start the camera image again by pressing the "play" button.

IMPLEMENTATION

Touch Projector is implemented on a standard Apple iPhone 3G. It offers a screen diagonal of 3.5" and a display resolution of 320 × 480 pixels. Touch Projector is implemented in Objective-C. Live video is captured using the built-in camera at a resolution of 304 × 400 pixels.

A dedicated machine runs the environment manager. It is implemented in C# using the .NET Framework 3.5. Using a 3.0 GHz Core Duo machine as manager, we are able to run the image processing at about 15 frames per second (fps). However, the iPhone's limited transmission bandwidth only allows up to 8 fps. Future mobile devices (including new iPhone revisions) will likely offer higher bandwidth.

When a target display is started, it first sends a discovery message to the network and waits for the environment manager's response including its IP address and port. Subsequently, a connection is established through which the display sends its content. Similarly to the environment manager, the target display's software is also implemented in C# using the .NET Framework 3.5.

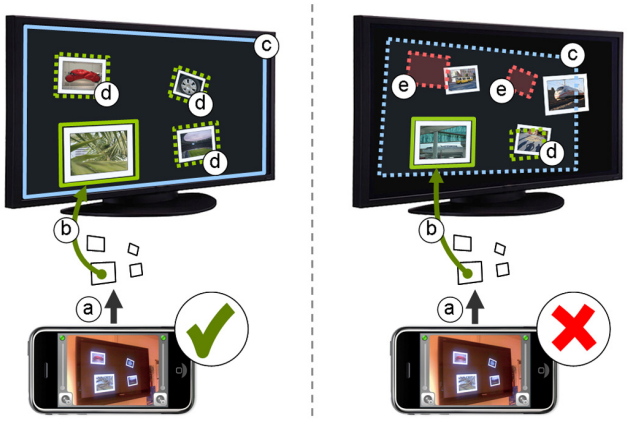

Figure 8. Tracking and display identification in Touch Projector works by looking at the closest match between the camera image and all known displays: (a) the system extracts polygons in the camera image. (b) The one closest to the video's center is matched to a display item through image differencing. (c) The resulting transformation between the polygon in the camera image and the real coordinates is calculated to identify the display boundaries. The remaining polygons are matched either correctly (d) or not (e) to those on the display. The left display matches the video image whereas the right is incorrect.

Detecting the target display

To allow a user to interact with a target screen, Touch Projector determines which on-screen object it is currently pointed at (see Figure 8). The mobile device permanently sends video frames to the environment manager. Depending on the previous frame, the environment manager decides which strategy to use for the current frame: (1) if the device was not pointed at a screen before, the current video frame is fully processed. (2) If the environment manager detected a screen in the previous frame, it uses simple optical flow analysis to determine the current spatial relationship.

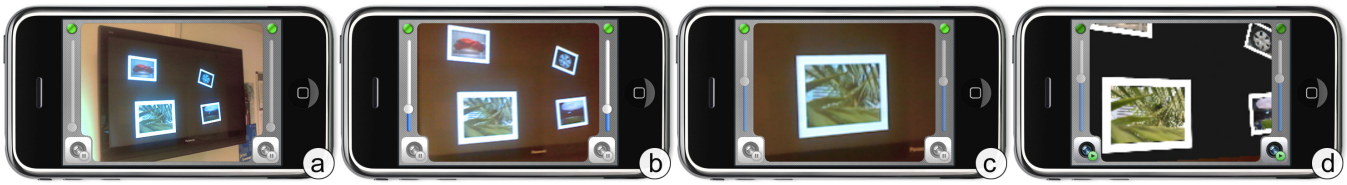

Figure 9. The different Touch Projector interfaces: (a) Original camera interface. (b) Manual zoom capabilities. (c) Automatic zooming. (d) Freezing the camera image with temporary overlay for precise interaction.

Full image processing

If no screen has been detected in the previous frame, the current video frame needs to be fully processed using the following steps: (1) reveal the polygon edges by increasing image contrast and performing a Hough transform. (2) For each identified polygon, the distortion caused by the camera's perspective is removed by transforming it into a rectangle with fixed dimensions. (3) The rectified polygon contents closest to the center of the video image are then compared with every object on all known displays using simple image differencing. (4) For the best match, the system computes a homography (i.e., the transformation between the camera image plane and the display image plane). It then tests whether the other polygons in the camera image correspond to items on the same display. (5) If they match well, the system successfully identified the target display. If the other polygons do not match, the system returns to step 4 on the next-best candidate until either a display has been identified or no possible matches are left.

If a display has been identified, the system chooses four points (i.e., corner points of all detected polygons) to compute the final homography. This minimizes calculation errors. The homography then allows the transformation of touch events into the target display's coordinate system. The feature points are further stored for subsequent frames.

Feature information from previous frame

If a display has been detected in the previous frame, the environment manager tries to detect the feature points used in the previous frame to calculate the homography. If they can be found in the current frame, the screen has been detected successfully and the homography (i.e., the spatial relationship) can be calculated as explained before. If at least one of the feature points cannot be detected in the current video frame, the system has to perform full image processing on the frame as explained above. If the environment manager still does not detect any screen it assumes that the Touch Projector device is not pointed at a screen.

Limitations of the current implementation

Touch Projector is subject to several limitations which result from the development stage of the prototype and not the underlying concept. The most prominent limitation is the interaction distance. Touch Projector needs to see at least one item fully in its viewfinder to detect the screen. With the iPhone's field of view of 45 degrees, the minimum distance is about 1.5 times the item's diagonal. Ultra-wide angle lenses could further reduce this minimum distance. On the other hand, the maximum distance is ten times the item's diagonal between the mobile device and the item itself. Future devices with higher camera resolutions could increase this distance substantially. The interaction speed is further crucial to the success of such systems. The iPhone's camera is particularly susceptible to motion blur. Moving the device faster than 50 pixels per frame impacts the recognition rate noticeably. Again, we assume that future devices with better cameras (e.g., better sensors, faster shutters) will address this in part.

USER STUDY

To validate our main design and the proposed extensions we conducted a user study. Participants acquired targets and dragged objects between screens using four different versions of Touch Projector: the naïve port presented at the beginning of the paper, as well as three improved versions, namely Manual Zoom, Auto Zoom and Freeze.

Interfaces

In our user study, we had four interface conditions all of which allowed interaction through video:

The **Original** condition enabled users to look and manipulate content through the original camera image (see Figure 9a). This system did not provide any zoom capabilities. The **Manual Zoom** condition allowed users to zoom in using up to 4× digital zoom when and to what level they desired (see Figure 9b). The **Auto Zoom** condition zoomed in automatically to keep a constant CD ratio independent of the screen's distance. In our study, the apparent size of display objects remained constant at 3cm (see Figure 9c). The **Freeze** and virtual preview condition allowed users to freeze the image by tapping on the *freeze button* (see the bottom corner of Figure 9d). The frozen image then switched to a computer-generated digital image of the target screen. Tapping the button again restarted the live video. Participants were free to choose whether to use freeze. This condition also included the automatic zoom feature.

Tasks

Participants performed two types of tasks. Both tasks required participants to use Touch Projector to interact with display content at a distance.

During the targeting task participants acquired targets on a distant screen. As illustrated in Figure 10, all trials began with a *start button* appearing on the screen. When participants tapped the start button, it disappeared, the target item

was shown on the screen and the timer started. Now participants acquired the target by pointing Touch Projector at the remote display and tapping on the target with their finger. If they missed the target an error was logged and participants had to try to acquire the target again. Selection of the correct object completed the trial and stopped the timer.

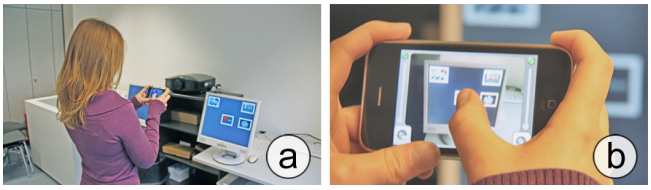

Figure 10. A participant performing the *targeting* task. She first starts by (a) tapping on the start button and then (b) selects the target item.

For each trial, the target was one of three apparent sizes: 0.75 cm, 1.5 cm and 3 cm on the Touch Projector screen. We varied apparent size to simulate large target screen distances that we did not have sufficient space for in our lab. We therefore kept the target screen distance constant and instead varied the target size on the screen.

During the dragging task participants dragged an object of fixed apparent size (3 cm on the Touch Projector) between distant screens. The setup contained two screens, as shown in Figure 11. At the beginning of each trial, the start button was shown on one screen and the target drop area on the other one. Tapping the start button initiated the trial, i.e., showed the item to be transferred as well as started the timer. Participants then aimed Touch Projector at the highlighted object and acquired the object with touch-and-hold. If participants acquired the wrong object an error was logged and participants had to repeat the trial. Participants then moved Touch Projector until the destination screen was visible in the live video image. Participants released the object by lifting off their finger, which "initiated the transfer". If the center of the object was located within the target area, the trial was completed. We measured task time and percentage of the object located outside the target area, which we call the *docking offset*.

Figure 11. A participant performing the *dragging* task. We chose three different angles between the two screens: (a) 45°, (b) 90° and (c) 180°.

Similar to the targeting task, we varied the apparent size of objects on the target screen. The objects on the source screen were always 3 cm, while the destination screen contained 3 cm, 1.5 cm, or 0.75 cm target areas. In addition, the angular distance from the source screen to the target screen was 45° (slightly left of the source screen), 90° (directly left of the participant), or 180° (behind the participant) as shown in Figure 11. As in the first task, participants had to acquire these targets as quickly and accurately as possible.

Experimental design

We used a within-subjects design in our experiment. In the targeting task we used a 4 *Techniques* (*Original*, *Manual Zoom*, *Automatic Zoom*, and *Freeze*) × 3 *Apparent Sizes* (*0.75 cm*, *1.5 cm*, and *3 cm*) design. The dragging task used a 4 *Techniques* (*Original*, *Manual Zoom*, *Automatic Zoom*, and *Freeze*) × 3 *Apparent Sizes* on target display (*0.75 cm*, *1.5 cm*, and *3 cm*) × 3 *Angles* (*45°*, *90°*, and *180°*) design.

Technique was counterbalanced across participants in the first task. In the second task, the presentation of *Technique* and *Angle* was counterbalanced across participants. In both tasks, the three *Apparent Sizes* were presented in random order within each block. Each task consisted of one practice block and three timed blocks for each Technique. Participants received up-front training. Each participant completed the study in 60 minutes or less. About 25% of the entire time was spent on the first task, 75% on the second. However, *targeting* is part of the second task (*dragging*).

Participants

Twelve volunteers (4 female), ranging in age from 22 to 30 years and from 162 to 205 centimeters in height, were recruited from our institution. One participant was left handed. Eleven of them had at least some previous experience with touch-based mobile phones.

Hypotheses

We hypothesized that each of the three modifications would lead to an improvement in user performance for small apparent sizes. For small apparent sizes we expected (H1) the zoom-enabled techniques to outperform the *Original* interface in terms of task time and error rate, (H2) *Auto Zoom* to outperform *Manual Zoom* in task time and (H3) *Freeze* to result in a lower docking offset than the other techniques.

RESULTS

We compared separate repeated measures ANOVA tests on mean completion times and error for each task. For the targeting task, error was measured as the number of failed trials. For the dragging task, error was the docking offset. To determine the nature of interaction effects we performed tests on subsets of the data. Post hoc pair-wise comparisons used Bonferroni corrected confidence intervals to retain comparisons against α=0.05. Unstated *p*-values are $p<0.05$.

Targeting task

Completion time: We found a significant main effect on task completion time for *Technique* ($F_{3,9}$=61.659, $p<0.001$, all pairs differ $p<0.025$) and *Apparent Size* ($F_{2,10}$=88.831, $p<0.001$, all pairs differ $p<0.002$). We further identified a significant interaction between *Technique* and *Apparent Size* ($F_{6,6}$=17.915, p=0.001).

Upon inspecting Figure 12a one can see that the task completion time disparity shrinks as *Apparent Size* increases. We separately analyzed each *Apparent Size* level and found that there is no significant main effect for *Technique* when *Apparent Size* is 3 cm. However, there is a significant main effect when the *Apparent Size* is smaller (p<0.002), indicating that this is the main source *Technique* × *Apparent Size* interaction found earlier. Overall, the slowest technique was *Original* (M=2198 ms, SD=775 ms), followed by *Manual Zoom* (M=1862 ms, SD=521 ms), *Freeze* (M=1592 ms, SD=299 ms) and *Auto Zoom* (M=1476 ms, SD=290 ms).

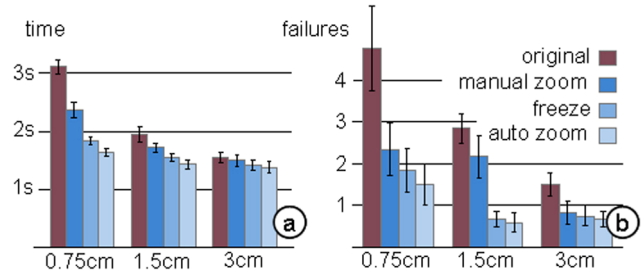

Figure 12. Results of *targeting* task: (a) Completion time and (b) number of failed trials by *Apparent Size*. Error bars indicate ± standard error of the mean.

Failed trials: We found a significant main effect on the number of failed trials for *Technique* ($F_{3,9}$=6.546, p=0.012, only *Freeze* and *Auto Zoom* differ from *Original* with p<0.015) and for *Apparent Size* ($F_{2,10}$=40.104, p<0.001, all pairs differ p<0.004 except *1.5 cm* and *3 cm*). We further found an interaction between *Technique* and *Distance* ($F_{6,6}$=92.533, p<0.001).

As with task completion time, Figure 12b indicates that the main source of the *Technique* × *Distance* interaction was that the amount of failures decreased as the *Apparent Size* increased. Overall, *Original* (M=3.0 failures, SD=2.5 failures) had the highest number of failed trials, followed by *Manual Zoom* (M=1.8, SD=1.8), *Freeze* (M=1.1, SD=1.3) and *Auto Zoom* (M=0.9, SD=1.2).

For both task completion time and number of failed trials, all techniques performed similarly when the *Apparent Size* was 3 cm. For all *Apparent Sizes*, the *Auto Zoom* and *Freeze* techniques performed similarly well. For *Apparent Sizes* of 1.5 cm and 0.75 cm the *Manual Zoom* and *Original* techniques performed significantly worse (p<0.01), with the *Original* technique performing substantially worse than all other techniques when *Apparent Size* is 0.75 cm in terms of task completion time (p<0.01). However, the number of failures was not significantly different at this level.

Dragging task

Completion time: We found significant overall main effects on task completion time for *Technique* ($F_{3,9}$=44.247, p<0.001, all pairs differ p<0.003 except *Auto Zoom* and *Freeze*), *Apparent Size* ($F_{2,10}$=69.015, p<0.001, all pairs differ p<0.001) and *Angle* ($F_{2,10}$=63.361, p<0.001, all pairs differ p<0.002). There is a significant interaction between *Technique* and *Apparent Size* only ($F_{6,6}$=18.777, p=0.001). The results are summarized in Figure 13a.

To discover the nature of the *Technique* × *Apparent Size* interaction we split the data based on *Apparent Size* levels and ran separate ANOVA tests. All techniques perform closely when *Apparent Size* is 3 cm. However, *Freeze* performs slightly worse causing significant differences from *Auto Zoom* and *Manual Zoom*. For the *Apparent Size* of 1.5 cm all pairs are significantly different (p<0.002) except *Auto Zoom* compared to *Freeze* and *Manual Zoom* compared to the *Original* technique. For the *Apparent Size* of 0.75 cm we observed significant differences between all pairs (p<0.040) except for *Auto Zoom* and *Freeze*. Overall, the slowest techniques were *Original* (M=11430 ms, SD=2785 ms) and *Manual Zoom* (M=9821 ms, SD=1014 ms). The fastest techniques were *Freeze* (M=9230 ms, SD=493 ms) and *Auto Zoom* (M=9024 ms, SD=516 ms).

Docking offset: We found significant main effects for *Technique* ($F_{3,9}$=4602.076, p<0.001, all pairs p<0.001) and *Apparent Size* ($F_{2,10}$=1121.254, p<0.001). We did not find a main effect for *Angle* or any other interaction effects. The results are summarized in Figure 13b.

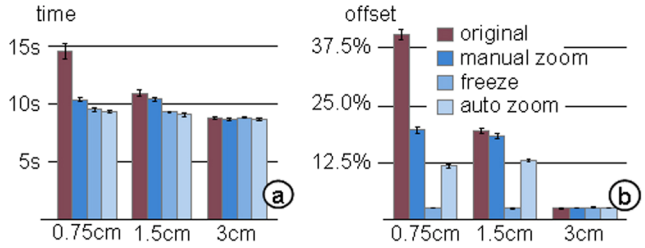

Figure 13. Results of *dragging* task: (a) Completion time and (b) docking offset by *Apparent Size*. Error bars indicate ± standard error of the mean.

There are two sources of the interaction between *Technique* and *Apparent Size*. First, there are minimal differences in means when *Apparent Size* is 3 cm (only *Auto Zoom* differs significantly from *Freeze*) and large significant differences (p<0.001) when *Apparent Size* is 1.5 cam and 0.75 cm. The second source of interaction is the consistently low docking offset for the *Freeze* technique for all *Apparent Sizes*. Overall, the least accurate of the techniques was *Original* (M=20.6%, SD=15.9%), followed by *Manual Zoom* (M=13.3%, SD=8.0%), *Auto Zoom* (M=8.9%, SD=4.8%) and *Freeze* (M=2.4%, SD=0.2%).

Subjective feedback

The mechanical nature of our tasks did not leave much space for thinking aloud. However, we did get a series of comments, suggestions and feature requests. The most prominent feature request mentioned by our participants was adding auditory or haptic feedback as indicator when a display has been detected. Several participants also requested to hold the device in a vertical way. The current

implementation of the display detection is not affected by the device orientation. However, the interface on the iPhone did not adapt to screen orientations, but we will include this in future versions. Overall, all our participants seemed to enjoy the interaction.

DISCUSSION

As hypothesized, all three improved techniques significantly outperformed the *Original* technique in both tasks for all but the largest apparent size (where there was no significant difference). When selecting a target, participants using the *Auto Zoom* technique were overall 49% faster / 70% less error-prone than when using the *Original* technique. For apparent sizes of 0.75 cm, participants were 90% faster / 68% less error-prone. In general, the zoom-enabled techniques were 34% faster / 59% less error-prone than the *Original* technique, which supports our first hypothesis.

In the targeting task, the *Auto Zoom* and *Freeze* technique performed best of all techniques with a slight advantage to *Auto Zoom* for small apparent sizes. We further found that *Manual Zoom* and the *Original* technique performed significantly worse for all small apparent sizes. Hence, the *Auto Zoom* technique also outperforms the *Manual Zoom*, which supports our second hypothesis. *Freeze* had a slightly higher task time compared to *Auto Zoom*. This can be explained by the fact that users had to press the pause button in order to freeze the image before they were able to acquire the target using the computer-generated overlay.

When dragging an object between screens, participants overall were 27% faster / 132% more accurate with the *Auto Zoom* technique compared to the *Original* technique. For the small apparent size they were 56% faster / 249% more accurate. The *Freeze* technique revealed its strength by being over 10 times more accurate than the *Original* technique. This supports our third hypothesis.

In the dragging task, only the *Freeze* condition has the advantage of retaining a low offset across all apparent sizes. However, the extremely low targeting offset of the *Freeze* technique was expected (see H3) as the instability of camera images increases with a higher zoom factor. *Auto Zoom* performs better than the other techniques in terms of task completion time (supporting H2). However, it does not allow for the highly precise target placement which can be achieved using the *Freeze* technique.

Our study shows that *Auto Zoom* is the best performing technique for targeting tasks. *Freeze*, however, outperforms *Auto Zoom* for precise manipulation tasks by keeping the image steady. This suggests that freezing the image temporarily should be an optional feature that complements automatic zooming (as implemented in the *Freeze* feature).

CONCLUSIONS

Touch Projector allows the manipulation of content shown on a distant display through touch input on live video. While this works well for targets with large apparent sizes on the mobile screen, small sizes lead to poor performance. We presented three extensions to the original idea that improve the performance in terms of task time and error rate.

In our experiment, we verified that zoom-enabled techniques outperform the naïve approach. Furthermore, the study revealed that freezing the live image significantly decreases the targeting offset and thus allows precise manipulation (i.e., translating, scaling, and rotating) of an item at a distance. Automatically zooming in to gain a higher apparent size also decreased the task time. The outcome of the experiment encourages using automatic zooming in general while allowing the user to temporarily freeze the image for high accuracy if required by the task.

In the future we plan to study the effects of completely computer-generated graphics on the mobile device as a replacement for the camera stream. Most importantly, this would enable the system to mimic an optical zoom with a much higher focal length on mobile devices. Additionally, we want to study the difference between giving feedback on the mobile device or the remote display.

ACKNOWLEDGMENTS

This work has been funded by the German state of Bavaria. We would like to thank the reviewers for their detailed comments and suggestions. We also thank the participants of our study for their time and patience. Furthermore, we would like to thank Doris Hausen, Christina Dicke and Christian Holz for their valuable feedback.

REFERENCES

1. Balakrishnan, R., and Baudisch, P. (2009). Special Issue on Ubiquitous Multi-Display Environments. *HCI Journal 24*, 1 & 2.
2. Ballagas, R., Rohs, M., and Sheridan, J.G. (2005). Sweep and point & shoot: phonecam-based interactions for large public displays. *Ext. Abstracts CHI 2005*, 1200–1203.
3. Ballagas, R., Borchers, J., Rohs, M., and Sheridan, J.G. (2006). The Smart Phone: a ubiquitous input device. *IEEE Pervasive Computing 5*, 1, 70–77.
4. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P., Bederson, B., and Zierlinger, A. (2003). Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch- and pen-operated systems. *Proc. Interact 2003*, 57–64.
5. Bier, E.A., Stone, M.C., Pier, K., Buxton, W., and DeRose, T.D. (1993). Toolglass and magic lenses: the see-through interface. *Proc. SIGGRAPH 1993*, 73–80.
6. Boring, S., Altendorfer, M., Broll, G., Hilliges, O., and Butz, A. (2007). Shoot & copy: phonecam-based information transfer from public displays onto mobile phones. *Proc. Mobility 2007*, 24–31.
7. Buxton, W., and Myers, B. (1986). A study in two-handed input. *Proc. CHI 1986*, 321–326.

8. Fitzmaurice, G.W. (1993). Situated information spaces and spatially aware palmtop computers. *Communications of the ACM 36*, 7, 39–49.
9. Forlines, C., Vogel, D., and Balakrishnan, R. (2006). HybridPointing: fluid switching between absolute and relative pointing with a direct input device. *Proc. UIST 2006*, 211–220.
10. Guimbretière, F., Martin, A., and Winograd, T. (2005) Benefits of merging command selection and direct manipulation. *ACM Transactions on Computer-Human Interaction 12*, 3, 460–476.
11. Hinckley, K., Ramos, G., Guimbretière, F., Baudisch, P., and Smith, M. (2004). Stitching: pen gestures that span multiple displays. *Proc. AVI 2004*, 23–31.
12. Johanson, B., Hutchins, G., Winograd, T., and Stone, M. (2002). PointRight: experience with flexible input redirection in interactive workspaces. *Proc. UIST 2002*, 227–234.
13. Kabbash, P., Buxton, W., Sellen, A. (1994). Two-handed input in a compound task. *Proc. CHI 1994*, 417–423.
14. Kato, H., and Billinghurst, M. (1999). Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. *Proc. IEEE and ACM International Workshop on Augmented Reality*, 85–94.
15. Latulipe, C., Kaplan, C.S., Clarke, C.L.A. (2005). Bimanual and unimanual image alignment: an evaluation of mouse-based techniques. *Proc. UIST 2005*, 123–131.
16. Liao, C., Liu, Q., Kimber, D., Chiu, P., Foote, J., and Wilcox, L. (2003). Shared interactive video for teleconferencing. *Proc. ACM Multimedia 2003*, 546–554.
17. Myers, B., Bhatnagar, R., Nichols, J., Peck, C.H., Kong, D., Miller, R., and Long, A.C. (2002). Interacting at a distance: measuring the performance of laser pointers and other devices. *Proc. CHI 2002*, 33–40.
18. Nacenta, M.A., Sallam, S., Champoux, B., Subramanian, S., and Gutwin, C. (2006). Perspective cursor: perspective-based interaction for multi-display environments. *Proc. CHI 2006*, 289–298.
19. Nacenta, M.A., Gutwin, C., Aliakseyeu, D., and Subramanian, S. (2009). There and back again: cross-display object movement in multi-display environments. *HCI Journal 24*, 1, 170–229.
20. Pears, N., Jackson, D., and Olivier, P. (2009). Smart phone interaction with registered displays. *IEEE Computer 8*, 2, 14–21.
21. Pierce, J.S., Forsberg, A.S., Conway, M.J., Hong, S., Zeleznik, R.C., and Mine, M.R. (1997). Image plane interaction techniques in 3D immersive environments. *Proc. Symposium on Interactive 3D Graphics*, 39-43.
22. Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. (1996). The go-go interaction technique: non-linear mapping for direct manipulation in VR. *Proc. UIST 1996*, 79–80.
23. Rekimoto, J. (1997). Pick-and-drop: a direct manipulation technique for multiple computer environments. *Proc. UIST 1997*, 31–39.
24. Robertson, G., Czerwinski, M., Baudisch, P., Meyers, B., Robbins, D., Smith, G., and Tan, D. (2005). The large-display user experience. *IEEE Computer Graphics and Applications*, 25, 4, 44–51.
25. Rohs, M., Schöning, J., Raubal, M., Essl, G., and Krüger, A. (2007). Map navigation with mobile devices: virtual versus physical movement with and without visual context. *Proc. ICMI 2007*, 146–153.
26. Sakamoto, D., Honda, K., Inami, M., and Igarashi, T. (2009). Sketch and run: a stroke-based interface for home robots. *Proc. CHI 2009*, 197–200.
27. Sears, A., and Shneiderman, B. (1991). High precision touchscreens: design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies 34*, 4, 593–613.
28. Seifried, T., Haller, M., Scott, S.D., Perteneder, C., Rendl, C., Sakamoto, D., Inami, M. (2009). CRISTAL: design and implementation of a remote control system based on a multi-touch display. *Proc. ITS 2009*, 33–40.
29. Shneiderman, B. (1983). Direct manipulation: a step beyond programming languages. *IEEE Computer 16*, 8, 57–69.
30. Shoemaker, G., Tang, A., and Booth, K.S. (2007). Shadow reaching: a new perspective on interaction for large displays. *Proc. UIST 2007*, 53–56.
31. Stoakley, R., Conway, M.J., and Pausch, R. (1995). Virtual reality on a WIM: interactive worlds in miniature. *Proc. CHI 1995*, 265–272.
32. Tan, D.S., Meyers, B., and Czerwinski, M. (2004). WinCuts: manipulating arbitrary window regions for more effective use of screen space. *Ext. Abstracts CHI 2004*, 1525–1528.
33. Tani, M., Yamaashi, K., Tanikoshi, K., Futakawa, M., and Tanifuji, S. (1992). Object-oriented video: interaction with real-world objects through live video. *Proc. CHI 1992*, 593–598.
34. Tsang, M., Fitzmaurice, G.W., Kurtenbach, G., Khan, A., Buxton, B. (2002). Boom chameleon: simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display. *Proc. UIST 2002*, 111–120.
35. Vogel, D., and Baudisch, P. (2007). Shift: a technique for operating pen-based interfaces using touch. *Proc. CHI 2007*, 657–666.
36. Wilson, A., and Shafer, S. (2003). XWand: UI for intelligent spaces. *Proc. CHI 2003*, 545–552.
37. Yee, K.-P. (2003). Peephole displays: pen interaction on spatially aware handheld computers. *Proc. CHI'03*, 1–8.