Interacting with Large Displays from a Distance with Vision-Tracked Multi-Finger Gestural Input

Shahzad Malik, Abhishek Ranjan, Ravin Balakrishnan

Department of Computer Science
University of Toronto
smalik | aranjan | ravin @ dgp.toronto.edu
www.dgp.toronto.edu

ABSTRACT

We explore the idea of using vision-based hand tracking over a constrained tabletop surface area to perform multi-finger and whole-hand gestural interactions with large displays from a distance. We develop bimanual techniques to support a variety of asymmetric and symmetric interactions, including fast targeting and navigation to all parts of a large display from the comfort of a desk and chair, as well as techniques that exploit the ability of the vision-based hand tracking system to provide multi-finger identification and full 2D hand segmentation. We also posit a design that allows for handling multiple concurrent users.

Categories and Subject Descriptors: H.5.2 [**User Interfaces**]: Interaction styles; I.3.6 **[Methodology and Techniques]**: Interaction techniques.

General Terms: Design, Human Factors.

Additional Keywords and Phrases: large wall, interaction, multi-point, touch surface, two hands, bimanual, symmetric, asymmetric, gesture, visual touchpad.

INTRODUCTION

The increased screen real estate provided by large wall displays allows for sophisticated single- and multi-user applications that cannot be easily accommodated with standard desktop monitors. However with this larger work area comes a number of challenges, particularly from a user interface perspective. While many innovative techniques have been proposed in the literature to deal with the difficulties in quickly accessing all parts of a large display, the majority focus on within arms-reach interactions that assume users will be standing close to the screen [12, 21, 27, 28]. However, consider a single-user design task that requires the visualization capabilities of a large display but also demands long hours. Similarly, consider a collaborative discussion where users gather around a large conference room table but also frequently need to display things on a large screen for others to see. In these *distant-contiguous* large screen situations [30], allowing the users to interact from the comfort of their chairs seems desirable. While a few such from-afar techniques have been proposed in the literature [16, 17, 18], many still assume mouse-based input and thus fast navigation and target acquisition tasks are still relatively inefficient compared to many arms-reach techniques.

In this paper we develop several one- and two-handed interaction techniques that support efficient large wall interactions from a distance, whereby a user is seated comfortably in front of the display at a desk or conference room table. Using real-time computer vision algorithms that track all ten fingers of a user's bare hands, we explore techniques that allow for a direct manipulation experience on large wall displays using finger manipulations and gestures, similar to a table-top display or touch-screen. A flat, rigid surface with a small identification tag and a large black rectangular region serves as a wireless touch-sensitive device over which a user can make finger manipulations and gestures, while two cameras mounted over-head are used to capture live video of the hands and black regions for real-time vision processing. Figure 1a shows our prototype touchpad, which is a simple piece of cardboard with the black region and identifier tag printed on regular paper. By attaching unique tags to different touchpads, the system also allows multiple users (each with their own touchpad) to be easily detected. Figure 1b shows the actual touchpad in use with a large projection display. Combined with two off-the-shelf web cameras, the system is extremely low-cost and very easy to implement.

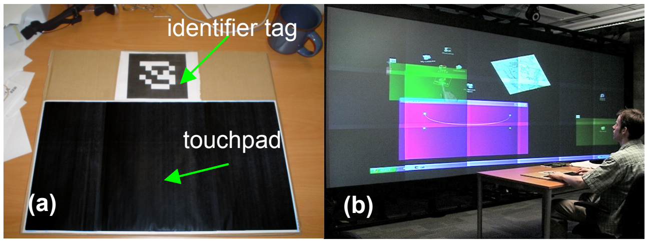

Figure 1. (a) The touch sensitive surface consisting of a unique tag and a solid black colored touch region; (b) A user working with the system on a large rear-projection display.

UIST'05, October 23–27, 2005, Seattle, Washington, USA.

RELATED WORK

Reaching distant targets and navigation of the entire display space are two of the major issues involved in interaction with large (> 10') upright wall displays. As such, there is a large body of literature that investigates these difficulties and proposes some effective solutions [12, 27, 28]. For example, Bezerianos et al. presented a tool called 'Vacuum' for quick access to distant items [3]. The user controls the area of influence of the tool so that distant objects that fall within the area of influence are transported closer to the user for easy selection. Similarly, Khan et al. introduced a widget called 'Frisbee' that uses the concept of a telescope to create a portal to another part of a large display for accessing remote objects [21]. Other techniques such as Drag-and-Pop and Drag-and-Pick [2] can be used for quickly activating distant icons on a graphical desktop, while shuffling and throwing [9] or flicking [34] allow objects to be moved to an approximate location at a specified distance or at the edge of the display.

The majority of these interaction techniques are suited to up-close, pen-based interactions in order to minimize a user's physical movements while standing in front of a large wall display. A number of researchers have also addressed the navigation and target acquisition issues when interacting from a distance. In the Pointright [18] and i-Room [17] systems the user can use a standard mouse as the input device and move the cursor across the entire display (consisting of different screens) seamlessly as though they were a single surface. Since they mainly focus on the problem of device-display integration, fast display navigation has not been addressed in detail. Khan et al. presented a technique called 'Spotlight' which allowed a user to control a large highlighted region across a large display from afar in order to direct the visual attention of an audience during a presentation [20]. While this technique has been found to be better than a regular cursor for highlighting targets, it is not clear how it should be used for reaching them efficiently.

Various vision-based techniques have been used for interaction with large scale displays. For example, the systems presented in [6] and [23] track a laser pointer and use it as an input device which facilitates interactions from a distance. While the laser pointer provides a very intuitive way to randomly access any portion of the wall sized display, natural hand-jitter makes it difficult to use for precise target acquisition tasks, particularly for smaller targets. Moreover, ordinary laser pointers have only two degrees of freedom which limit their use for complicated tasks. The VisionWand system [5] uses simple computer vision algorithms to track the colored tips of a simple plastic wand to interact with large wall displays both close-up and from a distance. A variety of postures and gestures are recognized in order to perform an array of interactions. A number of other systems use vision to track bare, unmarked hands using one or more cameras, with simple hand gestures for arms-reach interactions. For example, the Bare-Hand system [13] uses hand tracking technology to transform any ordinary display into a touch-sensitive surface. Similarly, the Touchlight system [33] uses two cameras to detect hand gestures over a semi-transparent upright surface for applications such as face-to-face video conferencing or augmented reality. The major advantage of such vision-based techniques is their ability to track multiple fingers uniquely, which allows for more degrees of freedom when compared to standard input devices such as a mouse. However, this advantage of vision-based techniques has not yet been fully leveraged for interactions with wall-sized displays.

The tabletop community has also been utilizing multi-finger input technology for a variety of direct manipulation applications. Rekimoto's SmartSkin technology [29] allows the detection of multiple contact points for tabletop displays, allowing full hand gestures to be recognized, but it cannot distinguish between different users. The DiamondTouch [7] technology offers similar functionality to the SmartSkin system, but with the added ability to also differentiate between multiple users. Wu and Balakrishnan demonstrated the capabilities of the DiamondTouch by presenting a multi-user room planning system that uses hand gestures for direct manipulation interactions [34]. While both of these technologies could be used to interact with large upright wall displays from afar in a manner similar to the touch surfaces found beneath many laptop keyboards, there are some shortcomings. Both the DiamondTouch and SmartSkin technologies require the hand to be in relatively close contact to the surface in order for a complete 2D hand image to be detected. As a result, it is difficult for an application to disambiguate which fingers are making contact with the surface. Therefore, when attempting to use such touch-sensitive surfaces for large wall interactions, the finger ambiguity and lack of 2D hand information makes it difficult for a user to visualize how the hand is being mapped to display space. The Visual Touchpad system [26] proposed a vision-based touch technology that simulates the functionality of the DiamondTouch or SmartSkin systems. However, since the system has access to an entire 2D hand image, it can resolve the finger ambiguity problem of the other systems. Additionally, the live images of the hands can be segmented from the video stream and then augmented over the workspace for an accurate visualization of the mapping between the touch surface and a display. Unfortunately, the Visual Touchpad directly maps the four corners of the touchpad to the four corners of a display. This causes serious problems when attempting fine positioning tasks on large wall displays, since a small amount of movement on the touchpad gets mapped to a large number of display pixels.

In short, most of the present interaction techniques for wall-sized displays are limited to up-close interactions using a pen or direct touch, while the limited number of systems allowing interaction from a distance suffer from one or more of the following issues: limited degrees of freedom, lack of visualization of degrees of freedom, inability to differentiate between the two hands and between fingers, or lack of proper balance between quick navigation and precise target acquisition. Based on these shortcomings, we have designed a vision-based bimanual interaction system that allows for quick navigation and precise target acquisition on large wall displays from afar using multi-finger manipulations and gestures.

DESIGN PRINCIPLES

In designing fluid interactions for a large wall display for users seated at a table, we have considered the following design issues:

Leverage both hands for multiple degrees of freedom: One of the benefits of large touch screens or tabletop displays is the natural direct manipulation experience they provide, as well as their potential for more complicated interactions using multiple fingers. We leverage this aspect of touch-screens and tabletops by using the Visual Touchpad [26] as our base input device, since it allows two-handed multi-point input as well as the ability to transparently render live video of the hands onto the display for a direct manipulation experience from afar.

Fast targeting to any point on the display: Touch-screen and tabletop display users can randomly access any point on the display by simply touching the desired location. As described earlier this is difficult to do when a separate touchpad surface is much smaller than the display to which it is directly mapped. We address this issue by using asymmetric two-handed input so that the dominant hand performs fine positioning towards a target while the non-dominant hand coarsely positions the space of the dominant hand.

Maximize comfort for from afar interaction: While our goal is to allow a user to interact with a large wall display while remaining seated at a table, we must still consider any potential discomforts that our interaction techniques may introduce. This includes allowing the user to adjust the position of the touchpad surface as well as minimizing awkward gestures.

Support for multiple concurrent users: In a conference room setting, it would be desirable to allow more than one user to access the display without affecting the work of others.

In addition to these design goals, we also consider the design issues outlined by Kjeldsen and Hartman [24] for vision-based user interfaces. In particular, our interaction techniques should consider the intuitiveness and learning curve required to perform a motion or gesture, the stability required by a user to perform a task, and the multiplexing ability offered to a user during the process of an operation.

SYSTEM OVERVIEW

Display Hardware and Software

We use a 5m wide x 1.8m high rear-projection display consisting of a 3x6 projector array, where each projector is connected one-to-one with a 2GHz Pentium4 computer running at a desktop resolution of 1024x768 pixels. Using the open source Chromium library [15], any standard OpenGL application can be distributed onto the projector array so that the projectors act as one single large display of up to 6144x2304 pixels.

Hand and Touchpad Tracking

Our hand tracking system is based upon the Visual Touchpad (VTP) device described in [26], which allows two unmarked hands to be tracked over top of a black rectangular surface using two off-the-shelf web cameras placed above the work area. Using simple computer vision algorithms, the system outputs the 2D tip position and orientation of any outstretched finger. By rectifying the black rectangular region as seen from both camera views so that it is axis-aligned, simple stereo disparity can be used to determine the distance of each finger above the touchpad surface (where a disparity of zero means the finger is directly touching the touchpad). Therefore, the system effectively acts as a low-cost, multi-point, touch-sensitive input device. We currently use a simple piece of cardboard with a 60x20 cm black region that resembles the shape and aspect ratio of our large screen.

The major advantage of the VTP over other touch-sensitive devices is the ability to extract the entire 2D image of each hand, which allows for differentiating between fingers. Additionally, the actual hand images can be extracted and rendered independently onto the screen as a visual proxy of a user's actual hands, providing richer feedback than a standard mouse cursor or even a virtual hand.

One problem with the original VTP was its requirement that the camera positions be fixed with respect to the touchpad surface. This limits the mobility of the device, and also prevents the detection of multiple devices/users from a single camera pair. In order to facilitate the use of multiple VTPs as well as make the device somewhat mobile while on a desk, we use the ARTag library [8] which allows up to 2048 unique 2D identifiers to be detected quickly and accurately in our captured camera images. By attaching such tags above the black rectangular region on each VTP (Figure 1), we can uniquely identify a large number of users. Additionally, the tag detection allows us to localize the position of the black rectangular region quickly, allowing the entire touchpad to be moved while the cameras remain fixed. This allows users to position the touchpad comfortably during interactions, supporting our third design goal.

The system runs on a 2GHz P4 computer, sufficient for tracking two touchpads/users quickly (<50ms/frame) using two 320x240 pixel web cameras per touchpad.

Postures and Gestures

Figure 2 shows the set of static postures and temporal gestures that our system can infer. Note that each of these gestures can be overloaded based on whether or not a particular fingertip is making contact with the touchpad surface, or is tapping/double-tapping the surface.

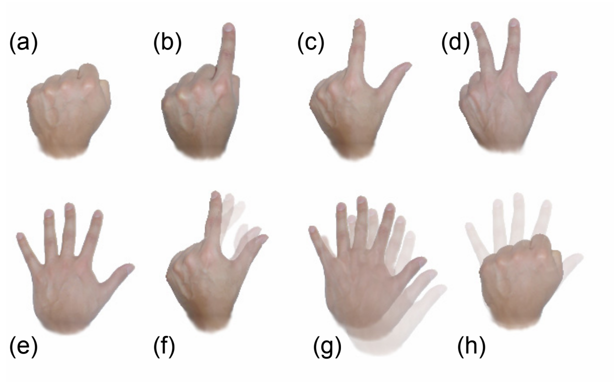

Figure 2. Postures and gestures recognized by our system. (a) Fist posture; (b) Pointing posture; (c) double-point posture; (d) triple-point posture; (e) five finger posture; (f) pinching posture; (g) five-finger slide gesture; (h) grabbing gesture.

INTERACTION TECHNIQUES

In the following sections we describe the bimanual interaction techniques that we have developed for fluidly interacting with large wall displays from afar. Without loss of generality, we assume that a user's right hand is the dominant hand while the left is the non-dominant hand.

Asymmetric Interactions

Asymmetric-dependent tasks, as proposed by Guiard [11], are those in which the dominant hand moves within a frame of reference that has been set by the non-dominant hand. In other words, the non-dominant hand can be engaged in coarse and less frequent actions, while the dominant hand will be used for faster, more frequent actions that require precision. It has been shown that such asymmetric-dependent tasks lead to the best performance due to their resemblance to the bimanual tasks humans perform in the real world [14, 19]. In this section we describe our asymmetric two-handed interaction techniques.

Coarse Positioning

Since allowing fast access to all parts of the screen is a fundamental issue in large display interaction, we have developed an asymmetric two-handed technique to address this problem.

Since the VTP can differentiate between the left and right hands, we are able to map the touchpad to the display differently for each hand. In asymmetric mode the left half of the touchpad is mapped to the four corners of the entire display (Figure 3a). Therefore, when the user makes a pointing gesture with the left hand index finger and touches the tip onto the touchpad surface, the corresponding position in display space is computed and the segmented video image of the left hand is instantly moved to that location. A panning icon also appears at the left index fingertip to denote that the finger can also be moved along the surface of the touchpad for smooth panning (Figure 8a). While this allows random access to almost any part of the display similar to a touch-screen, fine positioning is difficult due to the resolution differences between the touchpad and the display. In other words, mapping half of the 60cm width of our touchpad to the entire 5m width of the display means that even a 1cm change in the fingertip position results in a 16cm jump on the display. Additionally, our cameras introduce further inaccuracies depending on the capture resolution of the cameras (we currently capture at a resolution of 320x240).

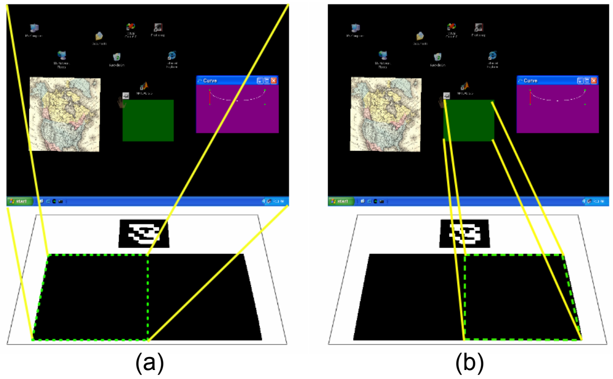

Figure 3. Touchpad mapping for asymmetric interactions for: (a) the left hand; (b) the right hand.

Workspaces and Fine Positioning

Following Guiard's asymmetric-dependent principles, we place a green-colored, semi-transparent, rectangular workspace at the left index finger position, with the right hand rendered inside of this workspace (Figure 4). Thus the right hand can be used to perform more accurate positioning and manipulation tasks *inside* of this workspace, while the left hand coarsely positions the entire workspace anywhere on the display. For such right hand interactions, the right half of the touchpad is mapped to the four corners of the workspace (Figure 3b). This configuration minimizes any interference that may occur if the hands begin to overlap.

Using this combination of two-handed coarse and fine positioning, a user can quickly access any part of the display with ease, which supports our second design goal.

Selecting/Moving/Rotating Single Objects

Kjeldsen and Hartman [24] suggest that direct pointing, control, and selection tasks are well-suited to vision-based hand tracking interfaces due to their low learning curve compared to systems based on complex gesture sets. We leverage this knowledge for the purpose of manipulating objects in our system.

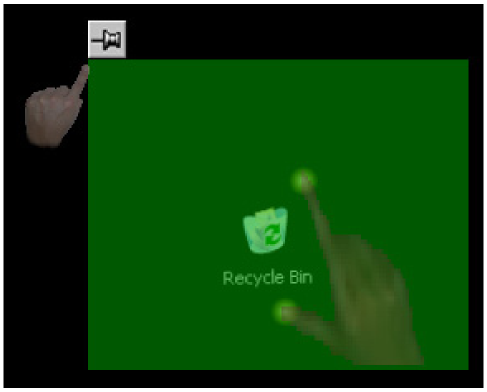

Figure 4. A workspace that can be coarsely positioned using the left hand (shown at top-left of the semi-transparent overlay), while the right hand performs fine manipulations inside of it.

To select an object inside of the workspace, a pointing gesture is made with the right index finger. When contact is made with the touchpad surface, any object underneath the on-screen fingertip becomes selected. In effect, the right hand in a pointing gesture can perform any operation that a single-button mouse could perform, where clicking is simulated by making contact with the touchpad surface. With the right hand, any selected object can then be moved locally within the workspace by simply moving the finger across the surface of the touchpad. This allows for precise positioning of the object. Additionally, objects can also be rotated if desired by using the finger orientation information provided by our tracking cameras, as described in [26].

To quickly move selected objects to areas outside of the workspace, the user can hold an object with the right index finger while the left index finger is used to move the position of the workspace as described earlier. The selected object will remain attached to the right index finger and thus remains within the workspace as it moves, thereby allowing the object to be coarsely placed anywhere on the screen quickly, but without interfering with any precision movements being carried out by the right hand In other words, the right hand does not have to transition between coarse and fine positioning as might be required in single hand techniques for large display interaction. Figure 5 illustrates this interaction.

Selecting Multiple Objects

In traditional graphical interfaces, selecting multiple objects such as icons usually requires dragging a box around a group of objects using a mouse button. For multiple random selections, however, a user is typically required to use a modifier key on the keyboard to individually select each desired object. While we can simulate such selections using a second finger as a modifier, we propose an alternative approach that leverages the high degree of freedom input provided by our tracker. By making a five-finger grabbing gesture with the right hand as shown in Figure 6, the object closest to the centre of the palm of the hand is "grabbed" and disappears from the workspace. To help visualize which object will be grabbed, a line is drawn from the centre of the hand to the closest object. Repeating this for a number of objects, a large number of randomly placed objects can be selected quickly and precisely

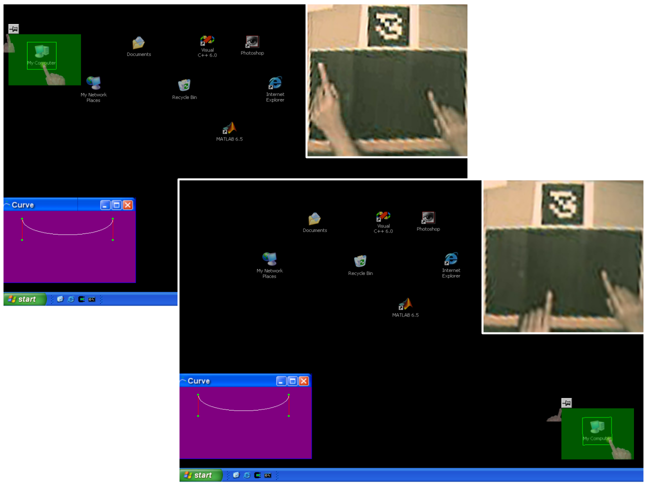

Figure 5. An example of fast object movement using two-hands. The icon at the top left of the display that is being held with the right hand is instantly moved to the bottom right using the left hand.

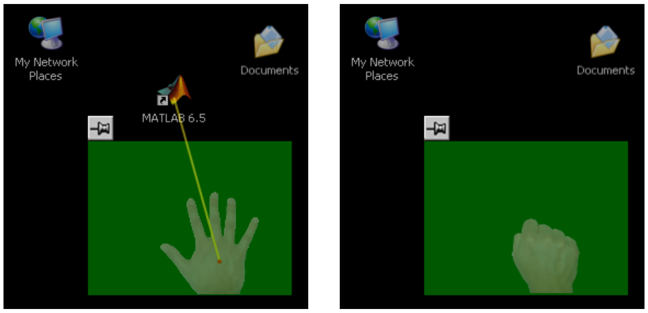

Figure 6. Grabbing the object closest to the hand.

As multiple objects are grabbed, they are placed in a last-in first-out queue at each of the fingertips starting from the thumb and progressing in order to the pinky finger. To place these objects back into the workspace the user can make and hold a five-finger gesture above the touchpad surface. When this is done, the objects assigned to each of the fingers are displayed over top of the hand image on-screen, in LIFO order from the tip (Figure 7). Therefore, by tapping one or more fingers onto the touchpad, the object closest to the tip of the tapped finger(s) will be placed back into the workspace at the tapped location. With five fingers a user can easily grab up to 15 objects without cluttering the display using our system. However, this will vary based on the size at which icons are rendered.

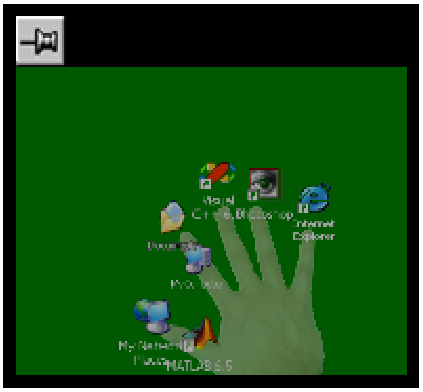

Figure 7. Placing multiply selected objects. The selected icons appear on each finger based on selection order. Tapping a finger releases the icon closest to the tip.

Resizing/Zooming/Rotating Workspaces

By default, the workspace is set to a size such that every pixel on the display can be reached using a combination of coarse and fine positioning. However, since the right hand operates in a space where the right half of the touchpad is mapped to the corners of the workspace, the user is limited to a granularity of a pixel. For precise object positioning this is ideal, but in some instances it might be desirable to work at a different granularity with the right hand.

To facilitate such instances, the left hand can be used to modify properties of the workspace. To resize the workspace, the user makes a pinching gesture with the left hand. A resizing widget then appears between the thumb and index finger of the on-screen representation of the hand to signify that a resize can be performed (Figure 8b). By increasing the distance between the two fingers, the workspace grows in both the horizontal and vertical directions (up to some predefined maximum size). Similarly, decreasing the distance between the fingers causes the workspace to shrink to some minimum size).

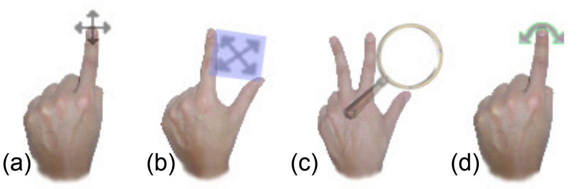

Figure 8. Widgets drawn beside the on-screen representation of the hand for modifying workspaces: (a) Panning; (b) Resizing; (b) Zooming; (d) Rotation.

Increasing the workspace size reduces the granularity with which the right hand operates, while decreasing the size increases the granularity. To counter the effect of a resize operation, the user can also modify the zoom level of the workspace. By placing the left hand in a triple-pinch-posture with all fingers touching the touchpad surface, a zoom lens widget appears between the left hand's thumb and index finger (Figure 8c). Raising the left index finger off the surface then causes a non-linear zoom-in of the workspace towards the center, where the speed of the zoom depends upon the amount of time the finger is held above the surface. Similarly, raising the left thumb instead of the index finger causes a non-linear zoom-out to be performed. By zooming out to a level below the default zoom setting, the workspace can provide a dollhouse [30] view of the entire display contents. This allows for fast access to any item on the screen, albeit in a smaller form, which can be useful in certain situations.

Finally, workspaces can also be rotated by extracting the left index finger orientation during a pointing posture held above the surface. We assume that if the finger is generally pointing in the vertical direction of the touchpad, no rotation should be performed. However, if the direction falls below -10 degrees then the workspace begins to rotate in the counter-clockwise direction. Similarly if the finger direction is above +20 degrees the workspace rotates in the clockwise direction. In both cases, a rotation dial appears at the tip of the left index finger to signify the mode change (Figure 8d). This allows workspaces to be positioned with the left hand as one would adjust a piece of paper in real life before writing on it. This allows a user to better position the right hand in order to more precisely manipulate an object. To avoid awkward orientations, however, we limit the amount of workspace rotation to +/-45 degrees from the vertical direction. Note that we chose unbalanced rotation thresholds since the left index finger generally points in the +5 degree direction from the vertical during typical touchpad usage. These should be reversed for left handed users.

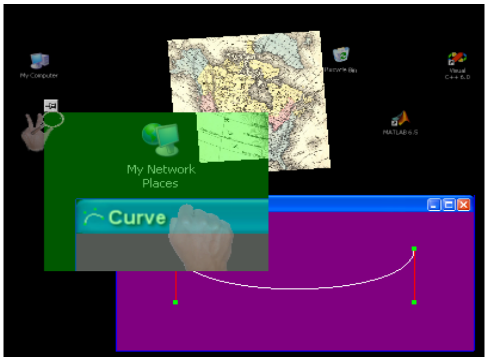

Figure 9. Zooming a workspace using three-fingers.

Subsequent movements of the workspace maintain the size, zoom level, and rotation settings that have been set, thereby mimicking the functionality of magic lenses as proposed by [4]. By combining resizing, zooming, and rotation operations, a user can work on different parts of the display with the desired amount of visual feedback and positioning granularity (Figure 9). These operations support our design goal of maintaining comfort for the user, since they allow the user to place a workspace and the right hand into a maximally efficient pose.

Pinned Workspaces

In many large display applications, a user may need to frequently move between two (or more) completely different regions of the screen. If the user desires working in each of these regions at different granularities, this would require constant zooming and resizing operations after each move. To remedy this problem, we allow workspaces to be pinned so that their position, size, and zoom setting are locked. To do this a user makes a double tap gesture with the left index finger in a pointing posture. This toggles the workspace to pinned mode, causing the right hand to become locked inside of the pinned workspace. An icon at the top-left of the workspace depicts the pinned/unpinned state of the workspace. The previously described interactions can then be performed inside of this pinned workspace as usual. If the left hand is again placed in a pointing posture, a transparent "ghost" workspace is shown emanating from the left index finger position. As the left index finger is moved further away from the top-left of the pinned workspace, the ghost workspace becomes more opaque up until the overlap between the ghost workspace and the pinned workspace falls below 25%. At this point, the ghost workspace becomes the active workspace, and the right hand smoothly transitions into the active workspace. The pinned workspace remains at its original location, but right hand operations can now be performed inside of the active workspace as before. The active workspace can then be pinned elsewhere to create other pinned workspaces. If the active workspace is brought back towards a previously pinned workspace, and the overlap is greater than 25%, the active workspace becomes a ghost workspace once again and the right hand transitions into the pinned workspace (Figure 10). In this manner, a user can quickly move between different parts of a large display without worrying about size or zoom settings. Additionally, single or multiple object selections can also be made between pinned workspaces. To delete a pinned workspace, the user can simply move into the workspace's area and then double tap with the left index finger. This removes the pinned workspace, and the ghost workspace becomes the active workspace. The concept of multiple workspaces combined with the asymmetric movement techniques further supports our second design goal of allowing fast access and targeting to all parts of the display, while simultaneously achieving our first design goal of leveraging both hands and multiple fingers effectively.

Symmetric Interactions

For certain tasks, a user may want to perform symmetric bimanual manipulations where both hands perform very similar functions in synergy. In this section we describe how we smoothly transition between asymmetric and symmetric interactions to support common symmetric tasks.

Transitioning from Asymmetric to Symmetric Interaction

By default, the system supports asymmetric interactions, where the left hand is rendered at the top left of the active workspace as a small multi-point cursor. To perform fine operations with the left hand in a manner similar to the right, the user first makes a five-finger sliding gesture (Figure 2g) towards the bottom-right corner of the touchpad. This causes the left hand to smoothly transition *into* the workspace so that the mapping for both the left and right hands is such that the four corners of the touchpad correspond to the four corners of the workspace (Figure 11). To transition back to asymmetric interaction a five-finger sliding gesture is again made with the left hand, but towards the top-left of the touchpad.

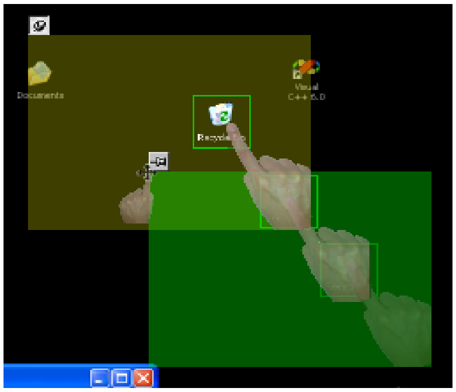

Figure 10. Transitioning into a pinned workspace. The hand smoothly transitions into the pinned workspace, taking any selected objects along with it.

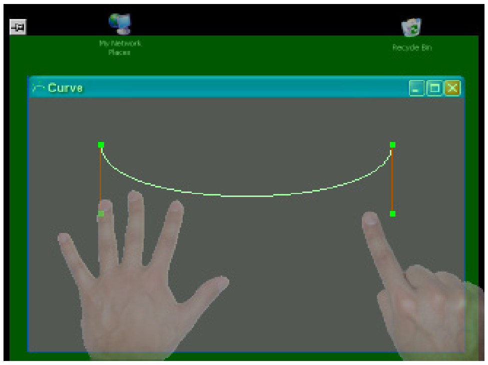

Figure 11. Mapping both hands into the same workspace for symmetric bimanual tasks.

DISCUSSION

Although we have not yet performed a formal evaluation of our interaction techniques, a number of graduate students in our research lab were asked to try the system in order to gauge some early feedback on its strengths and weaknesses. Each user was first given a 5 minute introduction to the interaction techniques, followed by 10 to 15 minutes of free experimentation time.

All users were quick to point out that the basic movement and selection techniques were very intuitive, largely due to each user's familiarity with touch-surfaces, tabletop displays, and/or tablets. Additionally, every user found the rendering of the hands on the display (along with the appropriate widgets and overlays) to be very compelling as well as informative, more so than the cursors typically used in large display interaction. Graham and MacKenzie [10] compared physical pointing tasks to virtual pointing tasks and found no difference in the initial movement times, but they did find that a user's ability to close-in on a virtual object was slower than for real objects. It's not clear that this directly applies to our system, since our rendered hands are neither real nor virtual in the abstract sense, but rather an accurate visual proxy of a user's real hands. We plan in the near future to more formally investigate the value of using such live hand images as is done in our system. Kirsh and Maglio [22] argued that certain cognitive and perceptual tasks are better solved by doing things in the real world as opposed to solving them mentally. For example, they describe the frequent translations and rotations that users perform on Tetris pieces as an example of users trying to gain a better understanding of the situation of the entire puzzle. Such *epistemic* actions can thus be used to uncover information about a problem that may be hard for a person to understand or solve completely in the head [22]. It is worth investigating whether or not the slight gesturing we perform with our hands in the real world to solve geometric problems also falls into the domain of epistemic actions, and if so, how it might apply to our large display interactions. This could provide some insight as to whether showing actual hands on the screen provides more than just a compelling experience for the user.

The use of the left index finger for coarse positioning of the workspace was found to be very intuitive by all users. However, some users felt that the default precision at which the right hand could manipulate objects was too coarse, thus requiring them to either reduce the size of the workspace or increase the zoom. This could be remedied by either using higher resolution cameras for the hand tracker or by moving the cameras closer to the touchpad surface. However, increasing the resolution would also increase the processing time as well as introduce noticeable lag on current CPUs.

While the workspace resizing gesture was found to be conceptually easily understood, one user complained that the three-finger gestures for zooming in and out were difficult and that the two-finger pinching gesture would be preferred for zooming. Unfortunately this would lead to an ambiguity with the current resizing gesture. Interestingly, Balakrishnan and MacKenzie [1] showed that a pinching posture where the thumb and index finger work together provides a higher bandwidth input than using a single index finger. In a similar manner, it would be useful to determine what input bandwidth could be had from the three-finger gesture, since this could allow us to optimize the gesture for other more suitable operations.

The multi-point grabbing gesture was well received by all users, but the queue-based placement gesture received mixed reviews. Many users found that placing objects precisely with the ring finger and pinky finger was difficult since both of these fingers are difficult to control independently from one another. As a result, attempting to place an object from one finger would sometimes inadvertently also place the object from the other finger. This leads us to believe that these two fingers should not be used for independent precision tasks, but rather as a group modifier for the remaining fingers' tasks. This, however, needs further analysis to be confirmed. Another problem users had with the placement gesture was the queue arrangement. Users felt that they shouldn't be required to think ahead about the order of object placement during the grabbing phase, which the LIFO queue forced them to do. One interesting suggestion was to allow a user to use their left hand to rearrange the ordering of objects in each finger.

While our interaction techniques currently don't provide any special support for collaborative tasks, an interesting side effect of using pinned workspaces in our system is their automatic support for multiple concurrent users. By pinning a workspace a user is effectively asking for exclusive access to a portion of the display. Therefore we can restrict other users from accessing a pinned lens that already has a user inside of it in a manner that is conceptually similar to the "carved" regions described in the Dynamo system [16]. This further supports our design goal of allowing multiple concurrent users to interact without interference on the same display.

One unexpected feature of our transparent workspaces is their automatic "spotlight" functionality [20]. Using a combination of workspace positioning with the left hand, pointing with the right hand, and speaking out loud, users could easily sway the attention of a small audience to a certain part of the large display extremely quickly. We plan to leverage this feature in the future more directly.

While the vision-based aspect of our system provided us with functionality that cannot be performed on standard touch-sensitive hardware (such as multi-point finger detection without ambiguity, as well as a full 2D hand image), there are still a few downfalls. In particular, the arch-enemy of vision algorithms is darkness. As a result, darkening a conference room is not an option if one plans to use our system for interaction. Aside from table lamps there is no quick solution to this problem. Ultimately we imagine that one could build a stand-alone vision-based device that consists of a glass touch surface, with an array of infrared cameras and lights embedded underneath the glass. This would not only resolve the issue of dark rooms, but it would also remove the need to have cameras hanging overhead in every room where one would want to use the system. A related problem with most vision algorithms is the difficulty when segmenting objects under varying lighting conditions and shadows. Since we use simple background subtraction to extract the hands from above a black surface, our system is quite reliable under a wide range of illumination conditions. Another possibility is to combine the strengths of our system with a touch-sensitive surface such as the DiamondTouch or SmartSkin for more robustness. The TactaPad by Tactiva [31] is one recent step in this direction, since they combine a touch-sensitive surface with a single camera that can replace a mouse cursor with a silhouette of a user's actual hands.

Vision-based tracking algorithms that track local features from frame-to-frame are prone to tracking failure when features become lost. Since the Visual Touchpad does not use any temporal information to extract fingertip positions, fast hand movements do not cause the tracker to fail. However, depending on the quality of the cameras, fast movement may cause motion blur which does confuse the fingertip detector. In such situations, the system simply uses the last valid fingertip positions and attempts to recover the actual positions in subsequent frames (before some predefined timeout expires).

Finally, our current setup places two 320x240 cameras high enough above the work area so that two touchpads can be detected accurately. Since our hand tracker requires a large amount of processing time, we have found that detecting more than two users seriously affects both speed and tracking accuracy. To detect more users we suggest adding extra machines and camera pairs, and then exchanging hand position information with a central node that manages a shared application. However, as processing power continues to increase, a single machine will be able to handle more than two users as well as higher resolution cameras.

CONCLUSIONS AND FUTURE WORK

This work investigated a number of techniques for interacting with large displays from afar using a vision-based hand and touchpad tracking system. By allowing users to sit comfortably at a table in front of a large display, traditional selection and navigation techniques become inefficient and other more appropriate methods must be developed. We presented a set of such approaches that leverage people's natural abilities to manipulate real-world items with their hands asymmetrically. Our current design satisfies our original design principles of: (1) leveraging two hands and multiple fingers for both natural and high degree of freedom input, (2) allowing fast targeting to any part of the display, (3) maximizing comfort for from afar interactions, and (4) supporting multiple users.

In the future, we would like to investigate how to further integrate multiple users onto a large display using our system. In particular, with the high degree of freedom input provided by two hands, it would be interesting to investigate what sort of collaborative tasks could be performed by two or more users working together. Another fruitful direction for research might be to investigate how vision algorithms could be further leveraged for tasks other than just detecting hands. In a manner similar to the DigitalDesk [32], we could very easily place other objects onto the touchpad surface such as documents or other tangible objects, and then project them onto the large display. This opens up the possibility of using real tools to perform virtual tasks in more natural ways.

ACKNOWLEDGEMENTS

We thank John Hancock, Anand Agarawala, Joe Laszlo, and Bowen Hui for assistance with video production, and members of the DGP lab at the University of Toronto (www.dgp.toronto.edu).

REFERENCES

1. Balakrishnan, R., and MacKenzie, I. S. (1997). Performance Differences in the Fingers, Wrist, and Forearm in Computer Input Control. In *Proceedings of ACM CHI Conference*. p. 303-310.

2. 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**.** In *Proceedings of Interact.* p. 57-64.

3. Bezerianos, A and Balakrishnan, R. (2005 – in press). The Vacuum: Facilitating the Manipulation of Distant Objects. In *Proceedings of ACM CHI Conference.*

4. Bier, E. A., Stone, M. C., Pier, K., Buxton, W., DeRose, T. D. (1993). Toolglass and Magic Lenses: The See-through Interface. In *Proceedings of ACM SIGGRAPH*. p. 73-80.

5. Cao, X., and Balakrishnan, R. (2003). VisionWand: Interaction Techniques for Large Displays Using a Passive Wand Tracked in 3D. In *Proceedings of ACM UIST Symposium*. p. 173-182.

6. Davis, J., Chen, X. (2002). Lumipoint: Multi-User Laser-Based Interaction on Large Tiled Displays. In *Displays*, Volume 23, Issue 5.

7. Dietz, P., and Leigh, D. (2001). DiamondTouch: A Multi-user Touch Technology. In *Proceedings of ACM UIST Symposium.* p. 219-226.

8. Fiala, M. (2004). ARTag Revision 1, A Fiducial Marker System Using Digital Techniques. *NRC/ERB-1117 Technical Report*. National Research Council of Canada.

9. Geißler, J. (1998). Shuffle, Throw or Take It! Working Efficiently with an Interactive Wall. (1998). In *ACM CHI Conference, Extended Abstracts*. p. 265-266.

10. Graham, E., MacKenzie, C. (1996). Physical versus Virtual Pointing. In *Proceedings of ACM CHI Conference*. p. 292-299.

11. Guiard, Y. (1987). Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. In *Journal of Motor Behavior*, 19, p. 486-517.

12. Guimbretière, F., Stone, M., Winograd, T. (2001). Fluid Interaction with High-resolution Wall-size Displays. In *Proceedings of ACM UIST Symposium*. p. 21-30.

13. Hardenberg, C. V., Bérard, F. (2001). Bare-Hand Human-Computer Interaction. In *Proceedings of the Workshop on Perceptive User Interface*.

14. Hinckley, K., Pausch, R., Proffitt, D., Patten, J., Kassell, N. (1997). Cooperative Bimanual Action. In *Proceedings of ACM CHI Conference*. p. 27-34.

15. Humphreys, G., Houston, M., Ng, R., Frank, R., et al. (2002). Chromium: A Stream-Processing Framework for Interactive Rendering on Clusters. In *ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH*. p. 693-702.

16. Izadi, S., Brignull, H., Rodden, T., Rogers, Y., Underwood, M. (2003). Dynamo: A Public Interactive Surface Supporting the Cooperative Sharing and Exchange of Media. In *Proceedings of ACM UIST Symposium*. p. 159-168.

17. Johanson, B., Fox, A., and Winograd, T. (2002). The Interactive Workspace Project: Experiences with Ubiquitous Computing Rooms. In *IEEE Pervasive Computing*. p. 67-74.

18. Johanson, B., Hutchins, G., Winograd, T., Stone, M. (2002). Pointright: Experience with Flexible Input Redirection in Interactive Workspaces. In *Proceedings of ACM UIST Symposium*. p. 227-234.

19. Kabbash, P., Buxton, W., and Sellen, A. (1994). Two-handed Input in a Compound Task. In *Proceedings of ACM CHI Conference*. p. 17-423.

20. Khan, A., Matejka, J., Fitzmaurice, G., Kurtenbach, G. (2005 – in press). Spotlight: Directing Users' Visual Attention on Large Displays. In *Proceedings of the ACM CHI Conference*.

21. Khan, A., Fitzmaurice, G., Almeida, D., Burtnyk, N., Kurtenbach, G. (2004). A Remote Control Interface for Large Displays. In *Proceedings of ACM UIST Symposium*. p. 127-136.

22. Kirsh, D., & Maglio, P. (1994). On Distinguishing Epistemic from Pragmatic Action. In *Cognitive Science,* 18. p. 513-549.

23. Kirstein, C. and Muller, H. (1998). Interaction with a Projection Screen using a Camera-tracked Laser Pointer. In *Proceedings of Multimedia Modeling*. p. 191.

24. Kjeldsen, R., and Hartman, J. (2001). Design Issues for Vision-based Computer Interaction Systems. In *Proceedings of the Workshop on Perceptive User Interfaces*.

25. Kurtenbach, G., Fitzmaurice, G., Baudel, T., Buxton, W. (1997). The Design of a GUI Paradigm based on Tablets, Two Hands, and Transparency. In *Proceedings of ACM CHI Conference*. p. 35-42.

26. Malik, S. and Laszlo, J. (2004). Visual Touchpad: A Two-handed Gestural Input Device. In *Proceedings of the ACM International Conference on Multimodal Interfaces*. p. 289-296.

27. Mynatt, E., Igarashi, T., Edwards, W., LaMarca, A. (1999). Flatland: New Dimensions in Office Whiteboards. In *Proceedings of ACM CHI Conference*. p. 346-353.

28. Pederson, E., McCall, K., Moran, T., Halasz, F. (1993). Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings. In *Proceedings of ACM CHI Conference*. p. 391-398.

29. Rekimoto, J. (2002). SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. *In Proceedings of ACM CHI Conference*. p. 113-120.

30. Swaminathan, K, and Sato, S. (1997). Interaction Design for Large Displays. *Interactions,* Volume 4, Issue 1.

31. Tactiva. (2005). TactaPad. http://www.tactiva.com.

32. Wellner, P. (1993). Interacting with Paper on the DigitalDesk. In *Communications of the ACM*, Volume 36, Issue 7. p. 87-96.

33. Wilson, A. (2004). Touchlight: An Imaging Touch Screen and Display for Gesture-Based Interaction. In *Proceedings of the International Conference on Multimodal Interfaces*. p. 69-76.

34. Wu, M., and Balakrishnan, R. (2003). Multi-finger and Whole Hand Gestural Interaction Techniques for Multi-user Tabletop Displays. In *Proceedings of ACM UIST Symposium.* p. 193-202.