

Perspective Cursor: Perspective-Based Interaction for Multi-Display Environments

Miguel A. Nacenta, Samer Sallam, Bernard Champoux, Sriram Subramanian, and Carl Gutwin

Computer Science Department, University of Saskatchewan

110 Science Place, Saskatoon, SK, S7N 5C9

{nacenta,sallam,champoux,sriram,gutwin}@cs.usask.ca

ABSTRACT

Multi-display environments and smart meeting rooms are now becoming more common. These environments build a shared display space from variety of devices: tablets, projected surfaces, tabletops, and traditional monitors. Since the different display surfaces are usually not organized in a single plane, traditional schemes for stitching the displays together can cause problems for interaction. However, there is a more natural way to compose display space – using perspective. In this paper, we develop interaction techniques for multi-display environments that are based on the user’s perspective of the room. We designed the Perspective Cursor, a mapping of cursor to display space that appears natural and logical from wherever the user is located. We conducted an experiment to compare two perspective-based techniques, the Perspective Cursor and a beam-based technique, with traditional stitched displays. We found that both perspective techniques were significantly faster for targeting tasks than the traditional technique, and that Perspective Cursor was the most preferred method. Our results show that integrating perspective into the design of multi-display environments can substantially improve performance.

Author Keywords

Multi-display interaction techniques, direct-manipulation interfaces, laser pointing, multi-monitor environments.

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces - *Interaction Styles; Input devices and strategies; Theory and methods.*

INTRODUCTION

Computing environments with many and diverse displays are becoming common. The archetype of these multi-display environments is the Smart Office, where it is usual to see interconnected tablets, wall-mounted displays,

laptops, and projected surfaces all being used concurrently and cooperatively. Some research projects have already explored and highlighted the benefits of multi-display working environments [11, 30, 31].

Many desktop interaction techniques do not work well in these new systems because they do not deal with either the discontinuity inherent in multi-display interaction or the intrinsic characteristics of different-sized displays [27, 32]. Recent research has tried to address problems related to display heterogeneity by creating specific interaction techniques for each display type: for small displays [12, 34], large displays [1, 15, 32, 20], interactive table-top surfaces [10, 25, 33] and multi-display situations [2, 17, 18, 26, 27].

While each of these techniques generally work well for certain display configurations and input devices, we can not expect users to adapt their interaction styles every time that they switch displays. Moreover, the transition in control from one display to another should be seamless enough that inter-display interactions don’t produce a significant overhead. We believe that it is possible to provide seamless control over multiple displays by using the spatial relationships between the display’s surfaces and the user – that is, by using *perspective*.

Perspective-based multi-display interaction techniques are techniques in which the position and orientation of each display relative to the user determines how control is applied. For example, control-to-display ratio could be dynamically modified to provide more control resolution for displays that are closer to the user, since users usually need more control in displays that can be seen in more detail. Perspective helps to solve several of the problems arising from controlling several heterogeneous displays from one input device. For example, perspective adapts naturally to different levels of required resolution, to the different visibilities and sizes of the displays, and to situations where one display overlaps another.

The best known perspective-based technique is the laser pointer. If we use a laser beam as control for the pointer in a multi-display system, for example, we will be able to act on the different displays with different control resolutions depending on how far away we are. However, laser pointer

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2006, April 22–28, 2006, Montréal, Québec, Canada.

Copyright 2006 ACM 1-59593-178-3/06/0004...\$5.00.

techniques suffer from poor accuracy and stability [21, 24], and can be very tiring for sustained interaction.

To overcome the limitations of existing techniques, we designed the Perspective Cursor: a novel perspective-based interaction technique that uses a relative-positioning input device (e.g., a mouse) together with the user's point of view to determine which displays are contiguous in the user's field of view.

In this paper, we present a study that compared Perspective Cursor, Virtual Beam (an implementation of the laser beam concept) and the standard interaction technique for multiple displays (Stitching of control spaces), which does not use perspective. We found that perspective-based techniques offer an intrinsic performance advantage for pointing tasks that involve several displays, particularly when the displays are overlapping or at large angles to one another. We also found that Perspective Cursor outperforms Virtual Beam due to the stability of the relative control device. Our results suggest that perspective-based techniques can be extremely valuable for providing seamless interaction in multi-display environments.

In the rest of this paper we review previous work on current interaction techniques for pointing, then introduce the concept of perspective for multi-display environments, and present the design of the Perspective Cursor. We then report on the empirical study and discuss the implications of our findings for the design of multi-display spaces.

RELATED WORK

The problem of multi-display interaction has been addressed in many different ways in past research. Some multi-display techniques allow the user to perform special gestures or input commands that, when issued in the one display's input space, control or import objects from other displays. Examples of these techniques are Pick-and-drop [26], Sync-tap [28] and Stitching [17]. Another variant of the same idea consists in performing a gesture with the devices themselves (e.g., bringing two tablets together), which virtually connects the displays [31, 16, 19].

However, these techniques normally need physical access to the different displays, which might not be possible. Laser beam and finger-pointing techniques [7, 9, 20, 25] provide an alternative for interacting with distant or inaccessible displays. Another straightforward solution to the same problem is to provide a more or less faithful virtual representation of the actual display setting that can then be manipulated from a local device [6, 22, 32].

A last group of techniques for multi-display interactions use the input devices originally associated with one display or device to remotely control another. For example, a mouse could be used across several displays [5, 8, 18], the movements of the mouse or a pen could be amplified to extend to other displays [13, 15, 22, 27], or the controls in one device could act as a remote control for a distant display [30].

MAPPING CONTROL SPACE TO SEVERAL DISPLAYS

One of the biggest challenges in the design of multi-display systems is to provide a way to support direct manipulation of different physical surfaces (displays) with interaction techniques that offer seamless control [11]. One way to do this is to distribute parts of the control space of an input device among the different displays, as is currently done in mouse-controlled multi-display machines.

Many operating systems allow users to configure how the control space of one mouse is assigned to several displays. The cursor goes from the edge of one display to the border of another; this is nothing more than a partitioning of the mouse position into zones for the control of the different displays (clutching not considered). This assignment of control space to displays in a more or less arbitrary way is what we call *Stitching of control spaces*. Figure 1 shows multi-screen configuration utilities from MS Windows® and MacOS X®, which allow the user to tell the system about the spatial configuration of the monitors, making the transitions relatively intuitive [29].

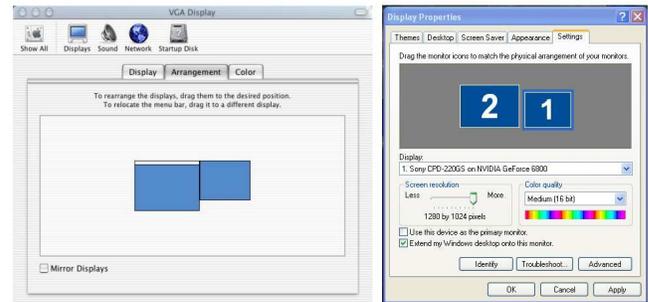


Figure 1. Multi-monitor set-up dialog of two operating systems

Stitching the virtual spaces like this, however, has three main setbacks: configuration, control resolution and spatial consistency.

Configuration

Stitching of control spaces is semi-static, which means that every time that the physical location of the displays is changed, the virtual position has to be reconfigured. This might be unimportant for desktop settings such as that of Figure 2, but is relevant in highly dynamic environments like a meeting room, where there might be tens of mobile displays (e.g., Laptops, PDAs etc.) This problem has been addressed in the past by providing the stitching of spaces in an explicit way [16, 17], or by using interaction techniques that require a device to be activated on the affected displays [26, 28]. However, these kind of explicit gestures require time and/or physical access to both surfaces, which is not always the case (e.g., very big or distant displays).

Control resolution

Commercial versions of the stitched spaces do not account for differences in resolution of the displays, or inter-display spaces (the frames and the distance between displays). It has already been shown that a more accurate virtual

representation of the physical space provides better performance in multi-display interaction [2, 29].

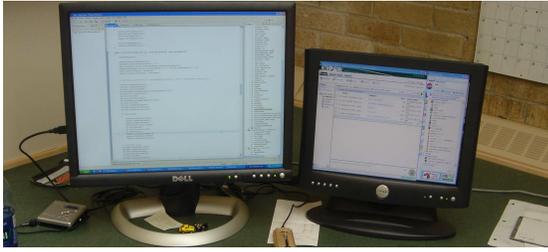


Figure 2. A typical multi-monitor setting

Spatial consistency

It is not possible to provide a consistent stitching of spaces if the screens are not all aligned in the same plane [6]. An example of this is shown in Figure 3, where the logical stitching is different depending on the position of the user. This is just a consequence of the fact that stitching spaces together is a simplistic way of mapping a 3D space into a flat 2D control space. It is very difficult to do that mapping in a meaningful way when there are several displays at non-orthogonal angles, or when displays overlap as in Figure 3.

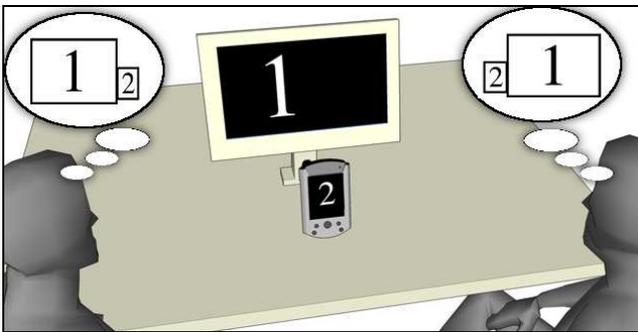


Figure 3. The optimal stitching of the two displays depends on the point of view.

PERSPECTIVE IN DISPLAY CONTROL

Perspective is defined as the appearance to the eye of objects in respect to their relative distance and positions. Perspective-based interaction techniques are multi-display techniques that use information about the location, orientation and distance of displays in the environment, relative to the point of view of the user, in order to provide control that is better adapted to what the user actually perceives.

Imagine that we want to provide control to all the displays of a meeting room with several projected screens, traditional monitors, and mobile displays. It would be difficult to make an assignment of control space to the tens or hundreds of potential configurations of this environment; but by incorporating the idea of perspective, we can solve many of the problems created by the multiplicity, heterogeneity, and reconfigurability of these displays.

For example, selecting a particular screen by pointing at it with a laser pointer is not a problem because humans are used to pointing, and we understand how the properties of 3D space and the orientation of the laser are going to affect the projection of the red dot. In this case, the control space (i.e., all the possible angles of the laser) is naturally distributed among all visible displays according to perspective – that is, according to the spatial relationships (orientation, distance, position) of each display to the beam.

If a screen is very near, we can rotate the laser in a wider angle without moving out of the screen, which means more resolution of control inside that screen. This is also fairly natural in the sense that we normally require more resolution and detail in objects that we can see better because they are closer, and conversely, we don't want too much resolution if we want to get the 'big picture' on a distant display.

It is reasonable to think that we will require more precise control for elements that are more visible because they are closer (i.e., they occupy a wider visible area); thus visibility and control are coupled. Laser-beaming naturally takes advantage of this relationship, unlike Stitching of control spaces, which keeps resolution constant regardless of the distance or the orientation of the display.

One problem with laser beams or finger pointing techniques is that they are, in general, not very precise. Human motor abilities constrain accuracy in pointing, forcing designers to filter the signal or implement interaction techniques with dwell time or other workarounds [20, 21, 24]. In addition, laser beam and finger pointing can be very tiring in certain situations, such as when we need to keep the pointer on the screen for a long time.

PERSPECTIVE CURSOR

The current state of the art makes choosing between accuracy and seamlessness a tradeoff. We can either have the seamlessness of the laser beam without the accuracy, flexibility and convenience of the mouse, or a mouse-based interaction technique that is accurate but uses a non-intuitive mechanism to stitch the spaces.

We overcome these two limitations in Perspective Cursor, a new perspective-based interaction technique that uses a relative positioning input device (e.g., a mouse or a trackball) together with the user's point of view to determine how displays are located in the field of view.

Perspective Cursor works as follows. We obtain in real time the 3D position coordinates of the head of the user (but *not* the orientation or the gaze direction) and at the same time, we maintain a three-dimensional model of the whole environment, with the actual position of all the screens. The model, together with the point-of-view coordinate of the user's head, lets us determine which displays are contiguous in the field of view, something very different to displays actually being contiguous in 3D space (Figure 4A).

The position and movement of the pointer is calculated from the point of view of the user, so that the user perceives the movement of the pointer across displays as continuous, even when the actual movement of the pointer considered in three dimensional space is not. Figure 4 shows several inter-display transitions that illustrate how the pointer moves between two displays when the displays are A) contiguous from the point-of-view of the user, B) overlapping from the point-of-view of the user and C) separate from the point-of-view of the user.

As can be observed in figure 4.C, the pointer travels through the empty space to get from one display into the next. Actually, the cursor can be in any position around the user, even if there is no screen there to show the graphical representation of the cursor.

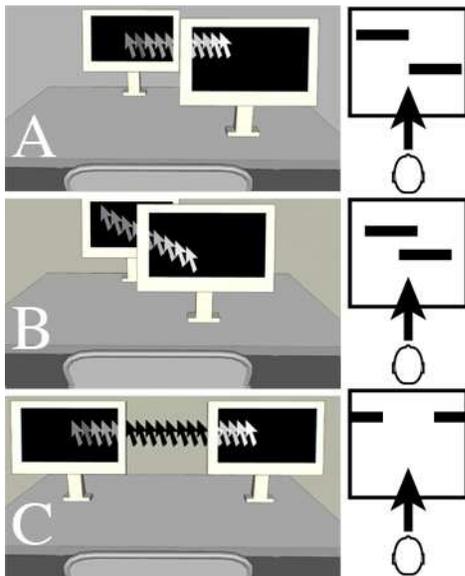


Figure 4. Examples of display transitions of Perspective Cursor. A) The displays are in different planes, but appear contiguous to the user. B) Displays that overlap each other. C) The cursor travels across the non-displayable space to reach the other display (the black cursors are only illustrative)

There are not many environments in which the users are completely surrounded by displays, meaning that users might lose the pointer in non-displayable space. The solution that we implemented is a perspective variant of halos [3]. Halos are circles centered on the cursor that are big enough in radius to appear, at least partially, in at least one of the screens. By looking at the displayed part of the circle, its position and its curvature, the users can tell how far and in which direction the Perspective Cursor is located. When the cursor is barely out of one display, the displayed arc section of the halo is highly curved, showing most of the circle. If the cursor is very far away, the arc seen will resemble a straight line.

Perspective Cursor is different from Stitching in the technology that it requires. Stitching can work without any extra device, but Perspective Cursor is based on the

assumption that the system knows the spatial relationships between the user’s head and the visible displays, and thus requires some kind of tracking device. In the discussion section we analyze some of the implications of the tracking requirements in the costs and applicability of the technique.

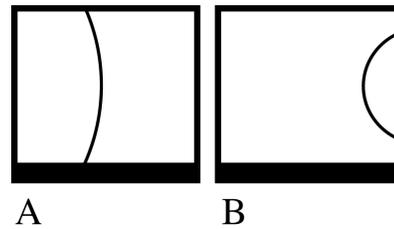


Figure 5. Two examples of halos. A) the cursor is far to the left of the screen. B) the cursor is close to the right of the screen.

EMPIRICAL STUDY

In order to validate the value of perspective for multi-display interactions and the characteristics of the new technique, we designed an experiment in which we compared two perspective-based techniques, Perspective Cursor and Virtual Beam, with a non-perspective interaction technique: Stitching of control spaces.

Experimental setting

For the experiment, we developed a prototype single-user multi-display environment through which we can test most kinds of inter-display transitions of the cursor. The setting consists of three fixed displays and one mobile display. The three fixed displays are a large vertical wall-projected screen, a projected tabletop display and a regular flat monitor. The mobile display is a tablet PC.

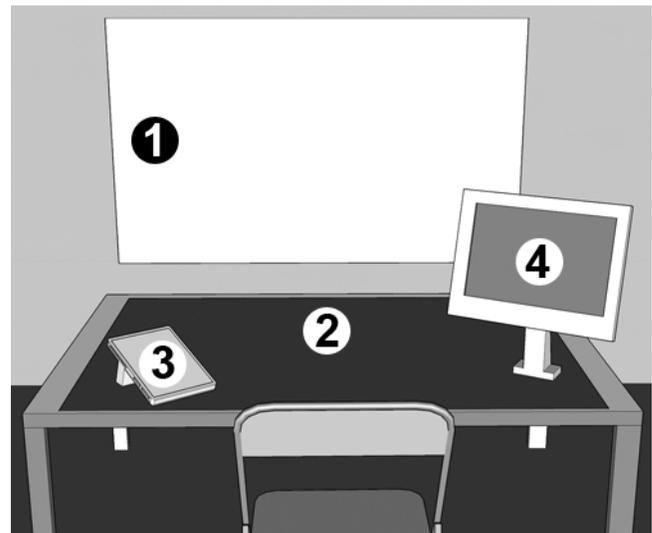


Figure 6. Experimental setting. 1) wall display 2) table-top display 3) Tablet PC 4) flat screen.

Figure 6 shows the physical locations of all the elements. The tabletop display (2) is a projected table with an image of 1024x768 pixels and 124.5x158cm in size. The wall display (1) has the same resolution but the projection is

slightly smaller (136x101.5cm). The flat screen (4) is a 15” LCD monitor with a resolution of 1024x768. The tablet PC’s display (3) is 15.5x21cm with an image of 768x1024 pixels.

We use a total of three computers to control all the displays. The main application resides in a Pentium IV PC that also controls the two big displays. The flat panel and the tablet PC are controlled by independent machines connected to the main application by a dedicated Ethernet network.

For relative-positioning control we use a wireless mouse. Position tracking is provided by a Polhemus® Liberty tracker with three tethered 6-DOF sensors. One sensor is attached to a baseball cap that measures the user’s head position, another is attached to the tablet PC, and one, in the shape of a pen with a button, serves as the virtual laser pointer.

The system kept an updated 3D model of the whole setting, including the displays, the position of the user’s head, the position and orientation of the pen (laser pointer) and the position and orientation of the mobile display. We must note that although the tracking technology that we used is affected by metallic and magnetic objects, the setting was designed so that accuracy of tracking was not an issue, except for the case of the tablet PC when using Virtual Beam, which we discuss later.

Three techniques were implemented in this prototype for the experiment: Virtual Beam, Perspective Cursor and Control-Stitched Displays (see video figure).

Virtual Beam

Our implementation of a laser pointer uses a 6-DOF sensor in the shape of a pen with a button close to the tip. To obtain the position of the cursor we virtually intersect a mathematical line coming from the tip of the pen in the longitudinal direction with the virtual model of the room. If there is a display in the way, the two-dimensional coordinate point of the intersection relative to that display is considered the current position of the pointer. If the line intersects more than one display, the display closest to the pen is chosen as the one displaying the cursor. When the line does not intersect any display, nothing is shown.

In short, the pen works as a laser pointer but for the fact that it controls the system’s pointer instead of a red dot, and that it does not display anything when pointed to a space without displays in the way. The button in the pen generates the same kind of events as does a mouse button.

Due to technology constraints, the pen could not be too close to the tablet PC without appreciable distortion (distortion appeared at distances of around 4cm). All subjects of the study were instructed not to bring the pen too close to the tablet PC to avoid this effect.

Perspective Cursor

Perspective Cursor uses the users’ head position (but not orientation) as the origin of the intersecting line discussed above. The orientation of the line is determined by the movements of the mouse so that a vertical movement of the mouse results in an increase or decrease of the angle of the line with respect to the equator. Conversely, a horizontal movement of the mouse changes the longitudinal orientation of the line. Wherever the virtual line intersects the surface of a display, there lies the Perspective Cursor.

The cursor keeps constant size relative to the user, i.e., the image of the cursor varies in size and shape depending on the position and orientation of the surface where it is being displayed, but it projects the same image on the user’s retina. The size of the cursor was calculated to be about three times the size of a normal cursor seen in a 1024 by 768 screen at a normal viewing distance (40cm), covering an angle of around 2 degrees (angular size of the cursor was multiplied in the video figure for illustration purposes).

If the position of the head changes but the mouse is not moved, the cursor stays in the same place of the same display. To prevent users from losing Perspective Cursor in inter-display (blank) space, we used a variant of the Halo technique [3] adapted for 3D environments.

Stitched Control Spaces

In this technique, the movement of the mouse is related to the changes in the coordinates of the cursor in a linear fashion. When the cursor reaches the border of a particular display the system checks if there is another display assigned to that end. If so, the cursor continues moving across the new display. If there is no other display stitched to that end, the cursor just stays in the border (hard limits).

Figure 7 depicts the actual stitching implemented for our setting, which was designed to be as close as possible to a “flattening” of the room’s 3D model into a 2D map.

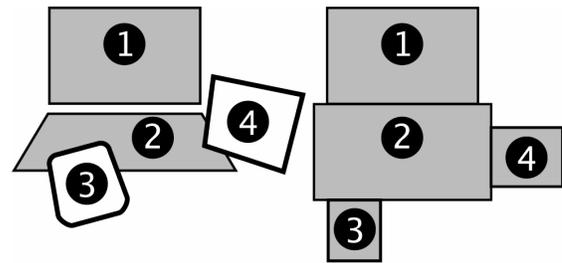


Figure 7. Multi-display environment and its 2D stitching of control spaces.

The C/D ratio in centimeters is kept constant, meaning that the small displays take less mouse movement to cross, although the number of pixels is the same.

Tasks

The subjects were asked to click on an origin icon (70x70 pixels) and then again in a destination icon (same size) as fast as they could, but without sacrificing accuracy. Both

icons were visible and seen by the user before each task started. The pairs of origin/destination icons were selected according to the results of a pilot study that provided us with four groups of tasks that represent different kinds of multi-display interactions: *simple across-displays*, *complex across-displays*, *within-display* and, *high-distortion within-display*.

Simple across-displays tasks. In these tasks the spatial relationship between the origin display and the destination display are very simple (Figure 8A). In our setting this is the case only for the tabletop display and the wall display (1 and 2 in Figure 6). The transition between these two displays is easy because both are more or less the same size and the connecting borders are parallel.

Complex across-displays tasks. In this group of tasks the origin and destination displays are not aligned in any way, and they might be of very different sizes (Figure 8B).

Within-display tasks. These are tasks with the two icons in the same display, i.e. mono-display tasks (Figure 8C).

High-distortion within-display tasks. In a display that is located very close to the user and in a parallel angle with the line of sight (e.g., our table top) there are regions of the display (the corners closest to the observer) that suffer from a strong perspective effect or *scherzo* that affects control perception. As the pilot study suggested that tasks operating in these regions would yield specific effects, we created another group testing this kind of tasks (Figure 8D).

The system considered a trial a miss if the second click did not fall inside the area of the destination icon, giving distinctive auditory feedback than a hit. Only the time between a click on the origin and a second click was measured.

Experiment design

The experiment was conducted with 12 right-handed participants (2 females and 10 males) between the ages of 19 and 35. All participants had experience with graphical user interfaces. Each subject was tested individually. Each experiment took approximately 70 minutes to complete.

The experiment used a 3x4 within-participants factorial design with planned comparisons. The factors were:

- Interaction Technique (Perspective Cursor, Virtual Beam, Stitched Control Spaces)
- Task type (Simple across-displays, complex across-displays, within-display, and high-distortion within-display).

The experiment comprised 3 blocks of trials, one for each interaction technique (Perspective Cursor, Virtual Beam and Stitched control spaces). Each block was split into two sets of trials; a training set and a test set. The training set had 32 trials, two per task, while the test set consisted of 8 trials for each of the 16 tasks in a random order, for a total of 128 test trials per technique (32 per task group). Each

subject provided a total of 384 valid time measurements. The order of the conditions was balanced across subjects (2 in each possible order of interaction technique).

The number of trials was determined through a conservative a-priori power analysis (power = 0.8, estimated standard deviation = 0.62) that made our experimental design capable of detecting differences in means larger than 15%.

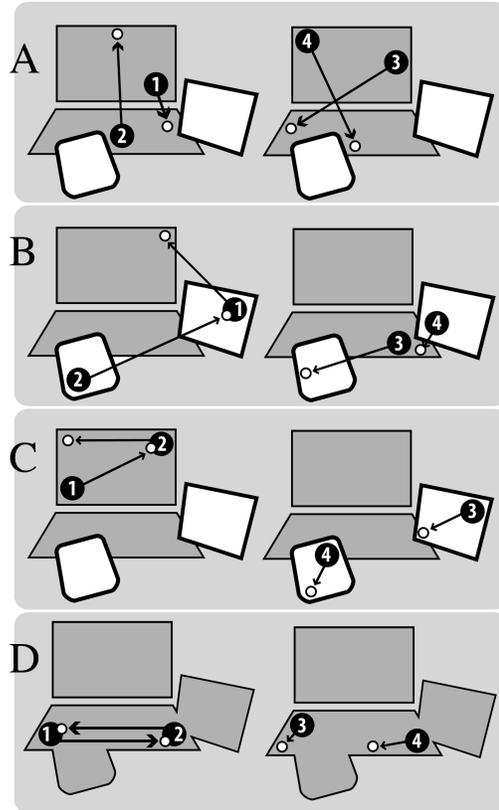


Figure 8. Task types A) simple across-displays B) complex across-displays C) within-display D) high-distortion within-display.

For each trial, completion time and hit/miss information was recorded. At the end of all trials the subjects were asked to complete a questionnaire evaluating the three techniques in performance, accuracy and preference. They were also asked to fill out a workload assessment form for each technique.

RESULTS

Three sources of data were gathered: completion time, accuracy and user preference, and workload assessment.

Completion time

A two-way repeated-measures ANOVA test over all the successful trials with *interaction technique* and *task group* as factors showed that both factors have a main effect in completion time of the task ($F_{2,72}=36.25$, $p < 0.0001$; $F_{3,72}=453$ $p < 0.0001$), and that there is also an interaction between the two factors ($F_{11,72} = 20.48$, $p < 0.0001$).

Figure 9 shows the average completion times and standard errors for the three techniques grouped by task type. As the general ANOVA test indicated that there was an interaction between *interaction technique* and *task group* we proceeded to analyze the effects of *interaction technique* for each of the conditions of *task group*.

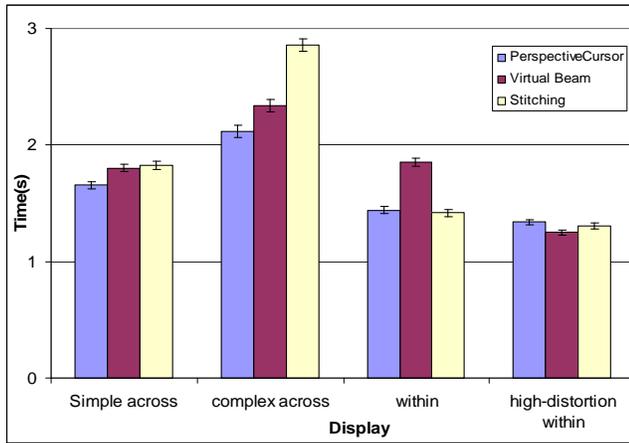


Figure 9. Time of completion and standard error (stem) in the different conditions.

Simple across-displays tasks

For simple transition tasks the ANOVA test revealed that there were differences in performance amongst the techniques ($F_{2,24} = 8.31, p < 0.0003$). The Tukey-HSD multiple-comparisons post-hoc test showed that for these group of tasks, Perspective Cursor ($t_{avg} = 1.6563s$) is significantly faster than Virtual Beam ($t_{avg} = 1.803s$) and also faster than Stitching of control spaces, ($t_{avg} = 1.8263$) but these two are not significantly different from each other.

Complex across-display tasks

For complex transition tasks, the ANOVA test revealed that there were also differences in performance amongst the techniques ($F_{2,24} = 50.99, p < 0.0001$). The Tukey-HSD multiple-comparisons post-hoc test showed that all task groups were significantly different from each other. Perspective Cursor was the fastest ($t_{avg} = 2.12s$), followed by Virtual Beam ($t_{avg} = 2.34s$), and Stitching ($t_{avg} = 2.86s$).

Within-display tasks

For the tasks involving only one display, ANOVA revealed that there were also differences in performance amongst the techniques ($F_{2,24} = 62.67, p < 0.0001$). The Tukey-HSD test showed that the two fastest techniques, Stitching of control spaces ($t_{avg} = 1.418s$) and Perspective Cursor ($t_{avg} = 1.44s$) were not significantly different from each other, but both were significantly faster than Virtual Beam ($t_{avg} = 1.85s$).

High-distortion within-display tasks

In the tasks that took place in areas of high perspective distortion, the ANOVA test revealed differences in performance ($F_{2,24} = 3.69, p < 0.0253$). Virtual Beam was the fastest ($t_{avg} = 1.2499$), not significantly faster than

Stitching ($t_{avg} = 1.3072$) but significantly faster than the Perspective Cursor ($t_{avg} = 1.3383$). Stitching and Perspective Cursor time averages were not significantly different.

Accuracy

In terms of overall accuracy Perspective Cursor was the best interaction technique with 45 misses, followed by Stitching with 57, and Virtual Beam with 154. Figure 10 shows the distribution of misses over different task types. In the high-distortion-within-display tasks there were 5 misses for each interaction technique. In all other task types Virtual Beam had the most number of misses with more than 35 misses per task.

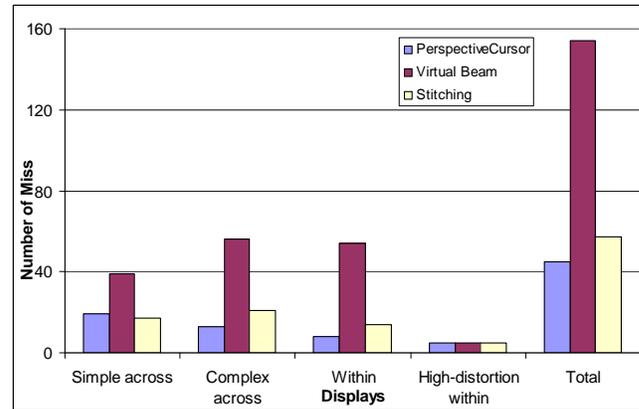


Figure 10. Number of misses per condition (of a total of 4608).

User preference and workload assessment

After finishing the tasks, users were asked to rank the techniques in order of subjective speed, accuracy and preference. Most users perceived Perspective Cursor as the fastest technique (9 first places and 3 second places) over Virtual Beam (3 first, 6 second, 3 third) and Stitching (3 second, 9 third). Perspective Cursor was also considered the most accurate technique (10 first, 3 second) followed by Stitching of control spaces (2 first, 5 second, 5 third), and Virtual Beam (5 second, 7 third).

When asked to rank the techniques by preference, Perspective Cursor was preferred by all but one user (who ranked it second). Virtual Beam received one first place vote, 8 second and 3 third, followed by Stitching, which received three second places and 9 thirds.

The users were also asked to fill out a workload assessment questionnaire. Analysis of the questionnaire showed that there were significant differences in user's perception of frustration, mental load and physical effort for the different techniques. Across these categories users considered Perspective Cursor less frustrating, easier to handle mentally and less physically tiring. Of particular interest is the assessment of physical effort in which Virtual Beam received an average of 5.91 out of 7, much higher than Stitching (3.41) and Perspective Cursor (2.41).

DISCUSSION

Below we will look at our main results and provide some explanation of why they occurred.

Perspective-based techniques vs. Stitching

The experiment found that the three techniques perform differently for across-displays tasks. Perspective Cursor is the fastest when several displays are involved (up to 26% faster than Stitching control spaces and 8% faster than Virtual Beam). Virtual Beam is better than Stitching when the relative position of the displays involved does not allow a straightforward stitching. In this kind of interaction Stitching of control spaces is confusing for the users.

Perspective-based techniques are faster because they provide an intuitive layout of control space. We observed that users had difficulties remembering how to access one display from another when using the Stitching of control spaces technique. Several subjects reported that they needed to plan the movements of the mouse ahead according to the stitching scheme, what we call a *maze effect*.

As we expected, a simple layout of monitors is easier for the Stitching technique, but Perspective Cursor still beats Stitching for these transitions (Perspective Cursor is 8% faster on average). One might think that the blank space that the Perspective Cursor has to cross between displays increases the interaction completion time, but consistent with what Baudisch et al. report [2], it is the lack of that space in Stitched Control that makes the transition less natural and slower, as the users end up overshooting much more often than with other techniques.

It should be noted also that the 3D geometry characteristics of perspective-based techniques allowed a seamless interaction across displays of very different resolutions without an explicit change in C/D ratio.

Perspective Cursor

In all but the high-distortion tasks Perspective Cursor was the best technique, or at least not significantly worse than the best. Most importantly, Perspective Cursor was as fast as Stitching in the simple within-display tasks, which means that the multi-display capabilities of the technique are not traded off for a poorer performance in the standard single-display interactions that we are used to.

The overall results for Perspective Cursor show that there is value in using a relative control device like the mouse in combination with perspective. Users seem also to appreciate it, as all but one ranked it best. We think that Perspective Cursor, although relatively complicated to implement compared to a non-perspective technique, is a better option for control of multi-display environments than the existing alternatives.

One possible problem of Perspective Cursor is the possibility of losing the cursor in non-displayable space. We were worried that users could have trouble with this feature but the halos seemed to work well, and when the

users lost the cursor, they were capable of bringing it back to a display very quickly by looking at the halos.

It must also be mentioned that the relative-positioning nature of Perspective Cursor allows for further target-acquisition optimization as in [4, 14, 35].

Virtual Beam

For tasks that involved complex display transitions Virtual Beam proved of value (almost 20% faster than Stitching). The experiment also provided weak evidence that in situations of high perspective distortion (closest corners of a non-perpendicular display) this technique is preferable to mouse-based techniques.

However, the accuracy of Virtual Beam was far below the other two (89% of success compared to 96% of Perspective Cursor and Stitching of control spaces). This problem is due to the inherent inaccuracy of the device and has been reported many times [21, 24]. Although there are ways to improve this accuracy by filtering or changing the interaction techniques [9, 25, 20], our main focus for this experiment was on performance and so we decided to include the technique without modifications that may introduce feedback lags or arbitrary delays.

Another issue of the beaming techniques is how we provide a button click. If the button is in the device itself, accuracy is further decreased by the clicking movement at the moment when most stability is required: at target acquisition. This problem can be solved if we perform the clicking gesture using the non-dominant hand, but this raises other kinds of problems for real-life environments because we usually need the non-dominant hand for other purposes (e.g., holding another device, gesturing, etc.).

Another drawback of Virtual Beaming made evident by the data collected is that it is a very tiring technique. Several users reported this, and the technique was rated the most physically demanding. Although we agree that in a real-life situation the use of a laser pointer or a pen would not be as intensive as in our experiment, the effect should be considered for applications that require intensive pointing for long periods of time.

It must also be mentioned that the technological limitation that reduced accuracy when the pen was too close to the tablet might have had an effect on the trials that involved the tablet.

What you see is what you can control

One important aspect of perspective-based techniques is that they provide control only over the display surfaces that are visible, and in the degree in which they are visible. This means that perspective techniques are not adequate for environments in which the multi-display interaction is intended for full resolution control of non-visible displays or machines from a single interface. For these situations it would be better to use remote-control techniques like Mighty Mouse [8].

Implications for collaborative environments

Perspective Cursor poses two problems for mutual awareness in co-located cooperative environments: predictability of the cursor movement and gesture visibility. First, some CSCW systems might benefit if the users can naturally acquire awareness of other users' actions. Perspective Cursor makes this more difficult because the movement of the cursor is more difficult to understand from a point of view different from that of the user in control.

Second, if awareness of the users' actions is important, beaming techniques in which the actions are highly visible and easy to interpret have an advantage over mouse-based techniques, in which the gestures are much less obvious.

Implications for privacy

In perspective-based techniques displays are only accessible if they are visible, enforcing a natural privacy protection derived from the real world. For example, if the owner of a laptop does not want somebody to see or act in the contents of her display, she will turn the laptop so that the display is parallel to the line-of-sight of the potential intruder. However, it is possible that further privacy protection rules would have to be implemented for certain kinds of environments (e.g., public spaces or very sensitive content).

Applicability and costs

As mentioned above, Perspective Cursor requires tracking of the user's head relative to the position and orientation of any display in the room. There exist several alternatives to implement this with current technology: 3D magnetic trackers, computer vision tracking, or active sensors. However, these technologies are still expensive and they are not free of problems (e.g., tethered sensors, interference from metallic objects) which may preclude their use in current systems. Nevertheless we believe that cost-effective solutions for this particular problem are attainable in the short term. In [23] we analyze possible ways to provide affordable solutions using computer vision.

We also believe that there are current applications that might already justify the cost of current solutions, e.g., television studios, command and control rooms, etc.

Generalizability of the results

In this study we took a close look at different techniques for multi-display pointing. We included displays of several sizes in several positions, which made the controlled environment reasonably similar to real conditions, however, there is still much to learn about these techniques in more general situations, in particular: how do the techniques perform in tasks other than pointing (e.g. text selection, drawing)? Are these techniques equally useful in multi-user environments? Can the techniques be adapted to other kinds of input devices? These questions have to be answered through future experiments.

LESSONS FOR DESIGNERS

There are six main lessons from this study:

- Perspective-based techniques should be considered when designing multi-display systems, especially if there are mobile displays involved.
- Perspective Cursor is effective for systems that require time-efficient interactions, and is strongly preferred by users.
- However, Perspective Cursor adds implementation complexity, and may not promote awareness in co-located collaborative environments.
- The Perspective Cursor technique requires an indication of the position of the cursor for when it is out of display space (i.e., when it is in blind-space).
- Beam-based techniques are intuitive and a good choice for multi-display interactions, but they must be implemented with mechanisms that improve accuracy.
- Stitched Control Spaces is a reasonable alternative for multi-display interaction if the setting is static and there is a simple 2D mapping of the location of the displays.

CONCLUSIONS

Multi-display environments and smart meeting rooms bring together several independent systems into a single display space. Traditional means of stitching these devices together often do not adequately represent the position and orientation of the devices, particularly when people look at the displays from different locations. To address this problem, we used the idea of perspective to design new interaction techniques for multi-display environments. The Perspective Cursor maps ordinary mouse input to the display space based on the user's current perspective: the cursor tracks correctly across displays of different resolutions, and appears where it should when displays overlap. We compared both Perspective Cursor and a Virtual Beam technique to a traditional multi-display setup, and found that both perspective-based techniques provided significant performance gains. In addition, Perspective Cursor showed advantages over the Beam technique, and was the most preferred technique.

In the future, we plan to look at other applications of perspective in multi-display environments, develop other perspective-based techniques, and test Perspective Cursor in more realistic tasks. We also plan to investigate the techniques in collaborative settings with multiple co-located users and multiple cursors.

ACKNOWLEDGEMENTS

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would like to thank Susana López, Marta Besteiro and Kate Skipsey for their help, and the reviewers for their insightful comments.

REFERENCES

1. Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tandler, P. Bederson, B., and Zierlinger, A. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing

- Remote Screen Content on Touch- and Pen-operated Systems. *Proc. Interact 2003*, 57-64.
2. Baudisch, P., Cutrell, E., Hinckley, K. and Gruen, R., Mouse Ether: Accelerating the Acquisition of Targets Across Multi-Monitor Displays. *Proc ACM CHI 2004*, 1379-1382.
 3. Baudisch, P., and Rosenholtz, R. Halo: A Technique for Visualizing Off-Screen Locations. *Proc. CHI 2003*, 481-488.
 4. Balakrishnan, R. "Beating" Fitts' Law: virtual enhancements for pointing facilitation. *IJHCS*, 61, 6, (2004), 857-874.
 5. Benko, H. and Feiner, S. Multi-monitor mouse. *Proc. CHI 2005*, 1208-1211.
 6. Biehl, J.T. and Bailey, B.P. ARIS: An Interface for Application Relocation in an Interactive Space, *Proc. Graphics Interface*, (2004), 107-116.
 7. Bolt, R.A. "Put-that-there": Voice and gesture at the graphics interface. *Proc. SIGGRAPH '80*, 262-270.
 8. Booth, K., Fisher, B., Lin, R., & Argue, R. The "mighty mouse" multi-screen collaboration tool. *Proc. UIST 2002*, 209-212.
 9. Davis, J., and Chen, X. LumiPoint: Multi-user laser-based interaction on large tiled displays. *Displays 2002*, 23(5):205-211.
 10. Deitz, P. and Leigh D. DiamondTouch: A Multi-User Touch Technology. *Proc. UIST 2001*, 219-226.
 11. Fox, A., Johanson, B., Hanrahan, P., and Winograd, T. Integrating Information Appliances into an Interactive Workspace. *IEEE CG&A* 20, 3, (2000), 54-65.
 12. Furnas, G. W., Generalized Eye Views. *Proc. CHI 1986* 16-23.
 13. Geißler, J. Shuffle, throw or take it! Working Efficiently with an Interactive Wall. *Proc. CHI 1998*, 265-266.
 14. Grossman, T., and Balakrishnan, R. The Bubble Cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. *Proc. CHI 2005*, 281-290.
 15. Hascoët, M. Throwing models for large displays. *Proc. HCI 2003*, 73-77.
 16. Hinckley, K. Synchronous Gestures for Multiple Users and Computers. *Proc. UIST 2003*, 149-158.
 17. Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., and Smith, M. Stitching: pen gestures that span multiple displays. *Proc. ACM AVI 2004*, 23-31.
 18. Johanson, B., Hutchins, G., Winograd, T., Stone, M. PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. *Proc. UIST 2002*, 227-234
 19. Kohtake, N., Ohsawa, R., Yonezawa, T., Matsukura, Y. Iway, M. Takashio, K., Tokuda, H. u-Texture: Self-Organizable Universal Panels for Creating Smart Surroundings. *Proc. UbiComp 2005* , 19-36.
 20. Myers, B.A., Peck, C.H., Nichols, J., Kong, D., and Miller, R. Interacting at a Distance Using Semantic Snarfing. *Proc. UbiComp 2001*, 305-314.
 21. Myers, B.A., R. Bhatnagar, J. Nichols, C. H. Peck, D. Kong, R. Miller, and A. C. Long. Interacting At a Distance: Measuring the Performance of Laser Pointers and Other Devices. *Proc. CHI 2002*, 33-40.
 22. Nacenta, M., Aliakseyeu, D., Subramanian, S., and Gutwin, C. A Comparison of Techniques for Multi-Display Reaching. *Proc. CHI 2005*, 371-380.
 23. Nacenta, M. Computer Vision approaches to solve the screen pose acquisition problem for Perspective Cursor. *Tech. Rep. HCI-TR-06-01, Comp. Sci. Dept., U. of Saskatchewan*, (2006).
 24. Olsen Jr., D.R. and Nielsen, T. Laser Pointer Interaction. *Proc. CHI 2001*, 17-22.
 25. Parker, J. K., Mandryk, R. L, Inkpen, K. M. TractorBeam: Seamless Integration of Local and Remote Pointing for Tabletop Displays. *Proc. Graphics Interface* (2005), 33-40.
 26. Rekimoto, J. Pick-and-Drop A Direct Manipulation Technique for Multiple Computer Environments. *Proc. UIST 1997*, 31-39.
 27. Rekimoto, J. and Saitoh, M. Augmented surfaces: a spatially continuous work space for hybrid computing environments. *Proc. CHI 1999*, 378-385.
 28. Rekimoto, J., Ayatsuka, Y., and Kohno, M. SyncTap: An interaction technique for mobile networking. *Proc. Mobile HCI* (2003), 104-115.
 29. Robertson, G., Czerwinski, M., Baudisch, P., Meyers, B., Robbins, D., Smith, G., and Tan, D. The Large-Display User Experience. *IEEE CG&A*, 25, 4, (2005), 44-51.
 30. Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R. H., and Nahrstedt, K. A Middleware Infrastructure for Active Spaces. *IEEE Pervasive Computing*, 1, 4 (2002), 74-83.
 31. Streitz, N.A., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., and Steinmetz, R. i-LAND: An interactive Landscape for Creativity and Innovation. *Proc. CHI 1999*, 120-127.
 32. Swaminathan, K. and Sato, S. Interaction design for large displays. *Interactions*, 4, 1, January 1997, 15-24.
 33. Wu, M. and Balakrishnan, R. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *Proc. UIST '2003*, 193-202.
 34. Yee, K.P. Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers. *Proc. CHI 2003*, 1-8.
 35. Zhai, S., Morimoto, C., Ihde, S. Manual and Gaze Input Cascaded (MAGIC) Pointing. *Proc. CHI 1999*, 246-253.