Experiences with Mouse Control in Multi-Display Environments

Manuela Waldner, Dieter Schmalstieg
Institute for Computer Graphics and Vision
Graz University of Technology, Austria
{waldner | schmalstieg}@icg.tugraz.at

ABSTRACT

It is now increasingly common to extend private workstations with large public displays into a shared multi-display environment. Mouse-based interaction across multiple displays provides a convenient way to quickly shift between private work on the personal monitor and tightly coupled collaboration on shared display spaces. However, mouse pointer navigation can be negatively influenced by display factors in the environment and thereby limits fluid interaction across displays. In this paper, we present experiences with mouse-controlled multi-display environments. Based on an experiment comparing four mouse pointer navigation techniques, we show limitations of mouse-controlled interaction in multi-display environments and suggest improvements to enhance the user interface experience with low-cost multi-display settings.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Input devices and strategies (e.g., mouse, touchscreen)*

General Terms

Design, Experimentation, Human Factors

Keywords

multi-display environment, mouse control

Figure 1: Public projected displays as extension of private workspaces in an office environment.

1. INTRODUCTION

The availability of affordable display technology has given rise to a wide variety of multi-display environments (MDE). Especially in conventional office and team spaces, users can extend private workspaces with shared interaction spaces by projecting imagery onto unused wall and table spaces (*c.f.*, Figure 1). The resulting combination of private and shared displays has been found to cause a better coordination of distributed work and a more equitable task work load [5].

Numerous observations have shown that users frequently switch between a tightly and loosely coupled collaboration style (*e.g.*, [11]). In an MDE, this change of collaboration style requires the user to switch between private and shared display spaces to share information, discuss shared information artifacts, or to return to private work. In a conventional office setting, performing these switches by moving the mouse pointer across display boundaries is convenient, as the user does not need to change interaction technique when using the MDE as extension of the personal workspace, and distant displays can be accessed without physically moving [4]. Furthermore, interference of multi-user input is kept to a minimum, as users do not occlude displayed content when interacting on public display space – in contrast to direct input methods. However, previous research has shown that mouse pointer navigation in MDEs is negatively influenced by a number of display factors, such as depth offsets between displays [10], adjacent displays at relative angles higher than 45° [9], physical distance and size-resolution mismatches [2], as well as non-optimal seating arrangements [13].

In this paper we report findings from an experiment comparing four mouse pointer navigation techniques in a heterogeneous multi-display setup [12]. We will discuss limitations of mouse-controlled MDEs and suggest improvements for mouse pointer navigation techniques.

2. COMPARISON OF TECHNIQUES

In conventional multi-monitor systems, adjacent display edges are usually "stitched" to create a seamless interaction area. This approach has also been applied to MDEs with more complex display arrangements, such as in Augmented Surfaces [8] or PointRight [6]. MouseEther [2] additionally incorporates visual discontinuities introcued by monitor bezels and display-size resolution mismatches into the motor space. Perspective Cursor [7] extended this approach by evaluating mouse input events from a tracked user's perspective of the environment. Thereby, it additionally introduces a non-uniform control/display (C/D) gain when navigating within a single display, caused by perspective foreshortening. Instead of implicitly triggering a transition by crossing connected display edges, pointer warping techniques (*e.g.* [3]) and interactive miniature views (*e.g.* [4]) allow the user to redirect input explicitly to a target display.

From this short overview, it is evident that mouse pointer navigation in MDEs covers a large design space. We grouped the differences between the presented navigation techniques into the following four categories:
Trigger: how input redirection is triggered (implicitly by moving the mouse pointer across a display edge or explicitly by pressing a trigger).
Cross-display movement: how display-less space is bridged (warping the mouse pointer across the gap or continuous movement by considering the physical display-less space).
Outcome: where the mouse pointer re-appears on the target display (*e.g.*, at the display edge or at the center).
C/D gain: how the C/D gain is adjusted when moving within a display (*e.g.*, standard C/D gain or perspective).

We assume that the optimal choice of parameters for navigation techniques is dependent on the individual display geometries (such as differences in size and resolution), the arrangement (such as distance and angle between displays), as well as the individual user preferences.

3. EXPERIMENT

We conducted a single-user experiment with 20 participants. We compared four navigation techniques differing in the above mentioned parameters in a heterogeneous multi-display setup [12]:

Path navigation (*path*) is similar to stitching, as it virtually connects the closest display edges of adjacent display pairs. Paths do not necessarily cover the entire display edges, but are limited to intervals determined from the corner point projections of the adjacent edge. Overlapping paths are prioritized according to their display-to-display proximity. Resulting paths represent point-to-point mapping areas between paired displays. To aid the path-finding process for the user, we visualized connected edge portions with color-coded lines.

Similar to Perspective Cursor [7], free navigation (*free*) takes into account the individual users' perspectives as pointer movement is adapted to their estimated focal planes. However, instead of letting the mouse pointer continuously move within display-less space, free navigation warps the mouse across display gaps. The outcome position on the target display is determined by extrapolating the last mouse motion and intersecting the resulting ray with the display boundaries on the user's focal plane representation of the environment. If no target display was found along the ray, input redirection is not triggered. As we did not employ head-tracking, the perspective representation of the environment does not change when the user moves.

The world-in-miniature (*WIM*) control is a GUI presenting a 3D model of the environment textured with live desktop contents. The view is initialized from the user's perspective and can be interactively modified. The WIM window is invoked by a shortcut and appears at the current mouse pointer location. To redirect mouse input, the user clicks the desired target location within the miniature view. As a result, the mouse pointer will be redirected to the associated display position in the physical environment.

Pointer warping (*warp*) relocates the mouse pointer to the center of a display with a given ID. Relocation is triggered by pressing a modifier key and the desired target display ID number on the keyboard. Table 1 illustrates the differences of the four employed navigation techniques.

	Path	**Free**	**WIM**	**Warp**
Trigger	implicit (edge interval)	implicit (edge)	explicit (GUI)	explicit (key-press)
Outcome	point-to-point edge-mapping	ray-edge intersection	selected in GUI	display center
C/D gain	standard	perspective	standard	standard

Table 1: The experimental conditions differ in how input redirection is triggered, the outcome position on the target display, and the C/D gain adjustment. All techniques warp the mouse pointer across display-less space.

The display setup was chosen, so that each cross-display transition covered varying display factors, such as change of display size, different display angles, depth disparities, whether displays are placed distant from each other (and thus are not accessible without crossing intermediate displays), and whether one of the displays lies outside the user's field of view. Figure 2 shows the setup.

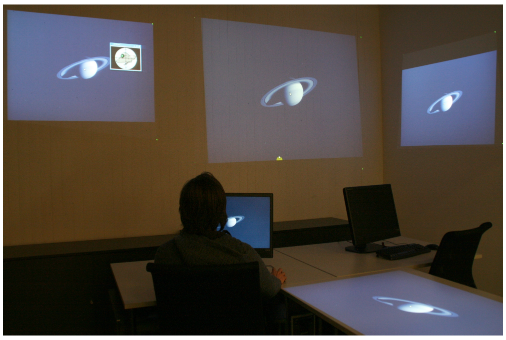

Figure 2: Experimental setup.

For each navigation technique, users had to accomplish a target selection task with targets appearing sequentially at different display locations. After performing the task for each of the four navigation techniques, users were asked to accomplish a combined navigation condition where they could choose between path navigation and free navigation as mouse-based navigation technique and were free to invoke the WIM control or pointer warping at any time. We collected task completion times between two target selections as performance measures, employed input logging, asked users to fill out a questionnaire rating the four techniques, and conducted a semi-structured interview at the end of the experiment.

3.1 Performance

We measured task completion times between two consecutive target selections and conducted a 4(navigation technique) x 9(transition defined by displays of two subsequent targets) repeated measures ANOVA. WIM was the slowest technique (average completion time across all transitions $t_{wim} = 5.84s$), followed by free navigation ($t_{free} = 4.17s$), which was slower than both, path ($t_{path} = 3.67s$) and warp ($t_{warp} = 3.29s$). Pointer warping had the best performance for navigation between the monitor and the projected wall displays, as well as for navigation from and to the tabletop display, compared to the other navigation techniques. Both mouse-based navigation techniques (path and free) were faster for navigation between the wall displays than WIM and warp. However, performance of path and free differed when more complex transitions than traversing between wall displays had to be accomplished: Free was superior compared to path for navigation between monitor and wall displays, as well as when navigating from the tabletop display to a wall display. However, it suffered from severe performance fallbacks when navigating *to* the tabletop display and subsequently selecting a target there. WIM and warp had almost uniform task completion times across all display crossings, thus did not seem to be strongly affected by changing display factors, as compared to path and free. Additional information on task completion times outcomes can be found in [12].

3.2 Usage frequencies

When giving our users the choice, eleven out of twenty users decided for path navigation, nine for free navigation as mouse-based navigation techniques. Overall, 64% of all display crossings were performed with the mouse, 27% with pointer warping, and 4% with the WIM control. Usage of mouse-based transitions was high for navigation between wall displays (85% on average for adjacent displays and 75% for jumping between the outermost wall displays). The mouse was also the main choice for navigation between monitor and wall displays (64%). However, pointer warping was employed more often than mouse-based transitions whenever the tabletop display was involved (47% and 29%, respectively).

3.3 Preference and user feedback

After the experiment, users were asked to evaluate the four navigation techniques on a seven-point Likert scale. A one-way repeated measures ANOVA did not reveal any significant differences across the ratings. On average, pointer warping was rated highest ($r_{warp} = 5.55$) and free navigation lowest ($r_{free} = 4.55$). In the interview, most users mentioned they preferred the combined condition.

When interviewing the users, we found that preferences were indeed diverse. Some users appreciated the clear, visualized structure given by path navigation. Others rated it as "too restrictive". Similarly, seven users stated they found free navigation "very intuitive" while two users "did not understand the concept at all". Several users also reported that the mouse pointer speed was too slow with free navigation. Free navigation was the only technique for which the C/D gain was altered from the standard device settings, as it was adjusted for the perspective field of view of the user. One user mentioned that "the display outline is my reference frame", so he would expect the mouse to move with consistent speed within the individual display boundaries, irrespective of the physical size. The perspective mapping was also the main reason for performance fallbacks of free navigation, as navigation on the tabletop display resulted in a skewed and rotated mouse navigation frame relative to the actual device space. Users mentioned that navigation on the tabletop display was the main reason to rate free navigation low. Another problem with free navigation was that any display border region was potentially connected to a target location. Depending on the current movement direction, the mouse pointer would be warped to a distant location on a remote display when involuntarily touching the display border. We assume that this problem will be even more severe when working with conventional desktop content, where important GUI-elements are often placed at the border of the display.

For the WIM control, users reported that the start-up latency was too high and that "too many mental steps" were involved. Some users also indicated that the window placement at the mouse pointer location was not appropriate. They argued that their focus was already on the target display when the WIM window would appear at the current pointer location, forcing them to look back and identify the target in the miniature view. Thus, some suggested to open the WIM consistently at the private monitor or synchronously on all displays. Pointer warping was generally appreciated by the users as it was perceived as fast option to change displays. Users also reported that pointer warping was convenient when loosing track of the mouse pointer, as it would consistently warp the mouse back to a pre-defined location. Considering usage frequencies, these reports are surprising, as pointer warping was less often employed than mouse-based transitions. When asking participants for their strategy when to employ pointer warping or the WIM control instead of mouse-based transitions, they mentioned long distance travels, "not easily accessible" displays, and tabletop display and monitor in particular.

4. DISCUSSION

Based on the findings and observations of the experiment, we will discuss implications and future research directions.

Implicit or explicit trigger? Our study suggests that triggering input redirection through pointer warping techniques leads to increased performance and is appreciated by participants across all experience levels. However, it also indicates that users rarely employ pointer warping when they have the choice. In fact, they choose pointer warping primarily to overcome subjectively complex transitions where extensive movement planning, physical effort, or adaptations of pointer movement directions (as for the tabletop) would be required when using the mouse. Thus, we recommend to provide pointer warping techniques as additional option when building mouse-controlled MDEs, so users can overcome complex display crossings and quickly relocate their pointer to a known position.

What about tabletop displays? Our initial design approach was to enable consistent navigation and manipulation techniques across all displays in the environment. However, seamless mouse pointer navigation proved to be unsuitable for accessing and manipulating tabletop content in a configuration as employed in our setup. As tabletop displays cannot be controlled from distance in a similar manner as wall displays, users need to gather around the tabletop display to view and manipulate content. It therefore seems reasonable to restrict interaction with tabletop content to direct manipulation techniques, such as touch input. However, to enable spontaneous collaboration, we require flexible user interfaces to relocate content from the remaining workspace to the tabletop display and vice versa. We assume that a world-in-miniature control is a promising tool for moving content across displays with different input capabilities. In a low-cost multi-display setting without touch-capable input devices, we recommend to refrain from seamless mouse-control across tabletop display borders. Instead, it seems more suitable to access tabletop displays by using pointer warping or a WIM control.

How can mouse pointer navigation in MDEs be improved? We will discuss the advantages and limitations of navigation techniques with respect to the navigation parameters introduced in section 2:
Trigger: Path navigation was rated as too restrictive by some users as the display-connecting paths were perceived as too small. On the other hand, the ability to leave the display at any display border position was a major problem for free navigation, as participants often lost their mouse pointer when they involuntarily touched the display border and thereby caused a transition to a remote display. The ideal solution would be to predict whether the user is actually intending to leave the display by analyzing the motion pattern. Thereby, we could preserve the display edges as valuable navigation aid when selecting items located at the boundaries of the displays [1], such as menu bars, while letting the user navigate quickly across display borders.
Outcome position: We observed that users sometimes had difficulties spotting the mouse pointer after performing a transition. We did provide the users with an animated dot to signal the outcome position on the remote display. However, when the target display was not in their field of view, they did not have a visual cue about the current mouse pointer location. It is worth investigating whether more sophisticated visual cues indicating the current mouse pointer location (*e.g.*, a technique like "anchored cursor" [8]) help the users finding their mouse pointer more easily. However, it is also important to find out whether more obtrusive visual cues interfere with collaborative work in a group.
C/D gain: The perspective C/D gain adjustment applied by free navigation caused serious navigation problems, such as targeting difficulties on the tabletop display, as well as low mouse pointer speed. While it seems useful to have a perspective representation of the environment to determine the outcome position on the remote display, having a perspective C/D gain does not seem to bring any advantage. In the future, we will therefore investigate different combinations of navigation parameters, combining positive aspects of path and free navigation.

5. CONCLUSION

Mouse pointer navigation is universally employable and can provide fine-grained input even on distant displays, while allowing to traverse quickly across multiple display boundaries. It seems to be an ideal choice to extend single-user workspaces with shared wall and tabletop displays. However, our experiment has shown some limitations of mouse-controlled MDEs including the ability to traverse to and from tabletop displays, involuntarily redirecting the mouse pointer, and finding complex paths across multiple displays. We discussed the implications and suggested some improvements for mouse-controlled MDEs. In the future, we will address the raised issues and shift our focus to collaborative settings.

6. ACKNOWLEDGMENTS

This work is supported by the Austrian Research Promotion Agency FFG (BRIDGE 822716). Thanks to Christian Pirchheim and Ernst Kruijff for their support.

7. REFERENCES

[1] C. Appert, O. Chapuis, and M. Beaudouin-Lafon. Evaluation of pointing performance on screen edges. In *Proc. AVI 2008*, pages 119–126. ACM, 2008.

[2] P. Baudisch, E. Cutrell, K. Hinckley, and R. Gruen. Mouse ether: accelerating the acquisition of targets across multi-monitor displays. In *Ext. Abstracts CHI 2004*, pages 1379–1382. ACM, 2004.

[3] H. Benko and S. Feiner. Multi-monitor mouse. In *Ext. Abstracts CHI 2005*, pages 1208–1211. ACM, 2005.

[4] J. T. Biehl and B. P. Bailey. ARIS: An Interface for Application Relocation in an Interactive Space. In *Proc. GI 2004*, pages 107–116, 2004.

[5] K. Inkpen, K. Hawkey, M. Kellar, R. M, K. Parker, D. Reilly, S. Scott, and T. Whalen. Exploring display factors that influence co-located collaboration: angle, size, number, and user arrangement. In *In Proc. HCI International*, 2005.

[6] B. Johanson, G. Hutchins, T. Winograd, and M. Stone. PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. In *Proc. UIST 2002*, pages 227–234, 2002.

[7] M. A. Nacenta, S. Sallam, B. Champoux, S. Subramanian, and C. Gutwin. Perspective Cursor: Perspective-Based Interaction for Multi-Display Environments. In *Proc. CHI 2006*, pages 289–298, 2006.

[8] J. Rekimoto and M. Saitoh. Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments. In *Proc. CHI 1999*, pages 378–385, 1999.

[9] R. Su and B. P. Bailey. Put them where? towards guidelines for positioning large displays in interactive workspaces. In *Proc. GI 2005*, pages 337–349, 2005.

[10] D. S. Tan and M. Czerwinski. Effects of visual separation and physical discontinuities when distributing information across multiple displays. In *Proc. Interact 2003*, pages 252–255, 2003.

[11] A. Tang, M. Tory, B. Po, P. Neumann, and S. Carpendale. Collaborative coupling over tabletop displays. In *Proc. CHI 2006*, pages 1181–1190. ACM, 2006.

[12] M. Waldner, C. Pirchheim, E. Kruijff, and D. Schmalstieg. Automatic configuration of spatially consistent mouse pointer navigation in multi-display environments. In *Proc. IUI 2010*, pages 397–400. ACM, 2010.

[13] J. R. Wallace, R. L. Mandryk, and K. M. Inkpen. Comparing content and input redirection in mdes. In *Proc. CSCW 2008*, pages 157–166. ACM, 2008.