Automatic Configuration of Spatially Consistent Mouse Pointer Navigation in Multi-Display Environments

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

ABSTRACT
Multi-display environments combine displays of various form factors into a common interaction space. Cross-display navigation techniques have to provide transitions to move the mouse pointer across display boundaries to reach distant display locations. A spatially consistent description of display relationships thereby supports fluid cross-display navigation. In this paper, we present two spatially consistent navigation techniques for seamless cross-display navigation in multi-user multi-display environments. These navigation techniques are automatically configured from a spatial model of the environment, which is generated in a camera-assisted calibration step. We describe the implementation in a distributed system and present results of a comparative experiment.

Author Keywords
multi-display environment, cross-display mouse navigation.

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

General Terms
Design, Experimentation, Human Factors

INTRODUCTION
Inexpensive large-format displays, such as large monitors and projectors, make it attractive to combine personal workspaces into an interactive teamspace operated by multiple collaborators. Such a multi-display environment (MDE) can be composed of displays of various form factors arranged arbitrarily in the environment.

To allow all users to operate all display spaces in the environment, cross-display navigation techniques to redirect user input to distant displays are required. Previous work has shown that cross-display mouse pointer navigation performance is affected by discontinuities in the spatial display arrangement [10, 7] and the user's location within the environment [6, 8]. *Spatially consistent* mouse pointer navigation techniques aim to accommodate for these discontinuities by creating cross-display *transitions* taking the spatial display arrangement into account. However, cross-display navigation solutions so far have either been restricted to a fixed environment with manual configuration of transitions [6] or by tracking the location of the user in a known display environment [8]. Ad-hoc usage of MDEs requires that users do not need to configure the navigation space manually while still having the flexibility to re-assemble the MDE according to their preferences. It also requires that users are not enforced to apply expensive and obtrusive head-worn tracking systems.

In this paper, we present methods for automatic generation of spatially consistent mouse pointer navigation frames and cross-display transitions for multi-user MDEs. To obtain the navigation frames, we rely on a three-dimensional model of the display environment, obtained in an automatic display registration process, and a user location estimation which does not require the user to wear head tracking devices. We implemented and evaluated two spatially consistent mouse pointer navigation techniques: *free navigation*, which creates 2D navigation frames from the estimated user locations and *path navigation*, which creates point-to-point mappings between virtually connected edge regions.

RELATED WORK
A popular approach to achieve seamless cross-display navigation is to virtually connect ("stitch") adjacent display edges, such as in MightyMouse [4], PointRight [6], Desktop Rover[1], and Synergy[2]. They all require the user to specify edge connections offline manually. In contrast to stitching, MouseEther [1] aims to minimize the visual discontinuity in motor space introduced by monitor bezels and display size-resolution mismatches on multi-monitor setups. Perspective Cursor [8] extends this approach to heterogeneous multi-display environments. As their display setup is non-planar, they require a 3D model of the environment and employ 3 DOF head-tracking to retrieve the user's position in the environment. Instead of implicitly triggering a transition by crossing connected display edges, pointer warping techniques (*e.g.* [2]) and interactive miniature views (*e.g.* [4, 3]) allow the user to redirect input explicitly to a target display.

[1]http://www.neslosoftware.com/desktopRover.html
[2]http://synergy2.sourceforge.net/

IUI'10, February 7–10, 2010, Hong Kong, China.

SPATIAL DISPLAY MODEL

To generate mouse pointer navigation frames automatically, the system needs to know the spatial display topology, as well as the user locations towards the displays for location-aware navigation control. Creating 3D models of display surfaces is a common technique for smart projector systems (*c.f.*, [5]) which usually employ the geometric information solely for compensation of projected imagery. However, this information has also high value for interaction techniques in MDEs – in particular for mouse pointer navigation. Our camera-assisted calibration step registers multiple projected displays, as well as monitors, to a common metric coordinate system and approximates display geometries by a polygonal model. If display arrangements change, as devices are added or removed, users can request a partial re-calibration. A full calibration step typically takes less than a minute.

Since we are interested in computer operation using conventional mouse and keyboard, it is reasonable to assume that users are seated at a known desk with a private "home" display. This assumption works well for conference rooms and open-plan offices. By default, monitors with connected mouse pointers are treated as private displays, accessible only for the host mouse pointer. Projected displays are public. We estimate user head locations to be at a certain distance from their home display's center with a viewing direction perpendicular to the display surface. As each user is uniquely associated with a personal mouse, our system is provided with a simple but sufficient identity management.

CROSS-DISPLAY NAVIGATION TECHNIQUES

Conventional stitching of desktops uses a very simple model of the display space. Displays are assumed to lie in one plane, with no gaps and no rotation between displays, as well as uniform resolution. Since this is clearly not adequate to represent more complex MDEs, Perspective Cursor [8] attempts to replace stitching with mouse navigation that operates in a perspectively correct space around the user. It builds on the idea of a mouse *ether* [1] accounting for the movement in the space between displays. This results in a perceived continuous movement in the real world, and was found to improve performance and pointing accuracy.

However, switching from a stitched planar to perspective navigation also incurs a number of problems: Due to perspective foreshortening, the mouse cursor will be subject to nonlinear control/display gain, in particular for displays viewed at a strongly oblique angle. In addition, if the user has to turn head and body to operate a (horizontal) display, this results in a dynamic change of the perspective navigation map or a non-intuitive mouse movement direction when using a static map. Occluded display regions and displays facing away from the user are inaccessible due to the perspective representation. Finally, perspective cursor control requires the user to navigate display-less space blindly. However, recent research [7] has shown that warping the mouse pointer across large gaps between displays is superior to a mouse ether approach. This finding is particularly important for multi-user setups, which often employ spatially separated displays (*e.g.*, a conference table in the center and surrounding wall displays). These considerations motivated us to investigate alternatives for navigation in complex MDE layouts, which combine the use of three-dimensional structure with the efficiency of warping.

Free Navigation

In a nutshell, *free navigation* works similar to perspective cursor navigation, but warps the mouse cursor across display-less space. First, a perspective map of all displays is computed from the estimated user location. The map is set up in such a way that navigation on the home display is not altered by this procedure. However, navigation in all other displays is subject to perspective effects. At runtime, each incoming mouse motion event is evaluated relative to this perspective mapping and converted to the target display's native pixel coordinates using per-display homographies. Unlike perspective cursor navigation, the user cannot navigate in display-less space. Instead, when leaving the display, the last motion on the source display is extrapolated to a ray. If this ray intersects other displays' edges, the intersection point closest to the current position is converted to the target display's device coordinates and input redirection is triggered (Figure 1).

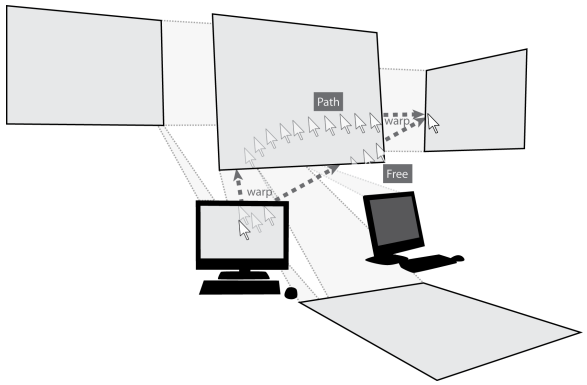

Figure 1. Exemplary mouse pointer paths for free navigation and path navigation. The areas connecting adjacent displays represent edge connections for path navigation

Path Navigation

Path navigation combines aspects of stitching and free navigation. Like stitching, navigation within a display operates in the normal 2D space of the display without any perspective effect. Like free navigation, it uses the spatial display model as reference frame to compute the mouse pointer paths. However, it does not incorporate the user perspective but rather creates a common navigation frame valid for all user locations. Warping across display boundaries is based on a set of edges connecting displays pair-wise in 3D (Figure 1). The connection algorithm starts by considering all possible display pairs and their closest edges. For each candidate pair, connecting edge intervals are computed from normal projections of the closest edges' corner points on the adjacent edge and vice versa. If the resulting interval is empty or smaller than a given threshold on both edges, the candidate is discarded. Otherwise, the normal distance of the edges at the midpoint of the intersection interval is used as a proximity measure. Spatial properties of adjacent displays,

such as connection of non-opposite edges or normal vectors facing away from each other, lead to penalties of the proximity score. Overlapping edge intervals are prioritized according to their proximity scores, so intervals with lower scores are trimmed or removed. Connected edge intervals between neighboring displays are visualized by pair-wise color-coded lines along the connected edge intervals.

The main advantage of path navigation is that it works equally well from any location within the environment and thereby creates a consistent navigation space for multiple users. Table 1 shows a continuum of cross-display navigation techniques from conventional stitching to Perspective Cursor [8].

Navigation technique	movement across displays	reference frame	movement in display
Perspective cursor	continuous	spatial	perspective
Free navigation	warping	spatial	perspective
Path navigation	warping	spatial	standard
Stitching	warping	device	standard

Table 1. Comparison of cross-display navigation techniques.

Implementation

Our mouse pointer navigation framework is implemented in the distributed MDE framework *Deskotheque* [9]. Input redirection is based on an extended version of the open-source mouse pointer sharing tool *Synergy*. In its original implementation, Synergy enables a single mouse and keyboard pair to be shared across multiple machines based on a client-server framework architecture. We added two major extensions to Synergy. As first extension, we implemented a navigation framework on top of the Synergy server which maintains the spatial descriptions of display relationships. As each mouse device is controlled by its own server process, the spatial descriptions of display relationships can be issued separately for each mouse pointer and user, respectively. This assures that each user is provided the appropriate 2D map for free navigation and that multiple users can employ different navigation modes simultaneously. As second extension, we added a coordination of multiple Synergy client instances on public host machines for the X Windows implementation. Multiple pointers on a single machine are coordinated by a "floor control", sequentially assigning the exclusively available core pointer functionality to the present pointers. Multiple pointers are rendered by a plug-in for the *Compiz*[3] window manager.

EXPERIMENT

We compared the implemented navigation techniques in a single-user experiment on a representative multi-user MDE. Twenty users (11 male, 9 female, aged 23 to 48) participated in the experiment. No participant was familiar with the system or the employed display setup. The setup consisted of four XGA projections and two single-monitor workstations of 24" with 1920x1200 pixels as shown in Figure 2. Displays were driven by commodity desktop computers connected via gigabit ethernet. The users were seated in front of the left workstation monitor, with mouse and keyboard placed in front of the monitor. The right monitor was private and therefore not accessible. There were no occlusions of displays from the user's point of view.

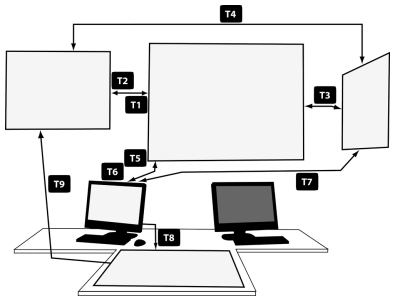

Figure 2. Experimental setup with transitions T1-T9.

We compared the following navigation techniques: free navigation (*free*), path navigation (*path*), a world-in-miniature control (*WIM*) to select the target display in the 3D model of the environment (similar to ARIS [3]), and a pointer warping technique (*warp*) redirecting the mouse pointer to a target display by pressing a keyboard shortcut. For *path*, navigation cues for the connected edge intervals were displayed. For each technique, users were asked to accomplish a target selection task starting from the home display. Targets appeared sequentially at different display locations, resulting in nine different transitions (*T1-T9*), illustrated in Figure 2. Each block was preceded by a short training phase. The order of navigation techniques and target appearances was balanced. Overall, the study lasted approximately one hour for each participant.

For task completion time, we measured the time between two target selections. Additionally, we collected subjective ratings of each technique on a seven-point Likert scale in a post-experiment questionnaire. Main effect and interaction analysis were performed at $\alpha = .05$ and Bonferroni adjustments were applied for post-hoc comparisons. One user test was declared as outlier and not included in the results.

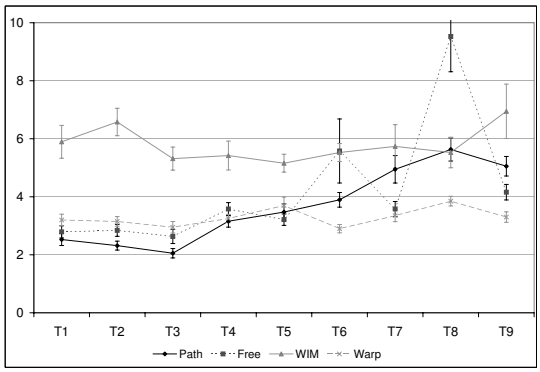

Figure 3. Task completion times (s) of all navigation techniques (average and standard error) for transitions T1 to T9.

A 4(navigation technique) x 9(transition) repeated measures ANOVA on target selection times showed main effects for navigation technique ($F_{3,54} = 33.801, p < .001$) and transi-

[3]http://www.compiz.org/

tion ($F_{8,114} = 18.439, p < .001$). There is also an interaction between navigation technique and transition ($F_{24,432} = 8.099, p < .001$). Overall, path was significantly faster than free. Warp was also faster than free, while WIM was the slowest technique overall. On average, path was faster than free for transitions between wall displays (T1-T4), which is significant for T2. It was also faster for accessing the tabletop display (T8). In contrast, free was faster than path for navigation *from* the tabletop display to the left projection wall (T9) and for navigation between monitor and the right projection wall (T7). Both, warp and WIM had almost uniform completion times across all transitions. Results are shown in Figure 3.

From the post-experiment questionnaire, we found no significant differences across preference scores ($F_{3,54} = 3.241, p = .461$). In an interview at the end of the experiment three users reported they preferred path navigation due to the given structure implied by the constraint, visualized paths, while three other users assessed path navigation as "exhausting" caused by the restrictive paths. Feedback for free navigation was similarly diverse: seven users rated free navigation as very intuitive while two users stated they did not understand the concept of free navigation at all. The main point of criticism was the mouse mapping on the table, leading to targeting difficulties on the tabletop display.

Path navigation seems to be more suitable for simple navigation tasks, such as between adjacent displays. In contrast, free navigation was superior for long and complex paths (T7 and T9) but had lower performance than path navigation overall. This contradicts findings by Nacenta *et al.* [8], who showed that perspective-based navigation is generally faster than stitching. Similar to our experiment, this difference was more distinct for complex navigation tasks. However, as Perspective Cursor was evaluated on a typical single-user workspace, there are two major differences in the experimental setup potentially leading to the divergent overall result: First, in their setup all displays were directly facing the user and, second, physical gaps between adjacent displays were comparably small. In our experiment, navigation to targets on the tabletop display (T8), which was partially located outside the user's field of view, caused serious problems for some of the participants. Many users mentioned a non-intuitive mapping of pointer movement on the table. This mapping is caused by the static perspective map, where the representation of the tabletop display is skewed and rotated relative to the actual device space (Figure 1). Navigation from the center wall to the monitor (T6) caused similar difficulties, as users sometimes involuntarily navigated to the tabletop display. It is probably sufficient to let users access displays located outside their immediate workspace exclusively by explicit pointer warping, which was the fastest technique in our experiment to access the tabletop display.

CONCLUSION

Deriving 2D mouse pointer navigation frames and transitions automatically from a 3D description of display spaces allows us to quickly build MDEs for collaborative workspaces tailored to the group size, architectural constraints, and the task to be accomplished by the team. Each team member is provided his or her own mouse and keyboard pair and can choose the preferred cross-display navigation technique, which is configured per mouse pointer. Our experiment has shown that estimating user positions and viewing directions from the 3D model is sufficient, as long as the displays are placed in front of the user. The results furthermore indicate that the "optimal" cross-display navigation technique depends on the user preference, which is strongly diverse, as well as the complexity of the display setup and the required mouse pointer travels. In the future we aim to extend our MDE to support more sophisticated tasks, like relocating content, in combination with the presented navigation techniques.

ACKNOWLEDGEMENTS

This work was funded by the Austrian Science Fund FWF (Y193 and W1209-N15) and FFG BRIDGE 822716.

REFERENCES

1. 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.
2. H. Benko and S. Feiner. Pointer warping in heterogeneous multi-monitor environments. In *Proc. GI 2007*, pages 111–117. ACM, 2007.
3. 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.
4. K. S. Booth, B. D. Fisher, C. J. R. Lin, and R. Argue. The "mighty mouse" multi-screen collaboration tool. In *Proc. UIST 2002*, pages 209–212. ACM, 2002.
5. M. Brown, A. Majumder, and R. Yang. Camera-Based Calibration Techniques for Seamless Multiprojector Displays. *IEEE Transactions on Visualization and Computer Graphics*, 11(2):193–206, 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, R. L. Mandryk, and C. Gutwin. Targeting across displayless space. In *Proc. CHI 2008*, pages 777–786. ACM, 2008.
8. 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.
9. C. Pirchheim, M. Waldner, and D. Schmalstieg. Improving spatial awareness in multi-display environments. In *Proc. VR 2009*, pages 123–126, 2009.
10. 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.