Deskotheque: Improved Spatial Awareness in Multi-Display Environments

Christian Pirchheim[*] Manuela Waldner[†] Dieter Schmalstieg[‡]

Graz University of Technology

Abstract

In this paper we present the multi-display environment Deskotheque, which combines personal and tiled projected displays into a continuous teamspace. Its main distinguishing factor is a fine-grained spatial (i. e., both geometric and topological) model of the display layout. Using this model, Deskotheque allows seamless mouse pointer navigation and application window sharing across the multi-display environment. Geometric compensation of casually aligned multi-projector displays supports a wide range of display configurations. Mouse pointer redirection and window migration are tightly integrated into the windowing system, while geometric compensation of projected imagery is accomplished by a 3D compositing window manager. Thus, Deskotheque provides sharing of unmodified desktop application windows across display and workstation boundaries without compromising hardware-accelerated rendering of 2D or 3D content on projected tiled displays with geometric compensation.

Keywords: Multi-Display Environment, Geometric Display Compensation, Collaboration, Mouse Pointer Navigation

Index Terms: H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces —Collaborative computing; I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms H.5.2 [Information Interfaces and Presentation]: User Interfaces—Windowing systems

1 Introduction

Inexpensive large format displays and projectors make it attractive to deploy multiple displays and thereby convert unused office spaces into collaborative teamspaces. The benefits of such a multi-display environment (MDE) have been studied by several research groups [11, 1, 2].

MDE research as mentioned above uses projectors only in very conventional ways, such as straight wall or tabletop displays. In existing offices with limited space and oddly shaped surfaces, such projector arrangements may not be feasible. The *Office of the Future* project [10] pioneered the use of geometric adaptation of projections to non-planar surfaces using projector-camera systems. Geometric adaptation of projectors has remained a popular research topic, but the geometric model that is implicitly acquired to achieve geometric adaptation has hardly been used to improve the user interface experience. Geometrically adaptive displays are typically designed for visualization of large 3D models on tiled projections, and do not provide application transparency.

However, the geometric model used in a geometrically adaptive display has high value for the collaborative user interface design. Not only does the geometric model provide freedom in choosing an appropriate physical projection area or combining multiple projections to achieve a desired aspect ratio and resolution, it also provides the foundation for seamless cross-display navigation. Automatic acquisition of a geometric model reduces the need for manual configuration of MDE properties, allows for dynamic reconfiguration, and enables multi-display interaction in tight and irregular office spaces.

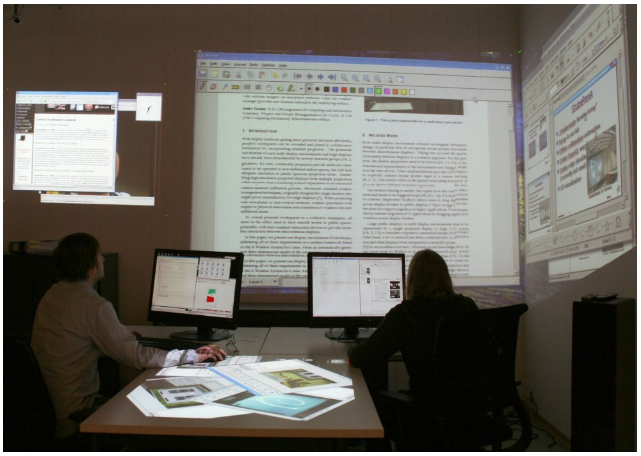


Figure 1: Deskotheque setup with multiple projections and personal workspaces.

Another particularly important goal of MDE research is application and network transparency – the system should let a single user work with unmodified desktop applications unaffected of display and machine boundaries, while the collaboration facilities of the MDE provide added value. Application transparency requires careful design of the new MDE facilities so that the operation of legacy applications is not disrupted.

The MDE Deskotheque described in this paper demonstrates the use of spatial awareness in a collaborative teamspace. The main contribution of Deskotheque is seamless multi-user mouse pointer navigation and window sharing across an irregular MDE in combination with geometric display compensation of projected displays. The system is based on the X Window System for Linux. It utilizes the inherent network transparency of the X Window System and the hardware-accelerated compositing window manager Beryl to achieve full application and network transparency without compromising 2D and 3D accelerated rendering.

2 Related Work

A common goal of multi-display environments is to create the illusion of a continuous interaction space by providing seamless mouse pointer navigation between discontinuous displays. Spatially consistent mouse pointer navigation can be achieved by stitching adjacent display borders [6, 11] or by creating a 2D representation of the display environment from the perspective of a tracked user [8]. Our system relies on the spatial relationship between displays to automatically derive mouse pointer navigation paths between adjacent display edge intervals.

The geometric relationship of individual displays is often determined manually [6, 8, 7] or by real-time tracking of mobile devices

[*] e-mail: pirchheim@icg.tugraz.at
[†] e-mail: waldner@icg.tugraz.at
[‡] e-mail: schmalstieg@icg.tugraz.at

[11]. None of these works report to use a camera-based calibration step to automatically extract the geometric relationship among multiple displays or individual display geometries.

Large public displays in multi-display environments are usually represented by large monitors [1], single projected displays [2, 8, 7, 11] or by proprietary displays [6] (e.g. interactive whiteboards). To our knowledge, Wallace et al. [16] presented the only multi display setup incorporating personal monitors in combination with an automatically aligned tiled display wall with software-based geometric compensation, but they do not include personal monitors in their calibration step and therefore lack a complete spatial model. In contrast, Deskotheque automatically obtains a model of the displays in an offline calibration step which is used for geometric compensation of projected displays, as well as for spatially consistent multi-user interaction.

Geometric compensation of projected displays requires the unmodified desktop image to undergo image-based transformations. Remote control software like VNC [12] or Microsoft RDP send compressed desktop pixels from a source host to a destination host. Several authors use modified VNC viewers to accomplish warping and blending on tiled projection walls (e.g. [17, 3]). An alternative for gathering window content is to exploit a windowing system's application redirection capabilities (e.g. [14]) and the usage of designated compositing window managers which access these window textures and accomplish a window composition to create the final desktop image. Deskotheque uses the OpenGL-based compositing window managers *Beryl*[1] and *Compiz*[2], which are available as standard components of popular Linux distributions, for hardware-accelerated warping and blending.

For sharing application windows across multiple displays and machines respectively, specialized ubiquitous computing middleware has been created, such as *Gaia* [1], *Roomware* [9], or *iRos* [5]. However, these frameworks usually do not support unmodified legacy applications. Conversely, *E-Conic* [7] employs rendering of unmodified window content provided by a dedicated VNC application server in a fullscreen OpenGL application. Similar to Wallace et al. [16], Deskotheque employs *XMove* [13] to dynamically change window assignment in an MDE exploiting the inherent network transparency of the X Window System.

3 SPATIAL AWARENESS

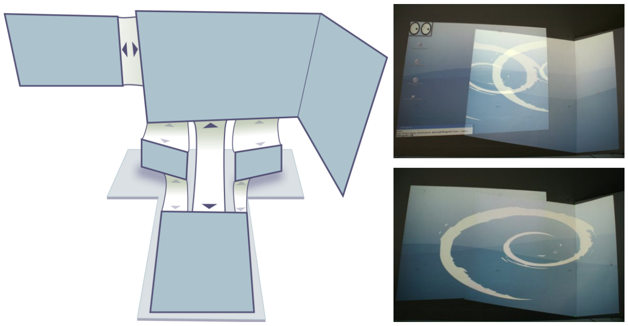
Our display environment interaction is adjusted to display geometries and spatial arrangement of displays. A three-dimensional model of the environment is automatically retrieved from a camera-assisted offline calibration step involving display-wise structured light patterns. This geometric information affects two conceptual layers:

On a global "*room*" level, the geometric model is used to infer display inter-connections. This enables us to support seamless navigation in the MDE with multiple mouse pointers and sharing as well as relocating of application windows (c.f. section 3.1).

On a local *display* level, geometric compensation of projected images provides adequate image quality for projected displays (c.f. section 3.2).

3.1 Seamless Navigation

From the geometric information, spatially consistent mouse pointer navigation paths can be automatically extracted. Deskotheque connects discontinuous displays at their closest adjacent edge portions (figure 2(a)). When reaching such a display edge, mouse pointer input will be redirected to the connected remote display. Connected edge portions are visually highlighted to ease the navigation task.



(a) (b)

Figure 2: (a) Spatially-aware mouse pointer navigation paths. (b) Two overlapping projections with geometric compensation and edge blending.

Cross display mouse pointer navigation facilitates dragging of application windows across disconnected displays, similar to the "hyperdragging" technique proposed by Rekimoto and Saitoh [11]. For large distance window movement, we also provide a menu function that lets a user choose a destination display host from a list integrated into the window's title bar.

By default, monitors are flagged as private while projected displays represent public display spaces. As a consequence, mouse pointer redirection and window sharing between private and public displays is restricted to the owner, i.e. the associated mouse pointer. Public displays are accessible for any user, either from their personal monitors, or from another public display.

3.2 Geometric Display Compensation

The design of public displays strongly depends on the task and data to be examined, as well as on the available room geometry. With a multi-display environment not providing automatic geometric compensation for projected displays, customized projected displays have to be carefully configured by hand.

To detect distorted projections and overlapping displays, individual display geometries are extracted from the 3D environment model. With multiple users in mind, we employ a geometric warping technique compensating the output image to appear like a physical rectangular wallpaper, on planar, as well as on multi-planar surfaces [15]. Tiled overlapping projections are warped to represent a rectangular display area. Intensity blending of the overlap area finally leads to a uniform projection. Figure 2(b) shows a desktop image without and with geometric compensation.

4 SYSTEM ARCHITECTURE

Deskotheque is implemented on top of the X Window System for Linux. It is tightly integrated into the windowing system and thus allows for application- and network-transparent cross-display mouse pointer navigation, application sharing, and geometric display compensation.

4.1 Deskotheque Framework

Deskotheque employs a distributed system involving multiple computer nodes ("*modules*") in a network. Modules may represent a single-user workstation consisting of a monitor with mouse and keyboard input, a tiled projected display without any input capabilities, or a self-contained device, such as a laptop. Individual modules are coordinated by a central master module, while the actual X client-server communication uses the standard peer-to-peer approach of the X protocol.

4.2 Framework Components

The framework functionality can roughly be sub-divided into three components: display management, mouse pointer redirection, and

[1] `http://www.beryl-project.org/`

[2] `http://www.compiz-fusion.org/`

window migration. These components largely operate on windowing system level. When designing these components, we obtained application transparency by leveraging existing components of the Linux and X Window System software infrastructure.

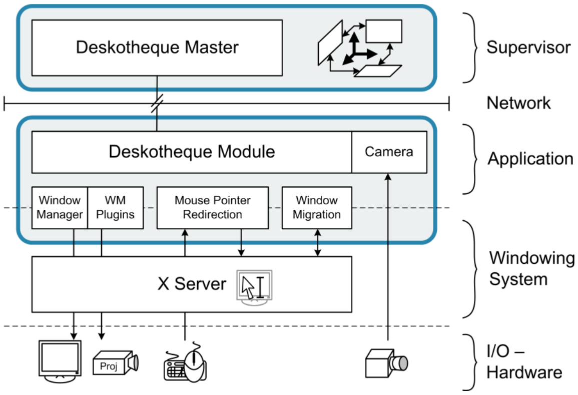


Figure 3: A Deskotheque module with associated components and input / output resources.

Each component was extended to serve as client application to the Deskotheque runtime communication (c.f. figure 3), which uses the distributed object library *Ice*[3]. It was strictly required that each component can be re-configured at runtime to react to changes in the environment. The master module updates individual components if modules are added, removed, or displays change location.

4.2.1 Display Management

Beryl is probably one of the most popular 3D compositing window managers for Linux. It employs an OpenGL extension to access window pixmaps stored in an off-screen buffer as OpenGL textures and thus makes efficient use of available graphics hardware for full hardware-accelerated 3D rendering. As a collection of plugins, advanced 2D and 3D window and desktop effects, such as window exposé and rendering of the desktop onto an rotatable 3D cube, are available.

The flexible plugin architecture of the Beryl window manager enables us to apply changes to the window management and desktop rendering without modifying the core implementation. Plugins can implement specific callback functions to directly influence rendering of individual windows, the desktop composition, or to query the X event loop.

Geometric compensation for multi projector displays requires the output image to be warped and blended before being sent to the framebuffer. Accessing the unmodified desktop image thus is essential to incorporate multi projector displays with legacy application support.

Our geometric projection compensation plugin to the Beryl window manager introduces an additional rendering pass, where the final desktop composition is rendered to an off-screen buffer to be warped and blended in the last pass (q.v. [15]). For the actual warping process, homography transformations are applied to the desktop texture. Each physical planar region has to be warped by a different homography matrix to compensate for oblique projection angles towards the individual wall portions. Overlapping projection areas are blended by rendering a full-screen texture on top of the desktop with intensity values coded in the alpha channel.

The display management component is not dependent on other framework components. Thus, Deskotheque can also be employed as stand-alone projected display with geometric compensation operated by a single module. Adding additional modules, mouse pointer redirection and window migration components provide multi-display interaction and collaboration facilities.

4.2.2 Mouse Pointer Redirection

Mouse pointer redirection is based on an extended version of the popular mouse and keyboard sharing tool *Synergy*[4]. The standard version allows sharing of a single mouse pointer and associated keyboard across multiple workstations. Synergy “stitches” displays together by connecting complete or partial display edges of different machines. As soon as the mouse pointer reaches a connected display edge the actual transition is invoked and the mouse pointer is “beamed” to the remote display. Subsequent to the relocation, Synergy grabs the input events on the local machine and redirects these events to the remote machine.

In Synergy’s original implementation, transition configurations are textually provided by the user at startup. The user specifies edges on the local and the remote display and, optionally, an interval along the edge. In Deskotheque, transitions between displays are dependent on the geometric model (see figure 2(a)) and subject to change at runtime, if modules are added or removed from the environment. Consequently, our modified Synergy implementation adopts to updated transition information at runtime.

In the Deskotheque environment, each module may provide a mouse and keyboard pair. Multiple users potentially move pointers without restrictions between displays and single displays may contain a varying number of pointers. In order to react to such situations, we made Synergy “multi-pointer aware”.

Per default, the X Window System provides a single *core* pointer per display and does not issue any input events for additional pointing devices. The *Xinput* extension of the X Window System provides an alternative device-aware *extension* event channel which exposes input events from additional pointing devices. This channel can be utilized for event redirection. A relocated core pointer is turned into an extension device on its host machine and thus does not issue core events to the local X Window System. Hence, the windowing system is disposable for receiving alternative core events, such as from a redirected remote pointer.

On each host, multiple Synergy instances are coordinated by the Deskotheque module. It establishes a simple floor control determining the *active* display pointer among the dynamic set of available pointers. The context switch comprises of changing the core device as well as warping the core pointer on the display. Extension pointer rendering is supported by the display manager component.

As the X Window System itself is unaware of multiple pointers, any floor control policy can only time-schedule the exclusively available window input focus to multiple pointers. Furthermore, concurrent pointer movement is handled sequentially – possibly resulting in pointer flickering. These obvious deficiencies have already been addressed by researchers [16] but their solutions usually require radical system constraints, such as a special window manager. Native support for multiple pointers for the X Window System is provided by the Multi-Pointer X [4] extension, which should soon be integrated in a major Linux release.

4.2.3 Window Migration

Deskotheque allows sharing of application windows between distributed displays and machines. We do not employ a designated application server to provide shareable application windows but rather enable the user to launch an application at any module and share this application across all participants. A designated application server implicates a potential bottleneck and restricts shareable applications to the server’s installed software.

The X Window System allows any X application (X client) to be operated by an X server on a remote machine by providing a

[3] http://www.zeroc.com/

[4] http://synergy2.sourceforge.net/

fully network transparent interface. However, X does not allow applications to change the X server at runtime. XMove operates as a so-called pseudoserver and thereby enables an application window to be migrated to a remote host at application runtime. The actual migration is transparent to the X server, as well as to the application itself. Migrated windows are fully functional and are subject to window manager operations, just like local applications.

Deskotheque establishes an XMove window migration, whenever a window is dragged over a display border or a destination host is selected from the window's title bar menu. In cooperation with the window manager and the mouse pointer redirection component, application windows can be dragged across display borders to adjacent displays along mouse transition paths. While Synergy triggers the mouse pointer transition and specifies the destination host, the window manager identifies the dragged window. These parameters are passed to the XMove component to invoke the window migration.

5 Results

The modular approach of our system allows for a quick configuration of various setups. Our example setup shown in figure 1 is driven by five PC workstations running Linux Ubuntu 7.04 and an additional master workstation. The two personal workspaces are connected to 22 inch monitors, while four XGA projectors are driving the projected displays. The wall projection covering the room corner was created by two overlapping projections. Two XGA cameras connected via gigabit ethernet were employed to capture structured light patterns during the calibration process. The initial calibration procedure takes less than a minute.

At runtime, the overall performance of legacy applications is hardly affected by running under the command of Deskotheque. With geometric compensation enabled, the frame rate is far above the physical display hardware limits. Using a 3D compositing window manager to apply geometric compensation allows for direct access of desktop imagery required for warping and blending on the graphics card. In contrast to common approaches to acquire the desktop image (e.g. via remote control software), no read back from the graphics card to main memory is required. This read back operation is expensive and severely limits the utility of such approaches in practice.

Shared applications suffer from a slight performance penalty when operated remotely caused by the XMove window operations due to the network latency and computational effort required for X protocol message manipulation. However, as no application server is required, this only affects applications that have been redirected by the user. Generally, network traffic within the Deskotheque framework is kept to a minimium: While framework components are configured by the master only at start-up or when the display configuration changes, the mouse pointer redirection component and the window migration component establish peer-to-peer connections solely on demand.

6 Conclusion

Deskotheque supports rapid deployment of MDE setups with heterogenous devices. The system maximizes the use of a spatial model of the environment by not only providing a continuous shared interaction space, but also by managing casually aligned tiled projected displays.

The tight integration into the windowing system allows for sharing of application windows and application-transparent display management. As geometric compensation is integrated into a hardware-accelerated 3D compositing window manager, there is no noticeable performance penalty when rendering the warped and blended desktop image.

In the future, the Deskotheque framework will help us to identify and investigate interaction design questions for heterogenous multi-display environments.

Acknowledgements

This project was funded in part by the Austrian Science Fund FWF under contracts Y193, and W1209-N15, and FIT-IT 813398. We would like to thank Albert Walzer for video production.

References

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

[2] J. T. Biehl, W. T. Baker, B. P. Bailey, D. S. Tan, K. M. Inkpen, and M. Czerwinski. IMPROMPTU: A New Interaction Framework for Supporting Collaboration in Multiple Display Environments and Its Field Evaluation for Co-located Software Development. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 939–948, 2008.

[3] D. Cotting, H. Fuchs, R. Ziegler, and M. H. Gross. Adaptive Instant Displays: Continuously Calibrated Projections Using Per-Pixel Light Control. *Computer Graphics Forum*, 24(3):705–714, 2005.

[4] P. Hutterer and B. H. Thomas. Groupware Support in the Windowing System. In *Proceedings of the eight Australasian conference on User interface*, pages 39–46, 2007.

[5] B. Johanson, A. Fox, and T. Winograd. The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *IEEE Pervasive Computing*, 1(2):67–74, 2002.

[6] B. Johanson, G. Hutchins, T. Winograd, and M. Stone. PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 227–234, 2002.

[7] M. A. Nacenta, S. Sakurai, T. Yamaguchi, Y. Miki, Y. Itoh, Y. Kitamura, S. Subramanian, and C. Gutwin. E-conic: a Perspective-Aware Interface for Multi-Display Environments. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 279–288, 2007.

[8] M. A. Nacenta, S. Sallam, B. Champoux, S. Subramanian, and C. Gutwin. Perspective Cursor: Perspective-Based Interaction for Multi-Display Environments. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 289–298, 2006.

[9] T. Prante, N. Streitz, and P. Tandler. Roomware: Computers Disappear and Interaction Evolves. *Computer*, 37(12):47–54, 2004.

[10] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 179–188, 1998.

[11] J. Rekimoto and M. Saitoh. Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 378–385, 1999.

[12] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper. Virtual Network Computing. *IEEE Internet Computing*, 02(1):33–38, 1998.

[13] E. Solomita, J. Kempf, and D. Duchamp. XMOVE: A Pseudoserver for X Window Movement. *The X Resource*, 11:143–170, 1994.

[14] M. van Dantzich, V. Gorokhovsky, and G. Robertson. Application Redirection: Hosting Windows Applications in 3D. In *Proceedings of the 1999 workshop on new paradigms in information visualization and manipulation*, pages 87–91, 1999.

[15] M. Waldner, C. Pirchheim, and D. Schmalstieg. Multi Projector Displays Using a 3D Compositing Window Manager. In *Proceedings of the 2008 workshop on Immersive projection technologies/Emerging display technologiges*, pages 1–4, 2008.

[16] G. Wallace, P. Bi, K. Li, and O. Anshus. A Multi-Cursor X Window Manager Supporting Control Room Collaboration. Technical Report TR-707-04, Computer Science, Princeton University, 2004.

[17] R. Yang, D. Gotz, J. Hensley, H. Towles, and M. S. Brown. PixelFlex: A Reconfigurable Multi-Projector Display System. In *Proceedings of the conference on Visualization*, pages 167–174, 2001.