Interaction Models for Multi-Display Slideshows

Patrick Chiu, Surapong Lertsithichai*, Qiong Liu
FX Palo Alto Laboratory, 3400 Hillview Ave., Bldg. 4, Palo Alto, CA, USA
{chiu, liu}@fxpal.com
*Silpakorn University, Faculty of Architecture, Bangkok, Thailand
surapong@post.harvard.edu

Abstract
We present an investigation of interaction models for slideshow applications in a multi-display environment. Three models are examined: Direct Manipulation, Billiard Ball, and Flow. These concepts can be demonstrated by the ModSlideShow prototype, which is designed as a configurable modular display system where each display unit communicates with its neighbors and fundamental operations that act locally can be composed to support the higher level interaction models. We also describe the gesture input scheme, animation feedback, and other enhancements.

1. Introduction

As displays become pervasive in business and home environments, a variety of configurations can be formed from multiple displays and it is challenging to design effective ways to interact with them. In the business setting, wall displays are often used for presenting slides. The main issues are how to make it easy for the speaker to move the slides around the displays and for the audience to follow the slides. In the living room at home, multiple TV displays can be used to show photos, video or multimedia content. Here, the style of interaction should be fun and entertaining.

A basic design issue with using multiple displays together is whether they are modeled as a single continuous surface or as discrete loosely coupled surfaces. The first model has been investigated in several research projects (e.g. [2], [3], [4], [6]); the basic idea is that a continuous surface is formed by mapping the edge of each display to the nearest edge of the nearest display. This model works particularly well as an extension of a work surface using a desktop metaphor. The second model is reminiscent of old lecture halls equipped with an array of chalkboards that slide around. For working with presentation slides, photos or multimedia, the second model is more suitable because it is natural for each display to show the content in full-screen mode. Previous work on connectable tables [5] and our work on meeting room wall displays [1] examined some of the issues.

In this demo, we show the ModSlideShow prototype [1] which employs a discrete model for supporting slide presentations in meeting rooms and showing photos (or videos) in living rooms. It can handle gesture input and provide animation for feedback. From our experience with designing and testing ModSlideShow, we have identified several models for moving content around a multi-display environment: Direct Manipulation, Billiard Ball, and Flow. We also describe the implementation and future applications.

Pervasive 2007 Invited Demo (May 13-16, 2007, Toronto, Canada)

2. Interaction Models

We assume that the user can perform input on at least one of the displays (e.g. touch panel, stylus, mouse & keyboard, etc.), or send content to one of the displays along a specified physical direction. Note that in general, displays and projectors do not come equipped with input devices. We also assume that the content is displayed full-screen (e.g. slides or photos). The user may be standing at a podium in a meeting room, sitting on a couch in a living room, or wandering around to interact with the various displays.

Due to the large physical distances involved in a multi-display environment, a user may be limited to interacting with one display at a time. The main novelty of the ModSlideShow application is making use of secondary effects of an operation to coordinate the content among the multiple displays. Moreover, it is sufficient to employ only *local* interactions between adjacent neighboring displays to support a wide range of useful scenarios.

The simplest scenario can be thought of as the *Direct Manipulation* model. In this model there are no secondary effects. The user can move a slide from a display to a neighbor display by making a gesture directly on the slide toward the target display. (Alternatively, the user can use an arrow key on a keyboard or some other input device attached to the display.) Examples of this interaction on a 3-display configuration are illustrated in Fig. 1(a) and 1(c). When the source of the content is on the center display (Fig. 1(a)), this works fine; however, if the source is on one of the side displays (Fig. 1(c)) this model is not effective because the it is difficult to put content on the far display. For more than 3 displays, this model can be very inefficient and tedious.

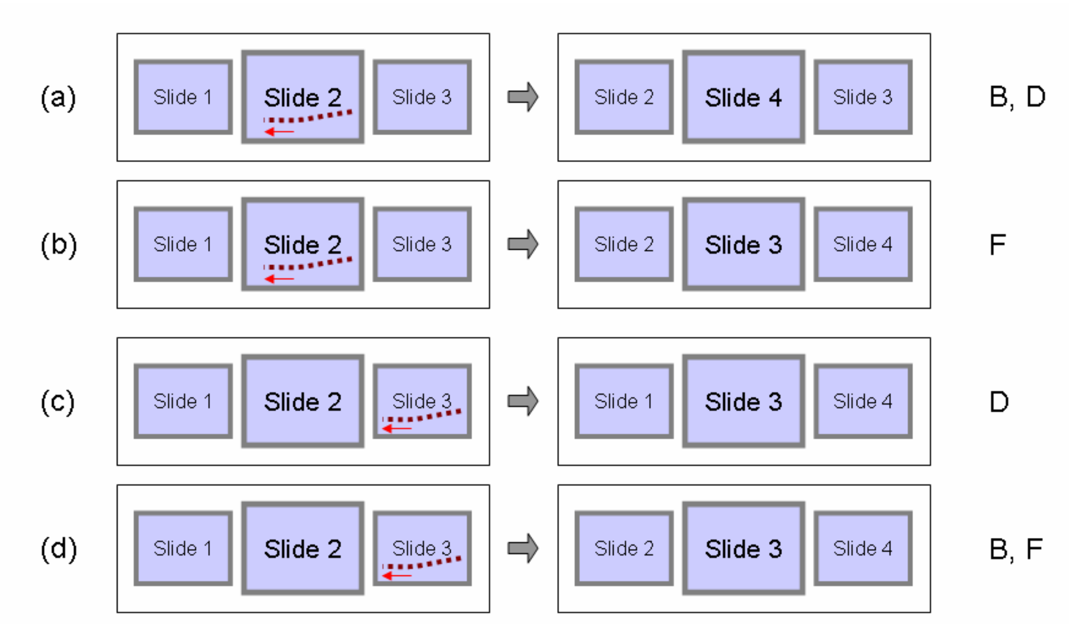

Figure 1. Examples of "Move Left" operations and the results that can occur with a 3-display configuration under the different models: Billiard Ball (B), Direct Manipulation (D), Flow (F). The source of the content is on the on the center display for (a), and on the rightmost display for (b), (c), (d).

The *Billiard Ball* model, as the name suggests, is analogous to a physical model in which objects collide and push one another along a direction vector. In this model the secondary effect is triggered when a slide is sent to a display along a specified direction (e.g. left, right, up, down). The receiving display then sends its slide to the adjacent neighbor along the same direction; this secondary effect is propagated locally display by display until there are no more displays. See Fig. 1(a) and 1(d). This model works well for content that has more complicated structures or special images that needs to be set aside (e.g. an overview or topic slide).

The *Flow* model is analogous to liquid flowing. In contrast to the Billiard Ball model, the space that opens up during a slide movement is filled up by pulling in a slide from the display in the opposite direction of the target direction. There are two local actions that are propagated: pushing along the forward direction and pulling from the backward direction. See Fig. 1(b) and 1(d). Flow also takes much less effort than Direct Manipulation. An example is the task of showing the most recent N slides on N displays, where a single flow move operation is equivalent to (N-1) direct manipulation move operations. This model is highly effective for slides or photos that are meant to be shown sequentially.

3. Enhancements

Through rapid prototyping and testing, new interaction techniques were developed and suggested by users for improving ModSlideShow. Initially, there was a basic gesture set (and the associated keyboard equivalents). See Fig. 2. This was enhanced by adding support for flicking the gestures, enabling a user to move slides farther based on the gesture motion acceleration.

One user suggested a less obvious enhancement: a transpose operation that swapped two neighboring slides. Its gestures are shown in Fig. 2. Note that since the transposition operations generate the permutations, the slides can be moved on the displays to any choice of desired positions.

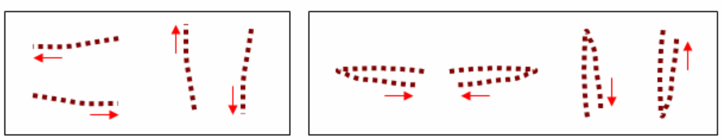

Figure 2. The basic gesture set {LEFT, RIGHT, UP, DOWN} and the enhancements {TRANSPOSE_LEFT, TRANSPOSE_RIGHT, TRANSPOSE_UP, TRANSPOSE_DOWN }.

Animation is employed as an effective way to provide feedback to the user and the audience viewing the content. When a slide leaves one display and enters another display, a smooth animation shows the movement. Unlike gratuitous animated transitions in single-display slide presentations, in a multi-display system the animation effect helps people see the movement of the slides and follow the presentation.

4. Implementation

ModSlideShow is built as a *modular display system* in which each display unit communicates locally with its adjacent neighbors. The topology is peer-to-peer. The fundamental operations are PUSH and PULL, with an option of whether or not to PROPAGATE. With these fundamental operations, it is possible to derive other operations such as TRANSPOSE, and at a higher level the Billiard Ball, Direct Manipulation, and Flow models can be realized.

Once the displays are physically set up in the environment, each instance of the ModSlideShow software is configured by entering its neighbors' computer names (or IP addresses) and physical relationships {LEFT, RIGHT, UP, DOWN}. A set of displays can also be assigned a group name.

Figure 3. A meeting room with 3 displays.

Each display unit can be set up with a source, which points to a list of content elements (e.g. a directory with JPG or TIF images). A user can enter the source through a dialog box. Typically, only one of the displays will have a content source, usually the display that the user interacts the most with in the environment.

ModSlideShow is implemented in Java, and the communication is via XML-RPC.

5. Conclusion & Future Work

We presented a study of three interaction models for multi-display slideshows and the flexible ModSlideShow system that is capable of supporting these models. Other models tailored to more specialized applications include a multi-cast model for sending a slide simultaneously to a set of displays for large audiences, a mirror model for teleconferences, and an augmented-reality model where the user can move content around using a "window-on-the-world" style metaphor. See [1] for details. In practice, choosing the appropriate model depends on the physical configurations of the displays in the environment and the type of applications that will be run.

Interesting areas for further exploration include utilizing sensors to automatically detect and configure a set of displays that are in close proximity, and showing dynamic content from RSS feeds and blogs in a multi-display environment.

6. References

[1] Chiu, P., Liu, Q., Boreczky, J., Foote, J., Fuse, T., Kimber, D., Lertsithichai, S., Liao, C. Manipulating and annotating slides in a multi-display environment. *Proceedings of INTERACT ''03*, pp. 583-590.

[2] Johanson, B., Hutchins, G., Winograd, T. & Stone, M. PointRight: Experience with flexible input redirection in interactive workspaces. *Proceedings of UIST '02*, pp. 227-234.

[3] Rekimoto, J. Pick-and-Drop: A direct manipulation technique for multiple computer environments. *Proceedings of UIST '97*, pp. 31-39.

[4] Streitz, N.A., Geißler, J. & Holmer, T. Roomware for cooperative buildings: Integrated design of architectural spaces and information spaces, *Proceedings of CoBuild '98*, pp. 4-21.

[5] Tandler, P., Prante, T., Muller-Tomfelde, C., Streitz, Steinmetz, R. ConnecTables: Dynamic coupling of displays for the flexible creation of shared workspaces. *Proceedings of UIST '01*, pp.11-20.

[6] Tani, M., Horita, M., Yamaashi, K., Tanikoshi, K. & Futakawa, M. Courtyard: Integrating shared overview on a large screen per-user detail on individual screens. *Proceedings of CHI '94*, pp. 44-50.