LumiPoint: Multi-User Laser-Based Interaction on Large Tiled Displays

Xing Chen, James Davis
Computer Graphics Laboratory, Stanford University
{xcchen, jedavis}@graphics.stanford.edu

Fig. 1. A large visualization display on which multiple users can interact using laser pointers.

Abstract. Projection technology has made large tiled displays an exciting feature of advanced visualization systems. However, one challenge to using such systems effectively is the development of input technologies that comfortably allow collaborative interaction with the displayed data on these large surfaces. We have developed an input system that allows any number of users to interact simultaneously with data on a large display surface. In this system, each user utilizes a laser pointer as a pen or pointing device. Using computer vision techniques, the system determines the beginning, continuation, and end of all currently active pointer strokes. The coordinates of each stroke are available for use by visualization applications. This system scales well with display size, display resolution, and number of users.

1 Introduction

Large tiled displays are well-suited for visualization applications for multiple reasons. Their high resolution enables the detailed display and exploration of complex data sets. In addition, the large display surface, which is accessible by several people, makes collaborative interaction with the data possible. Existing large-format systems primarily focus on rendering and displaying large amounts of data, but have few convenient provisions for interaction.

Enabling natural interaction on large tiled displays is a key challenge in advanced visualization systems. Since users tend to stand in front of the display, rather than sit at a desk, traditional keyboard and mouse interfaces are unsuitable. In addition the large size makes it cumbersome or impossible to physically reach all parts of the display. The scalable nature of tiled display technology in terms of size and resolution is another challenge. Input technology must be similarly scalable, in order to provide high quality interaction over the entire display surface. Since the large format of tiled displays encourages multiple people to collaborate, input technology should also scale in number of users. On a wall size display, it is easy to imagine not just a few, but many users simultaneously interacting with a data set. Many researchers in human-computer interaction have shown that collaborative work can improve productivity [11][10][4][2][20]. An ideal interface would allow multiple people to comfortably collaborate directly on the display surface.

In this paper we present an input system that was developed specifically to provide interaction with a tiled, back-projected display in our lab called the "Interactive Mural". The high resolution display surface is 6 ft by 2 ft and covered by eight projectors. It is bright enough to be used in normal office lighting and appears as a single seamless display device to users in front of the screen (Figure 1). It is designed to support complex visualization of large heterogeneous databases. The display rendering is powered by an eight processor shared memory SGI Origin, with two Infinite Reality graphics pipes [9].

In this system, each individual uses a laser pointer to directly manipulate data displayed on the screen. Cameras oriented towards the display surface observe all laser spots on the display directly. Each camera is connected to a CPU that digitizes the incoming video streams and finds all laser spot locations within the camera field of view at each time step. This data is communicated to a central estimator that determines, based on time coherence and motion dynamics, whether the laser spot is the beginning, continuation or end of a stroke. A stroke is the continuous path of a laser spot on the display surface from the appearance of the spot to its disappearance. This information, together with an estimate of the position, velocity and acceleration of each active stroke, are available for use either directly by a visualization application or indirectly after interpretation by a gesture recognition module.

This input system addresses some key challenges in large format visualization. The choice of laser pointers as the input apparatus provides an intuitive metaphor for a wall-like display. In addition, laser pointers allow wireless, remote access to all parts of the display surface. As the tiled-display increases in size and resolution, this input system is easily reconfigurable. As many cameras as necessary can be added in order to ensure the desired coverage and input resolution. Furthermore, since multiple laser spots can be tracked, as many users as desired can participate in the interaction simultaneously. This contribution is important because it removes a fundamental previous limitation, allowing not only a collaborative discussion, but also collaborative work on a single large display surface.

2 Related Work

Many large tiled displays have been built, but as previously noted, these systems typically do not provide for natural user interaction. [9][18][15] It was precisely this lack of existing scalable large format interaction that motivated our work.

Existing research on human interaction for back projected displays tends to be focused on smaller, single projector systems. In this context the need for a scalable technology does not arise. Some previous systems employ the idea of a camera mounted behind a projection screen to detect user input. The HoloWall system [12] uses infrared cameras to detect hands or objects that come into contact with a display surface. The Liveboard project [6][14] uses a rear mounted photo-effect diode that detects the location of an LED instrumented stylus. The LED intensity is modulated, allowing the system to identify a few different users using frequency multiplexing. Other computer vision based systems exist that attempt to use a camera to detect user motion or gestures. For instance, the VIEP system [17] included vision based input in a collaborative work space. However, none of these systems address the need to scale in terms of display size, display resolution, and the number of users.

A number of display surface input technologies make use of specialized touch-sensitive hardware to allow user input. For instance, touch screens are a common solution. Microtouch [13] and SmartTech [19] are two commercial vendors of whiteboard sized touch screens. These devices typically allow only a single user and have a limited and fixed size or resolution. Strickon and Paradiso [21] introduced a system that uses a projected sheet of laser light just above the display surface. A linear CCD detects the light reflected when an object penetrates this plane. None of these systems address the accommodation of multiple users. In addition, systems based on touch-sensitive hardware require physically touching the screen, which may be difficult or impossible on very large displays.

Many digital whiteboard technologies have been coupled with a single projector for an interactive display system. For instance, Rekimoto [16] introduced a whiteboard system that specifically addresses the needs of multiple user interaction. Rather than allowing multiple users to interact directly on the display surface, this system uses multiple display devices. Each user carries their own palmtop computer that acts as a holding area for personal options and new content as it is created. Users can then copy this new content to the whiteboard when it is complete and ready for public viewing. The focus of Rekimoto's work is high-level interaction methodologies built upon the whiteboard’s underlying input technology. Our system addresses the need for underlying hardware and software technology that facilitates simultaneous multiple user interaction in environments where the display may be much larger than a typical whiteboard.

Our system allows users to interact with a wall size display, either by directly drawing on the surface or by pointing from a distance. In addition, we believe that our system is the first that scales in display size, display resolution, and number of users.

3 Hardware Architecture

The Interactive Mural is a high resolution display providing rendering and interaction with large complex data sets. A laser pointer input system provides the display’s primary mode of interaction. Figure 2 gives an overview of the hardware architecture that supports this laser-based input. Behind the display wall, where the projectors reside, a number of NTSC cameras are mounted. The cameras do not need to be synchronized, for reasons that will be described later. Laser pointer input appears as a bright spot on the back of the display surface, visible to the cameras. Individually, each camera only observes a fraction of the display. However, considered as a group, the overlapping fields of view of all cameras cover the entire display surface. Our initial system used two cameras, providing a pointer resolution of 1200x240 over the display surface. We later scaled the system to eight cameras in order to increase the effective pointer resolution. We chose a multi-camera architecture precisely because it allows the input system to scale in response to user requirements.

Laser pointers are typically more intense than other light sources. However in the case of extremely bright projectors, wavelength bandpass filters can optionally be installed to increase the laser to projector contrast.

The video stream from each camera is digitized at up to 60 Hz by an SGI Indy workstation. In addition to digitization, each CPU locates any laser spots in its field of view. The 2D location of each spot in local camera coordinates is sent over Ethernet to an SGI Onyx2 (195 MHz R10K), which integrates information from all cameras into a stream of input events.

4 Software Architecture

Whenever the laser appears and moves across the display surface a stroke is produced. Our laser input system observes all laser spots on the display screen from a collection of cameras and produces a consistent set of user input events, including “stroke begin”, “stroke end” and “current stroke position”. This process is challenging because each camera makes independent observations, and laser spots from each of several pointers may have no unique characteristics.

In our architecture, laser spots are identified in local coordinates by each camera-CPU pair. These locations are communicated to a central estimator which fuses independent measurements into a single stream of input events. Additional modules support differentiation and processing of laser strokes initiated by multiple simultaneous users.

Since this system is meant to support user interaction, efficiency is important. The latency introduced by our software processing pipeline is under 20ms when supporting three simultaneous users. This is sufficient for supporting interaction, and lower than the latency introduced by other parts of our visualization framework.

4.1 Locating the laser spot

Laser spots must be extracted from each observed video stream. As described in the last section, laser pointers appear much brighter than the projector display to our cameras. We can use simple thresholding and classify any pixel that is brighter than a threshold T as a laser spot. Since laser spots seen through our diffusive projection

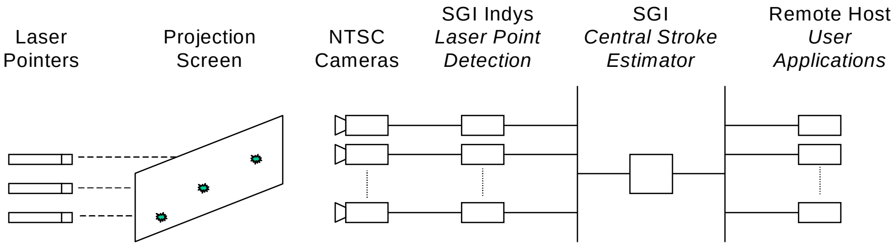

Fig. 2. Laser pointers are used as input devices on a wall size display. An array of cameras and CPUs placed behind the screen detects the projected laser spots. Information from individual cameras is filtered and aggregated by a central estimator to provide a stream of stroke events to visualization applications.

screen typically occupy more than one pixel on the camera CCD, we additionally aggregate regions of neighboring bright pixels into a single observed measurement. We take the mean location of participating pixels as the observed location. Typically a single laser point will produce a spot with a diameter of between three and twelve pixels. The exact size depends on the camera field of view and iris settings.

Conceptually, we check every pixel in a camera’s field of view to see if it should be classified as a laser spot. In practice, we can use a combination of subsampling and prediction to adjust the number of pixels inspected, depending on desired processor load. Since the size of a projected laser spot is larger than one pixel, we have found that checking approximately 30% of all pixels is adequate.

After extracting observed laser points in local camera coordinates, these locations are communicated via Ethernet to a stroke estimation server. The server uses all observations to maintain an estimate of the current number, location, and velocity of laser pointers in use.

4.2 Estimating Point Location

Laser interaction is detected by a set of independently observing cameras. Some method needs to be employed to merge these observations into a single consistent estimate of user input. The central stroke estimator in our system integrates data from all cameras into a single stream of information with respect to a common frame of reference.

Each camera observes laser points in its own frame of reference. We calibrate all the cameras [8] so that their positions and orientations relative to the display are known. Thus the 2D locations in the camera reference frame can be easily transformed to the display reference frame. However, some filtering of the data is necessary for the following two reasons. First, as the pointer traverses the screen, it will move in and out of each camera's field of view. The filter needs to merge these separate pointer positions into a single continuous stream of events. The events include the start, continuation and end of a stroke. Second, the 2D locations obtained by the low level point extraction routines are noisy, both because the point detection is imperfect and because the laser spots tend to jitter on the screen when pointers are used from a distance. The central estimator needs to filter the incoming data to produce a single smooth stroke absent of any pointer jitter.

The following example illustrates the necessity of using a filter. Consider a laser point projected into the overlapping region of two cameras and not moving at all (Figure 3). Both cameras report a measurement to the central server that is in turn converted to a position on the display. These two positions are two consecutive points on a stroke. If the cameras are perfectly calibrated, the two positions should be the same. However, in reality, there is always some slight error in the calibration, and measurements from cameras A and B will be converted to slightly different positions on the display. Therefore, without filtering, even if the actual laser pointer is not moving, the estimated pointer position will appear to jump back and forth rapidly between points A and B, in sync with the 60 Hz updates from each camera. Even if the miscalibration is only a few pixels, this effect is very distracting to the user.

4.3 Fusing Measurements

We use a Kalman filter to integrate our data and to estimate the pointer path. This is a well-known tool for parameter estimation of dynamic systems given direct or indirect noisy measurements. In this section we give an intuitive description of how a Kalman filter works. A comprehensive tutorial on the Kalman estimation framework can be found in [5].

In the specific case of tracking a single laser pointer, the "system states" to be estimated are the 2D position, velocity and acceleration $[\mathbf{p}, \mathbf{v}, \mathbf{a}]$ of the laser spot in screen space, and the noisy "measurement" is the 2D position $\mathbf{z}$ in camera image space. When a measurement is observed at time t, the Kalman filter performs the following steps: 1. *State prediction*: predict the new state at time t from the previous state based on a "constant acceleration" dynamic model: $\mathbf{a}_t=\mathbf{a}_{t-\Delta t}$, $\mathbf{v}_t=\mathbf{v}_{t-\Delta t} + \mathbf{a}_{t-\Delta t} \cdot \Delta t$, $\mathbf{p}_t=\mathbf{p}_{t-\Delta t} + \mathbf{v}_{t-\Delta t} \cdot \Delta t + \mathbf{a}_{t-\Delta t} \cdot (\Delta t)^2/2$; 2. *Measurement prediction*: predict the expected measurement at time t from the system state, i.e. $\mathbf{y}_t=\mathbf{H}\mathbf{p}_t$, where $\mathbf{y}$ is the predicted 2D position in camera space and $\mathbf{H}$ is the projection, based on each camera calibration, from screen coordinates to camera coordinates. 3. *Kalman state update*: compare the predicted measurement $\mathbf{y}$ and the actual measurement $\mathbf{z}$, and update the final estimate of the state based on observed discrepancies and the noise model of the state and measurement processes. After updating the system state, the newly estimated pointer location is output to any visualization applications that expect pointer input.

Conceptually, the final state estimate is a weighted average of the predicted and the observed location, optimal in a least square sense. This effectively removes the high frequency noise present in the observed point location. It should also be noted

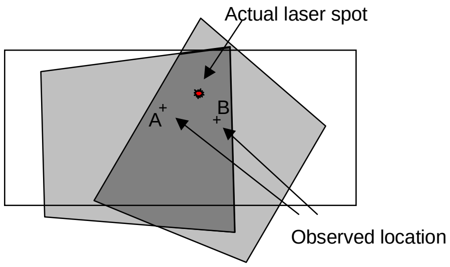

Fig. 3. Due to small miscalibrations, observations from camera A and B of the same laser point are transformed to different locations in the global coordinate space. Without filtering, user input will exhibit high frequency jitter between these locations.

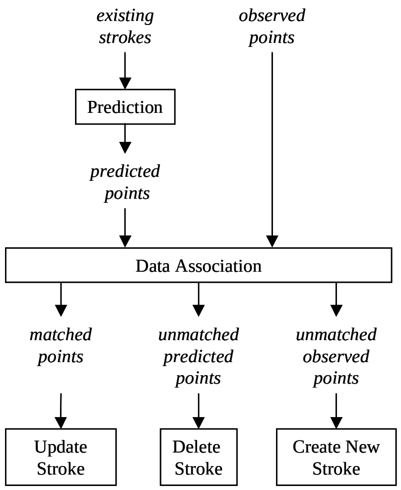

Fig. 4. The central estimator associates laser points observed by the tracking subsystem with points predicted by existing Kalman systems (strokes). Matched points revise the relevant stroke state, while unmatched points cause strokes to be added or deleted.

that since the dynamic model in the Kalman filtering framework explicitly describes how states vary with time, inputs to this filter need not be synchronized as long as they are time-stamped with respect to a universal time reference.

4.4 Managing Multiple Strokes

Thus far, we have only considered a single laser pointer in use. In practice, we may have numerous pointers simultaneously active. Laser spots originating from distinct strokes must be distinguished from one another, and information about each active stroke must be maintained.

Our method creates a separate Kalman system for each active pointer stroke. Whenever the central estimator receives a camera observation, data association needs to be employed to match each observed point with the correct stroke. We predict the expected position of each active laser stroke based on its previous position and velocity, and compare these to each observed position.

We first need to determine, for each observed point, whether it could be caused by any existing systems (strokes) in the filter. This determination is made by checking whether the observed point lies within an elliptical region around each predicted point. This region is the validation region that is standard in many Kalman filter based tracking applications [1]. If the observed point lies in the region it is considered “valid” and associated with an existing stroke. An observed point which is distant from any predicted point will be considered as the start of a new stroke.

For each valid observation, we associate the observed point with the closest predicted point. We have found that this data association technique works well in practice. Even strokes which intersect on the display surface are rarely interchanged. However, this process is not perfect, and there are several reasons that the wrong association could be chosen. Perhaps the user exerted such a large acceleration to the laser pointer that the actual observed value lies far from the prediction. An observation system with a high update rate mitigates this effect. Points do not move very much between fields when a 60 Hz update rate is maintained. Another difficulty arises when two laser points are positioned near each other. Because of the close proximity of the points, the likelihood is increased that the wrong associations will be chosen. Although this is a theoretical concern, it has little implication in practice, precisely because the points are close together. Even if the wrong association is chosen, the fact that it is nearby ensures that there will be minimal resulting impact on the estimated trajectory. Drastic errors are not made because the association is bounded by the size of the validation region. After finding associations we update the state of each Kalman system (stroke) that was observed by the current camera.

4.5 Adding and Deleting Active Strokes

After associating valid observations, there may remain observed points with no associated stroke, and strokes with no observed point. In these cases, systems may need to be added or deleted. Figure 4 shows the processing performed by the central estimator, including the conditions under which strokes are added and deleted.

When a new laser pointer becomes active, there is not yet an active stroke with which the observed laser point can be associated. After validating each observed point and associating them with one of the predicted states, we check for any observations left with no association. These observations are treated as the beginning of a new stroke. We initialize a new Kalman system with position at the observed location, and velocity zero. This new system is added to the existing set of active strokes.

When a stroke ends and the laser pointer is turned off, there is no explicit signal that indicates that this has occurred. Failure to observe a predicted point could mean any number of things. Perhaps the laser point lies outside the viewing area of the current camera, or possibly the low level tracking failed on this field and the laser point was simply missed. However, we do know that if the laser point is on, it should be observable by some camera. Since our tracking is fairly robust, we believe that the point will not be overlooked during low level tracking for more than a few frames. Thus, if we do not associate any observation with an active stroke for some duration, we declare the stroke finished and remove it from the active set of Kalman systems. We set this duration to 500 ms, but the exact value is a tradeoff between latency in receiving stroke end events and the reliability of the underlying observation system.

4.6 Supporting Multiple Users

After finding associations, updating the appropriate Kalman states, creating new strokes, and deleting inactive strokes, our system communicates the new information to user applications. In addition to stroke-begin and stroke-end events, our system sends the current position of each active stroke to user applications.

Our early multi-user applications tended to assume that each user would be identified uniquely based on laser pointer color. We have both red and green laser pointers, and additionally experimented with making light pens from a variety of LED colors. While this is the most straightforward way to retrofit existing single user applications, many interactions do not actually require knowledge of which physical pointer is actively in use. Rather, the context within which the strokes are placed in the application is of primary importance. For example, there may be no need to identify which user selected an option from a menu or edited a document, only that the action occurred. Furthermore, some interface technologies are fundamentally context free. For instance, Bier et. al. [3] discussed click-through-tools. This interaction technique explicitly replaces the standard user tool palette with on screen transparent lenses that affect underlying data when a user clicks on them. Additionally, even when applications store context such as preferences on a per user basis, physical tags may not be the only way to disambiguate context. For instance, Rekimoto [16] uses temporal coherence of strokes to determine which of several identical user styluses has generated a stroke on the whiteboard in the Pick-and-Drop system.

Given the variety of user interaction methodologies available to applications, we abandoned the naïve assumption that a physical identifier is required. Currently, in addition to laser color each event is tagged with an ID field, indicating to which stroke it belongs. Although the user application may not know from which physical laser pointer a stroke originates, user applications can resolve consistent click and drag sequences, even when multiple strokes of the same color are active in a single region of the work space. Currently, when multiple users interact using our system, the primary modality is with red laser pointers only.

5 Discussion

The fact that a laser pointer is by definition a pointing device, and is usually shaped liked a pen, makes it a natural and intuitive apparatus for interacting with a wall-like display. Our laser input system was originally installed as one of several input devices for the Interactive Mural. During the course of a year, users shifted from using wireless mice and trackballs to full time use of the laser tracking system. Additionally, we have observed that lab visitors immediately understand how to control applications using a laser pointer, while our previous inertial wireless mouse required instruction before users could use it effectively.

The remote nature of laser pointers is another important aspect of their use as an input device. While some users tend to approach the display and draw with the laser pointer as if they are using a whiteboard, it is not unusual for a collaborating party to make some small adjustment to the work without approaching the board. In addition, our system is useful on large displays where physically reaching all portions of the display is impossible.

The stroke locations reported by our system may also be used to generate some higher level input events. For example, we have a data presentation tool to visualize data stored in a hierarchical structure. At each node some combination of images, text, video or animated graphics is displayed on the projection wall. Stroke based gestures are used to navigate the data hierarchy. We make use of the existing Unistroke toolkit [7]. The output stroke positions of our tracking system pass first through the gesture recognition module, and recognized strokes are added as a symbolic event onto the application queue.

Our architecture provides an input device that scales with the associated tiled display. Our original installation used only two cameras. We have since installed both four and eight camera systems. In addition to our own tiled display, at the request of an outside research institution, we installed another laser based input system in their laboratory. Their tiled display uses twelve projectors and covers an 8 ft by 12 ft area. The laser tracking for this display is handled by four cameras attached to Wintel PCs with Matrox video digitizing boards. These systems work as expected and we feel confident that cameras can be scalably added in order to obtain any required resolution or range of coverage. The scalable nature of this technology makes it easily adaptable and deployable for displays with different size and resolution configuration.

The laser input architecture described here supports not just one or two, but many simultaneous users. We have exploited the systems capabilities with up to six active laser pointers, and feel that the collaboration enabled by multi-user interaction is an important component of our visualization framework.

6 Conclusion

We have presented an input technology designed for large-format *interactive* visualization systems. Laser pointers are used as an intuitive, wireless input device. The underlying hardware and software supports multiple users working collaboratively. Furthermore, it is scalable in terms of both the number of users, and the size and resolution of the display, making it particularly suitable for large tiled displays.

References

1. Bal-shalom, Y. and Fortmann, T. E. *Tracking and data association*, Acadamic Press, 1988.
2. Bier, E.A. and Freeman, S. MMM: a user interface architecture for shared editors on a single screen. *Proceedings of the ACM Symposium on User Interface Software and Technology*, p. 79-86, 1991.
3. Bier, E.A., Stone, M.C., Fishkin, K., Buxton, W., and Baudel, T. A taxonomy of see-through tools. *CHI'94 Conference Proceedings*, p. 358-64, 1994.
4. Bricker, L.J., Baker, M.J., and Tanimoto, S.L. Support for cooperatively controlled objects in multimedia applications. *CHI 97 Extended Abstracts*, p. 313-14, 1997.
5. Brown, R. G. and Hwang, P. Y. C. *Introduction to random signals and applied Kalman filtering*, 2nd Edition. J. Wiley, 1992.
6. Elrod, S., Bruce, R., Gold, R, Goldberg, D., Halasz, F., Janssen, W.; Lee, D., McCall, K., Pedersen, E., Pier, K., Tang, J., and Welch, B. Liveboard: a large interactive display supporting group meetings, presentations and remote collaboration. *CHI '92 Conference Proceedings*, p 599-607, 1992.
7. Goldberg, D. and Richardson, C. Touch-typing with a stylus. *Proceedings of INTERCHI '93*. p. 80-7, 1993.
8. Heikkilä, J. and Silvén, O. (1997) A Four-step Camera Calibration Procedure with Implicit Image Correction. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, p. 1106-1112, 1997.
9. Humphreys, G. and Hanrahan, P. A distributed graphics system for large tiled displays. *Visualization '99 Conference Proceedings*, 1999.
10. Inkpen, K., McGrenere, J., Booth, K.S., and Klawe, M. The effect of turn-taking protocols on children's learning in mouse-driven collaborative environments. *Proceedings of Graphic/Vision Interface '97*, p. 138-45, 1997.
11. Ishii, H. and Kobayashi, M. ClearBoard: a seamless medium for shared drawing and conversation with eye contact. *CHI'92 Conference Proceedings*, p. 525-32, 1992.
12. Matsushita, N. and Rekimoto, J. HoloWall: designing a finger, hand, body, and object sensitive wall. *Proceedings of UIST '97*, p. 209-10, 1997.
13. Microtouch Systems, http://www.microtouch.com/.
14. Pederson, E.R., McCall, K., Moran, T.P., and Halas, F.G. Tivoli: an electronic whiteboard for informal workgroup meetings. *Proceedings of INTERCHI '93*. p. 391-8, 1993.
15. Raskar, R, Welch, G., Cutts, M., Lake, A., Stesin, L., and Fuchs, H. The office of the future: a unified approach to image-based modeling and spatially immersive displays. *SIGGRAPH 98 Conference Proceedings*, p. 179-88, 1998.
16. Rekimoto, J. A multiple device approach for supporting whiteboard-based interactions. *CHI '98 Conference Proceedings*, p. 344 -351, 1998.
17. Rohall S. L. and Lahtinen, E. P. The VIEP system: interacting with collaborative multimedia", *UIST '96*, p. 59-66, 1996.
18. Samanta, R., Zheng, J., Funkhouser, T., Li, K., and Singh, J. P. Load balancing for multi-projector rendering systems. *SIGGRAPH/Eurographics Workshop on Graphics Hardware*, August, 1999. http://www.cs.princeton.edu/omnimedia/index.html.
19. Smart Technologies, Inc., http://www.smarttech.com.
20. Stewart, J., Bederson, B. B., and Druin, A. Single Display Groupware: A Model for Co-present Collaboration. *CHI '99 Conference Proceedings*, p. 286-93, 1999.
21. Stricken J. and Paradiso, J. Tracking hands above large interactive surfaces with a low-cost scanning laser rangefinder. *Proceedings of CHI'98*, p. 231-232, 1998.

.