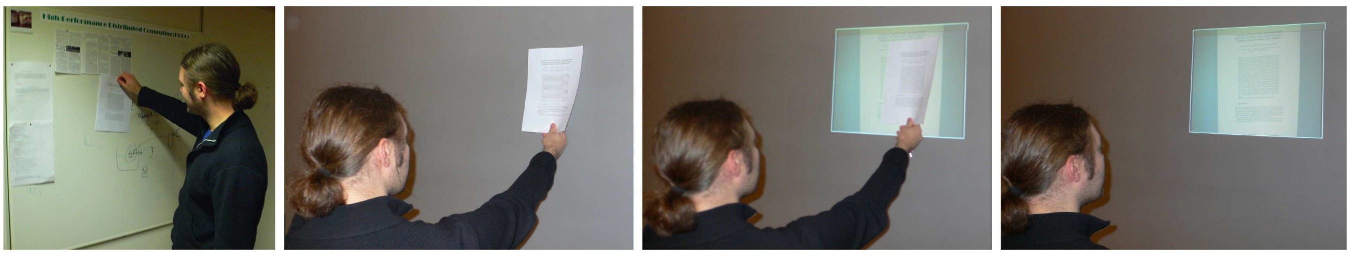
Blurring the line between real and digital: Pinning objects to wall-sized displays

Daniel Stødle*
Department of Computer Science
University of Tromsø, Norway

Otto J. Anshus†
Department of Computer Science
University of Tromsø, Norway



(a) (b) (c) (d)

Figure 1: (a) A user pinning a document to a combined white- and billboard. (b) A user pinning a document to the Wallboard. The document is held at the location where the user wants it to appear. (c) The content appears on the display wall. (d) The user removes the physical content, leaving the digitized version behind.

Abstract

Billboards are everywhere, enabling users to interact by leaving documents, images, ads or clippings for others to see. There is currently no simple and transparent way to replicate this interaction pattern in a wall-sized display context. Users must first employ devices like scanners or digital cameras to digitize the content they wish to share. Then the digitized content must be manually transferred to some computer, before the user can display and arrange it on the desktop. This paper presents a system that supports the classic billboard interaction pattern in a display wall context. The user briefly holds the content to digitize anywhere in front of the display wall, and an image of it appears at the same location. The system comprises a 6x3 m high-resolution wall-sized display, a gesture-based human-computer interface and a ceiling-mounted steerable camera, which together enable transparent and low latency object imaging.

1 Introduction

Wall-sized displays are becoming ever more common, with resolutions ranging from 10 to 100 megapixels and beyond [Li et al. 2000; Stolk and Wielinga 2006]. Display walls are typically built using a cluster of computers driving a set of tiled displays or projectors. Our display wall is built using 28 projectors and computers arranged in a 7x4 grid, forming a 22 megapixel, 7168x3072 display covering an area of 6x3 m.

There has been much work on moving whiteboard-style interaction to the realm of wall-sized displays, with commercial products such as the SMART Board [SMART Technologies] available. However, one fundamental issue that has yet to be addressed is making billboard-style interaction possible. On a billboard, users are less concerned about drawing or writing, and care more about leaving content of some kind behind for other users to see. This is typically done by simply fixing a document, news clipping, picture, advertisement or similar to the wall using pins, staples or magnets, as shown in Figure 1 (a).

Figure 2: The entire display wall being used in a billboard-like fashion.

The equivalent steps in current systems reduce to first digitizing the relevant content, either using a scanner or a digital camera. Then the content must be transferred in some way, with multimedia MMS messages, e-mail or BlueTooth file transfer among the many ways of doing this. Once transferred, the content must be brought up on the display wall, and manually placed at a location determined by the user. The entire process is time-consuming and requires a knowledgeable user.

*e-mail: daniels@cs.uit.no
†e-mail: otto@cs.uit.no

IPT/EDT 2008, Los Angeles, California, August 9–10, 2008.
© 2008 ACM 978-1-60558-211-5/08/0008 $5.00

This paper presents the design and implementation of Wallboard, a system that replicates the billboard-style interaction pattern. To achieve this, there are three important requirements that must be satisfied. (i) A user should not need to employ any devices, wear special gloves or be fitted with markers in order to "pin" content to the display wall, as the interaction should be as direct as it would be on a regular billboard. (ii) The content should appear on the display wall where the user is holding it, in order to match the behaviour of pinning content to a billboard. The user should be able to pin content anywhere on the display wall, and not be restricted to some designated region. (iii) As users expect to pin content to a billboard instantaneously, the time required to pin content to the display wall should also appear instantaneous to the user. Figure 1 (b)-(d) shows a user pinning a document to the Wallboard, and Figure 2 shows a large part of the display wall in use for imaged objects. Users are free to move and scale imaged objects once they appear on the display wall.

The main contribution of this paper is Wallboard, a scalable system for transparently imaging objects on wall-sized displays. The system is not limited to imaging objects, but also demonstrates how user content can be augmented with other kinds of data, including voice annotations and sensor measurements describing the content, captured in the moments preceding the action of pinning content to the display wall. The system demonstrates how the act of knowing *where* something is can sometimes be far more powerful than being able to identify exactly *what* that something is, while at the same time being a less complex and computationally expensive problem to solve.

2 Related work

There has been much work on creating digital whiteboards. In general, most of it has focused on ways of augmenting whiteboards with already-existing digital content, sharing content between different whiteboards (virtual or real), interaction styles or ways of supporting content creation. There has been little to no focus on making whiteboards act more like billboards, and in particular the act of pinning objects to the board.

The Xerox Liveboard system [Elrod et al. 1992] was one of the first digital whiteboards, upon which applications like Tivoli [Pedersen et al. 1993] were built. The Xerox Liveboard work identified aspects like image resolution as important to users, but also mentioned the need to "add [a] scanner." The Xerox Liveboard differs from our system in that it does not incorporate content from the environment into its applications.

In their work on Tangible Bits [Ishii and Ullmer 1997], the authors introduce the transBOARD. The transBOARD is a regular whiteboard augmented with sharing and storage capabilities through the use of a stroke recorder to store whiteboard contents. Physical content can be incorporated through the use of "phicons," barcode-tagged objects which represent real or virtual objects. This differs from Wallboard in that physical objects must be "attached" to such phicons before they can be used, and even then, do not actually appear on the transBOARD, but rather on a digital replica on a display nearby. Wallboard allows users to image any object, without manually having done so prior to pinning it to the display wall.

Mynatt et.al created Flatland [Mynatt et al. 1999], an augmented whiteboard intended for use in offices. The Immersive Whiteboard [Shae et al. 2001] is an attempt at bridging a physical whiteboard with a virtual counterpart. A video camera is used to create an avatar of the user, but can not be used to share other physical content, like documents or images.

There are many examples of surfaces that are active in the sense that they enable the inclusion of physical objects. The AMLCD panel [Abileah and Green 2007] enables the screen itself to scan documents, but is limited to capturing grayscale images of objects very close to the display. Microsoft's Surface [Bathiche and Wilson 2007] is a multi-touch enabled table that can sense devices like mobile phones and transfer images from the devices. Apart from not aiming to be a billboard, Microsoft Surface differs from Wallboard in that it is not possible to image arbitrary objects and have them appear on the surface. The EnhancedDesk [Koike et al. 2001] can recognize tagged documents placed on its surface, and augment the documents with interactive content. Recognition is done using a camera that looks for a matrix-code printed on the content to scan. No attempt is made at directly incorporating the imaged content; instead content must be tagged and recognized by the system. In [Klemmer et al. 2000], the authors demonstrate a desk that can digitize post-it notes. Their implementation only works with post-it notes, and requires the user to actually write the note on the digital desk. Our system can accomodate any content on a surface that is much larger than the desk demonstrated in [Klemmer et al. 2000].

Multi-touch and multi-point interaction has been an active field of research for several years, with commercial products like the Apple iPhone and Microsoft Surface available. There are many approaches to implementing multi-touch interfaces, including the use of electric capacitance in the Diamondtouch tabletop [Dietz and Leigh 2001], use of total internal reflection of infrared light [Han 2005], and the optical approach taken in [Morrison 2005]. The first two approaches require users to touch the canvas or screen. The device-free input system built for the Wallboard uses a touch-free approach, making for a cleaner surface and enabling the use of a flexible canvas. The approach taken by the SMART Board [Morrison 2005; SMART Technologies] is the one most similar to ours, but differs in its use of fewer and custom cameras with on-chip processing to perform object detection.

3 Design and implementation

Figure 3 illustrates the overall system architecture. The system comprises five major components: (i) a device-free input system, (ii) input analysis, (iii) camera and sensor control, (iv) the Wallboard application and (v) the Shout event system. Figure 4 illustrates how the various components are deployed.

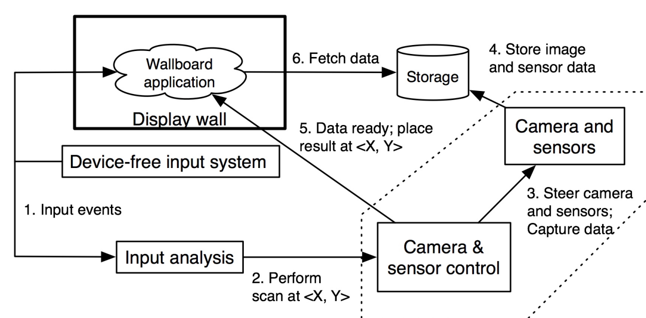


Figure 3: The overall system architecture and design.

3.1 Device-free input system

The device-free input system is used to enable multi-point, multi-user interaction with different applications running on the display wall, including Wallboard. The input system is built using 16 cameras mounted along the floor in front of the display wall's canvas. Images from each camera are analyzed in order to determine the location of objects intersecting two planes parallel to the display wall's canvas - the input system's "region of interest." Typical objects include hands, fingers or arms, although the interface does not distinguish between the different objects other than reporting different object radii. The intersection with each plane is found using triangulation, as shown in Figure 5.

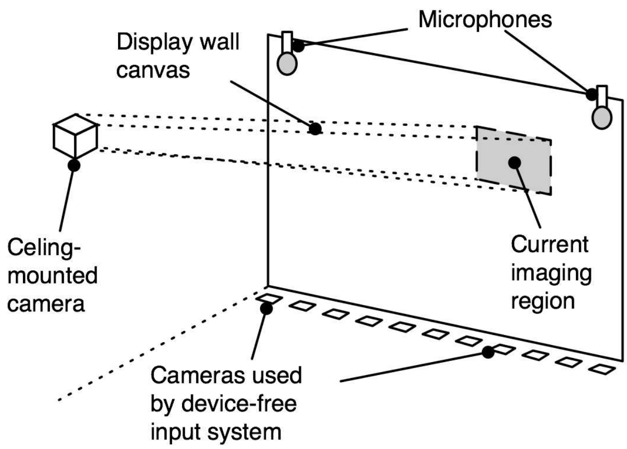


Figure 4: Schematic of the system deployment. Cameras are mounted along the floor to enable device-free interaction with the display wall. The camera used for imaging content is mounted in the ceiling at the back of the room. Microphones are deployed in front of the display wall canvas.

The 16 Unibrain Fire-i firewire cameras are connected in pairs to 8 Mac minis. Images from each camera are processed by applying two common techniques from computer vision: Background subtraction and thresholding. The result of processing a single image is a set of 1D positions (visible as the black dots inside the rectangles in Figure 5, and also shown in Figure 6), which when combined with data from the remaining cameras enable the triangulation of 2D object positions. The design and implementation of the device-free input system is based on the system presented by [Stødle et al. 2008].

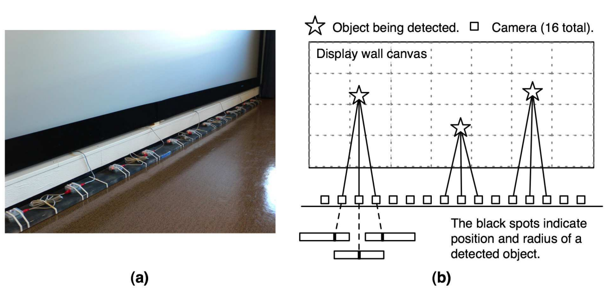


Figure 5: (a) The cameras for the device-free input system as they are mounted along the floor. (b) Triangulating object positions using image data gathered from each camera (triangulation shown for a single plane only).

The device-free input system is used for interacting with the Wallboard, as well as determining where to point the camera in order to image objects. This highlights an important principle employed throughout the design of Wallboard: Instead of determining *what* an object is, it is more fundamental to determine *where* it is. This principle is applied successfully in the design of the input system, and for Wallboard it makes it possible to easily determine where the content to image is located. In contrast, one could design a computer vision-based system where the area in front of the display wall is scanned by a camera to first identify the user, and then determine his location. Once the user's location has been found, the immediate surrounding area could be analyzed to find objects, before the camera can finally be accurately pointed at it and zoomed in. The latter approach would be more computationally demanding, time consuming and also very difficult to make reliable - if at all possible. It might also introduce assumptions about the content to image that are not ideal; for instance, assuming that content is always white and rectangular in shape.

3.2 Input analysis

The second component is the input analysis component. It is responsible for interpreting input events from the input system, and determining if a user is attempting to capture content for the Wallboard. For objects inside the device-free input system's region of interest, attributes like the object's movement and radius are used to determine whether to image the content or not. If the component determines that the content should be imaged, it sends an event to the camera and sensor control component, which will steer the camera and capture the targeted content.

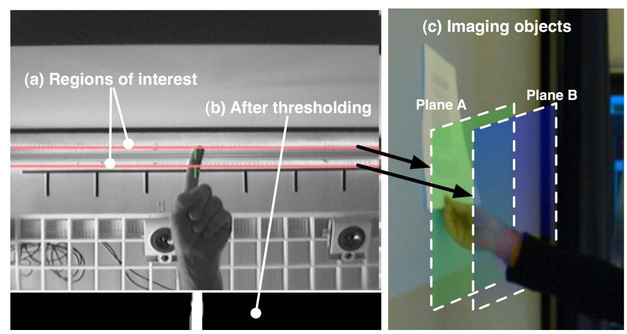


Figure 6: (a) The input image, showing the two planes in front of the display wall with a finger intersecting both. (b) The result after background subtraction and thresholding of one of the two planes. (c) The hand must intersect planes A and B, which are both parallel to the display wall surface, in order to target content for capture. In addition, the hand must remain stationary for one second and have a detected radius above an experimentally determined threshold.

The input analysis component is an important part, as it will essentially make or break the user's impression of the system. If it over-eagerly begins imaging content, spurious images will appear on the display wall. On the other hand, if it requires too much effort to invoke, users will end up frustrated with the system's behaviour. The input analysis component uses the following three factors, all supplied by the input system, to make a decision on whether or not to image an object: (i) The 2D position of an object (usually a finger, hand or arm) intersecting two planes in front of the wall (Figure 6). When a set of 2D positions is sufficiently close to each other, the remaining two factors are considered for that set. (ii) The width of the objects in the set. If the width is above an experimentally determined threshold, the object is tracked. (iii) If an object is tracked for more than 1 second and remains stationary, it will be interpreted as if a user wants to image the content held at the given location.

3.3 Camera and sensor control

The camera sensor and control component manages the camera and microphones in use by the system. When instructed to do so by the input analysis component, it will capture data from the camera. It will then notify the Wallboard application that it should fetch the newly captured data and position it on the display wall at the location where the user originally held the content to be captured. It also continuously records audio from the environment, with audio from the 15 preceding seconds being associated with the imaged content.

The camera used is a Canon VC-C4R with pan-tilt-zoom functionality and capable of generating images with a resolution of 720x540 in interlaced mode. It is mounted in the ceiling at the back of the room (see Figure 4), pointing towards the display wall. The camera is moved in response to a "scan" event from the input analysis component. To steer the camera, a mapping between the camera's pan and tilt coordinates to areas covered on the display wall is used. This mapping is created by determining the extreme values for pan and tilt at maximum zoom levels when aiming at the corners of the display wall. Linear interpolation is then used to map coordinates from the device-free input system to the camera's pan- and tilt-values, before the camera can be steered to the correct location.

One problem discovered in an earlier implementation of the system was that captured images were often affected by motion blur. Motion blur is introduced either by users being unable to hold the content to image stationary, lingering camera movement, or both. The problem is exacerbated by the fact that the camera in use is only capable of producing interlaced images. To handle this problem, the control component continuously captures images from the camera. Each new image captured is subtracted from the previous image, and used to calculate the average pixel intensity change, as well as the pixel intensity change's standard deviation. Whenever the standard deviation is below an experimentally determined threshold[1], that image will be eligible for being pinned to the display wall.

3.4 Wallboard

The fourth major component is the Wallboard application itself, whose main responsibility is to provide the graphical output on the display wall, as well as allow users to interact with imaged content. It accepts input events directly from the device-free input system, enabling multiple users to interact with it simultaneously using one or both hands. Since currently only one camera is in use, different users can not overlap imaging of content, but must interleave their use of the imaging feature. Wallboard also receives events from the camera sensor and control component, informing it when newly imaged content is available and where it should be placed on the display wall.

The Wallboard application is written in C using an in-development cluster-based backend to the Cairo [Worth and Packard 2003] rendering library. It responds to input events from the device-free input system, enabling users to not only image content, but also scale and move the resulting objects around on the display wall afterwards. When instructed to position an imaged object on screen, it will load the image representing it and place it at the coordinates given in the "fetch data" event. Using the event system it can also trigger playback of the audio associated with the imaged content.

3.5 The Shout event system

The fifth and final component is a network event system called Shout. Shout provides the other four components with the ability to send and receive events, and thus acts as an "event substrate" in between the components. It is designed to be both extendable and enable efficient event delivery. Shout is implemented in C, using a centralized event server to receive and distribute events from different clients. For efficiency and reduced bandwidth consumption, a binary format is used. The content and types of events is not pre-defined by the event system, but instead defined by the applications using the system. By default, a client receives all events, but event filters can be configured in order to limit the events received to specific types (such as the "fetch data" and "scan" events). To aid system efficiency, clients may also tell the server about which event types they intend to provide to the server. TCP is used for client-server communication.

4 Initial results

The latency for capturing content has been measured. The interval measured ranges from when the system determines that a user wants to capture content, until the content appears on the display wall – that is, not including the initial one-second delay used by the system to determine user intent. The methodology to measure this latency was as follows. An object was imaged by one of the authors 30 times at different locations on the display wall. Every time an event instructing Wallboard to image an object was received, a timer was started. That timer was stopped when a corresponding "fetch data" event was received, at which point the imaged content would appear on the display wall.

The results from this experiment yielded an average latency of 1.08 seconds, with a standard deviation of 0.26 seconds. The maximum observed latency was 1.73 seconds, and the minimum latency was 0.65 seconds.

5 Discussion

The latency for imaging objects is stems from the following factors. First, the camera requires some time to capture an image. Frames from the camera are captured using a frame grabber card, which provides new frames at a rate of 12.5 frames per second. At this rate, the time between an event prompting the control component to image content arrives, until the camera is ready with a new frame, can be up to 1.0 s / 12.5 frames/s = 0.08 s. With the actual latency about one order of magnitude higher, the camera frame rate is not the issue. The majority of the latency is due to two factors: (i) The camera is steered to target the content before an image is captured, and (ii) due to camera movement and stabilization, the technique used to avoid motion blur will prevent a number of the initial images from being recognized as valid images. The event system's latency has been measured at 1.9 milliseconds [Stødle et al. 2008], and thus contributes very little to the overall latency.

The use of a steerable camera to image content in the Wallboard system has some drawbacks. First, the pan- and tilt coordinates used to control where the camera is pointing, are far more coarse-grained than the coordinates provided by the device-free input system. This manifests itself as slight inaccuracies when imaging content, such as missing the top or one of the sides of a document. Second, it is quite common for parts of the fingers or hands to appear as part of the image representing the content. However, any other approach would require either (i) tagging all content in advance, which is impractical and violates the device-free aspects of the system design, or (ii) applying sophisticated computer vision techniques in an effort to recognize either the object or the fingers. The unwanted parts of the image could then be masked out. Finally, with the camera mounted in the ceiling, it is possible for the user to obscure the content to image when holding it at approximately waist-height or below.

The current implementation brings up a black box on the display wall, behind the content which is being imaged. This serves two purposes: First, it removes clutter from the resulting image by temporarily hiding other content at the location being captured. More importantly, however, it prevents light from the projectors leaking through the content being scanned. This is especially visible when imaging single sheets of paper. This effect could also be exploited for positive gain by the system, by allowing the color of the background to be changed. This could allow better capture of content like transparencies. It could also enable the system to respond to changes in the room's current light levels, enabling better camera

[1] The threshold used is affected mainly by camera noise and lighting factors.

exposure.

The Wallboard system is scalable along many different axes. It can be extended with additional cameras to enable several users to image objects simultaneously, and the device-free input system already easily accomodates more than one user - three persons may play Quake 3 Arena against each other at the same time on the display wall [Stødle et al. 2007]. The resolution of the images captured can be increased by using more expensive cameras without changing other parts of the system, and additional sensors may be added beyond the camera and microphones currently in use.

6 Conclusion

This paper has introduced Wallboard, a system that enables content to be moved from the real world to a display wall in a way that mimics the interaction pattern used to share information on a billboard. Without requiring the use of any devices, users briefly hold documents, images or other content in front of the display wall at the location where they want it to appear. A device-free input system determines where the object to capture is located, before a camera and associated sensors capture the object.

The system requires in total about two seconds to capture an image of the content a user wishes to place on the Wallboard. The first second is used to determine user intention ("does the user really want to capture this content and place it on the display wall?"), while the rest is caused by system latency incurred by camera movement and avoiding motion blur. An important design principle has been identified that greatly simplifies the implementation of both the device-free input system, and the process of determining where the content to capture is located: Instead of determining *what* an object is, it is more fundamental to determine *where* it is. Applying this principle enables Wallboard to avoid some very hard problems in computer vision (object recognition and pose estimation), while still resulting in a system that achieves the design requirements set out in the introduction: It mimics a billboard, it does not require the user to wear or use any devices, and the time required to pin content to the display wall is on the order of a few seconds.

Acknowledgements

The authors thank Espen S. Johnsen for creating the cluster-based backend to Cairo, John Markus Bjørndalen, Tor-Magne Stien Hagen and the technical staff at the CS department at the University of Tromsø. This work is supported by the Norwegian Research Council, projects No. 159936/V30 and No. 155550/420.

References

ABILEAH, A., AND GREEN, P. 2007. Optical sensors embedded within amlcd panel: design and applications. In *EDT '07: Proceedings of the 2007 workshop on Emerging displays technologies*, ACM, New York, NY, USA, 7.

BATHICHE, S., AND WILSON, A., 2007. Microsoft surface. http://www.microsoft.com/surface/.

DIETZ, P., AND LEIGH, D. 2001. DiamondTouch: a multi-user touch technology. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, ACM Press, New York, NY, USA, 219–226.

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. 1992. Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, 599–607.

HAN, J. Y. 2005. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, ACM Press, New York, NY, USA, 115–118.

ISHII, H., AND ULLMER, B. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, 234–241.

KLEMMER, S., NEWMAN, M. W., AND SAPIEN, R. 2000. The designer's outpost: a task-centered tangible interface for web site information design. In *CHI '00: CHI '00 extended abstracts on Human factors in computing systems*, ACM, New York, NY, USA, 333–334.

KOIKE, H., SATO, Y., AND KOBAYASHI, Y. 2001. Integrating paper and digital information on enhanceddesk: a method for realtime finger tracking on an augmented desk system. *ACM Trans. Comput.-Hum. Interact. 8*, 4, 307–322.

LI, K., CHEN, H., CHEN, Y., CLARK, D. W., COOK, P., DAMIANAKIS, S., ESSL, G., FINKELSTEIN, A., FUNKHOUSER, T., HOUSEL, T., KLEIN, A., LIU, Z., PRAUN, E., SAMANTA, R., SHEDD, B., SINGH, J. P., TZANETAKIS, G., AND ZHENG, J. 2000. Building and Using A Scalable Display Wall System. *IEEE Comput. Graph. Appl. 20*, 4, 29–37.

MORRISON, G. D. 2005. A camera-based input device for large interactive displays. *IEEE Computer Graphics and Applications 25*, 4, 52–57.

MYNATT, E. D., IGARASHI, T., EDWARDS, W. K., AND LAMARCA, A. 1999. Flatland: new dimensions in office whiteboards. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, New York, NY, USA, 346–353.

PEDERSEN, E. R., MCCALL, K., MORAN, T. P., AND HALASZ, F. G. 1993. Tivoli: an electronic whiteboard for informal workgroup meetings. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, ACM, New York, NY, USA, 391–398.

SHAE, Z.-Y., TSENG, B., AND LEUNG, W. H. 2001. Immersive whiteboard collaborative system. *Ann. Softw. Eng. 12*, 1, 193–212.

SMART TECHNOLOGIES. SMART board interactive whiteboards. http://www.smarttech.com/.

STØDLE, D., HAGEN, T.-M. S., BJØRNDALEN, J. M., AND ANSHUS, O. J. 2007. Gesture-based, touch-free multi-user gaming on wall-sized, high-resolution tiled displays. In *Proceedings of the 4th Intl. Symposium on Pervasive Gaming Applications, PerGames 2007*, 75–83.

STØDLE, D., HA, P. H., BJØRNDALEN, J. M., AND ANSHUS, O. J. 2008. Lessons learned using a camera cluster to detect and locate objects. In *Parallel Computing: Architectures, Algorithms and Applications. Proceedings of the International Conference ParCo 2007.*, IOS Press, vol. 15 of *Advances in Parallel Computing*, 71–78.

STOLK, B., AND WIELINGA, P. 2006. Building a 100 Mpixel graphics device for the OptIPuter. *Future Gener. Comput. Syst. 22*, 8, 972–975.

WORTH, C. D., AND PACKARD, K. 2003. Cairo: Cross-device rendering for vector graphics. In *Proceedings of the 2003 Linux Symposium*. http://cairographics.org/.