Data Handling Displays

Maxim Lazarov, Hamed Pirsiavash, Behzad Sajadi, Uddipan Mukherjee, Aditi Majumder
University of California, Irvine
Irvine, CA 92697
{mlazarov|hpirsiav|bsajadi|umukherj|majumder}@ics.uci.edu

Abstract

Imagine a world in which people can use their hand-held mobile devices to project and transfer content. Everyone can join the collaboration by simply bringing their mobile devices close to each other. People can grab data from each others' devices with simple hand gestures. Now imagine a large display created by tiling multiple displays where multiple users can interact with a large dynamically changing data set in a collocated, collaborative setting and the displays will take care of the data transfer and handling functions in a way that is transparent to the users.

In this paper we present a novel data-handling display which works as not only a display device but also as an interaction and data transfer module. We propose simple gesture based solutions to transfer information between these data-handling modules. We achieve high scalability by presenting a fully distributed architecture in which each device is responsible for its own data and also communicates and collaborates with other devices. We also show the usefulness of our work in visualizing large datasets and at the same time allowing multiple users to interact with the data.

1. Introduction

Recent advances in projection and camera technologies have made these devices increasingly portable. It is not difficult to envision displays made from portable projector-camera pairs in the near future. Many mobile phones and computers already come equipped with integrated cameras for video chat and handheld "pico-projectors" are already available to consumers. Given these advances in the miniaturization of projectors and cameras, it is entirely possible for projector-camera displays to also soon be portable.

We consider displays made from projector-camera pairs and show that such display units, coupled by a fully distributed architecture for handling data allow us to exchange data between displays. The advantage of such display modules is that they can be used in both portable and stationary displays alike, granting different capabilities in different situations. For example imagine two users with mobile devices that have integrated portable displays. The spatial proximity of the displays—indicated by an overlap in their areas of projection—allows for loose coupling between the devices. Then using gesture-based interaction with the displays—made possible by the augmenting camera—the users can transfer data across the devices. On the other hand, when several such displays are tiled to create a large display, their contiguous spatial arrangement allows a loose coupling between all the displays. The gesture-based interaction in this case can enable multi-user interaction with the display wall. The distributed architecture in this case allows efficient data handling across the different displays, critical when displaying large, dynamically-changing data.

Main Contributions: Our main contribution in this paper is to show the feasibility of a data-handling display realized by a distributed network of projector-camera systems. Augmented by a distributed gesture recognition mechanism, this enables efficient data transfers between the displays, enabling different capabilities in different applications.

2. Previous work

Our work shares a common vision with that of [18], though we expand on the rudimentary concepts presented therein into a scalable system. There have been many earlier publications in the direction of single and multi-user projection-based user interfaces. For single user, single projector systems, using a flashlight as an input device has been explored in [6, 13, 14] in order to interact with a large virtual workspace. Whenever a portion of the screen is lit by the flashlight the information on that part will become visible to the user. Cao and Balakrishnan [6] investigated the possibility of using the flashlight metaphor in order to achieve generic interaction techniques used for defining information spaces in a new environment on-the-fly. Beardsley et al. account for the distortions in the projected image for a single projector to create an undistorted stationary virtual desktop [1]. This is extended to multiple projectors by Raskar et al. [14, 15].

In addition to single user systems much research has been done in the area of single projector, multi-user

978-1-4244-3993-5/09/$25.00 ©2009 IEEE

systems which are usually used as collaborative groupware for collocated groups of users. These papers can be categorized into two major groups: the approaches that use display walls as described in [4, 8, 9, 19] and the ones that are using tabletop displays as described in [17, 21, 22]. These papers usually focus on handling multi-user collaboration paradigms in collocated groupware as discussed in [11] and also how to retain information privacy of the users utilizing shutter glasses, which is investigated in [18].

Our work falls in the category of multi-projector, multi-user interaction as in [5]. This work uses multiple handheld projectors to explore a shared physical space. When dealing with general displays instead of projectors, there have been more papers pertaining to connecting multiple display spaces. Pick-and-Drop [16] investigates a pen-based technique used to transfer data between devices by picking objects from one device and dropping them into another one. Toss-It, presented by Yatani et al. [23], introduces a new technique to transfer information between a PDA and other devices.

However, all these systems depend on a centralized camera-based-tracking system where a stereo pair of cameras sees all the different projectors which emit specialized LED based patterns to decipher the coupling between the input devices. Further, all these works use the handheld projector/pen/PDA as the input device where a user is coupled to a device and can hence only interact with the shared space through it.

In contrast, in our system, only human gestures are used as inputs, spatial proximity is used to couple the displays together as opposed to tracking the displays, and having a distributed network of projector-camera pairs opens up the possibility of distributed gesture recognition and tracking to enable interaction. Finally, there is little coupling between the users and the displays which automatically lends itself to a multi-user scenario, where a user may not have the ownership of a device.

3. System overview

The backbone of our system is built on our earlier work on creating scalable displays via a distributed network of "plug-and-play-projectors" (PPPs) [3]. A PPP is a self-sufficient display unit made of a projector, camera and computer (Figure 1). A scalable display is created by placing the projection areas of the PPPs in a spatially contiguous (overlapping) manner. The PPPs are initially unaware of the configuration that they are arranged in. Using visual communication via the cameras in the inter-projector overlap regions, a PPP starts detecting its neighbors whenever its associated camera perceives some other PPP in its coverage area. Using a distributed approach each PPP can discover its position in the display and communicate and propagate information throughout the entire display until convergence (geometric and photometric registration) is achieved. Following this the PPP can also set up a network-based communication channel across all the PPPs belonging to one display or a subset thereof via multicast or broadcast capabilities. Figure 2 shows an illustration of this distributed system architecture.

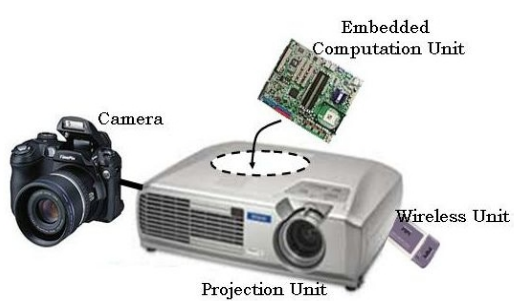


Figure 1: A Plug-and-Play Projector (PPP) consisting of a camera, a projector, and a processing unit.

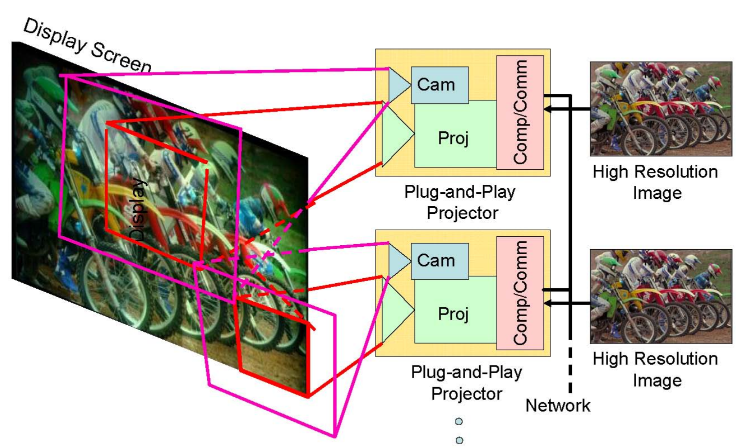


Figure 2: An illustration of a distributed system architecture for multi-projector displays.

This distributed architecture permits the easy detection of addition and removal of PPPs to a display. The neighboring PPPs can detect this via their cameras and broadcast a reconfiguration message following which the PPPs reconfigure and register themselves.

We propose using the same PPPs to create data handling displays that can now function as both input and output devices. We consider hand gestures as the primary source of input. Gestures can be detected using the same camera that is used for configuration identification and registration. We propose using the underlying distributed architecture proposed in [3] for interacting with the PPPs as well as using a distributed way to *recognize*, *communicate*, and *react to* gestures.

Distributed gesture recognition poses challenges when the gesture does not occur within a single display. However, note that since the recognition is a local phenomenon which only involves a single display or a few adjacent ones, having multiple users does not pose a problem.

In back projection systems, the screen blocks the view of the gestures in cameras mounted on the rear projectors themselves. This poses a challenge which is commonly resolved by the using infrared illuminators and cameras [7, 10], as will be described in Section 3.3.

Distributed gesture communication involves informing all the PPPs in the display or a subset thereof of the recognized gesture. The challenge lies in achieving this in a fashion that optimizes different parameters like speed, network congestion and reliability.

Finally, *distributed reaction* involves each PPP to reach an *independent* decision about how to react to the communicated message in such a manner that would result in a *globally acceptable* data transfer and display.

We have developed the first basic version of the above mentioned features resulting in two applications: (a) file handling in portable devices via front projection PPPs; (b) data handling in visualizing large, dynamically-changing datasets in a tiled display via rear projection PPPs.

3.1. File handling with front projection

Imagine two users holding their mobile devices which are equipped with small, hand-held PPPs. The users bring together the projection areas of their devices, causing them to overlap. Using the cameras, the PPPs can recognize simple hand gestures. Users can then use simple hand gestures to share data across the devices. On a lower level, the devices can use Wi-Fi or Bluetooth for communication and data transfer. Figure 3 demonstrates the simple process of moving a document from one mobile device to another in our system.

In our system, we use a PC to simulate the mobile device; the fundamental issues are the same. Figure 4 is a picture of our front projection setup.

The user can use a simple pointing gesture to select a file. He can first select a document in one folder and select a different folder on a different device to initiate a file transfer from one device to another. In addition to file transfer, users can use additional gestures to delete files or to capture new data. For example a user can show a capture gesture and then hold a physical document in front of the camera. The camera captures a picture of the document and a new object will be created that contains the captured image.

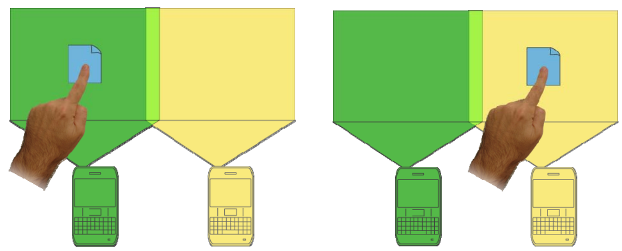


Figure 3: The process of moving a document from one mobile device to another one.

It is important to note that in our system we do not associate users with the particular displays. One user might interact with all of the displays or more than one user can interact with the same display.

3.2. Large visualization with back projection

A tiled display made of multiple rear projectors is commonly used for visualizing large datasets. Note that the same distributed gesture based data handling technique can also be used to interact with this type of data. The limitation we encounter is that the camera cannot detect visible light gestures reliably because the screen and projected image obstructs the hand.

To handle this situation, as in [7, 10], we augment our PPPs with an infrared (IR) camera and an IR illuminator. Note that while the regular cameras are used for registration and detection of addition and removal of PPPs, the IR cameras are used following registration when the display is being used for visualization purposes alone. The IR illuminator and the camera allow us to detect gestures when we touch the screen. Figure 5 shows our setup.

Users can now interact with the data. Imagine some map-like data which is being panned and zoomed by the users. In this work, we consider global interactions, i.e. an interaction that involves changing the display content on all PPPs similarly (e.g. translate the map 100 pixels to left or zoom the map to 1.5 times scale). This is most relevant for single user, but does not restrict multiple users to work together unless they perform different gestures at the same instant in time. A more sophisticated paradigm to handle multiple users, which is a part of our future work, may involve defining regions of local interactions and allowing different kinds of local interactions from multiple users in different parts of the data.

4. The First Prototype

In this paper, we present the very first version of the distributed gesture-based interaction and data handling we have developed using the PPP based distributed architecture [3].

4.1. Gesture detection

The PPP captures the scene using the camera and performs background cancelation on the captured images. Following this we get the silhouette of the hand performing the gesture.

In the rear-projection system, the use of IR illuminators and cameras does not require any dynamic background cancelation for gesture detection because our projected content appears only in the visible light spectrum, not seen by our IR cameras. It also makes gesture detection more robust because if we project dark content on the screen there might not be enough visible light for gesture detection. The cameras, being behind the screen along with the projectors prevent occlusion problems which are common in front projection systems. However our rear projection system does require the user to touch the screen for gesture detection. When the user touches the screen part of the light that reaches the front of the screen is reflected back to the IR camera we use for gesture detection.

Each PPP is responsible for detecting the gestures that occur within the view of its camera. When the gesture is happening at the overlap of two projectors they can detect that this is the same gesture using network communication in order to avoid duplicate execution of the gesture detection across two adjacent PPPs.

We assume simple 2D gestures and use an approach similar to the method described in [12] to classify the gestures. We first find the bounding box of silhouette of the hand to find its orientation. The bounding box will then be properly rotated in order to make the features rotation invariant. Following the rotation the gestures will be classified with metrics presented in [12]. We use a limited number of gestures in our systems and therefore simple classifications can be utilized effectively.

4.2. Gesture Communication

After detecting a gesture, the resulting action might affect the whole display. Therefore following each action with a global scope, the corresponding PPP will broadcast the action to its neighbors and the neighbors will keep doing the same thing. The system will reach a consensus after a few iterations. With this message passing mechanism that happens only between the neighbors we avoid network congestion that is usually a result of message broadcasting.

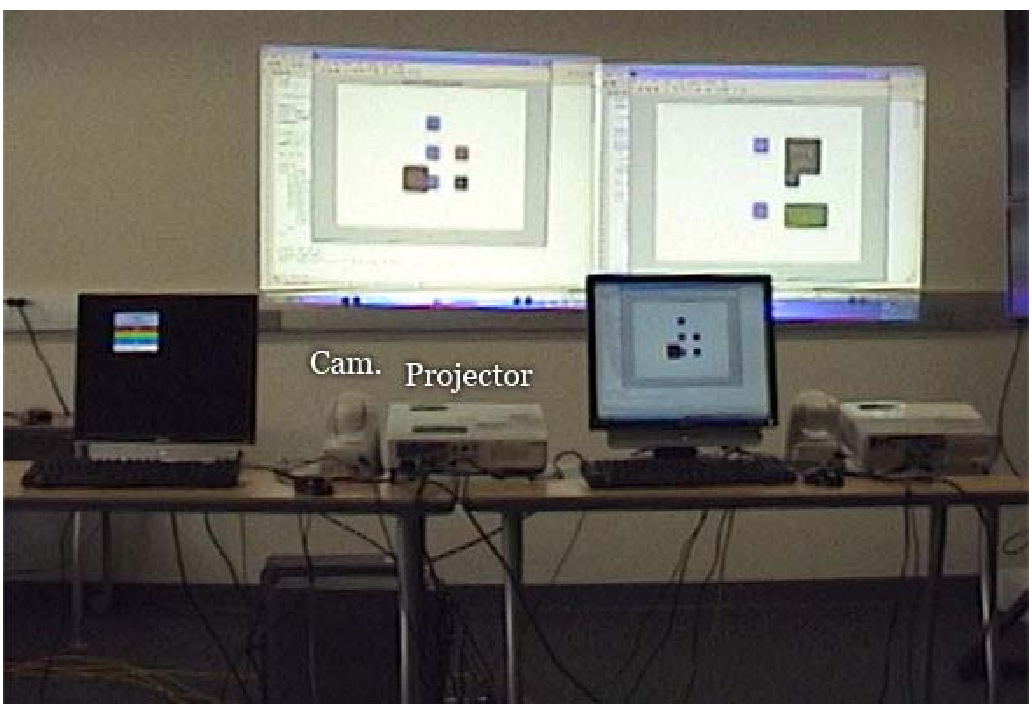


Figure 4: Our setup for the file handling with front projection system including two PPPs.

Figure 5: Our setup for the large dataset visualization with back projection system including four PPPs.

4.3. Data Handling Reaction

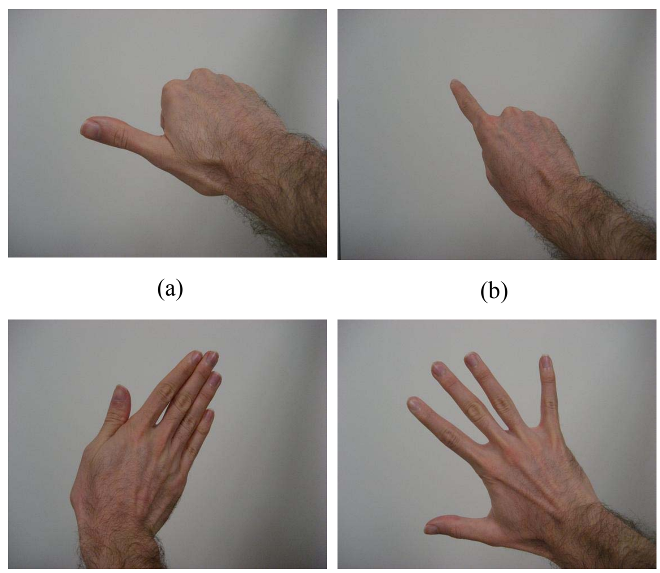
Once each PPP detects the gestures, the PPP realizes how the data should be moved on the unified display made by all of the PPPs to create the acceptable result for the user. Note that following the configuration identification step presented in [3], each PPP is aware of the part of the data it is responsible for and already has fetched the data. Similarly, when a PPP receives a transmitted gesture, based on the pre-defined interaction it is supposed to perform, it can compute the extra part of the data it should fetch from the server and in the position it should be displayed. Following this, the appropriate data is fetched by each PPP from the data server.

5. Implementation and results

As introduced in the previous section we have two different systems both utilizing hand gestures for user interaction but utilizing different projection mechanisms. In this section we will go through the implementation details of each of the systems and the way the users can interact with them.

5.1. File handling with front projection system

This system consists of two PPP units. Each of the units has a separate desktop. When units come spatially close to each other and start to overlap, users can start using file transfer abilities. Also each unit by itself gives the user the ability to capture documents and move or delete them on his own desktop.



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

Figure 6: Gestures used for the file handling with front projection system. Corresponding actions are: (a) cancel operation, (b) copy a document or paste it if something is already in the clipboard and the finger does not point to another document, (c) capture a new document, and (d) remove a document.

We use four basic gestures in this system for capture, select, paste, and delete. The corresponding gestures for each action can be found in Figure 6. Our gesture recognition is implemented in a non-optimized MATLAB code. Hence, for the gesture recognition to take place, the user should keep his hand steady for half of a second. This way the system can make sure that the user actually intends to do some action which prevents from many misinterpretations.

In this system cameras are located near the projectors in front of the display. The camera included in the PPP captures an image of the scene. Using the information gathered during the geometric and photometric registration phase [3], each PPP can geometrically warp and photometrically transform the image it is projecting appropriately to predict the scene without the gesture from the camera's point of view. This predicted image is then subtracted from the captured images in order to do a background cancelation and detect the silhouette of the hand.

After detecting the gestures, TCP is used for message passing in this system. The port number will be propagated together with the IP address after the PPPs visually recognized each other. Following that a TCP connection will be established between the neighboring PPPs and they can start to communicate. We chose TCP for this system since we need to transfer the actual user data (the files) along with the gestures which can be done reliably via TCP.

5.2. Rear projection system

Our rear projection system consists of four projectors in a two row, two column grid, each of them equipped with a camera located on top of the projector. The gestures in this system include translation and zooming which allow users to effectively browse different parts of the map. The gestures for this application are similar to gestures (c), and (d) in Figure 6 but they will be tracked.

In our system, instead of using a separate IR camera, following the distributed registration of the PPPs, we mount a simple visible light filter on the same camera to realize an IR camera. This transition takes place manually in our system but can be made automatic simply by using a mechanical actuator to switch between visible light and infrared filters. If the projector configuration changes (due to addition, removal or fault) the actuator can allow the cameras to be switched back to the normal mode (enabling the IR filter) for geometric and photometric calibration. We can also simply use an additional IR camera to capture gestures while an independent visible light camera monitors registration.

In order to perform static background cancelation required for the gesture detection the system captures the background before presence of any gesture to be used later. Since IR images are noisy, the system performs some preprocessing and then, by comparing the image with the background model, finds bounding box of the silhouette of the hand gesture.

In this application we need to track the gestures to interpret the intent of the user. This is achieved by computing and tracking the centroid, size and orientation of the gesture until the gesture is complete. The amount of movement or scaling is proportional to the difference of these values between the first and current frame.

For message passing we use a UDP based messaging to keep all the PPPs updated with the latest gestures detected by rest of the PPPs. First each PPP propagates its location, which is discovered during the geometric registration, and also its IP address. Afterwards a UDP connection will be established between every pair of neighbors. Since in this application we only need to transfer the gesture parameters in a timely manner and do not send any data along with it, we chose UDP, which affords us lower latency.

Each PPP keeps on sending its current detected gesture or a no-op when nothing is detected. Sending a no-op helps the neighbors to make sure that the PPP is still responding. All the PPPs use a polling mechanism to get the latest detected gesture and respond appropriatly. If a PPP stops responding due to a failure, it will be detected and handled via reconfiguration of the display in the same way as in [3].

5.3. Details of the utilized devices

We used two Epson PowerLite 1825 LCD projectors for our front projection setup and four similar projectors for our back projection setup. We used two Access IP cameras connected to two similar PCs for the front projection system and four Point Grey Firefly2 cameras connected to four heterogeneous PCs for the back projection system.

The photometric and geometric calibration process takes less than a minute and afterwards users can work with the system interactively.

6. Conclusion and future work

In this paper we presented data handling displays by extending the notion of the "plug-and-play projector" (PPP) and implementing a distributed gesture detection, communication and reaction system on top of the distributed network of PPPs proposed in [3]. We also used visual overlaps to group the PPPs instead of tracking them which requires a central tracking mechanism. Using visual overlap helped us to keep with the fully distributed paradigm of the system. We demonstrated usefulness of our approach by showing a front projection system used for information sharing between multiple desktops and a back projection tiled display for large dataset visualization.

However, this is the very first prototype and enormous possibilities for future work exist. We would like to be able to handle gestures which are not limited to scope of one projector and its overlap area e.g. a user can draw a large circle without considering how many projectors are included in his drawing. Then the area inside the circle can be considered as his personal workspace. We plan to study possible ways of defining the notion of personal working area in a large area display. This way a user can have his personal workspace when required and at the same time share his work with the rest of the users whenever he desires.

Finally we would like to seek other applications that can make use of such systems, such as office environments, and study a wider variety of gestures that suit these applications.

References

[1] Beardsley, P., Baar, J.V., Raskar, R., and Forlines, C. (2005). Interaction using a handheld projector. *IEEE Computer Graphics and Applications*, 25(1). p. 39-43.

[2] Bhasker, E., and Majumder, A. Self calibrating tiled displays. SID Symposium Digest of Technical Papers, 2007.

[3] Bhasker, E., Sinha, P., and Majumder, A. Asynchronous distributed calibration for scalable reconfigurable multi-projector displays. IEEE Transactions on Visualization and Computer Graphics (Visualization), 2006.

[4] Brignull, H., Izadi, S., Fitzpatrick, G., Rogers, Y., and Rodden, T. (2004). The introduction of a shared interactive surface into a communal space. *ACM CSCW Conference on Computer Supported Cooperative Work*. p. 49-58.

[5] Cao, X., Forlines, C., and Balakrishnan, R., Multi-user interaction using handheld projectors. Proceedings of the 20th annual ACM symposium on User interface software and technology, pages 43–52, 2007.

[6] Cao, X. and Balakrishnan, R. (2006). Interacting with dynamically defined information spaces using a handheld projector and a pen. *ACM UIST Symposium on User Interface Software and Technology*. p.225-234.

[7] Han, J. Y., Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. In Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology, ACM Press, New York, NY, 2005, 15-118

[8] Hilliges, O. and Terrenghi, L. (2006). Overcoming mode-changes on multi-user large displays with bimanual interaction. *MU3I Workshop on Multi-User and Ubiquitous User Interfaces*.

[9] Izadi, S., Brignull, H., Rodden, T., Rogers, Y., and Underwood, M. (2003). Dynamo: a public interactive surface supporting the cooperative sharing and exchange of media. *ACM UIST Symposium on User Interface Software and Technology*. p. 159-168.

[10] Kim, J., Park, J., Kim, H., and Lee, C., HCI(Human Computer Interaction) Using Multi-touch Tabletop Display, IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 2007. PacRim 2007.

[11] Morris, M.R., Ryall, K., Shen, C., Forlines, C., and Vernier, F. (2004). Beyond "social protocols": multiuser multiuser coordination policies for co-located groupware. *ACM CSCW Conference on Computer Supported Cooperative Work*. p. 262-265.

[12] Pirsiavash, H., Singh, V., Majumder, A., and Jain, A., "Shared Visualization Spaces for Environment to Environment Communication," Workshop on Media Arts, Science, and Technology (MAST), Santa Barbara, CA, Jan 2009.

[13] Rapp, S., Michelitsch, G., Osen, M., Williams, J., Barbisch, M., Bohan, R., Valsan, Z., and Emele, M. (2004). Spotlight Navigation: Interaction with a handheld projection device. *International Conference on Pervasive Computing, Video paper*.

[14] Raskar, R., Baar, J.v., Beardsley, P., Willwacher, T., Rao, S., and Forlines, C. (2003). iLamps: geometrically aware and self-configuring projectors. *ACM Transactions on Graphics*, 22(3). p. 809-818.

[15] Raskar, R., Beardsley, P., Baar, J.v., Wang, Y., Dietz, P., Lee, J., Leigh, D., and Willwacher, T. (2004). RFIG lamps: interacting with a self-describing world via photosensing wireless tags and projectors. *ACM Transactions on Graphics*, 23(3). p. 406-415.

[16] Rekimoto, J. (1997). Pick and drop: A direct manipulation technique for multiple computer environments. *ACM UIST Symposium on User Interface Software and Technology*. p. 31-39.

[17] Shen, C., Vernier, F., Forlines, C., and Ringel, M. (2004). DiamondSpin: An extensible toolkit for around the table interaction. *ACM CHI Conference on Human Factors in Computing Systems*. p. 167-174.

[18] Shoemaker, G. and Inkpen, K. (2001). Single display privacyware: Augmentng public displays with private information. *ACM CHI Conference on Human Factors in Computing Systems*. p. 522-529.

[19] Simon, A. (2005). First-person experience and usability of co-located interaction in a projection-based virtual environment. *ACM Symposium on Virtual Reality Software and Technology*. p. 23-30.

[20] Sugimoto, M., Miyahara, K., Inoue, H., and Tsunesada, Y. (2005). Hotaru: Intuitive Manipulation Techniques for Projected Displays of Mobile Devices. *INTERACT 2005*. p. 57-68.

[21] Wigdor, D., Leigh, D., Forlines, C., Shipman, S., Barnwell, J., Balakrishnan, R., and Shen, C. (2006). Under the table interaction. *ACM UIST Symposium on User Interface Software and Technology*. p. 259-268.

[22] Wu, M. and Balakrishnan, R. (2003). Multi-finger and whole hand gestural interaction techniques for multiuser tabletop displays. *ACM UIST Symposium on User Interface Software and Technology*. p. 193-202.

[23] Yatani, K., Tamura, K., Hiroki, K., Sugimoto, M., and Hasizume, H. (2005). Toss-it: intuitive information transfer techniques for mobile devices. *Extended Abstracts of the ACM CHI Conference on Human Factors in Computing Systems*. p. 1881-1884.