Geometrically Correct Imagery for Teleconferencing

Ruigang Yang,[*] Michael S. Brown[‡], W. Brent Seales[†] Henry Fuchs

Department of Computer Science
University of North Carolina at Chapel Hill

Abstract

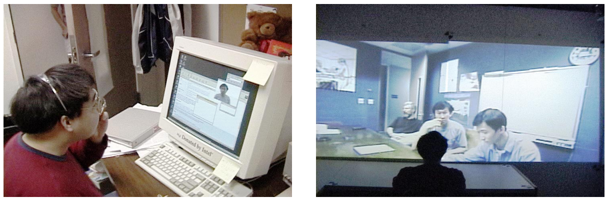
Current camera-monitor teleconferencing applications produce unrealistic imagery and break any sense of presence for the participants. Other capture/display technologies can be used to provide more compelling teleconferencing. However, complex geometries in capture/display systems make producing geometrically correct imagery difficult. It is usually impractical to detect, model and compensate for all effects introduced by the capture/display system. Most applications simply ignore these issues and rely on the user acceptance of the camera-monitor paradigm.

This paper presents a new and simple technique for producing geometrically correct imagery for teleconferencing environments. The necessary image transformations are derived by finding a mapping between a capture and display device for a fixed viewer location. The capture/display relationship is computed directly in device coordinates and completely avoids the need for any intermediate, complex representations of screen geometry, capture and display distortions, and viewer location. We describe our approach and demonstrate it via several prototype implementations that operate in real-time and provide a substantially more compelling sense of presence than the standard teleconferencing paradigm.

Keywords
Teleconference, telepresence, video conference, immersive display, graphics, real-time.

1 Introduction

Video-conferencing is a reality and is already widely available over dedicated lines at a number of institutions and businesses. Emerging technologies within the computer networking community are promising to make the Internet a viable medium for high-fidelity real-time video. While there has been wide interest in addressing the underlying transport technologies for real-time video, the issue of how best to capture and then present a video-stream has been neglected.



(a) Typical teleconferencing

(b) Compelling teleconferencing

Figure 1: (a) Typical *through-a-window* teleconferencing user-interface. (b) Compelling wide-field-of-view display created using the techniques presented in this paper. The *same* capture device was used for both (a) and (b).

Currently, the majority of video and teleconferencing application limits the capture of a video stream to a single sensor and the display to a CRT or flat-panel device. This camera-monitor interface does not provide a compelling or convincing *presence* to the participants [5]. The display scale and constrained field of view of these interfaces, allow for very limited interaction, providing the users with only a small "through-a-window" view of each other (Fig 1(a)). In this paper, we present a new and simple technique that enables more realistic and compelling teleconferencing applications. We relax constraints on both the capture device and the display environment, and show how to capture and then display the image data in a way that guarantees that the viewer will see perspectively correct imagery. We construct a scalable display environment by tiling a display surface with many light projectors (Fig 1 (b)). Image capture can be performed with any image sensor that maintains a common center of projection (COP). This can be a single camera with a wide-angle lens, or a panoramic image device composed of many individual cameras. Using an explicit capture-to-display surface mapping, we capture and render perspectively correct imagery in real-time for a stationary user. Our technique compensates for camera and display distortion, and allows the display geometry (surface and projector orientations) to be arbitrary. This means that any existing display surface, such the corner of an office or a set of white panels leaned against the wall, can be used as the display area. Our technique is easily scalable to very wide field of view (WFOV) environments. We demonstrate this scalability in section 6 by showing a WFOV teleconferencing environment we created using multiple light projectors and a common COP multi-camera rig.

[*] ryang@cs.unc.edu [‡] mbrown@dcs.uky.edu

[†] Visiting Research Associate Professor from the University of Kentucky

2 Related Work

There are numerous video conferencing products available. Hewitt [4] provides an excellent review of current video conferencing products and standards. Most video conferencing applications concentrate on networking issues and assume that the capture/display paradigm is that of a single camera and a CRT. Their emphasis is on image coding, network transport and collaborative user interfaces. These applications provide for the users a "video in a window"[2, 11] interface. Though they have improved significantly over the years, this *through-a-window* paradigm inhibits a realistic sense of presence between the participants.

There have been several approaches to make teleconferencing applications seem more natural and compelling. One interesting approach is the CU-SeeMe VR [3] system, which combines teleconferencing with virtual reality. In this system, live video streams from conference participants are textured onto planar surfaces. These planar surfaces, representing the participants, are allowed to roam around a shared virtual conference room. This is a first-order approach to using avatars for teleconferencing; participants see a 3D view of the 2D avatars within the virtual space. Although the system maintains the collaborative virtual space, the final output is still a desktop application, and the participants see the virtual world and the representation of others in it through a small window on a CRT.

Yamaashi addressed the limitation of the camera/monitor model by providing a user with two separate views: one wide-angle, and the other a controllable, detailed view [5]. The two views are separated into different viewports, and a simple user interface allows the user to pan the detailed view towards regions of interest. Although it enhances the user's awareness of the remote site, the geometric discontinuities of the viewports break any natural sense of presence. Yamaashi also points out the need for correcting the distortion for the wide-angle view.

Raskar et al. [7] proposed a teleconferencing and telecollaboration interface that moves away from the desktop metaphor and toward an immersive context. They proposed to extract reflectance and depth information dynamically for all visible "pixels" in a room, including walls, furniture, objects, and people. With an exhaustive representation of the environment, the system could exchange models over the network with a remote site with similar setup. These models would allow very realistic images of people and objects to be rendered and displayed within the correct geometric setting of the environment. A practical implementation of this system is not currently possible and requires solutions to difficult geometric and systems-level problems.

3 Background

Teleconferencing applications have three major components: video capture, transport, and display. We concentrate on video capture and its display, and assume a 30 frame-per-second delivery capability from the underlying network.

3.1 Video Capture

Traditional video conferencing systems generally have a one-to-one camera-to-display relationship, i.e., a camera at one site captures a video stream and sends it to an another site, which displays the video on some display device. This setup is duplicated at both sites for two-way communication. Generally, other than compression and decompression for bandwidth constraints, no other processing is applied to the video stream. Undesirable effects, such as camera lens distortion, are not removed, and appear as distortion in the final imagery. To lessen these effects, video-conferencing systems use narrow field of view (FOV) lenses, which help keep such distortion to a minimum. However, depending on the number of participants, a narrow FOV can limit the interaction between the users. To ameliorate this limitation, video conferencing cameras are sometimes mounted on controllable pan-tilt units, or even managed by a camera person, allowing the camera to continuously frame the object of attention.

In many cases capture distortion can be corrected and is not always undesirable. Nayar's Omni-Camera [10], which uses a parabolic mirror to reflect incoming light toward a single center of projection, is able to capture 360 degree horizontal FOV imagery. Software is available to *un*-distort the imagery to make it look correct. There are several commercially available devices that produce imagery in this manner [12, 14]. Systems that capitalize on introducing a known distortion and provide software/hardware for un-distorting to produce WFOV imagery certainly help to allow more participants to be viewed by the users. A primary drawback is that the teleconferencing application becomes dependent on a very specialized capture device.

3.2 Display

Teleconferencing imagery is typically viewed on a flat display, usually a CRT device. This presents two problems. First, the *scale* of the imagery is almost always much smaller than real-life. Second, when WFOV capture devices are used, CRT and flat panel displays are inadequate because their FOV is usually significantly smaller than the FOV of the capture devices. In order to view WFOV imagery, the user must scale the imagery down or scroll through the imagery via software. Using a wall of flat panel displays can increase the FOV, but introduces obvious seams between adjacent panels, again breaking the sense of presence.

One solution to these problems is to use light projectors, which can display large-scale imagery. Multiple light projectors can be overlapped or carefully abutted to produce WFOV displays. This idea has been used in a number of immersive display systems, such as the University of Illinois at Chicago's CAVE [1] and Trimensions theaters [13]. Apart from distortions at the capture device, light projector devices also introduce their own distortions. The distortion from using light projectors is often more pronounced than CRT's or LCD panel's, because the display surface is no longer an integrated part of the display device itself. Typical problems include non-planar display surfaces, off-axis projection, and key-stoning. Existing systems, like the CAVE, try to avoid display distortion by carefully mounting the projectors and the display surface, which greatly increases the setup and maintenance cost. Such display environments are extremely expensive and difficult for untrained users to set up, requiring precise construction. These environments also require high-end graphic workstations for image rendering. These constraints make these systems an undesirable solution for teleconferencing applications.

Recent techniques using light projector displays that are less rigidly constrained have emerged [9, 8]. These algorithms allow the formation of seamless imagery even while displaying on arbitrary surfaces. In our work, we expand those techniques and unify them with the capture device.

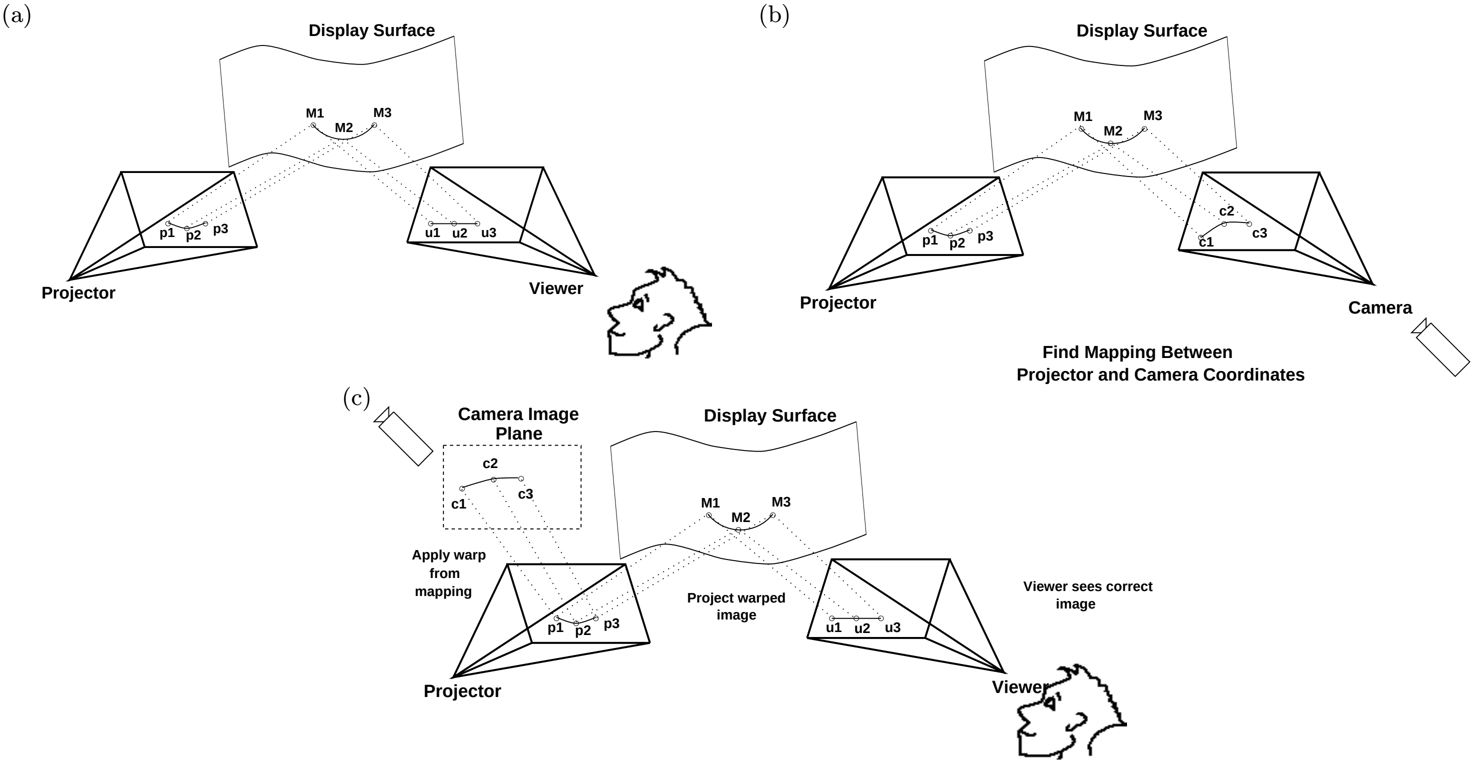


Figure 2: Outline of our method for producing geometrically correct imagery. (a) A projector illuminates three points on an arbitrary display surface. The points lie in a line for the observer. (b) The camera is positioned at the desired viewer location (*sweet spot*). A mapping is found between illuminated projector points and their corresponding camera coordinates. Note that due to camera distortion, the illuminated points no longer lie in a line. (c) Camera pixels are "pre-warped" to the appropriate projector pixels. The projected pre-warped imagery is un-warped by the projector and the display surface, producing correct imagery for the viewer positioned at the sweet spot.

4 Addressing Capture/Display Distortion

We first discuss the straightforward case of one camera and one light projector. If the camera performs pure perspective projection (i.e., behaves exactly as a pin-hole camera) and the projector is positioned such that its optical axis is orthogonal to a planar display surface, then the mapping between camera coordinates $C(u, v)$ and projector coordinates $P(x, y)$ can be described as a change in scale: $C(u, v) = P(su, sv)$, where s is the scale factor. Knowing the scale factor makes it possible to capture and display images so that they appear geometrically correct.

In practice, capture devices do not exhibit pure perspective projection and introduce perturbations of the mappings such that $C(u + \delta_u, v + \delta_v) = P(su, sv)$. It is often possible to model this perturbation with first and second order approximation to the distortion, which is generally radial distortion.

The situation becomes much more difficult to model with the introduction of the display's distortion, such as off-axis projection and a non-planar display surface. In order to produce correct imagery for a user in such an environment, the following parameters must be known [7] [1]:

- parameters for modeling the capture device
- parameters for modeling the projector device
- projector 3-D location
- display surface 3-D representation
- user 3-D location

Solving for these parameters is difficult and impractical. Even when all the parameters are known, rendering the imagery would be a computationally intensive operation, requiring a high-end graphics workstation for real-time execution.

We present a solution that avoids the difficulties of solving for the above parameters. Figure 2 outlines the idea. In Fig. 2(a) 3-D points M_1, M_2, M_3 are illuminated on the display surface by the projector pixels p_1, p_2, p_3. These points project to the viewer's image plane at u_1, u_2, u_3. In Fig. 2(b) we place camera where the viewer's eye is to be located. Due to distortion in the camera the viewer's points are perturbed such that $c_1 = u_1 + \delta_1$, $c_2 = u_2 + \delta_2$, $c_3 = u_3 + \delta_3$. From this location we find a mapping between the pixels in the projector and their viewed location in the camera, i.e., $p_i \rightarrow c_i$. Fig 2(c) shows the camera moved to a different location. The camera imagery is warped such that camera pixels c_i map to projector pixels p_i *before* the image is displayed by the projector. When the "pre-warped" image is projected, the display device and underlying display surface *undoes* the pre-warp and moves the camera pixels to the appropriate locations that will appear correct to the viewer.

[1] The technique by Raskar et. al was used to create perspectively correct imagery of synthetic data (rendered graphics) and assumed a distortion free projector. Their technique only required the last three items

This solution is a one-step method for finding the mapping that compensates for all of the geometric distortions introduced in the capture/display system. With this approach, producing correct imagery requires a 2-D warp of the captured coordinates c_i to projector coordinates p_i. This mapping can be obtained directly by turning on the projector's pixels p_i one-by-one and viewing them with the desired capture device.

The imagery will appear correct to the viewer when their eyes are located approximately where the capture device's center-of-projection was positioned when the mapping was computed. This location, commonly called the *sweet spot*, is the place from which the viewer should watch the projected imagery. As the viewer moves away from the sweet spot, the imagery will begin to appear distorted.

Our technique easily scales for multiple overlapped projectors. We find the mapping between each projector individually. Corresponding projector pixels in the overlapped regions are mapped to the same camera pixel resulting in geometrically seamless imagery between the projectors, without the need for image mosaicing or other image processing. This technique can also be apply to sensor devices that introduce a known distortion, such as compound camera devices or parabolic-mirrored cameras, just so long as the device maintains a common center of projection.

5 Implementation

We have implemented several capture/display configurations using the technique presented. Our applications are written in Visual C++ using the OpenGL API and run on Dell 410 and 610 400 Mhz Pentium II NT workstations, with Intergraph 3400 series graphics card for hardware texture-mapping and Matrox Meteor II frame-grabbers. Because the focus of this paper is the capture and display of imagery, we utilize an analog video network.

The capture/display configuration used various arrangements of JVC and/or Pulnix cameras with standard NTSC resoltion (640 × 480) and SHARP LCD 1024 × 768 LCD projectors for display.

Our software is has two components: finding the capture-to-display mapping (which we loosely call *calibration*), and the *rendering* application. Each runs separately.

5.1 Calibration Application

The calibration process collects the capture/display correspondences between the capture device (a single camera or a multi-camera rig) and display device (a set of fixed light projectors). Ideally, for every pixel $P(x, y)$ in the display device, we need to know its corresponding pixel $C(u, v)$ in the camera. It is often difficult and unnecessary to find correspondences for each pixel in the projectors. Usually the resolution of the camera is less than that of the projector, and a one-to-one mapping does not exist. Instead, we subsample the projector pixels from a given sweet spot. The calibration software allows the user to specify the sample resolution.

For our experiments we found that sampling resolution of every 32nd pixel up to every 8th display pixel was adequate. Since the resolution of the capture device is less than that of the projector, we use *features* in the projector image instead of individual pixels. Our features are 8×8 pixel blocks. The mapping is computed between the centroids of the features in the projector and the detected feature centroid in the capture device.

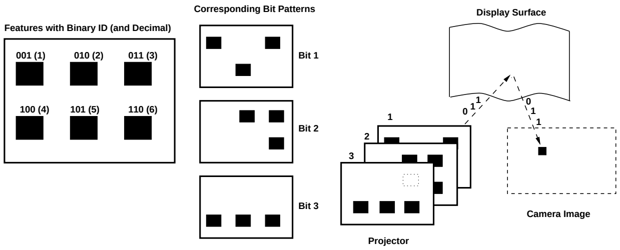


Figure 3: Binary encoded structured light scheme. (Left) Six binary encoded features and the corresponding bit patterns. (Right) Projected features are identified in the camera by their binary ID.

To facilitate feature detection, we use binary-coded structured light [6] to obtain fast and robust correspondence between features in the projector and the camera. Every feature is assigned a unique *id*. The *id* is coded in binary using n bits. We then create n black and white patterns as follows: every feature in the k^{th} pattern is colored white if the k^{th} bit of its *id* is 1, otherwise it is assigned the background color (black). Viewing these patterns with a camera, it is easy to compute which feature is being viewed by reconstructing its *id* from the viewed bit-pattern: n features can be identified by $log(n)$ patterns.

Fig 3 illustrates the principle of detecting features via binary-coded structured light. Synchronization is maintained by controlling the projector and camera from the same PC.

Moving the camera to the desired position (*sweet spot*), the structured-light calibration finds the capture-to-display coordinate correspondences. This capture-to-display mapping is saved and used by the rendering application.

5.2 Rendering Application

The rendering application uses the capture-to-display data to perform the image warping. From the sampled projector pixels $P(x, y)$ a 2-D triangulated mesh is created in the projector image plane using Deluanay triangulation. For each vertex, $P(x, y)$, its corresponding capture coordinate $C(u, v)$ is used as a texture coordinate. The camera's frame is used as a texture, getting updated every time a new frame is acquired. The texture-mapping achieves the image pre-warp. The pre-warped image is then displayed by the projector.

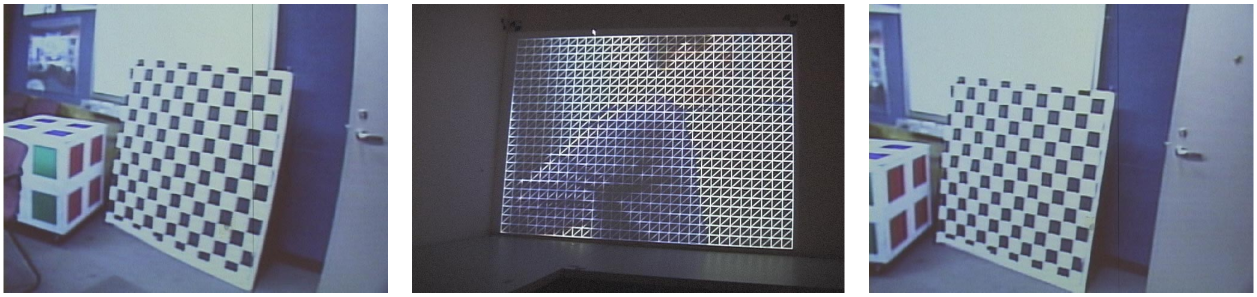
6 Results

This sections outlines several capture/display configurations we have implemented. Refer to table 1 for performance timings and sampling information. The following section provides further discussion on these experiments.

6.1 Experiment I: Fish-eye lens, one projector, and planar display surface

Our first experiment uses a camera with a 3.8mm fish-eye lens being displayed by one projector on a planar surface. Fig. 4 (a) shows the input from the camera displayed without any processing; radial distortion is obvious (note that straight lines appear curved).

Using the structure light technique outlined in Section 5.1, we find a mapping between display features and their corresponding locations in the camera. A 2-D triangle mesh is



(a) The direct output from the camera with a fish-eye lens projecting on a planar surface. There is noticeable radial distortion.

(b) The projector's textured mesh rendered in wire-frame mode.

(c) Applying the pre-warp via texture-mapping, the displayed imagery looks correct with the radial distortion removed.

Figure 4: Experiment I: Fish-eye lens camera, one projector, and a *planar* display surface

created in the projector's image plane. Each projector triangle vertex is assigned its corresponding camera coordinate as a texture coordinate. Fig. 4(b) shows the projector's textured mesh. The mesh is rendered in wire-frame mode for visibility. Fig. 4(c) shows the final result after pre-warping. Note that the radial distortion has been corrected.

6.2 Experiment II: Fish-eye lens, one projector, non-planar display surface

In the second experiment, we use the same capture/display configuration as before, but introduce a curved bump on the display surface, shown in Fig. 5(a). Using the previous capture-to-display mapping does not yield correct imagery (Fig. 5(b)), because the non-planar display surface introduces new distortion. After re-calibrating the new configuration, the updated mapping produces the correct imagery across the curved surface (Fig. 5(c)).

6.3 Experiment III: Fish eye lens and two projectors

For the third experiment, we use the same camera with two projectors with side-by-side overlap to create a WFOV display environment. Keeping the camera fixed at the desired viewer location, we calibrate each projector individually. Each projector generates its own capture-to-display mapping. The camera's video output is spliced to two PCs (one PC for each projector). Fig. 6(a) shows the resulting imagery, which is correct and geometrically seamless over the overlapped region of the projectors. Although our method produces *geometrically* seamless imagery in the overlapped region, the *photometric* seam in this region is visible because of double illumination, which leads to a bright strip. We use current intensity blending techniques [7, 9] to compensate for this artifact. The intensity of the overlapped regions are attenuated in each projector, such that a point illuminated by two projectors has the correct intensity value. Fig. 6(b) shows the result with blending.

6.4 Experiment IV: Multi-camera rig with multiple projectors

The fourth experiment uses a multi-camera and multi-projector system. The capture device is a common-center-of-projection multi-camera rig (known as the Camera-Cluster) designed at University of North Carolina at Chapel Hill. This cluster has twelve cameras which all share a common (virtual) COP. It is designed to provide a 360° horizontal FOV and 90° vertical FOV. Fig. 7 shows the Camera-Cluster and an example some of the cameras outputs. Only a portion of the output image (the region inside the grey partitions) is of interest.

In the camera cluster experiment, we use 10 cameras to cover a 220° horizontal FOV and 90° vertical FOV. We use five light projectors to create a 220° panoramic display environment. There are overlaps between projectors. Each projector is driven by a PC with multiple incoming video streams.

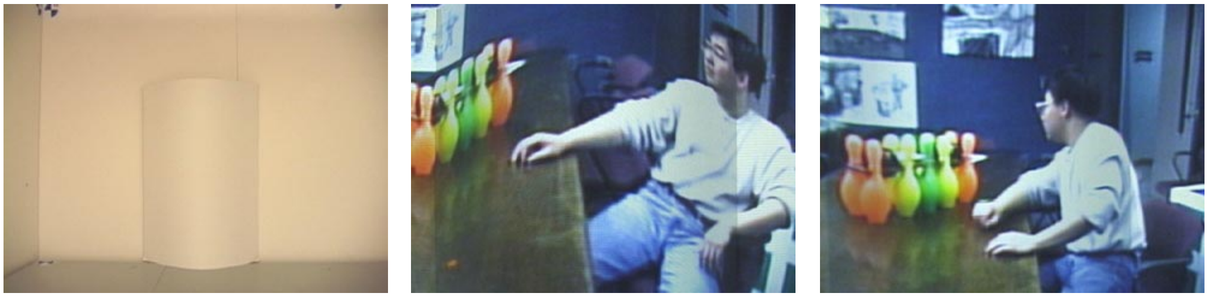
We positioned the Camera-Cluster at the desired sweet spot within the display setting, and performed the calibration one projector at a time. The calibration process took roughly an hour.

From the design of the Camera-Cluster and its positioning in our environment, one projector may need input from up to four cameras. Because of the large amount of data passed from video capture card to graphics card (up to four 640×480 32 bit video streams), the output imagery frame rate takes a performance hit. Table 6.4 describes the different setups and their corresponding performances. The frame rate is the average performance based on PCs with Pentium II 400 - 450 Mhz CPUs and Intergraph 3400 series OpenGL-accelerated graphics cards. We used one capture card (Matrox Meteor or Meteor II) per input channel.

Fig. 8(Left) shows a partial panoramic view of the captured conference room (there are still two projectors in the far left and right being clipped). It was taken away from the sweet spot to give an impression of the overall environment, because of this, it does not look geometrically correct. Fig.8(Right) is the geometrically corrected image take from the sweet spot.

7 Discussion

Our technique has several limitations, the first of which is that in order to produce correct imagery we must use the exact capture device for calibration and then for subsequent capture/display. This implies that for use with teleconferencing, we must calibrate with a camera and physically send

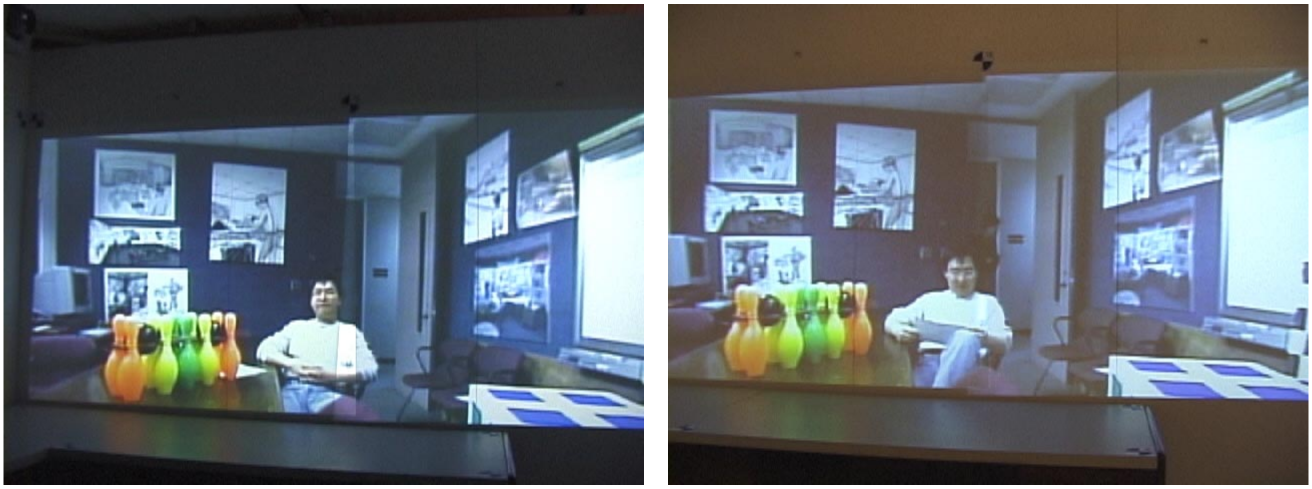


(a) We add a curved bump to the display surface.

(b) Projected imagery is distorted due to non-planar display surface.

(c) Corrected imagery based on a new capture-to-display mapping, which compensates for both camera distortion and the non-planar display surface.

Figure 5: Experiment II: Fish-eye lens camera, one projector, and a *non-planar* display surface. (a) We add a curved surface to the display surface. (b) The resulting imagery appears incorrect because of the display surface distortion. (c) After finding a new capture-to-display mapping, the resulting imagery looks correct.



(a) Two overlapped projectors without intensity blending.

(b) Two overlapped projectors with intensity blending.

Figure 6: Experiment III: Single fish-eye lens and two overlapping projectors. (a) Using one fish-eye camera, we find the capture-to-display mapping for two overlapped projectors. The resulting imagery looks *geometrically* correct, however, a *photometric* seam is visible in the overlapped region. (b) Using intensity blending in the overlapped region, we produce a visually seamless image.

Experiment	**Capture Device**	**Number of samples**	**Calibration time**	*fps* **no blending**	*fps* **blending**
I. Planar Surface	single fish-eye	720	40 *s*	24.02	–
II. Non-Planar Surface	single fish-eye	720	40 *s*	24.02	–
III. Overlapping Projs	Single fish-eye	720	80 *s*	24.02	20.11
IV. Multi-camera (a)	Two cameras	12000	7 minutes	17.33	15.63
IV. Multi-camera (b)	Four cameras	12000	14 minutes	9.99	9.99

Table 1: Performance Results: The table shows timing information, such as calibration time and rendered frames per second (fps), for our various experiments. It also lists the sample resolution for the experiments. Experiments with overlapping projectors give performance numbers with intensity blending on and off. Experiment IV is broken into two categories: (a) two cameras and (b) four camera configurations. The four cameras poor performance is mainly due to PCI bandwidth constraints on our PCs.

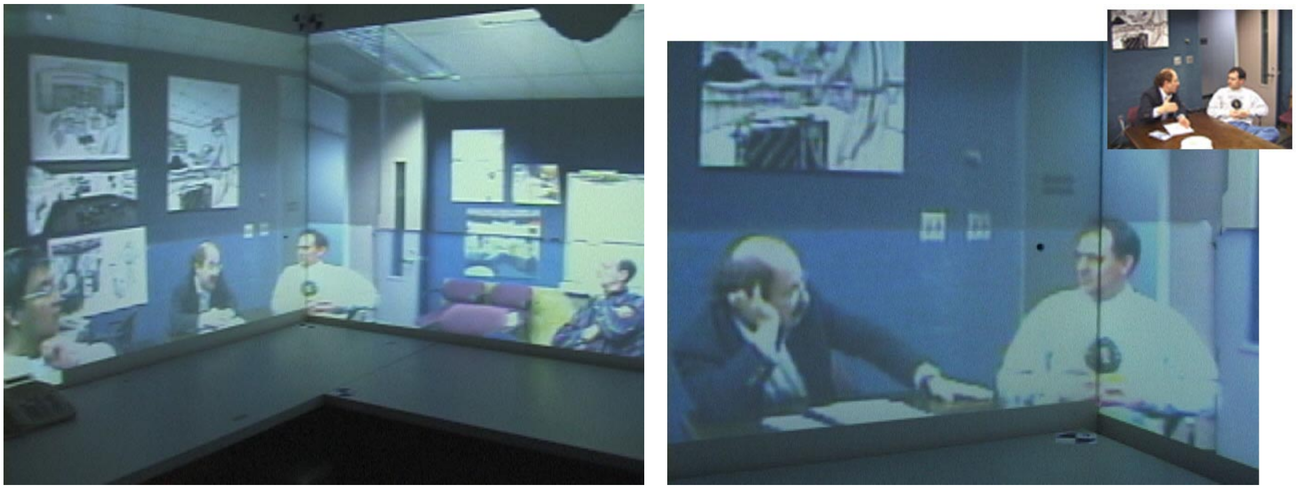


Figure 8: Experiment IV: (Left)Snap shot of multi-camera and multi-projector teleconference environment. There are two projectors in the far left that have been clipped. Note that this snapshot is not taken from the sweet spot and does not look perspectively correct. (Right) Snap-shot from the sweet spot, the image is perspectively correct. The small insert is a view taken roughly from the same position where the Camera-Cluster was placed at the remote-site.

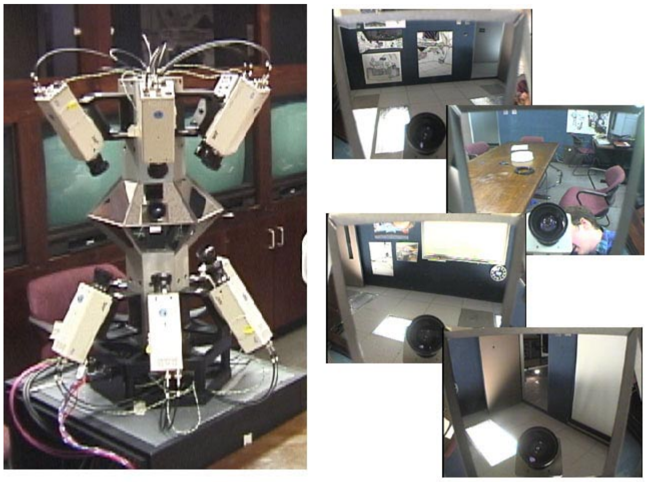


Figure 7: (Left) The Camera-Cluster (Right) Direct outputs of some of its twelve cameras

it to the remote-site. This is not a practical solution.

We have performed experiments calibrating with one camera and then capture/display using a second similar camera (i.e., same brand, model, and lens). Using the capture-to-display mapping obtained from the first camera, viewers were unable to distinguish the difference when the system ran using the second "similar" camera. Further experiments are necessary to determine to just how "similar" two devices need to be before they produce noticeable distortion. Our experiments indicate that it is very reasonable to simply have participants use the same model camera.

Second, our technique produces correct imagery for only one viewer location. This limits the technique to a system that supports one static viewer. However, many participants have viewed the setup outlined in our last experiment (Camera Cluster and multiple projectors). Many of those participants were unaware that they were not in the correct location until we informed them of the "sweet spot". Depending on the capture-to-display configuration during calibration, distortion as a result of moving away from the sweet spot varies greatly.

Lastly, we have found that many panoramic sensing devices that use multiple sensors to act like one logical device do not in fact provide a single common center of projection. The imagery produced in these cases will not be truly perspectively correct. In future work we will use several commercially available WFOV capture devices to get user reactions on the imagery they produce. We would like to determine how far from a common center of projection a multi-camera rig can be before the effects become noticeable to the viewers. There are also many photometric issues concerning multi-camera setups, such as color balance and focus between cameras.

8 Conclusions

We have presented a direct and efficient technique for producing geometrically correct imagery for a stationary viewer. Using an explicit capture-to-display device mapping, we compensate for the complex geometries and distortions of the capture/display model. This technique allows for the construction of realistic and compelling teleconferencing environments. We present results from several prototype systems we have implemented.

9 Acknowledgments

This work is supported by the National Science Foundation cooperative agreement ASC-8920219: "Science and Technology Center for Computer Graphics and Scientific Visualization", the "National Tele-Immersion Initiative" sponsored by Advanced Networks and Services, Link Foundation, and Intel Corporation.

We would like to thank Ramesh Raskar for proposing the use of the UNC Camera-Cluster use in a very-wide-field-of-view teleconferencing environment. We also gratefully acknowledge John Thomas, Jim Mahaney and David Harrison in the design and assembly of our test environment. We give special thanks to Herman Towles for useful discussions of implementation, hardware layout, and overall motivation.

References

[1] Cruz-Neria C., D.J. Sandin, and T.A. DeFanti. Surround-screen projection-based virtal reality: The design and implementation of the CAVE. *Computer Graphics, SIGGRAPH Annual Conference Proceedings*, 1994.

[2] Erikson H. Mbone: the multicast backbone. *Communications of the ACM*, 37(8):64–60, August 1994.

[3] Han J. and B. Smith. CU-SeeMe VR immersive desktop teleconferencing. In *Proceeding of ACM Multimedia*, pages 199–207, Boston, MA, 1996. ACM, ACM Press.

[4] Hewitt K. Desktop video conferencing product survey. *http://www3.ncsu.edu/dox/video.*

[5] Yamaashi K., J. Cooperstock, T. Narine, and W. Buxton. Beating the limitations of camera-monitor mediated telepresence with extra eyes. In *SIGCHI 96 Conference proceedings on Human factors in computer systems*, 1996.

[6] Altschuler M.D., B.R. Altschuler, and J. Taboada. Measuring surfaces space-coded by a laser-projected dot matrix. *Imaging Application for Automated Industrial Inspection*, 1979.

[7] Raskar R and G. Welch et al. The office of the future: A unified approach to image based modeling and spatially immersive displays. In *SIGGRAPH 98 Conference Proceedings*, Orlando FL, July 1998. ACM.

[8] Raskar R. and M. Brown et al. Multi-projector display using camera-based registration.

[9] Raskar R., G. Welch, and H. Fuchs. Seamless projection overlaps using image warping and intensity blending. In *4th Annual Conference on Visual Systems and Multimedia*, November 1998.

[10] Baker S. and S. Nayar. Catadioptric omnidirectional camera. In *Proceedings of CVPR.* 1997.

[11] Mcanne S. and V. Jacobson. A flexible framework for packet video. In *Proceeding of Multimedia*, San Francisco, November 1995. ACM, ACM Press.

[12] Kawanishi T., K. Yamazawa, H. Iwasa, H. Takemura, and N. Yokoya. Generation of high resolution stereo panoramic images by omnidirectional sensor using hexagonal pyramidal mirrors. In *ICPR '94*, 1994.

[13] Trimensions. http://www.trimension-inc.com.

[14] Nalwa Vic. His camera won't turn heads. *http://www.lucent.com/ideas2/innovations/docs/nalwa.html.*