Creating Adaptive Views for *Group* Video Teleconferencing – An Image-Based Approach

Ruigang Yang, Celso Kurashima, Andrew Nashel, Herman Towles, Anselmo Lastra, Henry Fuchs
Department of Computer Science, University of North Carolina at Chapel Hill

Abstract

We present a system and techniques for synthesizing views for many-to-many video teleconferencing. Instead of replicating one-to-one systems for each pair of users, or performing complex 3D scene acquisition, we rely upon user tolerance for soft discontinuities for our rendering techniques. Furthermore, we observed that the participants' eyes usually remain at a constant height (sitting height) during video teleconferencing, thus we only need to be able to synthesize new views on a horizontal plane. We demonstrate a real-time system that uses a linear array of cameras to perform Light Field style rendering. The simplicity and robustness of Light Fielding rendering, combined with the natural restrictions of limited view volume in video teleconferencing, allow us to synthesize photo-realistic views for a group of participants at interactive rate. **Categories and Subject Descriptors**: H.5.1[Multimedia Information System]: Video teleconferencing; **General Terms**: Design, Algorithms; **Keywords**: Group video teleconferencing, Light field rendering, Scene reconstruction.

1 Introduction

With recent rapid advances in network bandwidth and dropping costs for video equipment, video teleconferencing, a technology enabling communicating with people face-to-face over remote distances, has been widely deployed for business and education. Currently, the majority of video teleconferencing applications are designed for the *one-on-one* scenario that limits the capture of a video stream to a single sensor and the display to a CRT or flat-panel device. While widely used, this one-on-one interface does not provide a compelling or convincing presence to the participants [14].

Most group teleconferencing systems in use today are simply versions of the one-to-one system used by a group of people at each site. Such single camera/single display systems usually suffer from low resolution, small fields-of-view, and smaller than life-size displays. Attempts to overcome these limitations have involved replicating the one-to-one system for each set of participants, such as in [4, 13]. Typically, they use half-silver mirrors and cameras placed along the gaze direction to maintain geometry continuity and eye contact for multiple persons simultaneously. Although this may address some of the problems, these systems commonly produce discontinuities, or *hard* artifacts, in the display at camera boundaries. This results from the simple stitching or warping of acquired images from cameras with different centers of projection.

One solution that we and others have explored is a camera-mirror array creating a common, but virtual center of projection [15, 9]. Imagery from these systems is correct for a single static "sweet spot" and image distortion for the viewer increases with distance from the virtual center of projection. Thus such systems are best for a many-to-one conference but do not scale for group teleconferencing. Furthermore, precise alignment of many cameras and mirrors remains a manufacturing challenge.

The *Office of the Future* group at University of North Carolina at Chapel Hill in 1998 introduced a vision for the ultimate teleconferencing/collaboration interface [10]. In their long-term vision, an ordinary office is equipped with "a sea of cameras" and projectors [1]. The complete, dynamic 3D scene is extracted using computer vision techniques and transmitted over the network to a remote office. A unique view is then rendered in life size for each remote viewer. Thus collaborators in any locale would be able to interact with each other as if they were in a common room. The challenges in implementing this fully reconstructed many-to-many interface are enormous. With today's available hardware, a number of technical tradeoffs have to be made. Increasing the fidelity of scene acquisition leads to higher reconstruction latency and lower frame rates [8]. Such systems are also practically limited to two viewers by current 3D display technologies.

Figure 1: Our Multiple-Camera Group Video Teleconferencing Prototype in Use: The top photo left shows the remote participants. Our camera array to capture the participants is shown in top right. The bottom photo shows the local participants. The life-size, seamless image is synthesized using our method in Section 3.

In this paper, we present an alternative design for the many-to-many video teleconferencing interface. Instead of striving to synthesize a *perfect* view for *everyone* – which we do not believe practical in the near future, we try to provide the *best approximate* view for each local group *as a whole*, while maintaining geometric continuity and the sense of presence, without using any special hardware or limiting the number of participants or their locations. We focus on supporting collaboration between a small group of three to five people sitting on one side of a conference table, meeting with a remote group seated virtually across the table, through life-size, wide field-of-view imagery with only *soft* artifacts, such as incorrect viewpoints and small distortions, as shown in Figure 1. Another important goal is that users be unencumbered by tracking devices or special eye-wear for rendering.

2 Design Motivations

We believe that we will be able to provide a better view for the group as a whole if we can (a) eliminate any *hard* boundary in the synthesized views and (b) minimize the deviation of the average

distance to the center of projection of the synthesized view. While the first criterion is quite obvious, the second needs elaboration. A single-person teleconferencing system (as in Figure 2(a)) typically places the camera at position C to provide eye contact between the participants. Note that we are analyzing this system from the point of view of the participants at Location I, and that the camera is physically at Location II. An analogous camera position for the two-to-two conference system would be at C in Figure 2(b) , which is at the midpoint between the two viewers, P_1 and P_2. We believe a camera placed at C', where $C'P_1P_2$ forms an equilateral triangle, will be in a better position. If there are more participants, as shown in Figure 2 (c), the camera should be even farther away.

A high resolution, movable camera could be placed in Location II to capture the desired image, but in a typical situation there is hardly any room behind the screen to maneuver the camera. So the problem we are trying to solve is *to generate a seamless, high resolution image from the perspective of a user-driven virtual camera, using inputs from a number of fixed cameras*.

Unlike view synthesis in general, we use natural constraints in video teleconferencing, such as the limited viewing area and the limited depth variations of human participants, to simplify the synthesis problem and provide a practical and useable image-based approach for group video teleconferencing.

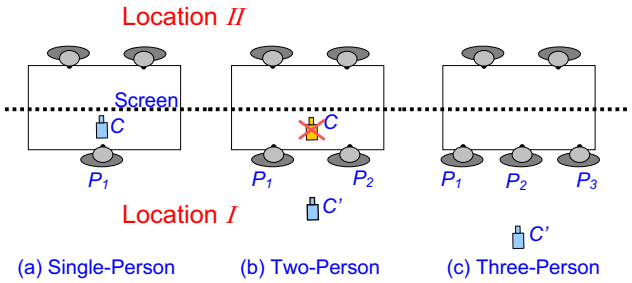


Figure 2: In a see-through-a-window conference design, our desired camera placement (C') as the number of participants grows. Most commercial systems, optimized for a single participant, use a single fixed camera as in C.

3 Image-Based Methods

Image-Based Modeling and Rendering (IBMR) methods have become a popular alternative to synthesize novel views. The key for IBMR is to reconstruct the *plenoptic function* that describes the flow of light in all positions in all directions [7]. With a complete plenoptic function, novel views can be easily synthesized by plugging the location and directions for the novel views into the plenoptic function. A class of IBMR methods, called *Light Field Rendering* (LFR), uses many images to pre-record the plenoptic function [5, 2, 11]. LFR methods often achieve a stunning level of realism without using any geometric information. Encouraged by these recent advances, we explore the possibilities of creating high-resolution, seamless virtual views using LFR techniques.

3.1 Perspective View Method

We observed that during a video teleconferencing session, the participant's view point is quite limited, usually at the eye level, with small lateral motions. Thus we can use a 1D linear array of cameras to capture a compact light field, which we refer to as the *Line Light Field*. This compact 1D setup makes real-time capture, transmission, and rendering possible. To achieve the best result, it is desirable to place the camera array horizontally at eye level using half-silvered mirror or actively controlled screen [3]. Novel views at eye level can be changed interactively, allowing the participants to view the remote scene from side to side, or from near to far to gain a sense of 3D. Furthermore, we can synthesize large FOV images using cameras that do *not* share a common center of projection.

Similar to the original LFR paper [5], we parameterize the captured light field by a line (the camera array) and a plane (the focal plane). We allow the user to control the position of the focal plane and the virtual viewpoint to achieve optimal viewing. With our linear camera setup, the blending weight only varies in the horizontal direction. A simpler blending scheme that only uses two nearest cameras for linear viewpoint motions was introduced by Sloan et al [12].

3.2 Orthogonal View Method

We find that the synthesized view from our perspective method is relatively blurry. This effect was caused by the under sampling in our camera system. We would like to improve the picture quality without increasing the number of cameras in use.

As we discussed at the beginning of Section 2, we desire to create a continuous, high-resolution, wide field-of-view image from a perspective further behind the screen, a good compromise for a group of people. As the number of participants grows, we would like to push the center of projection further away, so that every one has the same distance to the center of projection. If we push this idea to the extreme, we eventually want to display an orthogonal view. Unfortunately, normal cameras are designed to take perspective images. But we can create an orthogonal view from an array of cameras. For our 1D linear camera array, if we take the vertical scan line going through the image center for each camera, and piece them side by side, we can get horizontally orthogonal images. In practice, we can always use a small vertical strip of each camera due to the limited resolution of the display device, as well as the human visual system.

This thinking results in an extremely simple view synthesis method. For each camera image, we take out a narrow band in the middle, and juxtapose these bands. We also introduce a small amount of overlap between adjacent bands to accommodate for small registration errors and avoid the harsh boundaries for color mismatches. Unlike the perspective view method in the previous section, there is little inter-camera dependency, since the final color of each pixel in the synthesized view depends on at most two cameras' images. Thus it is possible to distribute (not replicate) the input image data to a number of PCs to create wide FOV high resolution imagery.

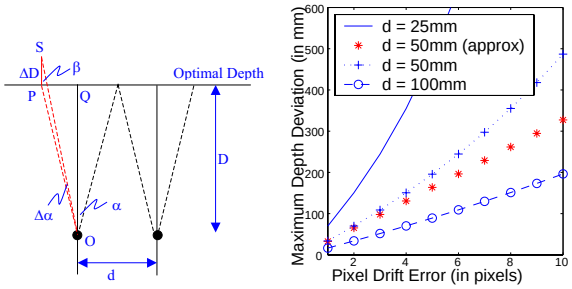
3.3 Sampling Requirement Analysis for Orthogonal Views

In the ideal case, when we only use a single vertical scan line through the image center from each input image to composite a horizontally orthogonal image, then no matter how far away the object is, its projection on the synthesized view remains the same. This means we can generate correct imagery without knowing the locations of the scene objects, thus avoiding the difficult scene reconstruction problem. But this is not practical since it would require thousands of cameras to create a single image. So we use a narrow band of columns to approximate the orthogonal view. If we back project the narrow bands into space, they will intersect at a certain distance, which we call the optimal depth D. Only the objects at the optimal depth will have the correct imagery in the synthesized views. Objects that are closer could be lost and objects that are further will have duplicates. We define an error tolerance measure (e) in terms of pixel drift, i.e., the distance from a pixel's ideal location in the synthesized view. For a given configuration, we would like to find out how much error there will be, or conversely, given an error tolerance measure, how many cameras are needed.

Inspired by the sampling analysis for LFR in [6], we attack this problem using a geometric approach. We assume that all of the cameras are mounted on a horizontal rail and regularly spaced. The optical axes of the cameras are parallel on a horizonal plane. Let us define the following parameters:

- Camera's field of view FOV
- Camera's horizontal resolution W (in number of pixels)
- Inter-camera distance d

The problem we try to solve here can be stated as follows: Given a set of camera configuration parameters, and a desired error tolerance e , what is the maximum depth deviation ΔD from the optimal depth D.



(a) Geometric Setup

(b) The maximum depth deviation with respect to pixel drift error

Figure 3: Error Analysis for Creating Orthogonal Views

From Figure 3(a), it is easy to see that $\alpha = tan^{-1}(\frac{d}{2}/D)$, $\beta = \angle OPS = 90 + (90 - \alpha) = 180 - \alpha$. After some trigonometry manipulations, we get

$$\Delta D = \frac{sin(\Delta\alpha)\sqrt{(d/2)^2 + D^2}}{sin(\alpha - \Delta\alpha)}$$

We can then approximate the angular deviation $\Delta\alpha$ in term of pixel drift e, where $\Delta\alpha = (e/W)FOV$. That leads to:

$$\Delta D = \frac{sin(e/W * FOV)\sqrt{(d/2)^2 + D^2}}{sin(\alpha - e/W * FOV)}, \quad (1)$$

where FOV is expressed in radians. Furthermore, since $sin(\alpha) = (d/2)/\sqrt{(d/2)^2 + D^2}$, $(e/W)FOV$ is usually a very small number and $(e/W)FOV \ll \alpha, d \ll D$, we can approximate Equation 1 as

$$\Delta D = \frac{e}{W} FOV \frac{D^2}{d/2} \quad (2)$$

We can derive a similar equation in case S is closer to the camera instead of farther away.

Let us assume $FOV = 30^o, W = 640,$ and $D = 1000$ mm. Figure 3(b) shows the maximum depth deviation with respect to pixel drift error under different camera placements $d = 25, 50, 100$ mm. The red line shows the results computed using the rough approximation (Equation 2), while the rest are computed using Equation 1. Note these are "one-sided" numbers, i.e., they only represent how much *further* away the real depth can be. The total distance variation is roughly twice as long. From the results we can see that it is indeed possible and practical to create crisp orthogonal images for depth variation under 400 millimeters, a reasonable value to accommodate normal human motions during a conference.

4 Implementation and Results

We have implemented our methods under the Windows environment. Our current prototype includes a total of 11 Sony digital firewire cameras arranged in a linear array, as shown in Figure 1. These cameras are regularly placed at 65 millimeter apart, very close to the minimum distance allowed by the form factor of the camera body. We are experimenting with inexpensive digital cameras, such as the *iBot* from Orange Micro (http://www.orangemicro.com/ibot.html), which offers full VGA-resolution, non-interlaced, digital color image at a cost much less than 100 dollars each. With these cameras, we can make a similar camera rig well under one thousand dollars. Currently, all cameras are synchronized by a wire controlled from a PC and fully calibrated using the method from [17].

The rest of our prototype includes a number of PCs interconnected through 100Mbit Ethernet. Six of them are video servers. Each of them is connected to a maximum of two Sony cameras and is used to capture and JPEG-encode the raw image data at full VGA resolution. Note that our system design is very flexible; we could easy increase the number of geometry servers or rendering modules as the number of participants increases or there is need to increase the screen size.

Our image-based method uses all eleven cameras. We can achieve an update rate of 8-10 FPS for QVGA images, and 4-7 FPS for VGA images (the rendering is fully interactive, over 30 FPS). The bottleneck is in image capture. We can only capture synchronized VGA resolution images at 7-8 FPS with two cameras on the same 1394 bus. This is caused by the 1394 bus bandwidth limitation and the transfer characteristics of the digital cameras under external trigging.

We first show the results from our perspective view method in Figure 4; note the obvious parallax in these pictures. In Figure 5, we compare the results between the perspective view method and the orthogonal view method. The first one is synthesized by the perspective method. The color band below is the color-coded blending weights for each camera. The second one is the orthogonal view and its blending weights. Note that we extend the column width for the last and first image to increase the field of view for the orthogonal image. It is quite obvious that the second one is crisper even in the blended part in the middle.

Figure 4: Side Views of the Perspective Method.

To create life-size images, we use *PixelFlex* [16], a reconfigurable projector array, as the display device. PixelFlex is composed of computer-controlled ceiling-mounted projectors and rendering PCs. Working collectively, these projectors function as a single logical display. PixelFlex closes the loop in our entire system – presenting the conferees with seamless, wide field-of-view images beyond XGA resolution. We use three projectors in PixelFlex and the orthogonal view synthesis method to create the title photos shown at the beginning of this paper (Figure 1). In a teleconferencing session with fewer participants, we use the view-dependent perspective method to synthesized desired views, shown in Figure 6. We put a stationary book to illustrate the view dependent effect when the local conferees move to different spots.

5 Discussion and Conclusions

We have obtained some very realistic results in our prototype system. With smaller,inexpensive cameras becoming available, we believe our methods provide a useful solution in the near term.

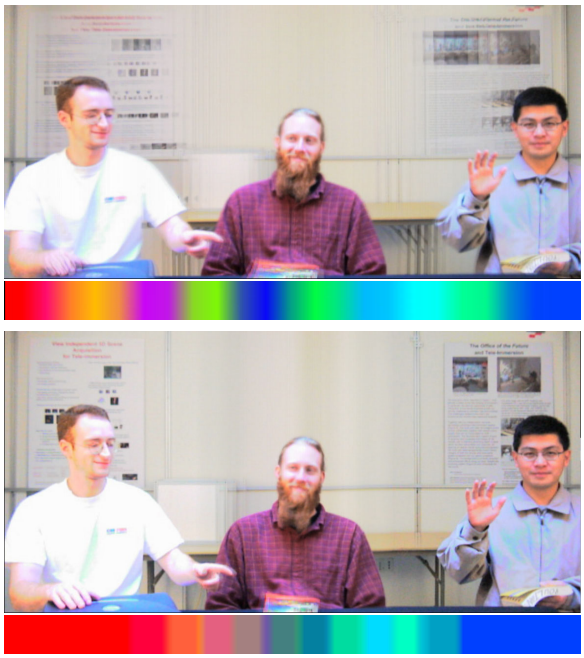


Figure 5: Comparison of Perspective View vs. Orthogonal Views. The narrow color bands below the images show the color-coded blending weights. Each camera is assigned a unique color, the color band is the weighted sum from all cameras.

Figure 6: View dependent effect when the local conferees move to different spots. Smaller images show the synthesized views. Notice that we have placed a book in the scene. When the local conferees are at right, as in the top image, the view point of the synthesized view is from the right, revealing the front cover of the book. When the conferee moves to left, as in the bottom image, the view changes accordingly, revealing the back cover of the book.

The bottleneck for our methods is the bandwidth, both the network bandwidth and of PC's internal bus bandwidth. Using orthogonal views could alleviate this problem since there is less data dependency between adjacent pixels in the synthesized views. Thus we could distribute and parallelize the rendering task to a number of PCs using a simple screen-space partition.

Looking into the future, we might achieve the best results by estimating some simple geometry using computer vision methods. For example, we could use a plane fitting algorithm to automatically adjust the focal plane position in the perspective method. Another major piece of future work is the validation of our assumption about the soft-discontinuity preference. In group video teleconferencing, do we prefer to have a continuous view of the entire group, with some ghosting in the near or far field; or rather prefer to have many one-on-one direct video feeds on monitors side by side, which contain obvious geometry discontinuities? Though we have a strong belief that most of us will prefer the former, the final answer to this question requires a rigorous user study.

In conclusion, we present a system and techniques for synthesizing views for many-to-many video teleconferencing. Instead of replicating one-to-one systems for each pair of users, or performing complex 3D scene acquisition, we rely upon user tolerance for soft discontinuities for our rendering techniques. We strive to create continuous (though not necessarily geometrically correct), high resolution, wide field-of-view imagery using casually-placed fixed cameras. We demonstrate a real-time system that uses a linear array of cameras to perform Light Field style rendering. We believe that such algorithms will lead to rendering of the best approximate views for groups of people, using currently available hardware and without limiting the number and position of participants, to achieve a flexible and scalable solution for group video teleconferencing.

This research was funded by Sandia National Laboratories under the Department of Energy's ASCI VIEWS program.

References

[1] Henry Fuchs, Gary Bishop, Kevin Arthur, Leonard McMillan, Ruzena Bajcsy, Sang Lee, Hany Farid, and Takeo Kanade. Virtual Space Teleconferencing Using a Sea of Cameras. In *Proceedings of the First International Symposium on Medical Robotics and Computer Assisted Surgery*, Pittsburgh, PA, Sept. 1994.

[2] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The Lumigraph. In *Proceedings of SIGGRAPH 1996*, pages 43–54, New Orleans, August 1996.

[3] Andreas M. Kunz and Christian P. Spagno. Technical System for Collaborative Work. In *Proceedings of Workshop on Virtual Environments 2002*, May 2002.

[4] L.C.Desilva, M.Tahara, K.Aizawa, and M.Hatori. A Multiple person eye contact (MPEC) teleconferencing system. In *Proceedings of IEEE International Conference on Image Processing*, volume II, pages 608–610, Octerber 1995.

[5] M. Levoy and P. Hanrahan. Light Field Rendering. In *Proceedings of SIGGRAPH 1996*, pages 31–42, New Orleans, August 1996.

[6] Z.-C. Lin and H.-Y. Shum. On the numbers of samples needed in light field rendering with constant-depth assumption. In *Proceedings of CVPR*, 2000.

[7] L. McMillan and Gary Bishop. Plenoptic Modeling: An Image-Based Rendering System. In *Proceedings of SIGGRAPH 1995*, pages 39–46, 1995.

[8] J. Mulligan and K. Daniilidis. View-independent Scene Acquisition for Tele-Presence. Technical Report MS-CIS-00-16, Computer and Information Science Dept., U. of Pennsylvania, 2000.

[9] PanoramTech. Panoram Technologies. http://www.panoramtech.com/.

[10] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. *Computer Graphics*, 32(Annual Conference Series):179–188, 1998.

[11] H. Y. Shum and L. W. He. Rendering with Concentric Mosaics. In *Proceedings of SIGGRAPH 1997*, pages 299–306, 1997.

[12] P.-P. Sloan, M. F. Cohen, and S. J. Gortler. Time Critical Lumigraph Rendering. In *Symp. on Interactive 3D Graphics*, April 1997.

[13] TeleSuite. TeleSuite Video Teleconferencing Systems. http://www.telesuite.com/models.html.

[14] K. Yamaashi, 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.

[15] R. Yang, M. S. Brown, W. B. Seales, and H. Fuchs. Geometrically Correct Imagery for Teleconferencing. In *Proceedings of ACM Multimedia 99*, pages 179–186, Orlando, November 1999.

[16] R. Yang, D. Gotz, J. Hensley, H. Towles, and M. Brown. PixelFlex: A Reconfigurable Multi-Projector Display System. In *Proceeding of IEEE Visualization 2001*, pages 167–174, San Diego, CA, 2001.

[17] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.