The Coliseum Immersive Teleconferencing System

H. Harlyn Baker, Donald Tanguay, Irwin Sobel, Dan Gelb, Michael E. Goss,
W. Bruce Culbertson and Thomas Malzbender

Hewlett-Packard Laboratories
1501 Page Mill Rd., MS 1181
Palo Alto, CA 94304

ABSTRACT

We describe Coliseum, a desktop system for immersive teleconferencing. Five cameras attached to a desktop LCD monitor are directed at a participant. View synthesis methods produce arbitrary-perspective renderings of the participant from these video streams, and transmit them to other participants. Combining these renderings in a shared synthetic environment gives the illusion of having remote participants interacting in a common space.

Categories and Subject Descriptors

H.4.3 [**Information Systems Applications**]: Communications Applications – *Computer conferencing, teleconferencing, and videoconferencing*.

General Terms Algorithms, Experimentation

Keywords Telepresence, Teleconferencing, View synthesis.

1. INTRODUCTION

Automatic construction of 3D models from multiple images has seen remarkable progress over the last few years. This is attributable partly to Moore's Law – the doubling of computing power and storage every 18 months – but more to the development of novel approaches to the problem, such as the use of color statistics within a volumetric framework [Seitz 97, Culbertson 99]. While earlier methods produced off-line high-quality reconstructions of a static scene imaged by multiple calibrated cameras, recent methods have advanced to real-time, on-line reconstructions [Matusik 00]. Although these techniques have been limited to reconstructing the visual hull of an object, when combined with view-dependent texture mapping they can produce fairly convincing displays of time varying content. Building on the MIT Image-based Visual Hulls (IBVH) system [Matusik 00], we have leveraged these methods in building a multi-participant desktop teleconferencing system.

Our system, named Coliseum, consists of 5 cameras, about 30 degrees apart, directed at a user positioned before an LCD monitor (Figure 1). With intrinsic and extrinsic calibration, view synthesis methods can be employed to produce renderings of the user from novel perspectives. Unlike views of avatars, these displays can communicate in real time such personal features as gesture, expression, and body language. Each user is assigned an initial location in a shared virtual environment, and is then free to move around. Displays are updated to this view, with renderings of others present reflecting this changed perspective. In this manner, an illusion is maintained of a shared environment responsive to participant movements.

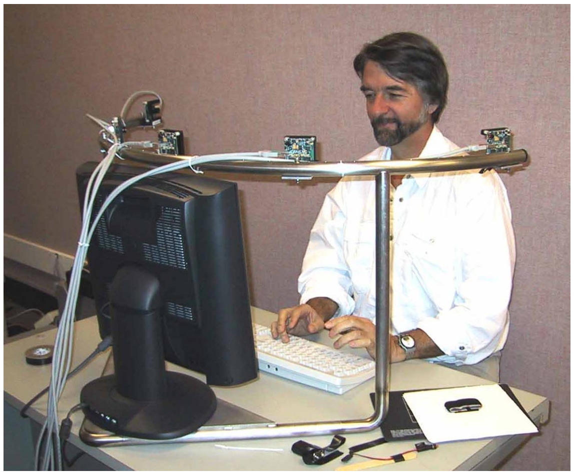

Figure 1. Desktop Coliseum system with user.

2. RELATED WORK

The pursuit of videoconferencing has been long and accomplished, although in-large-part less than successful in its commercialization. Some of the more notable earlier works addressed multi-participant issues, with a focus on user interface and human factors concerns [Gaver 93]. Our intention has been to push the envelope on all dimensions of the technology – display framerate and resolution, response latency, communication sensitivity, supported modalities, etc. – to establish a platform from which, in partnership with human factors and remote collaboration experts, we may better understand and deliver on the requirements of this domain.

Close to our approach is the Virtue project [Schreer 01], where sets of stereo cameras in a dedicated workspace enable

International Workshop on Immersive Telepresence '02, December 6, 2002,
Juan Les Pins, France.

{harlyn.baker, donald.tanguay, irwin.sobel, mike.goss, dan.gelb, bruce.culbertson, tom.malzbender}@hp.com

embedding a fixed number of users (3) in a shared synthetic 3D environment. Coliseum differs from Virtue in that it is a desktop system and supports participant mobility. The National Tele-Immersion Initiative [Lanier 01] developed a room-based multi-camera immersive teleconference system, again using real-time range to augment video display in a shared synthetic environment. Both of these systems use stereo correspondence to provide pixel-based range in the scene. While stereo ranging can give good definition of feature position, artifacts seem unavoidable, and seriously degrade perceptual quality.

In Coliseum, we acquire approximate participant geometry through the IBVH methodology. This approach has less dependence on true geometry, relying on rough shape that is painted with camera-acquired imagery for view-dependent texture mapping. In each video stream, the user (foreground) is segmented from the back-ground using pixel color statistics. The foregrounds from all video streams are backprojected and intersected to produce a visual hull that can be rendered and textured from arbitrary viewpoints to achieve participant depictions.

3. THE COLISEUM SYSTEM

Our initial implementation of the Coliseum system copied the room-scaled multiple-PC system of MIT, but aimed at a desktop situation using five low-cost small-format firewire CCD cameras, each attached to a computer (Figure 1). A sixth computer was used for camera integration and hull construction, and a seventh for network arbitration and synchronization.

In aiming the IBVH methodology at multi-participant 3D videoconferencing, a number of considerations became apparent. Clearly, such a multi-PC configuration lacks commercial viability. Scaling the system for desktop use with more limited computing resources provoked a review of the algorithms, data representations, and program structure. This review made it apparent that redesign and reimplementation of the Coliseum system would be necessary to reach our system goals.

3.1 IBVH for Single-PC Operation

The core of Coliseum's processing is split between computation on the acquired reference images and integration of these into the visual hull and its resultant display in the desired image. Redesigning the system for the single-PC close-range videoconferencing application – where the processing for all five cameras must be squeezed into a single machine – necessitated reconsideration of all computational and architectural elements of the task.

- The system must run on a commodity platform – a single personal computer.
- The analysis must meet throughput constraints for a variety of host processors – providing users with sufficiently low-latency visualization at sufficiently high framerates with high-enough resolution to make the experience acceptable.
- An evolving system with multiple contributors demands a modular organization with well-defined interfaces.
- The task involves seriously intensive computing and data handling that must be coordinated across multiple input and output streams (cameras, networks, and computers). A principal requirement is to balance and optimize processing resources across the pipelined system.

We have developed a new single-PC version of the Coliseum system that meets these requirements. Our principal improvements have occurred in the control/architecture structuring (providing a modular programming style with primitives for dataflow optimization and synchronization), in the low-level image analysis (what we do at each camera and how we calibrate the multi-camera system for operation), and in the interconnection organization among participating computers (how they communicate with each other in a conference setting). We highlight these changes below.

3.2 System Structure

Videoconferencing is a streaming media application. In support of this requirement, we have developed an infrastructure that facilitates building such data-intensive bandwidth-demanding systems. In the style of Microsoft's DirectShow, this capability allows the programmer to translate task dependencies into a computational dataflow graph in which nodes of atomic computation are connected by streaming media links. This framework abstracts application building, and delivers software engineering benefits including modularity, extensibility, and reusability. Since each module is atomic, the computational graph itself determines the amount of parallelism in an application. This framework automatically exploits the inherent parallelism in a graph, removing from the programmer the burden of parallelization and synchronization. Multiple instances of these computational graphs are combined to communicate together over a network in connecting remote participants.

Using this dataflow framework, the algorithmic core of our application has the structure depicted in Figure 2. This is a pipeline with four stages: acquisition, 2D image analysis, 3D reconstruction, and rendering. First, the acquisition phase captures a synchronized set of images from 5 cameras. Then, the 2D processing phase constructs foreground silhouettes by differencing input images and 2D background models. Next, the reconstruction phase uses the silhouettes to construct the image-based visual hull for the required novel viewpoints. Finally, the render phase paints the visual hull according to visibility constraints to produce the desired output images.

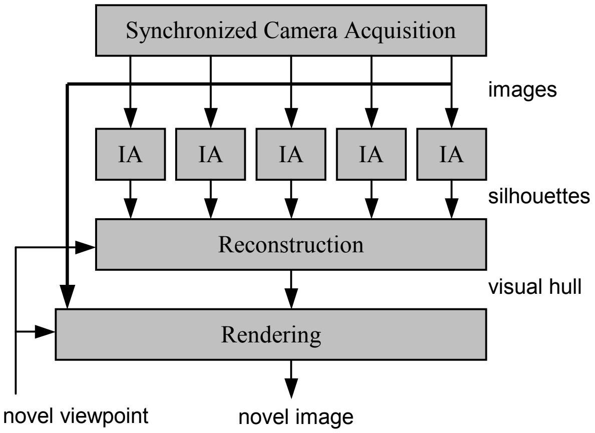

Figure 2. Algorithmic architecture: acquisition, image analysis, reconstruction, rendering.

3.3 Computational Issues

The heavy computational nature of this application, along with the significant burden of processing five video streams simultaneously, makes the size of our challenge clear. Every advantage must be made of algorithm design and processing efficiencies.

3.3.1 Image Analysis

Major system throughput improvements for single-PC operation have come through a recasting of the low-level image analysis methods. Table 1 sketches the changes employed at each camera in reaching our current level of performance. Several principles guided our redesign of this low-level processing:

- Touch a pixel as few times as necessary (i.e., once).
- Minimize data copying (i.e., don't).
- Use *lazy evaluation* to eliminate computation that may be found to have been unnecessary.
- Provide handles for trading quality for speed, so host capability can determine display/interaction characteristics.

Elements of our single-PC processing that embodied these guidelines include:

- Reading the camera raw Bayer mosaic (providing nearly five running VGA cameras on a single 1394 bus).
- Reducing foreground contour complexity by using piecewise linear approximations.
- Correcting lens distortion on foreground contours rather than on the acquired camera data (tens of vertices *vs*. 0.3M pixels).
- Resampling color texture for viewpoint-specific rendering on demand (once the needed source pixels are identified).

Figure 3 shows the results of foreground contouring, displayed with their visual hull in the space of the five Coliseum cameras.

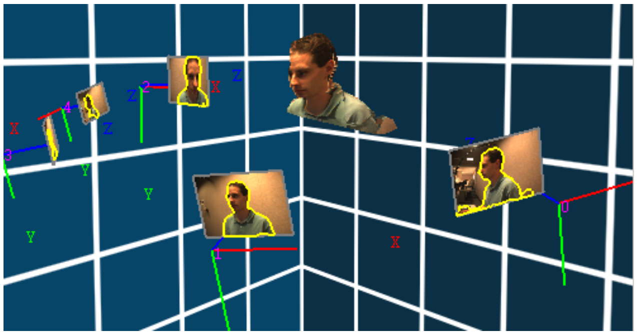

Figure 3. View of user in Coliseum space: Five cameras surround the rendered user. Each camera shows its coordinate system (in RGB), video frame, and foreground contour.

3.3.2 Camera Calibration

Combining all camera inputs to produce images depicting 3D subjects requires knowledge of camera characteristics and how they are positioned with respect to each other. These parameters describe lens distortion, cameras intrinsics and extrinsics, and color transform matrices. Any and all of these parameterizations may be needed at certain stages for video conference computations.

In Coliseum we use a single uncalibrated target that has proved satisfactory for obtaining all of the needed parameters (see Figure 4). It is a 10-inch cube with four colored squares on each face (24 colors plus black and white). Linear features (sides of squares) provide observations for determining lens distortion, while the shapes of squares themselves provide for estimating the intrinsic parameters. The squares' colors allow each face to be identified and oriented, so the observed colors can be used to determine each camera's color transform, and the corners provide image coordinate observations for determining extrinsic parameters. Waving the target around before the Coliseum rig is sufficient for obtaining the needed parameters.

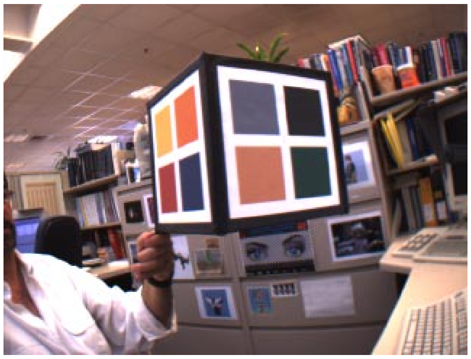

Figure 4. Calibration Target

3.3.3 Model Construction and Rendering

Single-PC Coliseum's 3D display construction differs in several ways from that of multi-PC Coliseum. Rather than approximating ray intersections as the point of closest approach, we derived an exact calculation that requires fewer operations. An interpolating revision to boundary sampling has reduced hull construction costs, and a provision for caching intermediate computations eliminates recalculation while the desired viewpoint is stationary.

Our rendering method remains the same across both platforms. For each point on the suface of the visual hull, we determine the set of cameras for which this point is visible. From this set, we select the camera closest in angular distance to the novel viewpoint, and then color the surface point with data from the selected camera. We have observed artifacts with this approach, however, and are investigating alternative painting methods.

Table 1. Revisions in image analysis strategy

	Multi-PC Implementation	Single-PC Implementation	Motivation
Foreground Discrimination	Foreground/background binarization in QVGA-YUV422 using pixel mean and variance (a background model)	Binarization in VGA Bayer RGGB	o No input-image color conversion before it is known to be needed o No downsampling (enables full VGA texturing) -> Fast
Foreground Segmentation	Edge detection via pixel multi-neighbor testing Edges pooled and binned	Dispatch table with caching Explicit connected contours	o Pixels are touched just once o Foreground is objects with image-based properties -> Fast
Foreground Representation	Binned edge elements Labeled pixels (Foreground/Background)	Connected edge elements enclosing subject	o Select among foreground objects using image properties o Enables tracking objects

3.4 Participant Communication

In Coliseum, a group of users sharing a common virtual environment is called a *session*. Coliseum can support multiple simultaneous sessions. A user belongs to one session, which is created when its first user joins and persists until its last user leaves. Users can come and go during the life of the session.

Session management is performed through the *Hub* subsystem, built using the Microsoft DirectPlay API. A Hub *host* process for each session runs on a central server and processes connections and disconnections. A Session Connection application shows participants a list of active sessions on a management server, enabling the user to join an existing session or initiate a new one. The host process performs the necessary actions, notifying members when other users join or leave.

To avoid the overhead of a central server, communications among users during a session are peer to peer. When a new user connects to a session, the local portion of the Hub subsystem determines compatible media types between itself and other users, and notifies the local and remote media transmission and reception modules. These media modules communicate directly using datagram protocols. A multi-stream UDP protocol allows coordination of different media-type network transmissions.

The user interacts with a 3D VRML virtual environment application augmented by rendered displays of the other participants. Coliseum hosts create IBVH renderings of their participants and transmit these to all participating sites. This MPEG4-compressed video is transmitted over UDP to the receiver, where it is decompressed and composited into the scene. Participants move themselves to change their view of the scene, with other users being informed of their altered positions. Figure 5 shows a still from a three-participant Coliseum conference.

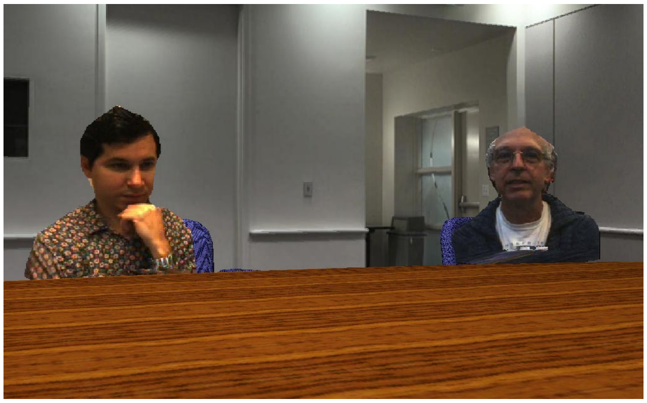

Figure 5. Two users sharing a virtual environment with a third

4. PERFORMANCE

For a two-person conference, our multi-PC IBVH system ran at about 10 Hz on QVGA camera images producing 224 x 224 subject displays, using 6 computers (plus one needed for network arbitration and synchronization) – all 733MHz P3s. The rearchitected single-PC system runs at a similar framerate on a dual-processor 733 MHz P3. Using a dual-procesor 2GHz P4, we have demonstrated two-party operation at over 20 Hz operation on VGA imagery.

Parameters accessible to adjust framerate with respect to processor capability include:

- Foreground sampling density (default is VGA processed as non-overlapping 2x2 Bayer quads)
- Deviation from linearity allowed in piecewise linear approximations (default is 2.0 pixels)
- Desired display image size (default is 300x300)
- Interpolation sampling density in computing the hull representation (default is 4x4)
- Acquisition framerate (default is 15 Hz, ideal is 30Hz, 1394 bus limit is currently about 28 Hz)

In a delivery system, these parameters will be adjusted by the application itself in optimizing performance with respect to processor speed.

5. SUMMARY

We have developed an immersive videoconference system that supports multiuser remote collaboration over networks. Framerate performance has been attained through judicious algorithmic, representational, and architectural system structurings. With Coliseum in place, we are now entering the phase of this work where we will be testing and evaluating a variety of performance, algorithmic, and user interface issues in establishing an effective capability for immersive videoconferencing and remote collaboration. Participating remote sites include HP Labs Bristol and Georgia Institute of Technology.

6. ACKNOWLEDGMENTS

Mike Harville, Ian Robinson, and Greg Slabaugh have also contributed to this project. Wojciech Matusik, Chris Buehler, and Leonard McMillan provided guidance on the original IBVH system.

7. REFERENCES

[Culbertson 99] W.B.Culbertson, T.Malzbender, and G.Slabaugh, “Generalized Voxel Coloring,” *Intl. Wrkshp on Vision Algorithms*, Springer-Verlag LNCS 1883, 100-115 (1999).

[Gaver 93] W. Gaver, A.J. Sellen, C. Heath, P. Luff, “Multiple views in a media space: One is not enough,” *Proc. INTERCHI, ACM Conf. Human Factors in Computing Systems*, Amsterdam, Netherlands, 335-341 (1993).

[Lanier 01] J. Lanier, “Virtually There,” *Scientific American*, April, 66-75 (2001).

[Matusik 00] W. Matusik, C. Buehler, R. Raskar, S. Gortler, L. McMillan, “Image-based Visual Hulls,” *SIGGRAPH*, 369-374 (2000).

[Schreer 01] O. Schreer, N. Brandenburg, S. Askar, E. Trucco, “A Virtual 3D Video-Conferencing System Providing Semi-Immersive Telepresence: A Real-Time Solution in Hardware and Software,” *Proc. Intl. Conf. eWork and eBusiness*, Venice, Italy, 184-190 (2001).

[Seitz 97] S. Seitz, C. Dyer, “Photorealistic Scene Reconstruction by Voxel Coloring,” *Proc. Comp. Vision Pattern Recognition Conf.*, 1067-1073 (1997).