A Framework for Collaborative Real-Time 3D Teleimmersion in a Geographically Distributed Environment

Gregorij Kurillo, Ramanarayan Vasudevan, Edgar Lobaton, and Ruzena Bajcsy
Department of Electrical Engineering and Computer Science
University of California, Berkeley
{gregorij, ramv, lobaton, bajcsy} @ eecs.berkeley.edu

Abstract

In this paper, we present a framework for immersive 3D video conferencing and geographically distributed collaboration. Our multi-camera system performs a full-body 3D reconstruction of users in real time and renders their image in a virtual space allowing remote interaction between users and the virtual environment. The paper features an overview of the technology and algorithms used for calibration, capturing, and reconstruction. We introduce stereo mapping using adaptive triangulation which allows for fast (under 25 ms) and robust real-time 3D reconstruction. The chosen representation of the data provides high compression ratios for transfer to a remote site. The algorithm produces partial 3D meshes, instead of dense point clouds, which are combined on the renderer to create a unified model of the user. We have successfully demonstrated the use of our system in various applications such as remote dancing and immersive Tai Chi learning.

1. Introduction

Teleimmersion is an emerging technology that enables users to collaborate remotely and experience the benefits of a face-to-face meeting. The users must be able to establish eye contact, recognize gestures, and communicate subtly via body language and facial expressions. It should allow users to feel immersed in the environment with their remote collaborators. The teleimmersive technology combines virtual reality for rendering and display purpose, computer vision for image capturing and 3D reconstruction, and various networking techniques for transmitting data between remote sites in real-time with minimal delay.

Though most existing videoconferencing systems make some attempt to humanize remote communication, the majority are unable to provide the immersive component of actual face-to-face communication. These systems rely on two-dimensional video streams between remote users and present these streams on several displays. Some attempts have been made to create a more immersive experience using large displays, gaze preservation through multi-camera capturing [18], and matching environments (e.g., tables, chairs) between the remote locations which create the illusion of continuity of the physical space into the screen. The experience of telepresence can be further enhanced using virtual reality where the remote users are rendered inside a shared virtual environment. Virtual meeting space allows for the possibility of collaborative work on 3D data such as medical data (MRI, CT scan, 4D ultrasound, 3D x-ray), scientific data, design models (e.g., CAD models, building designs), remote training (e.g., oil rigs, military applications), remote teaching of physical activities (e.g., rehabilitation, dance), and many others.

A critical aspect of the teleimmersive experience is the realistic representation of users inside the virtual space. Immersive virtual realities often employ avatars, to represent the human user inside the computer generated environment [9, 23]. An avatar is often designed as a simplified version of the user's physical features, while its movements are controlled via tracking of the user in the real world. Markers (e.g., body suit with tracking devices) which may interfere with users movement are generally employed to perform this tracking task. The actions of the avatar in the virtual environment are generally simulated through complex dynamic equations [9]. Therefore, using a limited number of markers, restricts the information available to reconstruct the dynamics of movement. Due to the limitations of the model, the appearance, and the movement ability, the overall interactive experience becomes highly constrained. In contrast to avatars, a full body 3D reconstruction can realistically represent the user's appearance and full dynamics of movement, such as facial expressions, chest deformation during breathing, and movement of hair or clothing.

In this paper, we present a multi-camera system based on dense stereo mapping that allows real-time 360° full-body 3D reconstruction. The data can be displayed locally

978-0-7695-3454-1/08 $25.00 © 2008 IEEE
DOI 10.1109/ISM.2008.32

or transferred to a remote site for display and interaction. In the past the presented framework has been successfully used in remote dancing applications [17], learning of Tai Chi movements [2, 19], and remote manipulation of virtual objects between two users [12]. The main focus of this paper is to describe the recent significant improvements in the reconstruction quality, speed, compression, and rendering of 3D data.

2. Related Work

Several attempts have been made in the past to develop real-time 3D reconstruction systems to capture the human body. Most real-time approaches fall into one of three categories based on their computational approach: (1) silhouette-based reconstruction, (2) voxel-based methods with space sampling, and (3) image-based reconstruction with dense stereo depth-maps. In the silhouette-based reconstruction [6], 3D information is obtained via visual hulls that are formed by intersecting generalized cones between a silhouette and the camera center. The voxel-based method [6], on the other hand, determines depth by sampling a uniform grid of space using color consistency. Finally, the vision-based reconstruction [20, 8] creates dense stereo depth-maps by correlating slightly displaced views of the same scene.

One of the first teleimmersive systems was presented by the researchers at University of Pennsylvania [16]. Their system consisted of several stereo camera triplets used for the image-based reconstruction of the upper body, which allowed a local user to communicate to remote users while sitting behind a desk. A simplified version of the desktop teleimmersive system based on the reconstruction from silhouettes was proposed by Baker et al. [3] who used five different views to obtain 3D model of the user. Kanade et al. [10] used large number of cameras distributed around the room to capture full-body movement in real time. Blue-c teleimmersion system, presented by Gross et al. [5], allowed full-body reconstruction based on silhouettes obtained by several cameras arranged around the user. The user was located in a cave-like environment with cameras recording images through active projection panels. Reconstruction from the silhouettes provides faster stereo reconstruction as compared to the image-based methods; however, this approach is limited in accuracy, ability to reconstruct concave surfaces, and discrimination of occlusions by objects or several persons inside the viewing area. Voxel-based interactive virtual reality system presented by Hasenfratz et al. [6] featured fast (25 fps) real-time reconstruction of the human body on a 1.6 Ghz Pentium 4 processor but were limited in acquiring details such as clothing and facial features.

In this paper, we address some of the drawbacks of the image-based teleimmersion system presented by Jung and Bajcsy [8]. Their system is based on trinocular 3D reconstruction using a region-based stereo algorithm running real-time reconstruction on a background subtracted image at 5 to 7 frames per second (FPS). Their proposed method has two shortcomings, its slow speed and its unreliability in homogeneous regions resulting in holes. For example, if we look at Figure 1, we see a sample image (top left), and the result of the algorithm proposed by Jung (bottom left) which took over 600 milliseconds to compute on a two 3 Ghz Xeon processors and the result of our algorithm (bottom right) which took less than 40 milliseconds to converge on the same machine. Lighter gray corresponds to nearby objects, darker gray corresponds to more distant objects, and black corresponds to areas of uncertainty. Note the hole inside of the person in Jung and Bajcsy's result. Our results are due to a representation (top right of Figure 1) that we introduce in this paper which produces a partial 3D mesh instead of dense point clouds, and an improved camera calibration scheme which is not only more accurate, but also much faster. We will describe how this representation allows for real-time stereo reconstruction, real-time interpolation, and efficient compression.

This paper is organized as follows: first we describe our overall system, second we describe our hardware setup in detail, and then our software setup in detail. Finally, we describe our overall system performance and its applications.

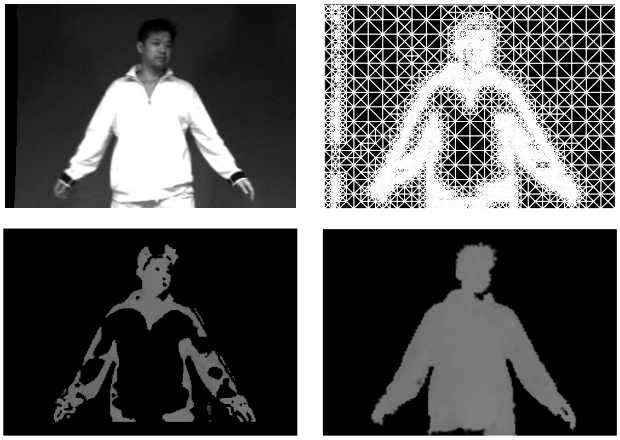

Figure 1: Comparison of the results for disparity map calculation from the algorithm proposed by [8] (bottom left) and the one proposed in this paper (bottom right). Base image (top left) and triangulation used for our reconstruction (top right) are also shown.

3. System Overview

The purpose of our system is to allow for collaboration between geographically distributed users by creating

a shared virtual environment. All users interact in the system via their local stations (Figure 2). Each system owns a virtual representation of the locally reconstructed 3D object. However, in order to properly model interaction between objects in the shared virtual environment and allow for flexibility in the visualization of these objects, each station must maintain a local copy of the entire virtual space. Model manipulation and post-processing of data can then be performed locally. With these requirements in mind, each station must perform the following three tasks: compute a 3D reconstruction of local objects, communicate these objects to other stations, and visualize the virtual environment.

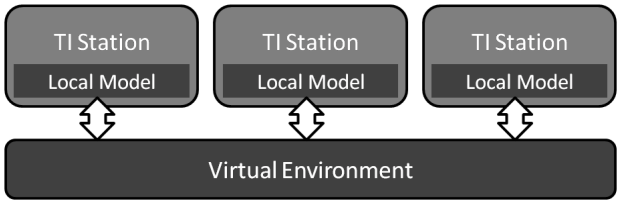

Figure 2: Diagram of the system depicting the local model maintained by each station and the shared virtual environment.

An appropriate choice of representation for the data should take into consideration the tasks mentioned above. The system should be able to reconstruct any object that is present at each station. Hence, no priors on the physical models can be assumed (i.e. no avatars to be fitted). Since we require real-time streaming of 3D data, we must choose a representation that allows for fast and efficient compression. Though each station creates a separate 3D model of an object based on views from multiple camera clusters (see section 4), in order to visualize the objects we need to integrate these views which can be expensive unless the representation allows for straightforward integration.

With these requirements in mind, it would be ideal to represent an object by a triangulation of its surface. Constructing a triangulation of an object in real-time can be a challenging task; nevertheless, in this paper we will describe how to build a triangulation rapidly, how this triangulation allows us to build dense 3D depth maps, and how we can use the triangulation to perform efficient data compression and consequently efficient reconstruction of the data packet all in real-time. Note we will not talk about how this triangulation will allow for improved visualization. However, after obtaining triangulations of the object corresponding to views from each camera cluster (see left plot in Figure 3), we can then integrate all of the local triangulations to construct a consistent 3D model in real-time (see right plot in Figure 3).

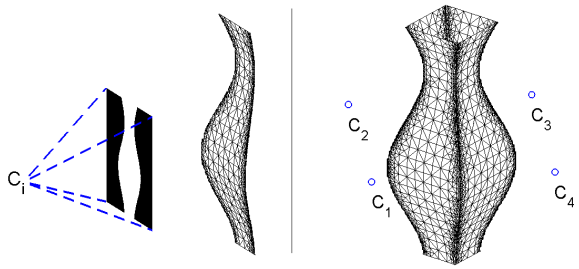

Figure 3: (Left) Triangulation corresponding to a single camera cluster. View information from camera cluster C_i is used to build a partial reconstruction of the 3D object. (Right) Integrated representation from multiple views is obtained combining partial reconstructions.

4. Hardware Overview

In this section, we present different components of our teleimmersion system (Figure 4).

4.1. System Configuration and Cameras

The teleimmersion apparatus consists of 48 Dragonfly cameras (Point Grey Research Inc, Vancouver, Canada), with the resolution of 640 $\times$ 480 pixels, which are arranged in twelve clusters. Each cluster contains three grayscale cameras for stereo reconstruction and a color camera for texture acquisition. The cameras, equipped with 6 and 3.8 mm lenses, are mounted on an aluminum frame. The clusters cover a 360° view of the user. The dimensions of the workspace are about 2.0 x 2.0 x 2.5 m^3. The cameras of each cluster are connected through IEEE 1394a interface to a dedicated server which performs image acquisition and stereo reconstruction. The bandwidth of the PCI bus limits the frame rate of the image acquisition to 20 frames per second. The server computers used for the reconstruction have two dual core Intel Xeon 2.33 Ghz, 2 GB of memory and 1 Gbps connection to Internet 2.

4.2. Synchronization

To achieve consistent stereo reconstruction the images captured by the cameras have to be precisely synchronized. The synchronization of the cameras is done through external triggering on the cameras. The camera triggering pins are connected to the parallel port of the triggering server which generates a signal to initiate image capturing while receiving TCP/IP messages from the cluster computers to notify the server once the reconstruction has been completed.

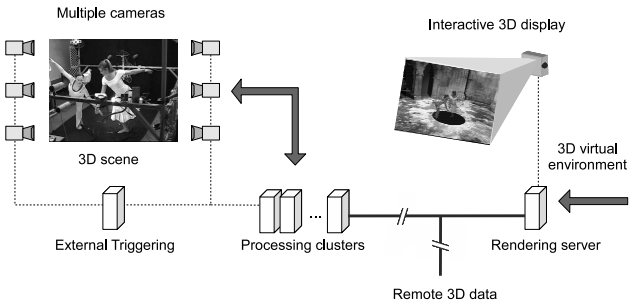

Figure 4: Components of the teleimmersion system. Capturing component is displayed to the left. Data is then processed using a computing cluster. The model is transmitted over the network for display at another teleimmersive station.

4.3. Illumination

The illumination of the workspace is accomplished by eight 55 W light fixtures with diffusive filters. Six lights are mounted from the top to illuminate the subject from above and two are located on the floor illuminating the users's lower part of the body. This arrangement of the lights minimizes shadows which can interfere with background subtraction and 3D reconstruction.

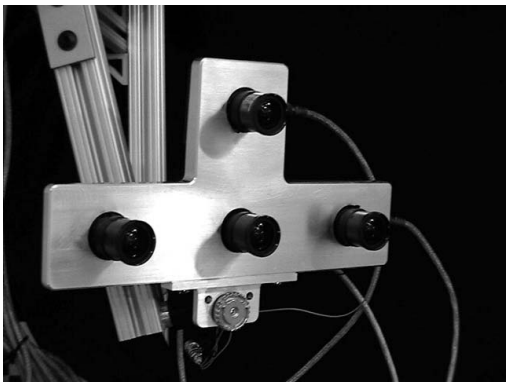

Figure 5: Stereo cluster with three grayscale cameras and a color camera on top of the fixture.

4.4. Display

The display system consist of a rendering computer with Intel Dual Core CPU, 2.66 Ghz, 2 GB of memory, and two NVIDIA GeForce 7900 GTX graphics cards. The renderer can receive compressed 3D data directly from the cluster computers in separate network streams or indirectly through a gateway computer which can also handle connections from a remote site. The renderer can output the data to a passive stereo display system consisting of two projectors with filters for circular polarization or/and other displays showing different view points of the virtual space. The users can use a wireless 3D mouse to interact with the graphic user interface designed to control the virtual environment.

5. Software Architecture

In this section we will present the algorithms for the calibration of the multi-camera system and for the real-time 3D reconstruction. We will compare the implemented software architecture with the initial design of the teleimmersion system [8].

5.1. Camera Calibration

The accuracy of 3D reconstruction greatly depends on the quality of the camera calibration. The erroneous calibration of the stereo cluster results in inadequate reconstruction of the depth while inaccurate calibration of the cluster positions results in imprecise overlap of the partial 3D meshes obtained from different views. With this in mind, we approach the calibration using two-steps.

In the first step of the calibration, all cameras are internally calibrated using well-known Tsai algorithm [22]. A planar checkerboard target is placed in different positions and orientations to generate a set of points for homography calculation. Initial guess of the internal parameters (i.e. focal length, optical center and distortion) is optimized using Levenberg-Marquardt algorithm [14]. In the second step, external calibration is performed to determine position and orientation of each cluster with regard to a reference camera.

In our calibration method we combine the idea of vision graphs to calibrate multi-camera systems with small overlap using virtual calibration object. Our algorithm requires cameras to share workspace volume at least pairwise. In contrast to other methods [4, 7, 21] our approach resolves Euclidean reconstruction (preserving metric information) and introduces novel parameters reduction in the case of two-point bar calibration.

Our external calibration algorithm can be summarized as follows [13]:

(a) image acquisition and sub-pixel marker detection on multiple cameras
(b) composition of adjacency matrix for weighted vision graph describing interconnections between the cameras (e.g. number of common points)
(c) computation of fundamental $\mathbf{F}$ and essential matrix $\mathbf{E}$ with RANSAC
(d) essential matrix decomposition into rotation and translation parameters defined up to a scale factor λ

(e) determination of the scale factor λ through triangulation and LM optimization
(f) optimal path search through the graph
(g) global optimization of the parameters using sparse bundle adjustment

Both calibration steps were implemented using C++ and OpenCV [1] computer vision library to allow fully automatic and fast calibration.

The cameras in each cluster were first internally calibrated using the checkerboard with 10×15 number of squares with 40 mm in size. The checkerboard was placed in 20 different positions and orientations. Automatic corner detection was implemented along with Tsai calibration method to obtain the intrinsic parameters for each camera and the geometric position and orientation of the camera inside the cluster.

For external calibration we used a rigid metal bar with two LED markers attached on each end. We chose Luxeon I LED (Philips Lumileds Lighting Company, San Jose, CA), with the brightness of 30.6 lm and 160° emitting angle. The distance between the markers was measured at 317 mm using a tape measure before the calibration. The marker locations were extracted in real time on each cluster and stored locally.

The complete external calibration of 12 cameras and 4738 collected 3D points, took 14 seconds on a personal computer with Intel Xeon 3.20 GHz processor and 1 GB of memory. The mean reprojection error between all the cameras was 0.3391 pixels with the standard deviation of 0.0365 pixels. Figure 6 shows the results of the external calibration where for clarity only the central camera of each cluster is presented. Camera #3 was chosen as the reference camera since its position and orientation correspond with the floor level.

The hierarchical approach allows more robust and flexible calibration as compared to simmultaneous external calibration of all cameras [21]. Since the cameras within the cluster are located close to each other (approximately 110 mm), the homography based method is more appropriate as compared to the fundamental matrix approach. On the other hand, the clusters are positioned more sparsely, therefore essential matrix (obtained from fundamental matrix) decompositions will yield more accurate results for geometric calibration of the clusters.

5.2. Reconstruction Algorithm

In this section, we describe how we represent the data to build rapid and accurate 3D depth maps. We consider a partition of the domain called Maubach's bisection scheme [15]. Decomposition begins with a coarse triangulation of the domain using right isoceles triangles (see right plot of

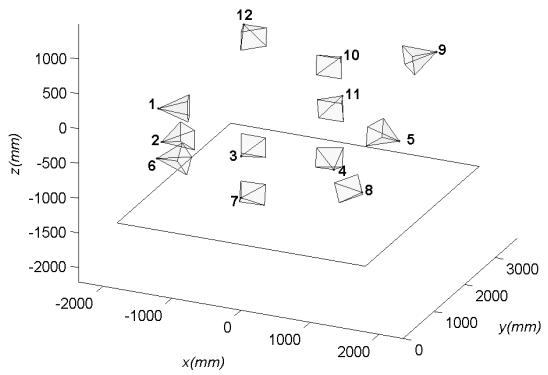

Figure 6: Three-dimensional layout of 12 stereo clusters after the calibration.

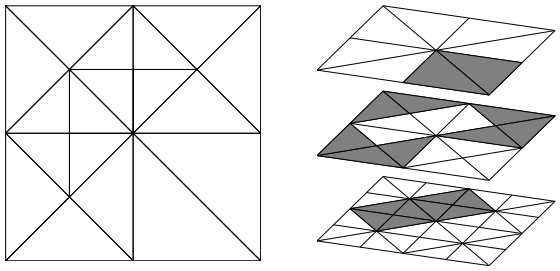

Figure 7: Illustration of Maubach's bisection scheme: sample triangulation (left), multi-resolution structure of triangulation at different levels of refinement (right). Highlighted triangular elements are the ones selected for sample triangulation at corresponding levels.

Figure 7). Next, each triangle is refined via a simple bisection step. Refinement occurs when a given criterion goes unsatisfied in a particular region. In our system, this criterion is the variance of the grayscale image in each triangle. Although this scheme would suffice in some sense, Maubach introduces an additional constraint: he requires that there be no floating nodes (i.e. no nodes in the middle of a triangle's edge). This sort of triangulation is referred to as conforming and it allows for the application of various Finite Element Methods [11].

This representation is advantageous in our system for three reasons. First, it reduces the dense stereo depth-map calculation from working pixel-by-pixel to working region-by-region. Second, since the correspondence part of the depth-map calculation can have difficulty in finding matches it is important to be able to interpolate quickly in order to get a dense depth-map. Finally, the triangulation

can be easily compressed as will be described in section 5.3.

By reducing the number of points upon which we must perform the correspondence calculation we can speed up the stereo calculation. Having a fast interpolation function is crucial in our setup because we perform stereo correspondence measurements on grayscale images which tend to reduce the reliability of the correspondence measurement when compared to performing the same calculations on color images. If we look at Figure 8, we see two sample images from the system in the top row, and disparity results at the bottom. For the disparity values, lighter gray indicates portions of the image that are close and darker gray indicates portions of the image that are further away. In the bottom left, we see the results right after the correspondence calculation on the triangulation. Notice the numerous holes in the calculation, but a region growing approach fills in these holes as we see in the bottom right image.

The speed of the various components of our system can be found in table 1, which compares our algorithm to Jung's algorithm [8]. Both algorithms use a triangulation sheme to reduce computation load. Our version uses Maubach's bisection scheme while Jung's uses Delaunay triangulation. Both algorithms also use the same correlation scheme to determine the disparity correspondences. Finally, our algorithm performs a post-processing step to fill-in holes (as described previously). An example of the rendered results can be found in Figure 10.

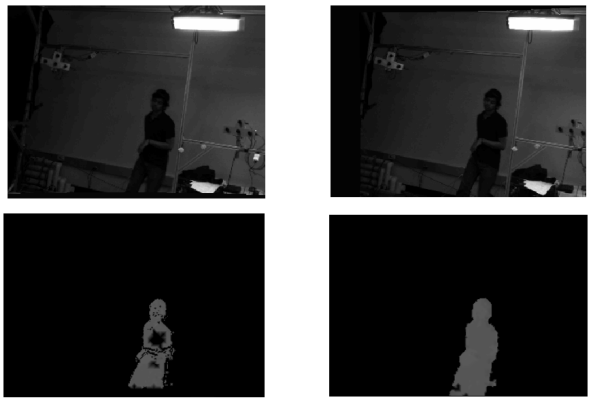

Figure 8: Images in top row are examples of a stereo pair from the system. The bottom left image is the result straight from the disparity computation after background subtraction, and the bottom right image is the result after the diffusion step.

5.3. Communication Protocols

Sharing the models between different stations requires transmitting the triangulation models in real time. A standard format for transmitting triangulation information consists of transmitting nodal values, and then specifying the triangles based on the node indices. In particular, considering RGB values (3 bytes) and disparity values (2 bytes) per pixel, we obtain a factor of 5 bytes per node. Each triangle needs to specify 3 vertices and each vertex needs at least 3 bytes (since there are more than 2^{16} possible nodes in the triangulation). Hence, there is a total of 9 bytes per triangle. This gives:

$$\begin{aligned} \text{Std. Package Size} \quad = \quad & (\text{No. of Nodes}) \times 5 \\ & +(\text{No. of Triangles}) \times 9. \end{aligned} \tag{1}$$

The first term is the tranmission cost for the values at each node, and the second term is the cost due to the triangulation structure.

Since our triangulation results from a bisection scheme, it is possible to specify how the triangulation was obtained instead of specifying the actual triangulation. That is, we can specify which triangles are bisected. This encoding scheme for the triangulation yields larger gains given enough triangles in the representation. No additional time for encoding a triangulation is required since the information for encoding the triangulation is generated at the same time as the triangulation is computed. Also the decoding of the instructions (i.e. the time for recreating the triangulation) is less that 1 ms for the images that we are considering.

The sizes of the packages generated over an image sequence after background subtraction are illustrated in Figure 9. For a typical image in the sequence see Figure 8. In the sequence used for Figure 9, no object is present in the field of view of the camera for the first 7 frames during which our compressed package consists of the triangulation information but no active nodes. Thereafter, an individual is present in the field of view of the camera. We observe an overall reduction in the package size between standard and proposed formats by a factor close to 3. We note that it is still possible to compress the transmitted node values. However, this has not been implemented in our protocol since off-line compression of the packages did not give a significant reduction in the package size.

Process	Jung's [Old]	Ours [Old]	Ours [New]
Triang.	110 ms	7.55 ms	3.83 ms
Disparity	240 ms	28.79 ms	15 ms
Post-Proc.	N/A	3.27 ms	1.78 ms
Total	350 ms	39.61 ms	21.39 ms
Rate	2.85 fps	25.25 fps	46.75 fps

Table 1: Comparison between algorithm used by S. Jung [8] and the one proposed in this paper. [Old] refers to the use of the hardware setup proposed by S. Jung (two 3 GHz Xeon Processors) and [New] refers to our setup as described in section 4. The images in Figure 1 are used for this comparison.

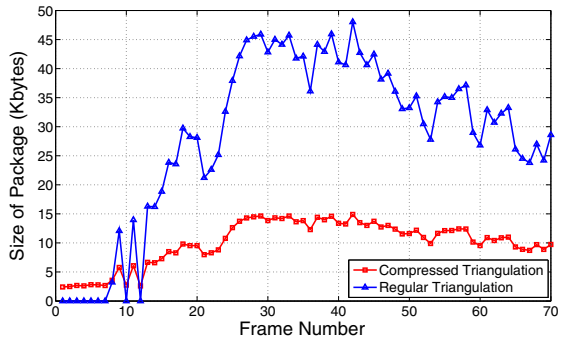

Figure 9: Comparison of package size between standard and our encoded format. Packages include triangulation information, RGB values (3 bytes per node), and disparity values (2 bytes per node).

From Figure 9 each frame is less than 15 Kbytes. For 12 clusters we have 180 Kbytes of data per frame. The proposed approach compresses data one frame at the time. A more significant compression can also be obtained by considering compressing over time (i.e. video compression). In this case, we can transmit updates between frames instead of transmitting the whole triangulation.

6. System Performance and Applications

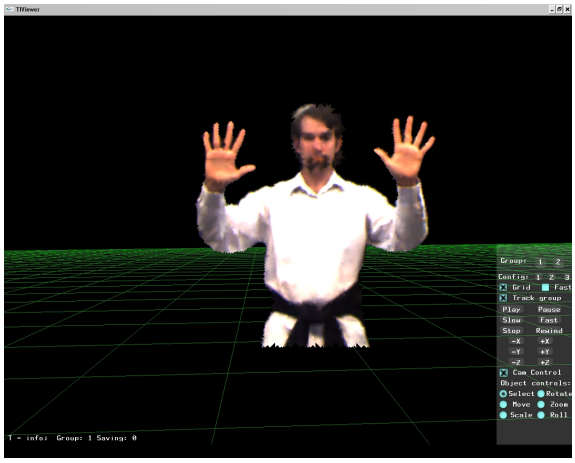

Figure 10: A sample rendering from a camera cluster.

An example of the rendering can be found in Figure 10. In the past we have performed a study of teaching Tai Chi using immersive virtual reality and compared it to learning from 2D video. We have captured three moves performed by a Tai Chi teacher. The teacher's 3D recording was projected into the virtual space simultaneously with the real-time data from a student. The students saw themselves on the screen with the teacher. In the study [2, 19] we have demonstrated that immersive virtual reality provides better learning of physical movements than a two-dimensional video. Similar approach could also be used for rehabilitation where a therapist's movement were recorded or streamed live while a remote patient would try to perform the exercises.

Projection of dislocated digitized users into the same virtual space in real time offers variety of use in art and dance. Over the past two years we have conducted several local and remote experiments with dancers [17, 24]. Image based 3D reconstruction does not require the dancers to wear any markers or body suits allowing them to freely move in the space. The system can capture dynamics of hair and clothing without relaying on models. The dancers had to accommodate to technical limitations of the system, such as frame rate and transmission delay. At the same time they introduced new techniques to take advantage of the technology, such as digital transformations (e.g., rotations), disappearing (e.g., moving beyond stereo range), dissolving into 'particles' (e.g., moving too close to the camera). Teleimmersion environment also introduced the novel concept of 'virtual touch' where the feedback for touch relays on visual information from the virtual environment. Movies and images are available at: `http://tele-immersion.citris-uc.org`

7. Conclusions and Future Work

In this paper we have addressed the major bottlenecks of the teleimmersion system described in [8], such as the reconstruction speed, noise, and data compression. We have significantly improved the frame rate of the stereo reconstruction by implementing the triangulation-based stereo algorithm. The algorithm also improves stereo reconstruction in homogeneous regions while preserving detailed information in textured areas. The bisection scheme used for the triangulation allows transfer of the structure instead of data for each triangle node or pixel providing much greater compression for data transmission over the network.

Our new hierarchical approach to calibration sped up the calibration process from several hours to several minutes. The system can be easily re-calibrated when changing the cluster positions using marker based calibration without the need to perform internal and external calibration on all cameras. This is an important future step for transferring the technology from controlled environments into real world scenarios.

Our initial research work [17] on the application level has been focused on artists as users of the teleimmersive technology. The virtual environment provides different digital effects (e.g., transformations and deformations of rendered images) that can be applied in real-time to manipulate what is displayed to the audience. The presented system

offers new possibilities for learning and training individuals to perform physical movements (e.g., dancing, physical therapy, and exercise). 3D data captured by our system can be used to analyze subject's movement. Extracted kinematic data can be applied as a feedback to the user during learning of motions. The users can be immersed inside computer generated existing or non-existing environments, such as ancient buildings and future architectural designs to allow interactive exploration. The system could also be combined with head-mounted display and head tracking to provide higher level of immersion. Finally, there are many applications of social networking and entertainment where the users could interact in real-time inside a common virtual environment, such as games supporting physical interaction, interactive music video, and 3D karaoke.

References

[1] Opencv: Open computer vision library. [web page] http://sourceforge.net/projects/opencvlibrary, November 2006.

[2] J. N. Bailenson, K. Patel, A. Nielsen, R. Bajcsy, S. Jung, and G. Kurillo. The effect of interactivity on learning physical actions in virtual reality. *Media Psychology*, page In Press, 2008.

[3] H. Baker, D. Tanguay, I. Sobel, D. Gelb, M. Gross, W. Culbertson, and T. Malzenbender. The coliseum immersive teleconferencing system. In *Proceedings of International Workshop on Immersive Telepresence, Juan-les-Pins, France*, 2002.

[4] X. Cheng, J. Davis, and P. Slusallek. Wide area camera calibration using virtual calibration objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, 2000.

[5] M. Gross, S. Würmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. V. Gool, S. Lang, K. Strehlke, A. V. Moere, and O. Staadt. blue-c: a spatially immersive display and 3d video portal for telepresence. *ACM Trans. Graph.*, 22(3):819–827, 2003.

[6] J. Hasenfratz, M. Lapierre, and F. Sillion. A real-time system for full-body interaction with virtual worlds. In *Proceedings of Eurographics Symposium on Virtual Environments*, pages 147–156. The Eurographics Association, 2004.

[7] I. Ihrke, L. Ahrenberg, and M. Magnor. External camera calibration for synchronized multi-video systems. In *Proceedings of 12th International Conference on Computer Graphics, Visualization and Computer Vision 2004*, volume 12, pages 537–544, Plzen, Czech Republic, February 2004.

[8] S. Jung and R. Bajcsy. A framework for constructing real-time immersive environments for training physical activities. *Journal of Multimedia*, 1(7):9–17, 2006.

[9] P. Kalra, N. Magnenat-Thalman, L. Moccozet, G. Sannier, A. Aubel, and D. Thalman. Real-time animation of realistic virtual humans. *IEEE Computer Graphics and Applications*, 18(25):42–56, 1998.

[10] T. Kanade, P. Rander, S. Vedula, and H. Saito. *Mixed Reality, Merging Real and Virtual Worlds*, chapter Virtualized reality: digitizing a 3D time varying event as is and in real time, pages 41–57. SpringerVerlag, 1999.

[11] H. Kirchner and H. Niemann. Finite element method for determination of optical flow. *Pattern Recognition Letters*, 13(2):131–141, 1992.

[12] G. Kurillo, R. Bajcsy, K. Nahrsted, and O. Kreylos. Immersive 3d environment for remote collaboration and training of physical activities. In *Proceedings of IEEE Virtual Reality Conference (VR 2008)*, pages 269–270, Reno, NV, March 8-12, 2008 2008. Accepted for Poster Presentation.

[13] G. Kurillo, Z. Li, and R. Bajcsy. Wide-area external multi-camera calibration using vision graphs and virtual calibration object. In *Proceedings of 2nd ACM/IEEE International Conference on Distributed Smart Cameras, Stanford University, USA*, In Press, 2008.

[14] M. Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++. [web page] http://www.ics.forth.gr/ lourakis/levmar, July 2004.

[15] J. Maubach. Local Bisection Refinement for N-Simplicial Grids Generated by Reflection. *SIAM Journal on Scientific Computing*, 16:210, 1995.

[16] J. Mulligan and K. Daniilidis. Real time trinocular stereo for tele-immersion. In *Proceedings of 2001 International Conference on Image Processing, Thessaloniki, Greece*, pages 959–962, 2001.

[17] K. Nahrstedt, R. Bajcsy, L. Wymore, G. Kurillo, K. Mezur, R. Sheppard, Z. Yang, and W. Wu. Symbiosis of tele-immersive environments with creative choreography. In *ACM Workshop on Supporting Creative Acts Beyond Dissemination, Associated with 6th ACM Creativity and Cognition Conference*, Washington D.C., June 13-15 2007.

[18] D. Nguyen and J. Canny. Multiview: spatially faithful group video conferencing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 799–808. ACM, New York, NY, 2005.

[19] K. Patel, J. N. Bailenson, S. Hack-Jung, R. Diankov, and R. Bajcsy. The effects of fully immersive virtual reality on the learning of physical tasks. In *Proceedings of the 9th Annual International Workshop on Presence, Ohio, USA*, pages 87–94, 2006.

[20] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002.

[21] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multi-camera self-calibration for virtual environments. *Presence*, 14(4):407–422, 2005.

[22] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA3(4):323–344, 1987.

[23] Y. Yang, X. Wang, and J. X. Chen. Rendering avatars in virtual reality: integrating a 3d model with 2d images. *Computing in Science and Engineering*, 4(1):86–91, 2002.

[24] Z. Yang, W. Wu, K. Nahrstedt, G. Kurillo, and R. Bajscy. Viewcast: view dissemination and management for multi-party 3d tele-immersive environments. In *Proceedings of ACM Multimedia, Augsburg, Germany*, pages 882–891, Sept 2007.