REAL-TIME STEREO-VISION SYSTEM FOR 3D TELEIMMERSIVE COLLABORATION

Ram Vasudevan , Zhong Zhou, Gregorij Kurillo, Edgar Lobaton, Ruzena Bajcsy, Klara Nahrstedt***

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
{ramv,gregorij,lobaton,bajcsy}@eecs.berkeley.edu

*Laboratory of Virtual Reality Technology and Systems
Beihang University
zz@vrlab.buaa.edu.cn

**Department of Computer Science
University of Illinois, Urbana Champagne
klara@cs.uiuc.edu

ABSTRACT

Though the variety of desktop real time stereo vision systems has grown considerably in the past several years, few make any verifiable claims about the accuracy of the algorithms used to construct 3D data or describe how the data generated by such systems, which is large in size, can be effectively distributed. In this paper, we describe a system that creates an accurate (on the order of a centimeter), 3D reconstruction of an environment in real time (under 30 ms) that also allows for remote interaction between users. This paper addresses how to reconstruct, compress, and visualize the 3D environment. In contrast to most commercial desktop real time stereo vision systems our algorithm produces 3D meshes instead of dense point clouds, which we show allows for better quality visualizations. The chosen representation of the data also allows for high compression ratios for transfer to remote sites. We demonstrate the accuracy and speed of our results on a variety of benchmarks.

Keywords— 3D video, compression, real time, teleimmersion

1. INTRODUCTION

3D display technology has improved considerably both in quality and popularity in recent years. Unfortunately, the development of technology to generate 3D content has lagged behind the development of such displays. Most of the content we now enjoy on stereo displays is either generated off line as in 3D movies or is synthetically generated as in video games. Accurate real-time generation of 3D data from real-life scenes has proved extremely difficult.

Approaches to real-time 3D content generation can be divided into two categories, those with active and those passive sensors. The active sensors incorporate laser or infrared devices as in time-of-flight cameras like the ZCam or Canesta [1].

This work was sponsored by the National Science Foundation under Grants 0703787, 0724681, and 0937060.

Edgar Lobaton is now with Department of Computer Science, University of North Carolina at Chapel Hill.

Fig. 1. Two users interacting with a virtual car model each captured with a different stereo camera and rendered inside a shared virtual environment using the stereo vision system presented in this paper.

However, such devices have significant shortcomings, such as low resolution, limited range, high noise, and albedo sensitivity [2]. Passive sensors, generally cameras, observe existing electromagnetic information and use that information to infer about the 3D world. Approaches to extract 3D content from cameras usually take three forms: visual-hull extraction, volumetric reconstruction, or image-based reconstructions. Of these approaches, the image-based one, where 3D information is extracted by comparing rectified images, can achieve much higher accuracy with less noise (see [3] a more extensive review of these three approaches). Though image-based stereo algorithms have been studied extensively (see [4] for a review of such algorithms), they have struggled to simultaneously achieve accuracy and real-time performance. We employ the image based approach and overcome its associated difficulties by proposing

978-1-4244-7493-6/10/$26.00 ©2010 IEEE

a real-time region-based algorithm via a multi-scale scene representation. In addition to achieving high accuracy and speed, this representation of the data allows for better visualization via texture mapping and high compression ratios for transfer of the data to remote sites.

In this paper, we present a real-time portable stereo vision system that creates accurate 3D reconstruction of users via a mesh with high-resolution dynamic texture mapping. We make two major contributions: first, a novel multi-scale representation that allows for the highly accurate reconstruction of a scene which is described in Section 4; second, a real-time texture compression and decompression technique that allows for high-quality visualization which is described in Section 5. The rest of the paper includes a brief overview of related work in Section 2 and an overall description of the various system components in Section 3. An example of the results achieved by our system presented in this paper can be found in Figure 1.

2. RELATED WORK

In this section, we briefly review related research efforts in generating real-time 3D video for desktop teleimmersion. The first such teleimmersive system was presented by researchers at the University of Pennsylvania [5] who used several stereo camera triplets for image-based reconstruction of the upper body. A local user was able to communicate to a remote user while preserving gaze. Another desktop teleimmersive system based on reconstruction from silhouettes was proposed by Baker et al. who used five different views to obtain a 3D model of the user via a visual-hull approach [6]. The system employed a single PC which performed 3D reconstruction and rendering of the users in a simple virtual meeting room. The compact system showed limited accuracy and speed. A similar system was proposed by Kauff and Schreer [7] who obtained the 3D video data by merging depth maps generated by multi-baseline algorithm from four views. The system featured a custom-built multiprocessor board. To generate a arbitrary virtual view of the remote user in virtual environment, view synthesis by 3D warping was utilized. The later approach was extended and presented by Schreer et al.[8] as part of a multi-user 3D conferencing system. Their algorithm combined volumetric reconstruction with depth estimation to balance the low accuracy. The system, however, required a large number of cameras to generate a 3D model of the user.

In this paper, we describe a desktop teleimmersive system that employs image-based reconstruction on a stereo pair to get accurate results in real-time. We combine local and global approaches for disparity calculation in a novel way to generate an accurate 3D mesh and apply high-resolution texture mapping to improve the final visual quality. We compare our approach against various benchmarks [9, 10] to illustrate the strength of our method. We also develop a compact representation of the texture information, the Border-Descriptor Inter-Frame Compression (BIFC) scheme, to achieve real-time performance with high compression ratios.

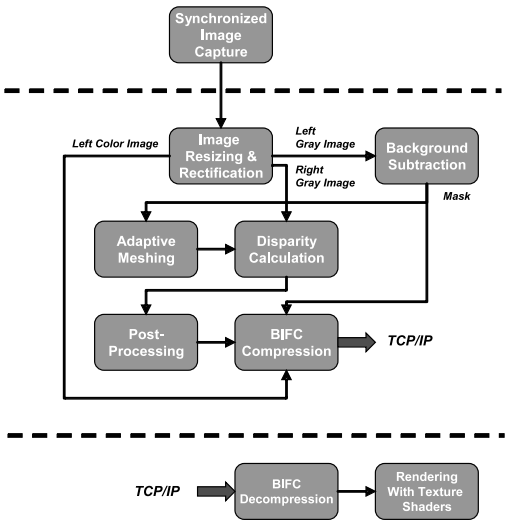


Fig. 2. Flow chart describing the algorithmic pipeline of the desktop system.

3. SYSTEM OVERVIEW

In this section, we provide an overview of the algorithmic pipeline of our platform which allows for collaboration between geographically distributed users by seamlessly integrating their 3D representation and virtual objects within a shared virtual environment. All users interact in the virtual environment via their local stations. In order to properly model interaction between objects in the shared virtual environment and allow for flexibility during visualization, each station maintains 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 three tasks: compute a 3D reconstruction of the local environment, communicate this 3D data to other stations, and visualize the virtual environment.

In our system, we employ a Bumblebee 2 camera (1024x768 resolution) developed by Point Grey, Inc., whose internal and external parameters are calibrated prior to use [11]. The images are resized to 320x240 and rectified. The 3D reconstruction is performed on these rectified resized images, and the dynamic texture is applied via the high resolution images. As we show in the next few sections, this decision does not have a detrimental effect on the accuracy of the reconstruction while guaranteeing high quality visualizations. Next, one of the images to be used for reconstruction is background subtracted and then meshed. Using this mesh, a 3D reconstruction is computed. This data is then post-processed to improve the accuracy. Finally the original color image is compressed using our compression technique which employs motion residuals for inter-frame compression. This package is then sent to the remote location where it is decompressed in real-time and passed to the rendering loop, which visualizes the depth information and texture maps the decompressed data. Figure 2 illustrates this algorithmic pipeline.

4. STEREO ALGORITHM

In this section, we describe a 3D reconstruction algorithm that first segments an image into regions and matches these regions rather than perform pixel by pixel matching which is generally inaccurate. This segmentation of the image has two benefits when compared to the traditional pixel matching approach. First, the segmentation into regions is done according to a criterion that complements the matching step. Namely, the matching step has the benefit of the knowledge of the scale at which to perform matching, which is unavailable while performing pixel by pixel matching. Second, the segmentation is done in a fashion to allow straightforward information sharing between pairs of segments to improve the overall accuracy of the initial estimate of the depth returned by the matching. After describing our algorithm, we compare the performance of our algorithm with several algorithms on a traditional benchmark.

4.1. Construction of the Representation

We begin by decomposing the image domain into a coarse representation of right isosceles triangle bases functions of a fixed largest possible size. Each triangle is then bisected if the variance of the gray scale image within each triangular region is higher than a given threshold. This type of segmentation is referred to as Maubach's bisection scheme [12].

In addition to this bisecting scheme, we introduce an additional constraint to allow for the application of a variety of standard post processing techniques: we require that there be no nodes in the middle of a triangle's edge. If after the bisection scheme such a node exists, then we bisect the triangle with the offending node at its triangle's edge, which ensures that the triangles in question satisfy our required criterion. This type of mesh is referred to as conforming and aids in the development of algorithms to quickly post process the depth maps created by our reconstruction algorithm.

4.2. Calculation of Depth

After the construction of this representation of the image, we can calculate the depth at the nodes of the triangles by employing a normalized cross correlation technique. Since window-based stereo aggregation methods, like cross correlation, implicitly assume that all pixels within the window have similar disparities, they struggle whenever windows straddle depth discontinuities. This results in the infamous foreground fattening effect. Fortunately, the aforementioned image partitioning scheme provides the necessary information to overcome this difficulty.

We assume depth varies smoothly within any image segment with homogeneous color. Fortunately, our mesh employs an identical assumption during its construction. The size of the image segment generated by the meshing dictates our stereo aggregation window size choice, since all elements in an image segment have similar depth. We do not assume that pixels in the same segment share the same depth, but rather that they lie on a locally planar surface. This method succeeds in our system for two reasons: first, it improves the robustness of our matching procedure by employing entire regions instead of single points and, second, it reduces the total number of points that must be matched which improves the overall efficiency of the matching procedure. Finally, we note that if a region has too low of a variance (i.e. the largest triangle size) or if there is an occlusion, then cross correlation performs extremely poorly. In this instance, we simply skip this region and rely on the result of the post processing step to fill in the depth in this region.

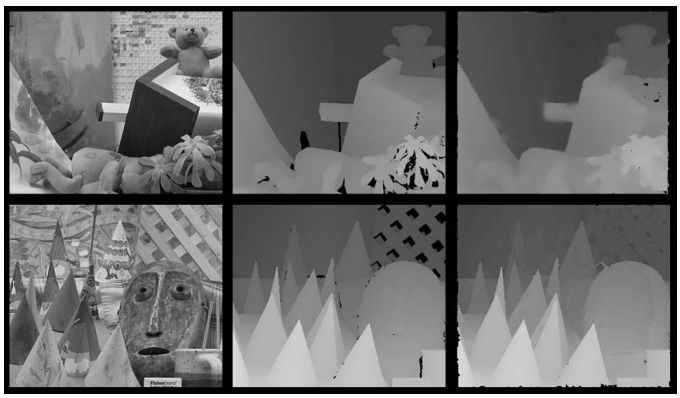


Fig. 4. Two images, each of size 450×375, from the benchmark developed by Scharstein et al. [10] (left column), the ground truth for these two images produced using a laser range finder (center column), and the output of our stereo algorithm (right column). Note that lighter gray values indicate that the object in the scene is closer, darker gray indicate that an object in the scene is further away, and black indicates areas of uncertainty.

Since the representation is conformal, the depth map can be post processed by exploiting an approximation to standard global optimization procedures such as anisotropic diffusion, which have been proven to improve the overall quality of depth reconstruction [13]. Since our mesh is conformal, depth values can pass between neighboring triangles via their nodes. Though these finite element methods generally converge slowly, they are proven to converge rapidly in a conformal representation [14]. In Figure 3, we show images to illustrate the steps of our stereo algorithm.

4.3. Results

At this point, we compare the effectiveness of our algorithm in calculating disparities. The benchmarks consist of dozens of pictures. The two images that the benchmark has identified as the most difficult are found in the left column of Figure 4. The accuracy of the measurements is calculated against a ground truth image, which can be found in the center column of Figure 4, generated by a laser range finder. Note that lighter gray values indicate that the object in the scene is closer and darker gray indicates than an object in the scene is further away. Black areas correspond to points where the disparity value is unknown.

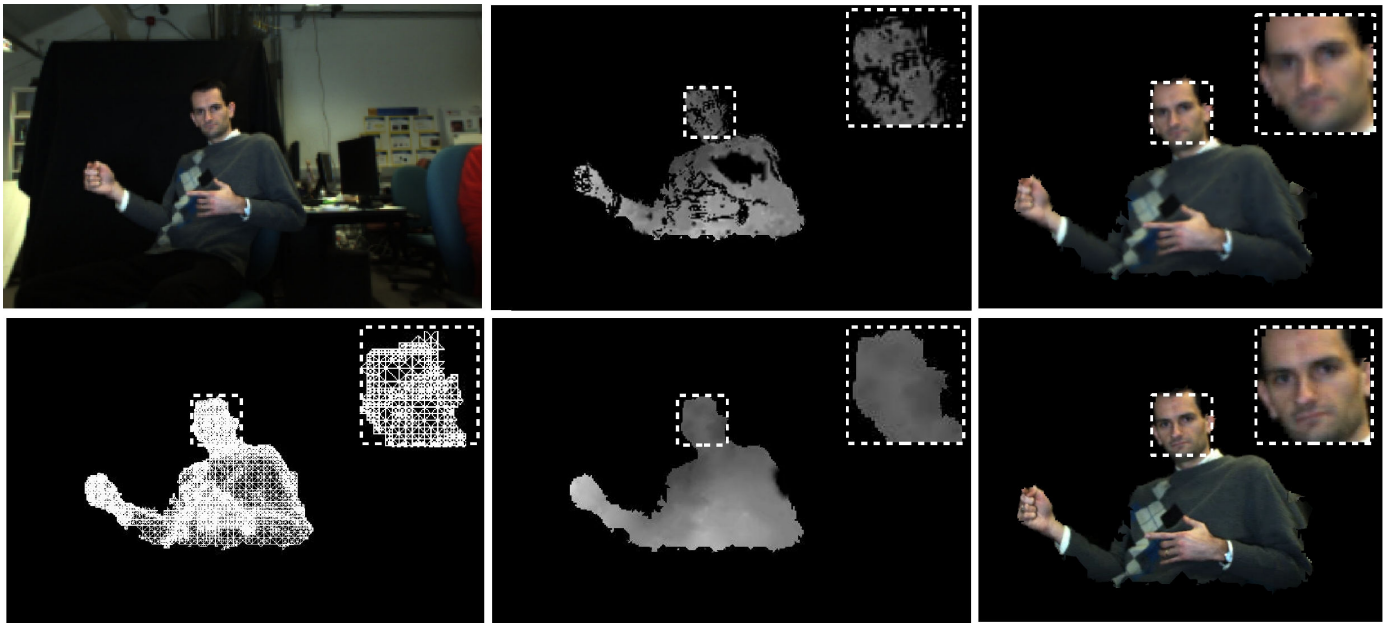


Fig. 3. A 320×240 image taken from a single camera in a stereo cluster (top-left), the mesh generated for this image (bottom-left), the pre-processed disparity image (top-middle), and the post-processed disparity image (bottom-middle), a rendering of the data using linear interpolation between the color value stored at each node (top-right), and a rendering of the data using the BIFC scheme and texture mapping (bottom-right). Note that lighter gray values indicate that the object in the scene is closer, darker gray indicate that an object in the scene is further away, and black indicates areas of uncertainty.

In this domain, error is calculated by the percentage of pixels that differ by more than a pixel, which is approximately the number of pixels that differ in their returned value by more than a single centimeter. The output of our stereo algorithm on the images found in the left column of Figure 4 calculated on two dual core 2.33 GHz machines can be found in the right column of the same figure. A quantitative comparison of our algorithm can be found in table 1. We include the most accurate performers on this benchmark in the same table. Wang et al. employed a dual core 1.6 GHz machine [15], Bleyer et al. employed a 2 GHz Pentium 4 machine [16], and Klaus et al. employed a dual core 2.21 GHz machine [17]. We arrive at comparable levels of accuracy as the top performers, but our algorithm takes anywhere between three hundred to two thousand times less time to produce an answer. These top performers arrive at a high level of accuracy by relying upon variants of global optimization technique, which are slow. We arrive at comparable levels of accuracy at a much faster speed on CPU by taking a hybrid approach: performing a local optimization technique (the region matching) and using a global optimization approximation to improve the initial results (anisotropic diffusion). Table 2 describes the average speed of our algorithm on a sequence of images taken from our system.

Process	Our's	Wang	Bleyer	Klaus
Teddy 1-Pixel Error	7.15%	8.31%	6.54%	7.06%
Teddy Speed	42.1ms	20s	100s	14s
Cone 1-Pixel Error	7.56%	7.18%	8.62%	7.92%
Cone Speed	53.8ms	20s	100s	25s

Table 1. A quantitative comparison of our algorithm against the top performers on the benchmark developed by Scharstein et al. [10]. The teddy and cone image correspond to the top and bottom rows of Figure 4 respectively. Our output was produced with approximately 40,000 triangles in both instances.

Triangulation	3.83 ms
Disparity	15.8 ms
Post-Processing	1.78 ms
Total	21.41 ms

Table 2. Average frame rate for a typical image sequence in the TI system on two dual core 2.33 GHz machines obtained using TI stereo pairs each with size 320×240 with approximately 10000 triangles per frame.

5. REAL-TIME COMPRESSION

Though the image partition described in the previous section provides a straightforward method to compress the depth information [18], if the partition is employed to compress the texture image it would result in poor visual appearance. If instead the uncompressed texture information was employed via texture mapping, then visual quality would remain unadulterated. Unfortunately, texture images are very large in size. An illustration of the difference between the two methods can be found in Figure 3.

5.1. BIFC Algorithm

In this section, we present a Border-Descriptor Inter-Frame Compression (BIFC) scheme, which employs inter-frame motion estimation. We first divide the image into macro blocks to increase the speed. Three types of macro blocks are defined in our scheme, (1) foreground blocks, (2) background blocks, and (3) border blocks for areas corresponding to the edge between the background and foreground. The border block is not considered in other bitmap-based compression methods. By employing the background subtraction, we can label each of the macro blocks.

Similar to MPEG encoding, we divide frames over time into two types: intra-frames and inter-frames. Intra-frames are compressed via a variant of JPEG compression. Namely, in intra-frames, only foreground and border blocks are compressed via JPEG compression. Therefore the intra-frame encoding structure consists of a sequence of macro block types, mask encoding, and the image encoding data. In order to perform inter-frame compression, we assume that users do not move too quickly between consecutive frames. This notion is made more clear in the next subsection. Under this assumption, the border blocks alone can be used to perform a motion estimate since they encode the most distinct features. Foreground blocks can then be filled in employing the border blocks. Following the block search, DCT transforms of block residuals are done as in JPEG compression. The inter-frame encoding structure is defined as a sequence of macro block type, mask encoding, image-residual encoding, and moving vector data.

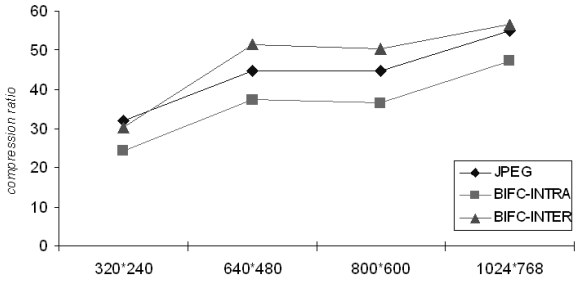
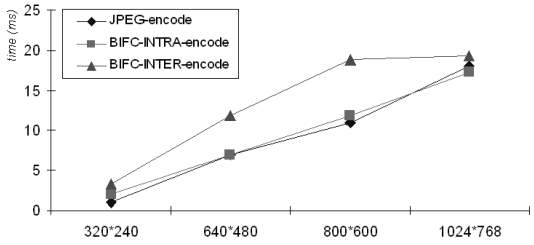
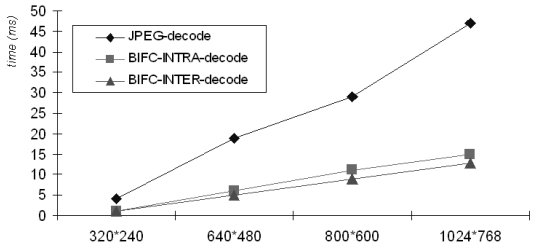


Fig. 5. Comparing compression for the same image sequence at different resolutions for different compression schemes



(a) Compression



(b) Decompression

Fig. 6. Average time cost for compression on the capturing side and decompression on the rendering side for images with large (75%) foreground coverage.

5.2. Compression Ratio and Speed

To illustrate the type of compression ratio and the speed of compression and decompression we consider the performance of our algorithm at various resolutions on our two dual core 2.33 GHz machine. The results in this paper are presented for about 75% foreground coverage of the entire image area. The key frame was calculated every 10 frames while the stereo reconstruction was performed at about 30 frames per second. The macro block size was set at 16 by 16 pixels and the search window for motion estimation was set at 32 by 32 pixels. The user movement is considered too quick if it moves beyond this 32 by 32 pixel search window. Figure 5 illustrates the compression ratio of the JPEG and BIFC scheme as a function of image resolution. For high foreground coverage the difference in the compression ratio between JPEG and BIFC intra-frame compression is small. On the otherhand, BIFC inter-frame compression has a distinct advantage over the JPEG compression technique.

In Figure 6 we compare the time cost for compression and decompression between JPEG and BIFC scheme for large foreground coverage (about 75%). The BIFC intra-frame compression has higher compression time than JPEG compression mainly due to the calculation of block type, while the inter-frame compression is close to or below the JPEG scheme. For decompression, the advantage of the BIFC compression scheme is obvious. Figure 3 illustrates the visual quality of our approach.

6. CONCLUSION

In this paper, we described a real-time stereo-vision system that creates a highly accurate 3D reconstruction of users to use in collaborative environments. Our novel data representation and 3D reconstruction algorithm offers a flexible, accurate, and fast solution to real-time scene (user) capture in 3D. The reconstruction algorithm described here is amongst the top performers on an industry wide benchmark for accuracy and it is easily one of the fastest reconstructions available. Using dynamic high-resolution texture mapping on lower-resolution mesh data we can leverage between the currently available computing power, network bandwidth, and visual quality required for face-to-face interactions in shared virtual environments.

Within our framework, the users are integrated into the virtual environment. Different digital effects (e.g., relighting, deformations) can be applied in real time to manipulate what is displayed to the remote users. The users can be immersed inside computer generated existing or non-existing environments, such as ancient buildings and future architectural designs, or merged with 3D medical (e.g. MRI) or other scientific data (e.g. seismic tomography of Earth crust) to allow interactive exploration.

7. REFERENCES

[1] A. Kolb, E. Barth, R. Koch, and R. Larsen, "Time-of-Flight Sensors in Computer Graphics," in *EUROGRAPHICS 2009*, M. Pauly and G. Greiner, Eds., 2009.

[2] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, "High-Quality Scanning Using Time-of-Flight Depth Superresolution," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08.*, 2009.

[3] Ramanarayan Vasudevan, Edgar Lobaton, Gregorij Kurillo, Ruzena Bajcsy, Tony Bernardin, Bernd Hamann, and Klara Nahrstedt, "A Methodology for Remote Virtual Interaction in Teleimmersive Environments," in *ACM Multimedia Systems*, 2010.

[4] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms," in *International Conference on Computer Vision and Pattern Recognition*, 2006, pp. 519–528.

[5] J. Mulligan and K. Daniilidis, "Real time Trinocular Stereo for Tele-immersion," in *Proceedings of International Conference on Image Processing*, 2001, pp. 959–962.

[6] H.H. Baker, D. Tanguay, I. Sobel, D. Gelb, M.E. Gross, W.B. Culbertson, and T. Malzenbender, "The Coliseum Immersive Teleconferencing System," in *Proceedings of International Workshop on Immersive Telepresence, Juanles-Pins, France*, 2002.

[7] P. Kauff and O. Schreer, "An Immersive 3D Video-Conferencing System Using Shared Virtual Team User Environments," in *Proceedings of the 4th International Conference on Collaborative Virtual Environments*, 2002, pp. 105–112.

[8] O. Schreer, I. Feldmann, N. Atzpadin, P. Eisert, P. Kauff, and H.J.W. Belt, "3D Presence: a System Concept for Multi-User and Multi-Party Immersive 3D Videoconferencing," in *European Conference on Visual Media Production*, Nov. 2008, pp. 1–8.

[9] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *International Journal of Computer Vision*, vol. 47, no. 1, pp. 7–42, 2002.

[10] D. Scharstein, R. Szeliski, and M. Coll, "High-Accuracy Stereo Depth Maps Using Structured Light," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003, vol. 1.

[11] G. Kurillo, Z. Li, and R. Bajcsy, "Framework for Hierarchical Calibration of Multi-Camera Systems for Teleimmersion," May 27-29, 2009 2009.

[12] J.M. Maubach, "Local Bisection Refinement for N-Simplicial Grids Generated by Reflection," *SIAM Journal on Scientific Computing*, vol. 16, pp. 210, 1995.

[13] P. Favaro, S. Osher, S. Soatto, and L. Vese, "3D Shape from Anistropic Diffusion," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2003, vol. 1.

[14] H. Kirchner and H. Niemann, "Finite Element Method for Determination of Optical Flow," *Pattern Recognition Letters*, vol. 13, no. 2, pp. 131–141, 1992.

[15] Z.F. Wang and Z.G. Zheng, "A Region Based Stereo Matching Algorithm Using Cooperative Optimization," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[16] M. Bleyer and M. Gelautz, "A Layered Stereo Matching Algorithm Using Image Segmentation and Global Visibility Constraints," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, no. 3, pp. 128–150, 2005.

[17] A. Klaus, M. Sormann, and K. Karner, "Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure," in *International Conference on Pattern Recognition*, 2006, vol. 2.

[18] G. Kurillo, R. Vasudevan, E. Lobaton, and R. Bajcsy, "A Framework for Collaborative Real-Time 3D Teleimmersion in a Geographically Distributed Environment," in *Proceedings of 10th IEEE International Symposium on Multimedia (ISM 2008)*, Berkeley, CA, December 15-17 2008, pp. 111–118.