Skeleton-Based Data Compression for Multi-Camera Tele-immersion System

Jyh-Ming Lien[1], Gregorij Kurillo[2], and Ruzena Bajcsy[2]

[1] George Mason University, Fairfax, VA, [2] University of California, Berkeley, CA

Abstract. Image-based full body 3D reconstruction for tele-immersive applications generates large amount of data points, which have to be sent through the network in real-time. In this paper we introduce a skeleton-based compression method using motion estimation where kinematic parameters of the human body are extracted from the point cloud data in each frame. First we address the issues regarding the data capturing and transfer to a remote site for the tele-immersive collaboration. We compare the results of the existing compression methods and the proposed skeleton-based compression technique. We examine robustness and efficiency of the algorithm through experimental results with our multi-camera tele-immersion system. The proposed skeleton-based method provides high and flexible compression ratios (from 50:1 to 5000:1) with reasonable reconstruction quality (peak signal-to-noise ratio from 28 to 31 dB).

1 Introduction

Tele-immersion (TI) is aimed to enable users in geographically distributed sites to collaborate and interact in real time inside a shared simulated environment as if they were in the same physical space [1]. In addition, the virtual environment can include different synthetic objects (e.g. three-dimensional models of buildings) or captured data (e.g. magnetic resonance image of a brain) which can be explored by the users through three-dimensional (3D) interaction. The TI technology combines virtual reality for rendering and display purpose, computer vision for image acquisition and 3D reconstruction, and various networking techniques for transmitting the data between the remote sites in real-time with the smallest delays possible. TI is aimed at different network applications, such as collaborative work in industry and research, remote evaluation of products for ergonomics, remote learning and training, coordination of physical activities (e.g. dancing [2], rehabilitation), and entertainment (e.g. games, interactive music videos). In addition, 3D data captured locally could be used for kinematic analysis of body movement (e.g. in medicine and rehabilitation) or in computer animation [3].

To render realistic model of the TI user inside the virtual space, real-time 3D capturing of the human body is needed. Human body can be captured using multi-camera system which allows extraction of depth information. Different approaches for real-time processing of depth information have been proposed. One of the first TI systems was presented by the researchers at University of Pennsylvania [4]. 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 tele-immersive system based on reconstruction from silhouettes was proposed by Baker et al. [5]. Blue-c tele-immersion system presented by Wurmlin et al. [6] allowed full-body reconstruction based on silhouettes obtained by several cameras arranged around the user. Reconstruction from the silhouettes provides faster stereo reconstruction as compared to the image-based methods; however, this approach lacks accuracy and discrimination of several persons or objects inside the system. The TI system presented in this paper introduces 360-degree full-body 3D reconstruction from images using twelve stereo triplets [7]. The captured 3D data are transferred using TCP/IP protocol to the rendering computer which reconstructs and displays point clouds inside a virtual environment at the local or remote sites in real time. The presented system has been successfully used in remote dancing applications [2] and learning of tai-chi movements [8][1].

One of the major bottlenecks of our current system for tele-immersion is the real-time transfer of large amount of data points, which are generated by the stereo reconstruction from multiple images, to the remote site. Current image/video-based compressor [2] in our TI system only allows transmission of data with the rate of 5 or 6 frames per second (fps). In the future we plan to improve the resolution of the captured images which will additionally increase the bandwidth requirements for transfer of data in real time. In this paper, we focus on the data compression using motion estimation for TI applications. The proposed compression method provides high and flexible compression ratios (from 50:1 to 5000:1) with reasonable reconstruction quality (peak signal-to-noise ratio from 28 to 31 dB). We describe a novel technique of skeleton-based compression of 3D point data captured by our system. Robustness and efficiency of the algorithm is examined theoretically and experimentally. We discuss how to compress the data using the estimated articulated motions and how to encode non-rigid motions using regular grids (residual maps). Finally, we propose several efficient ways to detect temporal coherence in our residual maps including accumulating residuals from a few frames and compress them using established video compression techniques.

2 Overview of the Tele-immersion Apparatus

Our tele-immersion apparatus consists of 48 Dragonfly cameras (Point Grey Research Inc, Vancouver, Canada) which are arranged in 12 clusters covering 360 degree view of the user(s). The cameras, equipped with 6 and 3.8 mm lenses, are mounted on an aluminum frame with dimensions of about 4.0 x 4.0 x 2.5 m^3 (Fig. 1a). The arrangement of cameras was optimized to increase the usable workspace coverage to about 2.0 x 2.0 x 2.5 m^3. Each cluster consists of three black and white cameras intended for stereo reconstruction and a color camera used for texture acquisition. The four cameras in each cluster are connected through a fire-wire interface to a dedicated personal computer. The cluster PCs are dual or quad CPU (Intel Xeon, 3.06 GHz) machines with 1GB of memory and 1 Gbps connection to Internet 2. To synchronize image acquisition on all 48 cameras, the trigger computer generates an analogue signal sent to all the cameras in each frame. The cluster computers notify the trigger computer via TCP/IP messaging when the image acquisition and reconstruction have been completed. Although the

[1] Movies and images are available at http://tele-immersion.citris-uc.org

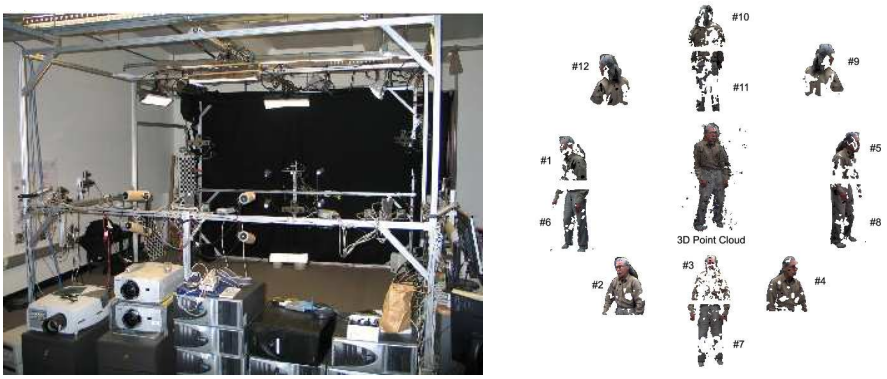


Fig. 1. (a) Multi-camera stereo system for tele-immersion consists of 48 cameras arranged in 12 stereo clusters. The captured images are processed by 12 multi-core computers to provide 360-degree 3D full-body reconstruction in real time. (b) Partial depth image data captured by each of the 12 stereo camera clusters, which are accurately calibrated to a reference coordinate system, is combined inside the renderer into a full-body 3D reconstruction.

cameras allow capturing of images with the resolution of 640 by 480 pixels, the images are resized to 320 by 240 pixels to reduce the computational load. Based on the intrinsic and extrinsic parameters of each camera obtained during the calibration, the image is first rectified and distortion of the lens is corrected. The background is subtracted using a modified Gaussian average algorithm [9]. In the next step, edges of foreground objects are extracted and regions with similar features (i.e. texture) are identified to perform region based correlation among the images captured by the stereo triplet [4]. Next, the depth map is computed from the three gray scale images using triangulation method. The reconstruction algorithm has been paralleled to exploit the multi-core technology. The image data is equally split between the CPUs to increase the speed of the 3D reconstruction process. The accuracy of the stereo reconstruction is mainly affected by the quality of calibration. The errors can occur due to lens distortion, misalignment between image and lens plane, and deviations of position and rotation between the stereo cameras [10],[11]. The calibration of our TI system is performed in two steps. The intrinsic calibration of camera parameters (i.e. distortion, optical center) is performed by Tsai algorithm [12] using a checkerboard. Extrinsic parameters of each camera (i.e. position, orientation) are obtained by capturing LED position in about 10,000 points located inside the overlapping volume between the calibrated and reference camera. The captured points are re-projected to the reference camera plane while non-linear optimization is used to reduce the errors. The quality of the stereo reconstruction is further affected by illumination conditions, texture and color of the objects, and arrangement of the camera clusters around the workspace. The illumination inside the tele-immersion apparatus is diffused by filters to reduce shadows and specular highlights which interfere with the stereo matching algorithm.

To transfer the data to the rendering computer, the acquired depth map (2 bytes per pixel) is combined with the color map (3 bytes per pixel) and compressed by run-length encoding and z-lib [13] loss-less compression algorithm [2]. The compressed data package is sent through the network using TCP/IP protocol. The size of each data package depends on the image coverage and ranges from about 25 to 50 KBs for imaging one person inside the system. Data streams from the twelve calibrated clusters are composed into a 3D image by a point-based rendering application developed using OpenGL. Based on camera calibration parameters the renderer combines the points received from each cluster into the full 3D reconstruction of the tele-immersion user (Fig. 1b). The complexity of the stereo reconstruction is currently limiting the frame rate to about 5 to 6 frames per second (fps). The required network bandwidth for this acquisition rate is below 5 Mbps. With increased frame rate and resolution of the captured data the bandwidth requirements can exceed 1 Gbps, therefore different compression techniques of the 3D data are needed.

3 Skeleton-based Compression

One of the major bottlenecks of our current system for tele-immersion is the transfer of large amount of data points. For example, data from images of 640x480 pixel depth and color maps from 10 camera clusters with a frame rate of 15 fps would require 220 MB/s network bandwidth. Several methods [6],[2] have been proposed to address this problem using image and video compression techniques, but the data volume remains to be prohibitively large. Currently, our compression method can only reach 5 to 6 fps when connected with another remote TI site. In this section we will discuss our solutions to address the problems in TI data compression from a model driven approach. The main idea of this work is to take advantage of prior knowledge of objects, e.g. human figures, in the TI environments and to represent their motions using smaller number of parameters, e.g., joint positions and angles. The data transfer can be significantly reduced by introducing compression algorithms based on extracted kinematic parameters (i.e. joint angles) of people captured by the stereo cameras.

We propose a new compression method based on human motion estimates. Instead of transmitting the point clouds, we can simply transmit the motion parameters. This approach is based on the assumption that most points move under rigid-body transform along with the skeleton. In reality, point movements may deviate from this assumption, such as muscle movements and hair or cloth deformations; therefore, we further compress the deviations from the rigid movements. As it will be shown in the remainder of the paper, the deviations (which is called “prediction residuals”) in most cases are small. An overview of this model driven approach is shown in the figure below (Fig. 2). Our compressor provides (flexible) high compression ratios (from 50:1 to 5000:1) with reasonable reconstruction quality. Our method can estimate motions from the data captured by our TI system in real time (10+ fps).

3.1 Skeleton and Motion Estimation

We extend Iterative Closest Points (ICP) [14] to estimate motion of dynamic point data in real time. Our approach assumes the skeleton(s) of the person(s) represented by

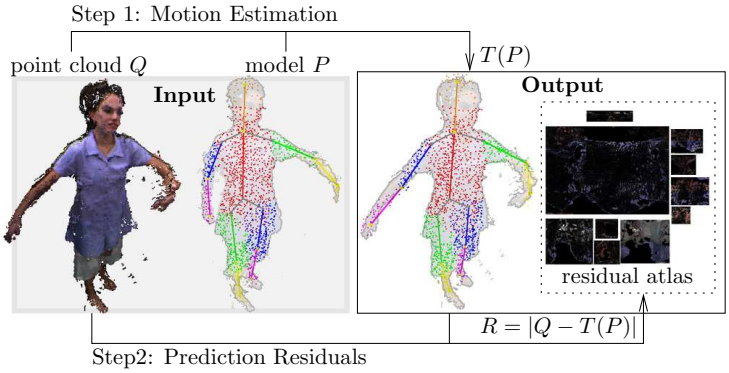


Fig. 2. Skeleton-based compression of the point data Q, where P is the skeleton, and T is the motion parameter (e.g., the skeleton configuration) that transforms P to Q. The prediction residuals R is the difference between $T(P)$ and Q.

points in the first frame is given. Several methods exist [15],[16] and can provide us an initial skeleton to start the process. We then apply a real-time tracking method which fits the skeleton to the point cloud data captured from the rest of the movements.

Extensive work has been done to track human motion in images (see surveys [17,18]). Our goal is to estimate motion from 3-D points. Algorithms 3.1 and 3.2 outline our approach. The input parameters of Algorithm 3.1 include the skeleton S and the points Q that we would like S to fit to. The algorithm starts by fitting the points (P_l) associated with the root link (l_{root}) of S to Q using ICP (see Algorithm 3.2) and then fits the rest of the links hierarchically, i.e., from the torso to the limbs. Finally, once all links are roughly aligned with the point cloud Q, we perform a global fitting which tries to minimize the *global* difference between the skeleton S and Q. Details of global fitting can be found in [19].

Algorithm 3.1: ARTICP(S, Q, τ)

cluster Q
q.push(l_{root})
while $q \neq \emptyset$
do
  $l \leftarrow q$.pop()
  $T \leftarrow$ICP(P_l, Q, τ)
  for each child c of l
  do
    apply T to c
    q.push(c)
global fitting

Algorithm 3.2: ICP(P, Q, τ)

repeat
  find corresponding points $\{(p_i \in P, q_i \in Q)\}$
  compute $error$ and T in Eq. 1
  $P = T(P)$
until $error < \tau$
return (T)

Given two point sets P and Q, ICP first computes corresponding pairs $\{(p_i \in P, q_i \in Q)\}$. Using these corresponding pairs, ICP computes a rigid-body transform T

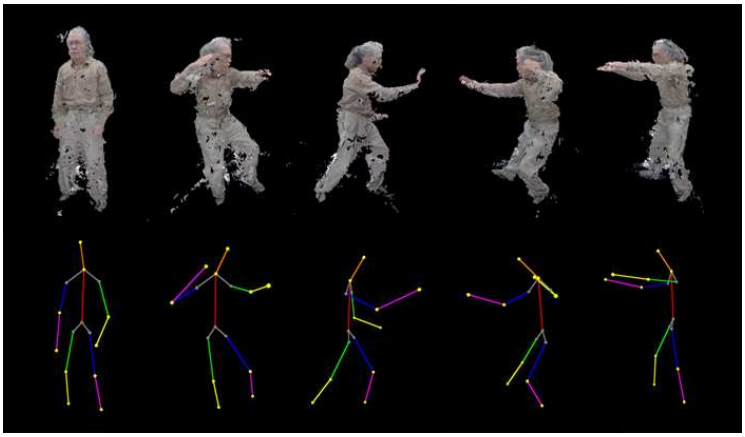


Fig. 3. Top: Stereo reconstruction of our tele-immersion system. Bottom: Real-time skeleton extraction from 3D point cloud data

such that the "matching error" defined in Eq. 1 between P and Q is minimized.

$$error = \underset{T}{\operatorname{argmin}} \sum_{i} |(T(p_i), q_i)|^2 \ . \tag{1}$$

The main step for minimize the matching error (see details in [20]) is to compute the cross-covariance matrix Σ_{PQ} of the corresponding pairs $\{p_i, q_i\}$,

$$\Sigma_{PQ} = \frac{1}{n} \sum_{i=1}^{n} [(p_i - \mu_p)(q_i - \mu_q)^t] \ , \tag{2}$$

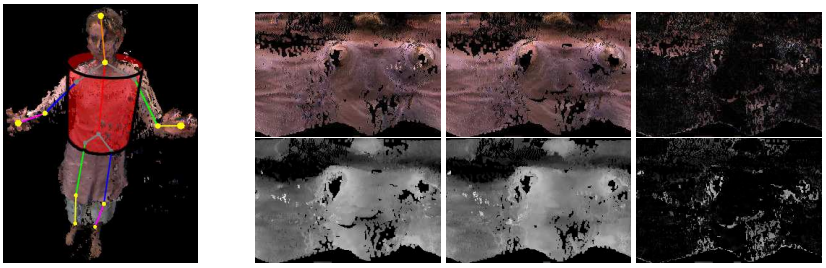
where μ_p and μ_q are the centers of $\{p_i\}$ and $\{q_i\}$, resp., and n is the size of $\{p_i, q_i\}$. As outlined in Algorithm 3.2, ICP iterates these steps until the error is small enough.

Figure 3 illustrates the result produced by our motion estimation from a sequence of tai-chi movements. The main features of our motion estimation include: (a) Hierarchical fitting for faster convergence, (b) Articulation constraint, (c) Monotonic convergence to local minimum guaranteed, and (d) Global error minimization. Due to the space limitation, we refer interested readers to [19] for detail.

3.2 Prediction Residuals

Motion estimation brings the skeleton S close to the current point cloud Q. However, due to several reasons, e.g., non-rigid movements and estimation errors, our model, the points P_S associated with the skeleton S, may not match Q exactly. We call the difference between P_S and Q "prediction residuals" (or simply residuals). Because P_S and Q are close to each other, we expect the residuals to be small. In this section, we present a method to compute the prediction residuals.

First, for each link l in the skeleton, we have two point set associated with l, namely, P_l and Q_l which are points from P_S and Q, respectively, and are closest to l than other



(A torso link) (a) (b) (c)

Fig. 4. A skeleton and the torso link is shown as a cylinder. (a) Color and depth maps at time $t-1$ of the torso link. The Y-axis of the maps is parallel to the link. (b) Color and depth maps at time t of the torso link. (c) The differences between the maps at times $t-1$ and t.

links of S Then we project both P_l and Q_l to a regular 2-D grid embedded in a cylindrical coordinate system defined by the link l (see Fig. 4). Because P_l and Q_l are now encoded in regular grids, we can easily compute the difference, which can be compressed using image compression techniques. Because this projection is invariant from a rigid-body transform, we only need to re-sample Q_l at each time step. We determine the size of a grid from the shape of a link l and the size of l's associated points P_l . We make sure that our grid size is at least $2|P_l|$ using the following formulation, i.e., the width and the height of the grid are $2\pi R_l S$ and $L_l S$, resp., where R_l and L_l are the radius and the length of the link l and $S = \sqrt{\frac{|P_l|}{\pi R_l L_l}}$. We call that a grid encodes 100% prediction residuals if the grid has size $2|P_l|$. As we will see later, we can tune the grid size to produce various compression ratios and qualities.

Table 1. Efficiency of skeleton-based motion estimation method. On average, our method achieves 11+ frames per second (fps).

motion	dancer 1 (Fig.3)	dancer 2	tai-chi student	tai-chi teacher
average fps	11.9 fps	11.5 fps	12.5 fps	12.6 fps

4 Results

In this section, we study the quality of our skeleton-driven compression on the TI data with various levels of prediction residuals. We also compare our model-driven compression to H.264 video compression [21] and Yang et al. method [2]. All the experimental results in this section are obtained using a Pentium4 3.2GHz CPU with 512 MB of RAM. We evaluate the results of our motion estimation method using four motion sequences captured by our TI system. These motions are performed by the dancers, one student and one tai-chi master (Fig.2). The data captured by our TI system have about

75,000 3D points in each frame. Table 1 shows that we can maintain at least 11 fps interactive rate in all studied cases.

4.1 Quality

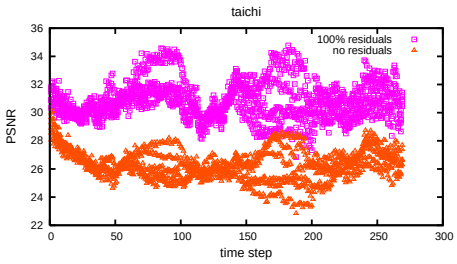
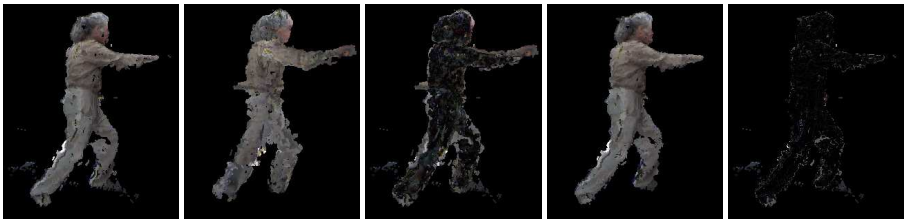


Fig. 5. PSNR values from the tai-chi motion. Each point in the plot indicates a PSNR value from a rendered image. For each time step, there are 12 points, which indicate 12 images rendered from two points with 100% and 0% residuals.

The quality of our compression method was measured as the difference between the point data before and after the model-driven compression. The ”peak signal-to-noise ratio” (PSNR) of the images rendered from uncompressed and compressed point data was computed. In our experiments, two sets of images (one for each point set) are rendered from six (60 degree separated) camera views in each frame. We compare the reconstruction quality by computing PSNRs w.r.t the uncompressed data. Typical PSNR values in image compression are between 20 and 40 dB. We considered three compression levels, i.e., compression without residuals and with 50% and with 100% residuals. We see that encoding residuals indeed generate better reconstruction by 4 dB as compared to the compression without residuals. Figure 5 shows that considering residuals always produces better reconstructions in all frames. Another important observation is that the compression quality remains to be the same (around 30 dB) during the entire motion. Figure 6 shows that the difference between the uncompressed and compressed frame is more visible when the prediction residuals are not considered.

4.2 Compression Ratio

The analysis of different compression ratios showed that our model-driven compression method can achieve 50:1 to 5000:1 compression ratios (Table 2). As we have shown earlier, our compression method can provide different compression ratios by varying the level of residuals considered during the encoding. Significantly higher compression ratio as compared to the other two methods tested is achievable due to a fundamental difference in encoding the data while maintaining reasonable reconstruction quality (Fig 6). Both, Yang et al.’s and H.264 algorithms, are image (or video)-based compressions, which take color and depth images as their input and output. Our model-driven compression, however, converts the color and depth images to motion parameters and prediction residuals. Despite high quality and high compression ratio of H.264 algorithm, the processing cannot be performed in real time for the amount of data that we considered in this work.



(a) (b) (c) (d) (e)

Fig. 6. Reconstructions from the compressed data and their differences with the uncompressed data. (a) Uncompressed data. (b) Compressed without residuals. (c) Difference between (a) and (b). (d) Compressed with residuals. (e) Difference between (a) and (d).

Table 2. Compression ratio. Both Yang et al.'s [2] and H.264 (we use and implementation from [22]) compression methods took the color and depths images as their input and output. The compression ratio of H.264 reported in this table is obtained using 75% of its best quality. We use jpeg and png libraries to compress color and depth residuals, respectively.

motion		**dancer 1**	**dancer 2**	**student**	**tai-chi master**
size before compression		142.82 MB	476.07 MB	1.14 GB	988.77 MB
compression ratio	Yang et al. [2]	11.36	11.73	10.23	14.41
	H.264 [22]	64.04	53.78	32.04	49.58
	no residuals	1490.31	3581.52	5839.59	5664.55
	25% residuals	195.62	173.52	183.80	183.82
	100% residuals	66.54	55.33	60.29	61.43

5 Conclusion

In this paper we have presented a skeleton-based data compression aimed for the use in tele-immersive environments. Data produced by the full-body 3D stereo reconstruction requires high network bandwidth for real-time transmission. Current implementation of the image/video-based compressor [2] in our tele-immersion system only allows transmission of data with the rate of 5 or 6 fps. Using model-based compression techniques can significantly reduce the amount of data transfer between the remote TI sites.

Using the real time (10+ fps) motion estimation technique described in this paper, we can compress data by converting point cloud data to a limited number of motion parameters while the non-rigid movements are encoded in a small set of regular grid maps. Prediction residuals are computed by projecting the points associated with each link to a regular 2D grid embedded in a cylindrical coordinate system defined by the skeleton link. Our experiments showed that our compressor provides adjustable high compression ratios (from 50:1 to 5000:1) with reasonable reconstruction quality with peak signal-to-noise ratio from 28 dB to 31 dB.

References

1. Lanier, J.: Virtually there. Scientific American **4** (2001) 52–61

2. Yang, Z., Cui, Y., Anwar, Z., Bocchino, R., Kiyanclar, N., Nahrstedt, K., Campbell, R., Yurcik, W.: Real-time 3d video compression for tele-immersive environments. In: Proceedings of SPIE/ACM Multimedia Computing and Networking (MMCN06), San Jose, CA. (2006)
3. Kalra, P., Magnenat-Thalman, N., Moccozet, L., Sannier, G., Aubel, A., Thalman, D.: Real-time animation of realistic virtual humans. IEEE Computer Graphics and Applications **18** (1998) 42–56
4. Mulligan, J., Daniilidis, K.: Real time trinocular stereo for tele-immersion. In: Proceedings of 2001 International Conference on Image Processing, Thessaloniki, Greece. (2001) 959–962
5. Baker, H., Tanguay, D., Sobel, I., Gelb, D., Gross, M., Culbertson, W., Malzenbender, T.: The coliseum immersive teleconferencing system. In: Proceedings of International Workshop on Immersive Telepresence, Juan-les-Pins, France. (2002)
6. Wrmlin, S., Lamboray, E., Gross, M.: 3d video fragments: dynamic point samples for real-time free-viewpoint video. Computers and Graphics **28** (2004) 3–14
7. Jung, S., Bajcsy, R.: A framework for constructing real-time immersive environments for training physical activities. Journal of Multimedia **1** (2006) 9–17
8. Patel, K., Bailenson, J.N., Hack-Jung, S., Diankov, R., Bajcsy, R.: 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. (2006) 87–94
9. Piccardi, M.: Background subtraction techniques: a review. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Hague, Netherlands. (2004) 3099–3104
10. Zhao, W., Nandhakumar, N.: Effects of camera alignment errors on stereoscopic depth estimates. Pattern Recognition **29** (1996) 2115–2126
11. Zhang, D., Nomura, Y., Fujii, S.: Error analysis and optimization of camera calibration. In: Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS 91), Osaka, Japan. (1991) 292–296
12. Tsai, R.: 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** (1987) 323–344
13. Zlib: Compression library (2005)
14. Besl, P., McKay, N.: A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992) 239256
15. Herda, L., Fua, P., Plankers, R., Boulic, R., Thalman, D.: Skeleton-based motion capture for robust reconstruction of human motion. In: Proceedings of Computer Animation Conference. (2000) 77–93
16. Theobalt, C., Magnor, M., Schuler, P., Seidel, H.: Multi-layer skeleton fitting for online human motion capture. In: Proceedings of 7th International Workshop on Vision, Modeling and Visualization (VMV 2002), Erlangen, Germany. (2002) 471–478
17. Aggarwal, J.K., Cai, Q.: Human motion analysis: a review. Comput. Vis. Image Underst. **73** (1999) 428–440
18. Gavrila, D.M.: The visual analysis of human movement: a survey. Comput. Vis. Image Underst. **73** (1999) 82–98
19. Lien, J.M., Bajcsy, R.: Skeleton-based compression of 3-d tele-immersion data. In: Proceedings of the ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC). (2007)
20. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14** (1992) 239–256
21. Wiegand, T., Sulivan, G., Bjntegaard, G., Luthra, A.: Overview of the h. 264/avc video coding standard. IEEE Transactions on Circuits and Systems for Video Technology **13** (2003) 560–576
22. Adobe: Qicktime 7.0 h.264 implementation (2006)