Vision, Video, and Graphics (2003)
P. Hall, P. Willis (Editors)

A Flexible and Versatile Studio for Synchronized Multi-view Video Recording

Christian Theobalt,[1] and Ming Li,[1] and Marcus A. Magnor[1] and Hans-Peter Seidel[1]

[1] Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany

Abstract

In recent years, the convergence of computer vision and computer graphics has put forth new research areas that work on scene reconstruction from and analysis of multi-view video footage. In free-viewpoint video, for example, new views of a scene are generated from an arbitrary viewpoint in real-time using a set of multi-view video streams as inputs. The analysis of real-world scenes from multi-view video to extract motion information or reflection models is another field of research that greatly benefits from high-quality input data. Building a recording setup for multi-view video involves a great effort on the hardware as well as the software side. The amount of image data to be processed is huge, a decent lighting and camera setup is essential for a naturalistic scene appearance and robust background subtraction, and the computing infrastructure has to enable real-time processing of the recorded material. This paper describes our recording setup for multi-view video acquisition that enables the synchronized recording of dynamic scenes from multiple camera positions under controlled conditions. The requirements to the room and their implementation in the separate components of the studio are described in detail. The efficiency and flexibility of the room is demonstrated on the basis of the results that we obtain with a real-time 3D scene reconstruction system, a system for non-intrusive optical motion capture and a model-based free-viewpoint video system for human actors.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications I.4.9 [Image Processing and Computer Vision]: Applications I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture I.4.5 [Image Processing and Computer Vision]: Reconstruction

1. Introduction

In recent years, one tendency in computer graphics is to include information that was measured in the real world into the rendering process. New research directions, such as image-based or video-based rendering, investigate the possibility of creating new realistic views of a scene from a set of real world images or video streams. For the realistic rendering of surface materials, the acquisition of surface reflection models from images of real objects is becoming more and more important.

In computer vision, analysis of and model reconstruction from images and video streams has long been in the focus of interest. The convergence of computer vision and computer graphics creates the new research area of surround vision that depends on high quality multi-view image data. Among the most difficult to acquire image material are synchronized video streams of dynamic scenes that are recorded from multiple camera positions.

The need of high quality multi-video data is motivated by the wide range of interesting application areas, some of which shall be mentioned briefly. In visual media, such as TV or feature films, an ever increasing application of Computer Graphics elements can be observed. Computer-animated human characters are widely used in movie productions and commercial spots. To make their appearance natural, human motion capture technology is used to acquire the human motion data from video footage of a moving real person.

The advent of increasingly powerful computers and the existence of high-end graphics hardware even in consumer PCs makes possible researching new immersive visual media. In free-viewpoint video, a real-world scene is recorded from multiple camera positions. Using the recorded video footage, a three-dimensional temporally varying model of the dynamic scene is reconstructed. A viewer is then given the freedom to interactively choose an arbitrary viewpoint onto the reconstructed 3D video.

While the reconstruction and analysis algorithms in human motion capture and free-viewpoint video are, by themselves, computationally challenging, the actual acquisition of the multi-view video footage requires considerable effort on the hardware as well as on the software side.

In this paper, a new multi-view video studio is described that is both flexible and versatile. It is designed for the acquisition of reference video data to be used in different surround vision applications, such as vision-based human motion capture, real-time 3D scene reconstruction and free-viewpoint video of human actors. It is demonstrated that using off-the-shelf hardware, an efficient and comparably inexpensive acquisition room can be designed.

The rest of the paper starts with a review of previous work (Sect. 2). After a description of the requirements and the studio concept in Sect. 3, the separate components of the studio such as the room layout (Sect. 4), the camera system (Sect. 5) and lighting (Sect. 6) are described. The computer hardware and the software subsystem are explained in Sect. 7 and Sect. 8. In Sect. 9 a real-time visual hull reconstruction system, a non-intrusive human motion capture method and a system for free-viewpoint video of human actors are described in more detail. The paper proceeds with a discussion in Sect. 10 and concludes in Sect. 11 with an outlook on future improvements of the studio.

2. Previous Work

The advent of new research areas in computer graphics and computer vision, such as image-based rendering and automatic analysis and reconstruction from video streams has created the need for high quality real-world image data. For acquisition of these data, complex special-purpose setups need to be built.

For realistic rendering of materials the acquisition of surface reflection properties from real object is essential. Different acquisition setups consisting of high-quality cameras and a set of light sources have been proposed in the literature [6, 26].

Whereas the set of still images under different illumination conditions obtained during the measurement of reflection properties is already very memory-intensive, the amount of data to be handled in multi-view video processing is even larger.

In video-based human motion capture, researchers use multiple video streams showing a moving person from different viewing directions to acquire the motion parameters. Commercial motion capture systems exist that use optical markers on the body in connection with several expensive high-resolution special purpose cameras [19]. Marker-free motion capture algorithms also exist that don't require any intrusion into the scene. In [4] the volumetric visual hull [12] of a person is reconstructed from multiple camera views, and an ellipsoidal body model is fitted to the motion data. The video acquisition takes place in a *3D Room* that allows recording with up to 48 cameras [11]. A similar setup for motion capture using reconstructed volumes is proposed in [14]. A different multi-camera system for volume reconstruction that uses a custom-made PC cluster for video processing is described in [2]. Several other video-based human motion capture systems exist that use multi-view camera data as input [7, 5].

In 3D or free-viewpoint video, multi-view video streams are used to reconstruct a time-varying model of a scene. The goal is to give a viewer the possibility to interactively choose his viewpoint onto the 3D model of the dynamic scene while it is rendered. A system for acquisition of multi-view images of a person for reconstruction of short motion sequences using conventional digital cameras is used in [27]. In a previous stage of their *3D Room,* Narajanan et. al. built a dome of over 50 cameras to reconstruct textured 3D models of dynamic scenes using dense stereo. To handle the huge amount of image data, the video streams are recorded on video tape first and digitized off-line. In [16] a multi-camera system is described to record a moving person for reconstruction of the polygonal visual hull in an off-line process. The previous paper is an extension of the original work on polygonal [17] and image-based visual hulls [18] that employ a multi-camera system for real-time 3D model reconstruction. A system for recording and editing of 3D videos based on the image-based visual hull algorithm is described in [28].

3. Our Studio Concept

The studio whose setup and application scenarios are described in this work is intended to be a universal acquisition environment for different research projects in surround vision. The applications in mind include video-based human motion capture, real-time 3D scene reconstruction and free-viewpoint video. Therefore, flexibility and versatility are important design criteria. To keep the cost as well as the administrative overhead moderate, off-the-shelf hardware is preferred over special-purpose equipment.

On the performance side, the requirements to the system are challenging. The setup must be able to acquire and process video data in real-time. At the same time it also has to provide the necessary storage bandwidth for recording and saving of multi-video streams. Intermediate storage of video data on analog media, such as video tapes, is not acceptable since we want to keep the hardware and administrative costs low and to prevent quality losses during the conversion into digital form.

Important design issues involve the applied camera equipment, the lighting, the supporting computing subsystem and the spatial layout of the video room. The several subsystems

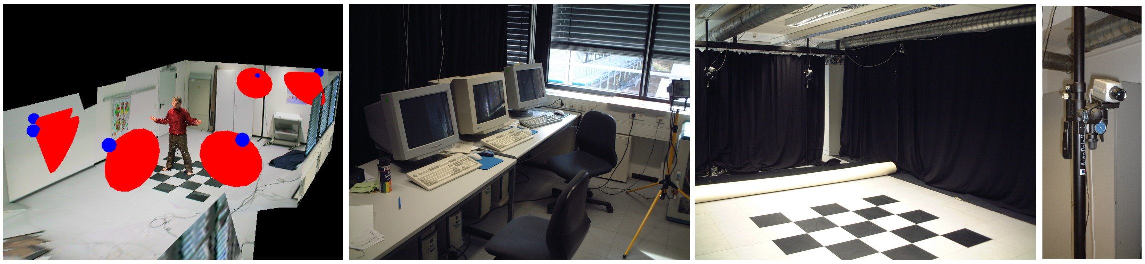

Figure 1: *Illustration of the acquisition room with cameras rendered as cones (1). The control room of the studio (2) and the large recording area with calibration pattern, cameras and all-around curtain (3). One of the video cameras mounted on a pole (4).*

of the camera room are described in more detail in the following sections.

4. Room Layout

The spatial dimensions of the room need to large enough to allow multi-view recording of dynamic scenes from a sufficient distance and from a large number of viewpoints. Hence, the studio is installed in a room of approximately 11 by 5 meters in size. The ceiling has a height of approximately 3m. Along one of the shorter walls an area of 1.5m by 5m is separated that serves as a control room of the studio. The remaining area of the studio, which is surrounded by opaque black curtains, is completely available for recording (Fig. 1).

5. Camera System

The cameras used in the studio need to fulfill the requirements specific to multi-view video applications. First, they must provide the possibility of external synchronization to make sure that the multi-view video streams are correctly registered in time. Second, the cameras have to deliver high frame-rates of at least 15 fps at a suitable resolution of 320x240 or 640x480 pixels. For scenes containing elements that move very rapidly, frame rates of at least 30 fps should be possible. Furthermore, the transfer of the image data to the computer needs to be efficient. Even with a high number of cameras, a sufficient bandwidth has to be available in the applied bus system.

Considering these requirements, we decided to use Sony$^{\mathrm{TM}}$ DFW-V500 IEEE1394 cameras. These CCD cameras provide a frame resolution of up to 640x480 pixels at 30 fps. External synchronization is possible via a trigger pulse sent to the camera through an external connector. The frames are provided in YUV 4:2:2 format, and they are digitally transferred to the host computer using the IEEE1394 interface. This way, the additional hardware overhead of a frame-grabber card is prevented. The bus bandwidth of 400 MBit/s is sufficient to control two of the cameras with one PC. The cameras provide a high number of adjustable parameters and an automatic white-balancing. Parameter sets for different applications can be stored in internal memory channels.

Currently, up to 8 cameras can be installed at arbitrary locations in the studio. For positioning, telescope poles (Manfrotto$^{\mathrm{TM}}$ Autopole [15]) with 3-degree-of-freedom mounting brackets (Manfrotto$^{\mathrm{TM}}$ Gear Head Junior [15]) are used that can be jammed between the floor and the ceiling (Fig. 1).

For most applications, the internal and external camera parameters of each of the 8 imaging devices must be known. To achieve this, a calibration procedure based on Tsai's algorithm is applied [24]. The external camera parameters are estimated using a 2×2 m checkerboard calibration pattern on the floor in the center of the room. The corners of the checkerboard are detected automatically by employing a sub-pixel corner detection algorithm on the camera images showing the pattern [9]. The internal camera parameters (center of origin in the image plane, effective focal length) can calculated from the large checkerboard pattern also by means of Tsai's calibration method. More accurate determination of the internal parameters is achieved by using a small checkerboard pattern attached to a wooden panel that can be positioned to cover a large part of each camera's field of view. Due to the sufficiently steep viewing angle between the cameras arranged around the scene and the pattern on the floor, the concurring effects of focal length and camera translation can be robustly distinguished.

6. Lighting

The studio lighting is an important issue for the quality of the produced video material. The applied equipment must be flexible enough to produce good lighting conditions under different scenarios (Sect. 9). In our free-viewpoint video and motion from video research, robust separation of the foreground object from the background is essential. Therefore, the number of shadows cast on the floor has to be minimized, whereas the scene has to remain luminous enough. Furthermore, the applied lighting shall give the scene a natural appearance, which is an important criterion for free-viewpoint video (Sect. 9).

To make background subtraction easier, a chroma-keying approach is traditionally applied. There, the studio background is painted in a uniform color, such as blue or green [27]. In the camera images, the foreground object is separated from the background by discarding all pixels that are close in color to the background color. There exists a number of problems with this approach, one of these undesired effects is that particularly in small studios the background color is reflected from the foreground object and gives it an unnatural appearance.

In our studio, a different setup is chosen (see also Sect. 8). To minimize the effects of external light on the center of the scene, and to minimize the visual appearance of shadows cast on the walls, the studio is equipped with an all-around opaque black curtain. The floor can be covered with a black matte carpet that can be easily removed to reveal the calibration pattern. The described covering of the floor and of the walls greatly improves the robustness of a color-based background subtraction scheme (Sect. 8). Shadows cast by a moving object or person are only slightly different in color from the background and cannot wrongly be classified as foreground. At the same time, unwanted reflection of the background color onto the foreground object, as it is often observed in methods employing a blue or green-screen [27], is prevented.

There exist three rows of light sources (Siteco$^{\text{TM}}$ louver luminaire [22]) on the ceiling that have a large spatial extent and produce a very uniform lighting in the center of the scene. These light sources illuminate objects in the center of the scene from the top at rather steep angles only. This is achieved by multiple reflectors that spread the light homogeneously downwards but prevent direct illumination of the camera lenses. This lighting setup allows a flexible positioning of the cameras while preventing direct recording of the light sources which would cause glares in the camera optics. In addition, sharp shadows and unwanted highlights on the recorded objects are prevented. For flexibility, 3 spotlights are also available that can be placed at arbitrary locations in the room.

7. Computer Infrastructure

For data processing, the studio is equipped wit 4 standard PCs that feature AMD 1 GHz Athlon$^{\text{TM}}$ CPUs, 768 MB of main memory and graphics cards with Nvidia GeForce3$^{\text{TM}}$ GPUs. Each computer has 3 IEEE1394 connectors. The operating system used is Linux with kernel version 2.2.18. The computers are connected via a 100 MBit/s Ethernet network. In the current setup, one PC is connected to two of the video cameras.

The computing equipment is flexible enough to allow different software architectures for the different applications. A common design principle is to exploit parallelism by using a client-server setup. The client computers acquire and preprocess the video frames in real-time. Final processing, such as 3D model reconstruction and visualization, is done on the server computer. The development of client-server systems is supported by the standard software environment in the studio (Sect. 8). The specific details of the software architectures for different research projects will be given in Sect. 9.

For synchronization of the cameras, a control box was built that distributes a trigger pulse from the parallel port of a host computer to up to 8 cameras. For trigger control, a Linux kernel driver was implemented.

8. Software Library

A standard software library was developed that allows applications in the studio to interface the available hardware components. For control of the cameras a high-level C++ library based on the libdc1394 [25] open source IEEE1394 package is available. Thread classes for asynchronous reading of camera frames and visualization of images on the screen are also provided. A standard set of image processing classes based on the Intel Image Processing Library [8] and the Open Source computer vision Library [9] is available for inclusion into the application code. A tool for camera calibration using Tsai's method [24] is available. Code for correction of first order radial lens distortion effects [10] as they are recovered during calibration is also provided.

For the development of client-server software that controls the cameras and processes the data, a set of reference implementations for sending and receiving of image and volume data is available. The implementation of client and server classes is based on the Adaptive Communication Environment (ACE) [21].

An important component of the basic software library are the background subtraction classes. We implemented a subtraction scheme for general backgrounds based on pixel color statistics originally proposed in [4]. The algorithm computes mean color and standard deviation of each background pixel from a sequence of images without a foreground object. Foreground pixels are identified by a large deviation of their color from the background statistics. If no additional criterion is applied, shadow pixels are wrongly classified as foreground. Shadows are characterized by a large intensity difference compared to the background, but only a small difference in hue. This criterion is applied to minimize the number of wrongly classified pixels in shadow.

The combination of the studio standard software and hardware makes possible the easy implementation of scalable application prototypes. Using a client-server architecture, systems are easily extendable by inclusion of more clients. Experiments can be done with a large number of different camera setups without major changes in the application code.

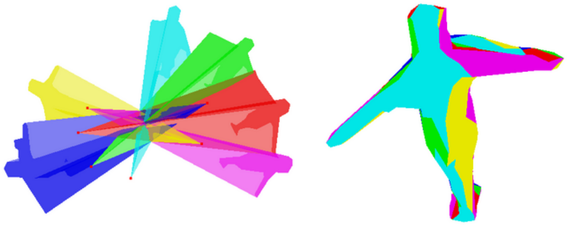

Figure 2: *Visual hull reconstruction. Six cones are generated from silhouette images taken from different viewpoints (l). Reconstructed 3D Object (r).*

9. Applications

In this section, the functionality of the room is demonstrated using research projects that were undertaken with the available equipment.

9.1. Visual Hull Reconstruction

Silhouettes of 3D objects provide strong cues for reconstructing 3D geometry from 2D images. The reconstruction method is well known as *shape-from-silhouette* [20]. Laurentini introduces the visual hull concept [12] to characterize the best geometry approximation that can be achieved by this method. Theoretically, the visual hull must be reconstructed using silhouette images from all possible viewpoints. However, in practice, we are limited to employ only a finite number of available images.

The basic principle of visual hull computation is fairly intuitive. Since the viewing information associated with each silhouette image is already known in the camera calibration step, we are able to back-project each silhouette into 3D space. This produces a generalized cone containing the actual 3D object in question. The intersection of such cones from all reference views gives an approximation of the visual hull. Fig. 2 illustrates this process.

There are two different approaches for visual hull reconstruction: volumetric and polyhedral. The former one tessellates a confined 3D space into voxels. Then each voxel is projected onto every reference image plane. The voxel, whose projection falls outside of any silhouette, will be carved away. The later approach first extracts 2D contours from silhouette image. Then explicit polyhedral representations of generalized cones are generated from all silhouette contours. Finally, by performing 3D intersection of these cones, a polyhedral visual hull is reconstructed.

We have implemented both of the above approaches in a distributed real-time client-server system (Fig. 3, see also Sect. 9.2). The silhouette images are obtained using the background subtraction classes explained in Sect. 8 and transferred to the server. An open source library [1] is applied to compute 3D polyhedral intersections. Using 6 cameras, the volumetric reconstruction runs at interactive frame rates (about 8-10 fps), whereas polyhedral reconstruction is achieved in real time (about 25fps). Using a new method exploiting off-the-shelf graphics hardware, polyhedral hulls can be reconstructed at up to 40 fps [13]. (Fig. 4). The system scales easily to a higher number of client computers, a hierarchical network structure is also possible for larger systems. The acquisition environment enables robust background subtraction while preserving a natural appearance of the scene. The implementation makes optimal use of the available software components.

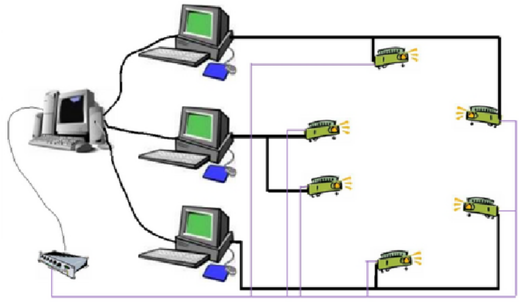

Figure 3: *Client-server architecture of the visual hull reconstruction system.*

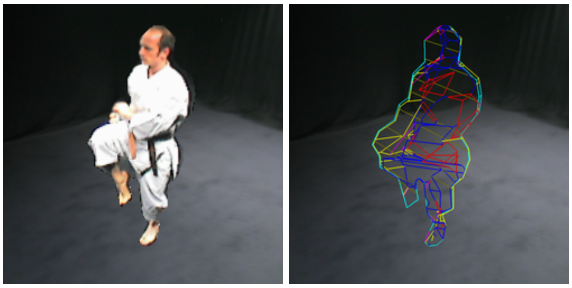

Figure 4: *Textured polygonal visual hull (l) and underlying polygons (r).*

9.2. Human Motion Capture

Video-based Human motion capture is the task of acquiring the parameters of human motion from video sequences of a moving person. In our work, we focus on the model-based acquisition of human motion without the application of optical markers on the person's body. In [23], we presented a system that combines the volumetric reconstruction of a person's volume from multiple silhouettes with a color-based feature tracking to fit a multi-layer kinematic skeleton model to the motion data at interactive frame rates. In Fig. 5, an overview of the motion capture system architecture is given.

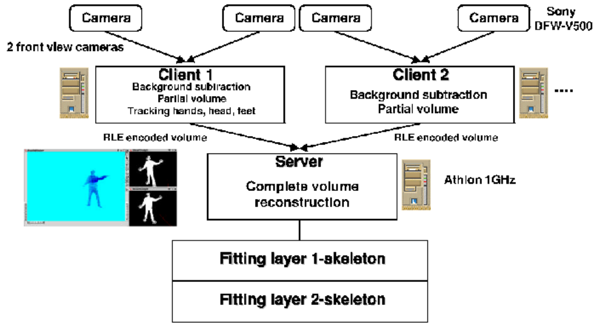

Figure 5: *Motion capture client-server architecture.*

The software is implemented as a distributed client-server application. Two cameras are connected to one PC on which the client application is running. Each client computer records video frames at a resolution of 320x240 pixels, and performs a background subtraction (Sect. 8) and silhouette computation in real-time. Furthermore, each client machine computes a partial, voxel-based visual hull from the two silhouette views available to it. To do this, the scene is subdivided into a regular grid of volume elements (voxels). Each of the voxels is projected into the silhouette views of the cameras and classified as occupied space if it projects into the foreground silhouette. This step is very fast since, in our static camera setup, the projections can be precomputed and stored in a lookup-table. By this step, a volumetric approximation to the visual hull is computed on each client computer that is run-length-encoded and transmitted to the server application.

On the client computer controlling the 2 cameras that observe the person from font, color-based trackers are running that follow the motion of the head, the hands and the feet. Via triangulation, the 3D locations of these features are computed and also transferred to the server. The homogeneous lighting in the studio minimizes color variations of hands, head and feet due to position changes of the body. This way, color-based tracking works robustly in a large area of the studio.

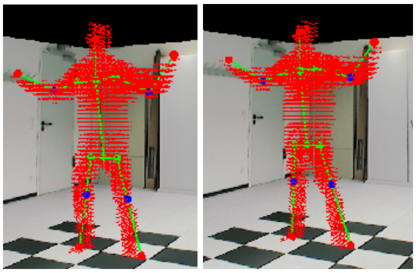

Figure 6: *Skeleton fitted to visual hull of a moving person.*

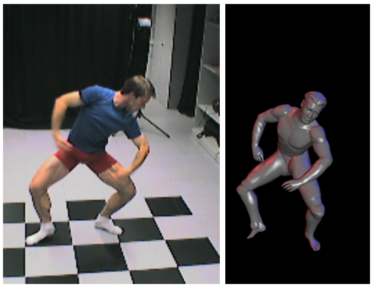

Figure 7: *An input video frame (l) and the body model pose recovered by the motion capture algorithm (r).*

On the server, the complete visual hull is reconstructed by intersection of the partial volumes. In a separate step, a two-layer kinematic skeleton is fitted to the motion data using the reconstructed 3D feature locations and the voxel-based volumes.

In Fig. 6 results obtained with our system are shown. The depicted visual hull was reconstructed from 4 camera views. The complete client-server system performing background subtraction, visual hull reconstruction, feature tracking and visual hull rendering can run at approximately 6-7 fps for a 64^3 voxel volume. For an average motion sequence, a fitting frame rate of 1-2 fps is achieved. Due to the combination of feature tracking and volumetric fitting, the system becomes more robust against problems that are typically observed with visual hull reconstruction approaches. Even in the presence of phantom volumes that are due to insufficient visibility, the fitting procedure can correctly recover the right body pose.

The motion capture system can efficiently make use of the resources available in camera studio. Finding good camera positions to minimize reconstruction artifacts in the volumes requires frequent repositioning of the cameras which is made easy by the telescope poles. The studio setup also guarantees that the background subtraction quality will not deteriorate in any new camera position. By relying on common software library components, the combination of the visual hull reconstruction with the model fitting step was simplified. The system is also scalable to a higher number of clients.

9.3. Free-Viewpoint Video

The goal of free-viewpoint video is to build a three-dimensional, time varying representation of a dynamic scene that was previously recorded by multiple video cameras. During playback of this three-dimensional video, the viewer can interactively choose his viewpoint onto the scene.

In our camera studio, we test a new model-based approach for recording, reconstructing and rendering of free-viewpoint video of human actors [3]. The system employs a generic human body model whose motion parameters are acquired by means of a marker-less silhouette-based human motion capture algorithm. For realistic modeling of the time-varying surface appearance of the actor, a multi-view texturing method is applied that reconstructs surface textures from camera images.

The production of a free-viewpoint video can be separated into three steps. The first step is the acquisition of the multi-view video material. Whereas for the previous two applications real-time recording and processing of the video streams is needed, in this scenario, off-line recording and efficient streaming to disk is required. The applied hardware setup is similar to the architecture depicted in Fig. 3. The implementation of the recording software is straightforward since the standard studio library provides almost all the necessary components. The client computers read out the video frames from the two connected cameras and stream them directly to disk. A server that simultaneously runs a client sends the trigger signals to all cameras. At a resolution of 320x240 pixels, 15 fps are achieved when recording with 8 cameras and 4 client computers. The video frames are written to disk in uncompressed raw RGB format.

The next step is the model-based reconstruction of the free-viewpoint video sequence. The main part of this step is a silhouette-based motion parameter estimation algorithm that applies features of latest consumer graphics hardware during tracking. In an initialization step, the generic human body model consisting of a triangle mesh surface representation and an underlying kinematic skeleton is adapted to the silhouettes of the person standing in a special pose. At each time step of the video, motion parameter estimation is done by means of a hierarchical optimization procedure that maximizes the overlap between the model and image silhouettes. The error metric to measure the fit is efficiently computed in graphics hardware.

The last step is the playback of the 3D video sequence. During playback, time-varying textures that are reconstructed from all available cameras are attached to the human body model. Rendering of the scene can be done in real-time and more than 30 fps can be easily achieved.

Fig. 7 shows one of the input camera views out of a sequence taken from 7 camera perspectives. The corresponding pose of the body model as it is found by the motion capture algorithm is also shown. Fig. 8 shows two screenshots of the renderer displaying a new viewpoint onto a reconstructed dynamic scene. Interactive editing of the recorded video by changing the body pose is also possible as it is shown in the left image.

The visual results of the rendering and the results of the motion capture prove, that the employed camera and lighting equipment are efficient in several aspects. The appearance of the surface textures is very natural while, at the same time, the background subtraction still works robustly enough to produce high-quality silhouettes.

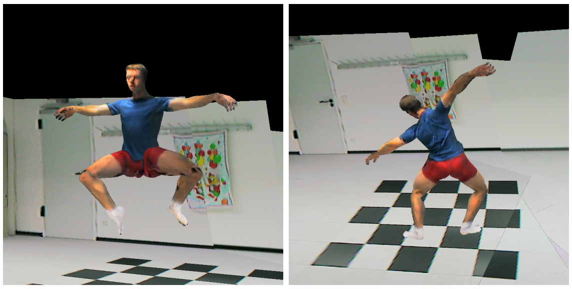

Figure 8: *Bullet-time pose as know from feature films created by manual editing (l), screenshot of a dancing sequence from a new camera position (r).*

10. Discussion

The results obtained with the previously described software systems show that the acquisition room provides a flexible infrastructure to develop conceptually different surround vision applications. The physical layout of the room allows the recording of comparably large dynamic scenes from multiple camera views. The commitment to a common basic software library simplifies the reuse of software components for other research projects.

11. Conclusion and Future Work

In this paper, the design and construction of an acquisition room for multi-view video is described. The design requirements are explained and their implementation in the different studio sub-systems analyzed. It is explained why building a room for multi-view video recording involves efficient solutions to different hardware and software problems. The amount of data to be processed is huge, hence an efficient computing subsystem is necessary that also provides sufficient I/O-bandwidth. The cameras have to fulfill the necessary frame rate, resolution and configurability requirements. The lighting equipment has to provide sufficient illumination for a natural scene appearance while the robustness of the background subtraction method must not be worsened.

We demonstrated the efficiency of our multi-view video studio by means of different research projects. The camera room proves to be a flexible environment for real-time acquisition and processing of multi-view video streams, as well as for off-line recording. The available hardware and software infrastructure enables the fast development of client-server software systems based on standard software components. The provided standard class library of image processing tools greatly simplifies the development of application prototypes. The application of off-the-shelf hardware keeps the cost of the whole environment moderate while not compromising versatility.

In the future, the acquisition room will be used, e.g., to record sample data that will also be made available to MPEG community that is currently in the process of developing a standard for 3D video. An extension of the static setup to a mobile setup by using several notebook computers is also being investigated.

References

1. P. Bekaert and O. Ceulemans. Boundary representation library. http://breplibrary.sourceforge.net.
2. E. Borovikov and L. Davis. A dristibuted system for real-time volume reconstruction. In *Proceedings of Intl. Workshop on Computer Architectures for Machine Perception*, page 183ff, 2000.
3. J. Carranza, C. Theobalt, M. Magnor, and H.P. Seidel. Free-viewpoint video of human actors. In *Proc. of SIGGRAPH03*, to appear. ACM, 2003.
4. K.M. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3D voxel reconstruction of human motions. In *Proc. of CVPR*, volume 2, pages 714 – 720, June 2000.
5. D.M. Gavrila and L.S. Davis. 3D model-based tracking of humans in action: A multi-view approach. In *Proc. of CVPR*, pages 73–80, 1996.
6. M. Goesele, H. Lensch, W. Heidrich, and H.P. Seidel. Building a photo studio for measurement purposes. In *Proceedings of VMV2000*, pages 231–238, 2000.
7. T. Horprasert, I. Haritaoglu, D. Harwood, L. Davis, C. Wren, and A. Pentland. Real-time 3d motion capture. In *Second Workshop on Perceptual Interfaces*, 1998.
8. Intel. IPL - Intel image processing library. http://www.intel.com/support/performancetools/libraries/ipl/.
9. Intel. Open source computer vision library. http://www.sourceforge.net/projects/opencvlibrary, 2002.
10. R. Jain, R. Kasturi, and B.G. Schunck. *Machine Vision*. McGraw-Hill, 1995.
11. T. Kanade, H. Saito, and S. Vedula. The 3d room : Digitizing time-varying 3d events by synchronized multiple video streams. Technical Report CMU-RI-TR-98-34, Robotics Institute - Carnegie Mellon University, 1998.
12. A. Laurentini. The visual hull concept for silhouette-based image understanding. *PAMI*, 16(2):150–162, February 1994.
13. M. Li, M. Magnor, and H.P. Seidel. Hardware-accelerated visual hull reconstruction and rendering. In *Proc. of Graphics Interface*. to appear, 2003.
14. J. Luck and D. Small. Real-time markerless motion tracking using linked kinematic chains. In *Proc. of CVPRIP*, 2002.
15. Manfrotto. http://www.manfrotto.com.
16. T. Matsuyama and T. Takai. Generation, visualization, and editing of 3D video. In *Proc. of 3DPVT'02*, page 234ff, 2002.
17. W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for real-time rendering. In *Proc. of 12th Eurographics Workshop on Rendering*, pages 116–126, 2001.
18. W. Matusik, C. Buehler, R. Raskar, S.J. Gortler, and L. McMillan. Image-based visual hulls. In *Proc. of SIGGRAPH00*, pages 369–374, 2000.
19. A. Menache. *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann, 1995.
20. M. Potmesil. Generating octree models of 3D objects from their silhouettes in a sequence of images. *CVGIP*, 40:1-20, 1987.
21. Several. ACE - ADAPTIVE communication environment. http://www.cs.wustl.edu/˜schmidt/ACE.html.
22. Siteco. http://www.siteco.de.
23. C. Theobalt, M. Magnor, P. Schueler, and H.P. Seidel. Combining 2D feature tracking and volume reconstruction for online video-based human motion capture. In *Proceedings of Pacific Graphics 2002*, pages 96–103, 2002.
24. R.Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *Proc. of CVPR*, pages 364–374, June 1986.
25. C. Urmson, D. Dennedy, D. Douxchamps, G. Peters, O. Ronneberger, and T. Evers. libdc1394 - open source IEEE1394 library. http://www.sourceforge.net/projects/libdc1394.
26. G.J. Ward. Measuring and modeling anisotropic reflection. In *Proceedings of SIGGRAPH92*, pages 265–272, 1992.
27. S. Weik and C.-E. Liedtke. Hierarchical 3d pose estimation for articulated human body models from a sequence of volume data. In *Robot Vision*, 2001.
28. S. Wuermlin, E. Lamboray, O.G. Staadt, and M.H. Gross. 3d video recorder. In *Proc. of Pacific Graphics 2002,*, pages 325–334, 2002.