ViPiD - Virtual 3D Person Models for Intuitive Dialog Systems

Patrick Klie Torsten Büschenfeld Jörn Ostermann
University of Hannover
Institut für Informationsverarbeitung
Appelstr. 9A, 30167 Hannover, Germany
Email: {klie|bfeld}@tnt.uni-hannover.de

***Abstract*— ViPiD is a complete framework for audio and 3D video capturing of one or several moving persons as well as the creation of 3D person models for intuitive dialog systems. Therefore we are setting up a multi-camera environment for 3D scene analysis, incorporating aspects such as 3D/4D reconstruction, motion estimation, virtual camera integration, coding of time variant 3D meshes and free viewpoint video.**

I. Introduction

Free viewpoint video is one of the major research fields in terms of convergence of computer graphics and computer vision. It is especially related to threedimensional television. We propose an architecture essentially consisting of a multi-camera system of 24 cameras and a stereoscopic display. The motion capturing process, which is concentrated on one or more moving foreground objects, will be done in a marker-free fashion.

This paper is structured as follows: Section II describes the overall system architecture including the cameras, synchronization, network architecture and the configurable cabin and chromatte background. Section III explains the algorithms to generate the virtual 3D person models.

A. Related Work

Many multi-camera systems have been built up in the past including appropriate software for acquisition, reconstruction, coding and rendering purposes, for instance at the Max-Planck-Institut für Informatik in Saarbrücken ([1], [2], [3], [4], [5]), at the ETH Zurich ([6], [7], [8], [9]), at the Microsoft Interactive Visual Media Group [10] and at the Stanford university [11].

II. System Architecture

A. Cameras

As depicted in Figure 1, the core of the multi-camera system is formed by 20 Prosilica EC1380C FireWire cameras, which are connected to 5 linux-driven servers, PS1 to PS5, by PCI-X 1394b/FireWire QuikFire iDT804PCI host adapters, with 4 external connectors. The cameras are equipped with Schneider CNG 1,4/8-0512, CM120 Cinegon lenses.

Additionally, we have two Thomson LDK6000MKII HD-cameras with Canon YJ12x6,5BKRS lenses and two Canon XL H1 HD/SD-cameras.

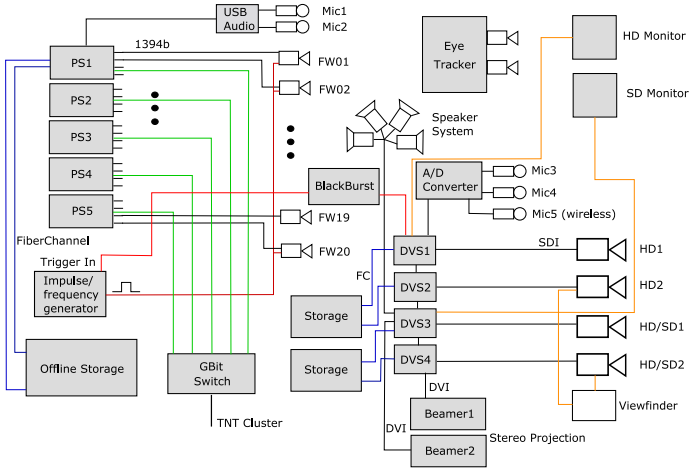


Fig. 1. Block diagram of the system architecture

B. Synchronization by external triggering

The FireWire cameras, as well as the HD/SD cameras, can be triggered externally, making network synchronization by FireWire or Ethernet obsolete. The trigger itself is a configurable frequency/impulse generator. An additional Kramer Blackburstgenerator enables the FireWire cameras to be synchronized together with the HD and SD cameras to extend the multi-camera system to 24 cameras.

C. Network architecture

Each of the five servers connected to the FireWire cameras has the following specifications: Dual Opteron 265 Dual-Core with 1.8GHz, 2GB RAM, eight hard drives with 250GB SATA configured as RAID0, one NX7800GTX PCIe graphics card and one 2GBit Fiber Channel Controller. Server PS1 is connected to one DVS Near-Line 4.0TB, 16x SATA HDD offline storage via two Fiber Channel connectors.

The two HD and two HD/SD cameras are connected to four Windows-driven servers, DVS1 to DVS4, with the following specifications: ProntoHD-SAN, Dual Xeon 3GHz, 2GB RAM and internal hard drive with 150GB. Two FC-SAN 16x70GB FC-HDD RAID0 offline storages are connected to the DVS-servers.

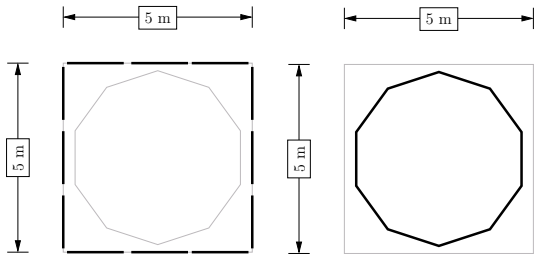
The five PS-servers and the DVS servers are connected to

a Gigaethernet Baseline-Switch to link each other and the rest of the workstations in the institute's linux cluster.

Using all cameras simultaneously in a 1K-resolution (1360x768) with 25 fps, the recording capacity of all storages summed up is around 102 minutes of uncompressed color video material. 143 minutes can be recorded when using one HD-camera in 1080p mode with 29.98 fps and 12 bit color resolution using the two FC-SAN offline storages (178 minutes with 24 fps).

D. Configurable cabin and chromatte background

Such a multi-camera environment must be flexible, in order to arrange several different setups. Our setup consists of twelve wall segments. The segments can be configured e.g. enclosing a rectangle with 3×3 segments (i.e. 5 m $\times$ 5 m) in size (Fig. 3). Another setup is a circular arrangement of the cabin with approximately 5 m diameter.



(a) Rectangular setup (b) Circular setup

Fig. 3. Top view of two sample setups of the cabin

Figure 2 shows important aspects of our environment. The rendered image depicts the circular setup. Cameras can be mounted flexibly in different positions as the left part of the image indicates. Yellow arrows show possible mounting points. In order to not restrict the vertical position of the camera in the center (emphasized in red), wall elements E_1 and E_2 can be turned by 90 degrees. Thus, free horizontal movement is possible (Fig. 4).

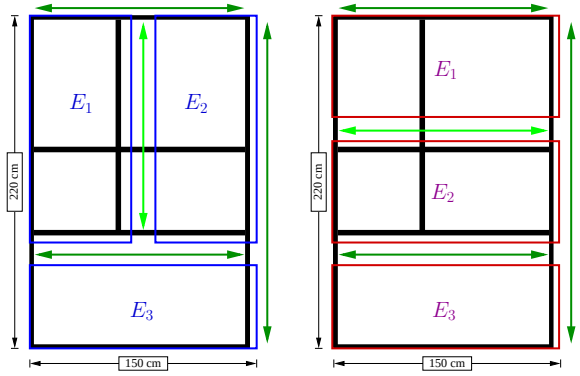


Fig. 4. Possible mount positions (green arrows) for cameras. Wall elements E_1 and E_2 can be turned by 90 degrees to either allow vertical or horizontal flexibility of camera in the center.

Each of the wall segments consists of an aluminum frame and three wooden elements (E_1, E_2, E_3). Two setups (blue and red) are possible in order to allow free vertical (blue) or horizontal (red) movement of the centrally mounted camera.

The magnified area at the bottom left of Figure 2 shows the camera mount in detail. The installation allows all three rotatory degrees of freedom. The rotational axes of the attachment are denoted by the yellow lines.

A major problem when focusing on 3D reconstruction of objects is segmentation. The object has to be separated from the background. We address this problem by using a special cloth that covers the cabin elements. This so called *Chromatte*™ can be seen in the left part of Figure 2 (dark grey cloth). In the final setup, the cloths of the elements are connected by Velcro® fastener.

Illumination of the Chromatte™ with green LEDs causes reflection only in the direction of the incoming light. Therefore, each camera is equipped with a ring of LEDs (see second magnification) to yield optimal background reflection.

The effect is shown in the pictures at the bottom right. Scattered light is reduced resulting in reliable segmentation.

E. Treadmill

The space to perform movement is limited to 5x5 meters in the basic configuration. Since the focus range of our FireWire cameras does not exceed 2 meters, we simulate movements with a treadmill placed into the center of the cabin. The movements are especially related to those of a walking or running person.

F. Miscellaneous

An Eyegaze Analysis System with a binocular eyetracking extension and the NYAN - eyegaze analysis suite is used to make subjective tests on image, video and 3D quality.

Audio capturing is done with four AudioTechnica AT835b microphones and one Sennheiser Bodypac wireless microphone. A M-Audio Fast Trak Pro USB-Audio Device, an Analog-AES/EBU Mindprint AN/DI Pro Audio A/D Converter and a Mutec MC1.1 digital audio format converter, will do the necessary audio conversions.

The recordings of the HD and SD cameras can be inspected with a Sony BVM-2016P 20" RGB Y/C SD-Monitor, a JVC DT-V1910CG SDI-Monitor and a Viewfinder. The results can be displayed on a stereo projection BlackScreen XPR3/3D.

III. Generation of Virtual 3D Persons

The generation of the virtual 3D persons will be made in several steps that are described in the following and depicted in Figure 5.

A. Multi-camera calibration

To enable metric reconstruction of 3D scenes, camera calibration is an important issue. Since calibration has to be done for the whole working volume, classical calibration patterns are not well suited for this task. We use a freely available software for multi-camera calibration [12]. Moving a point light within the working volume is used to simultaneously calibrate all cameras. Ihrke et al. [13] describe another approach, using a bright ball as reference object for calibration.

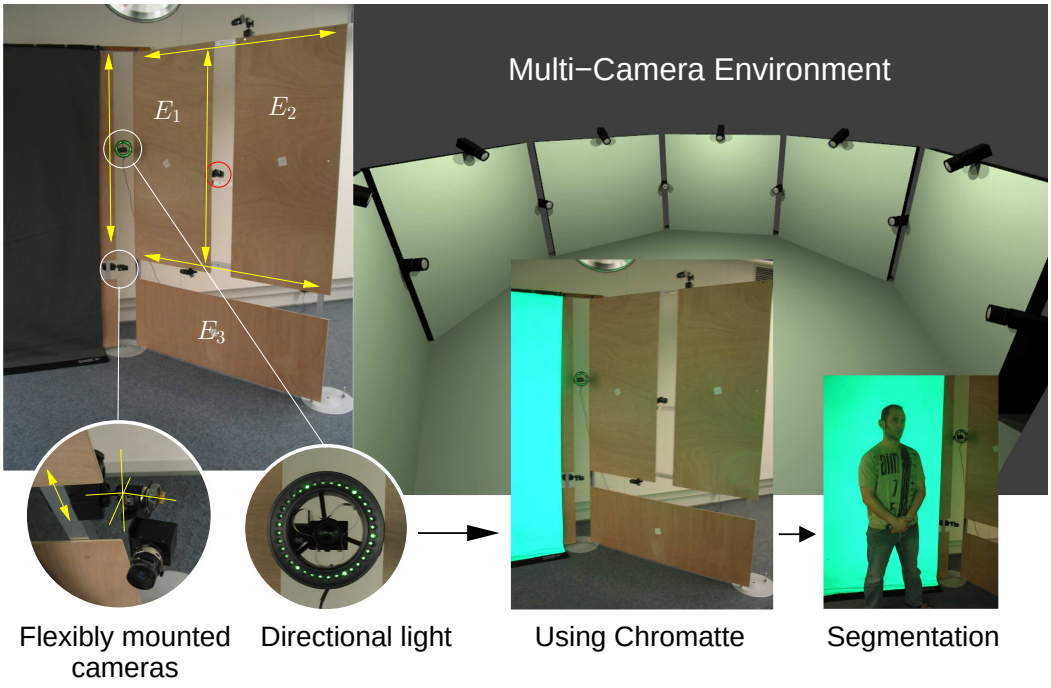


Fig. 2. This figure shows the assembly of our multi-camera environment. The final setup is depicted by the rendered picture. The left image shows the actual construction, emphasizing the flexibly mounted cameras and *Chromatte*™ cloth, improving segmentation (bottom right).

The point light is detected in each frame of the calibration sequence. Projection matrices and vectors of the projected points in 3D space compose the so called *scaled measurement matrix*. By utilizing the known projections and filling up missing points due to occlusion, the scaled measurement matrix can be factorized to obtain camera projection matrices.

B. Video Preprocessing

The chroma keying, segmentation, color calibration, radial undistortion and depth/disparity estimation will be done in the vertex and fragment shaders of the 7800GTX graphics cards additionally to CPU solutions to reduce the entire processing time. The shading language used will be GLSL.

The chroma keying step simplifies to thresholding in color space due to the usage of Chromatte™ which immediately yields the segmentation as an alpha mask or video data augmented by an alpha channel. Potential discoloring at the foreground borders is expected to be minimal but will be further reduced by despilling routines.

The different cameras have differing color response functions. To avoid color inconsistencies that would lead to texture artifacts on the 3D-models a color calibration for the different views will be applied. For this color calibration an approach similar to [14] will be used deploying linear least squares matching, rgb to rgb transforms or even general polynomial transforms directly on the color distributions. This process can be aided by the usage of color patterns such as the *GretagMacbeth [15] ColorChecker*™ providing well defined colors.

C. Shape-from-silhouette and depth maps

The result of the video preprocessing step consists of one or more segmented foreground objects, which can be processed by classical shape-from-silhouette techniques, to obtain 3D models as triangulated surfaces [16]. Silhouette intersection gives a very coarse representation of the objects. Mesh closeness can be further enhanced by applying dense depth map results to the silhouette to ”carve out” concavities [17].

D. Time-consistent dynamic 3D mesh generation and coding

Reconstruction will first be done statically producing one single 3D mesh for every time step. These meshes will have different connectivity from one time step to the next one. For coding and streaming purposes, these meshes are converted to dynamic mesh sequences with constant connectivity, to reduce bitrates. This time-consistency is achieved by combining feature point matching and tracking together, with a multi-view optical flow variant to obtain three-dimensional scene flow as proposed in [18] and mesh parameterizations (compare e.g. [19], [20], [21]), to perform the remeshing of current, with already existing, connectivity. Dynamic mesh coding is finally done with an existing connectivity-guided predictive compression scheme [22].

E. Multi-view texturing and dynamic surface light fields

To get realistic results, the dynamic mesh sequences will be textured in a multi-view fashion. These textures can be further enhanced by deploying dynamic surface light fields proposed in [23]. In this case, the plenoptic function consists of time

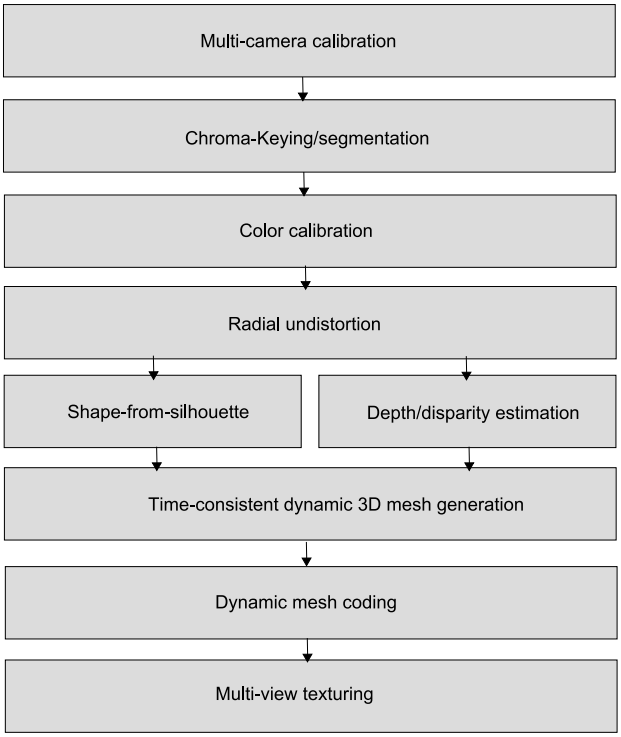


Fig. 5. Flow chart of the algorithms

varying rays emanating from surface points. To keep storage consumption controllable, a coarsification of the 3D mesh will be used as a base domain.

ACKNOWLEDGMENT

The authors would like to thank the EC IST 6th Framework 3DTV NoE for partially funding this work. Additional thanks go to the creators of the Openmesh library [24] providing a polygon mesh data structure, the creators and contributors of the OpenCV library [25], to Svodoba et al. for publishing their multi-camera self-calibration, to Robert B Davies for his matrix library Newmat [26] and to Gernot Ziegler for fruitful discussions about shader programming.

REFERENCES

[1] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel, "Free-viewpoint video of human actors," *ACM Trans. on Computer Graphics*, vol. 22, July 2003.

[2] B. Goldlücke, M. Magnor, and B. Wilburn, "Hardware-accelerated dynamic light field rendering," in *Proceedings Vision, Modeling and Visualization VMV 2002* (G. Greiner, H. Niemann, T. Ertl, B. Girod, and H.-P. Seidel, eds.), (Erlangen, Germany), pp. 455–462, aka, November 2002.

[3] C. Theobalt, G. Ziegler, M. Magnor, and H.-P. Seidel, "Model-based free-viewpoint video acquisition, rendering and encoding," *Proc. Picture Coding Symposium (PCS'04)*, San Francisco, USA, pp. 1–6, Dec. 2004.

[4] N. Ahmed, E. de Aguiar, C. Theobalt, M. Magnor, and H.-P. Seidel, "Automatic generation of personalized human avatars from multi-view video," *Proc. ACM Symposium on Virtual Reality Software and Technology (VRST'05)*, Monterey, USA, pp. –, Nov. 2005.

[5] M. Magnor and B. Goldlücke, "Spacetime-coherent geometry reconstruction from multiple video streams," *Proc. IEEE 3D Data Processing, Visualization, and Transmission (3DPVT'04)*, Thessaloniki, Greece, pp. 365–372, Sept. 2004.

[6] S. Würmlin, E. Lamboray, O. Staadt, and M. Gross, "3d video recorder: A system for recording and playing free-viewpoint video," in *Computer Graphics Forum*, vol. 22, Blackwell Publishing Ltd, Oxford, U.K., 2003.

[7] M. Gross, S. Würmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Meier-Koller, T. Svoboda, L. V. Gool, S. Lang, K. Strehlke, A. V. Moere, and O. Staadt, "blue-c: A spatially immersive display and 3d video portal for telepresence," *ACM Transactions on Graphics Proceedings of ACM SIGGRAPH 2003*, 2003.

[8] S. Würmlin, E. Lamboray, and M. Gross, "3d video fragments: Dynamic point samples for real-time free-viewpoint video," *Computers and Graphics*, vol. 28, no. 1, pp. 3–14, 2004.

[9] M. Waschbüsch, S. Würmlin, D. Cotting, F. Sadlo, and M. Gross, "Scalable 3d video of dynamic scenes," *The Visual Computer Proceedings of Pacific Graphics 2005*, vol. 21, pp. 629–638, October 2005.

[10] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, 2004.

[11] B. Wilburn, M. Smulski, K. Lee, and M. A. Horowitz, "The light field video camera," in *Proceedings of Media Processors 2002, SPIE Electronic Imaging*, 2002.

[12] T. Svoboda, D. Martinec, and T. Pajdla, "A convenient multi-camera self-calibration for virtual environments," *PRESENCE: Teleoperators and Virtual Environments*, vol. 14, pp. 407–422, August 2005.

[13] I. Ihrke, L. Ahrenberg, and M. Magnor, "External camera calibration for synchronized multi-video systems," in *WSCG '2004 : the 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2004 ; short communication papers proceedings*, vol. 12 of *Journal of WSCG*, (Plzen, Czech Republic), pp. 537–544, UNION Agency, February 2004.

[14] A. Ilie and G. Welch, "Ensuring color consistency across multiple cameras," in *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, (Washington, DC, USA), pp. 1268–1275, IEEE Computer Society, 2005.

[15] GretagMacbeth Color Management Solutions. `http://www.gretagmacbeth.com`.

[16] G. Eckert, J. Wingbermühle, and W. Niem, "Mesh based shape refinement for reconstructing 3d-objects from multiple images," in *Proc. 1st European Conference on Visual Media Production*, 2004.

[17] L. Falkenhagen and T. Wedi, "Improving block-based disparity estimation by considering the non-uniform distribution of the estimation error," in *3D Structure from Multiple Images of Large-Scale Environments"*, (Freiburg, Germany), pp. 93–108, proceedings of SMILE workshop, June 1998.

[18] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *ICCV (2)*, pp. 722–729, 1999.

[19] E. Praun, W. Sweldens, and P. Schröder, "Consistent mesh parameterizations," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, (New York, NY, USA), pp. 179–184, ACM Press, 2001.

[20] K. Hormann and G. Greiner, "MIPS: An efficient global parametrization method," in *Curve and Surface Design: Saint-Malo 1999* (P.-J. Laurent, P. Sablonnière, and L. L. Schumaker, eds.), Innovations in Applied Mathematics, pp. 153–162, Nashville, TN: Vanderbilt University Press, 2000.

[21] S. Yoshizawa, A. Belyaev, and H.-P. Seidel, "A fast and simple stretch-minimizing mesh parameterization," in *Shape Modeling International 2004 (SMI 2004)* (F. Giannini and A. Pasko, eds.), (Genova, Italy), pp. 200–208, CNR,Aim@Shape, IEEE, June 2004.

[22] N. Stefanoski and J. Ostermann, "Connectivity-guided predictive compression of dynamic 3d meshes," in *ICIP 2006*, October 2006.

[23] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, "Surface light fields for 3D photography," in *Siggraph 2000, Computer Graphics Proceedings* (K. Akeley, ed.), pp. 287–296, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[24] M. Botsch, S. Steinberg, S. Bischoff, and L. Kobbelt, "Openmesh – a generic and efficient polygon mesh data structure," 2002.

[25] Intel Corporation. Open Source Computer Vision Library. `http://www.intel.com/technology/computing/opencv/`.

[26] R. B. Davies, "Writing a matrix package in c++," in *OON-SKI'94: The second annual object-oriented numerics conference*, 1994.