07 - 10 September 2009

PROCEEDINGS

54. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

Information Technology and Electrical Engineering - Devices and Systems, Materials and Technologies for the Future

Faculty of Electrical Engineering and Information Technology

Startseite / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=14089

TECHNISCHE UNIVERSITÄT
ILMENAU

Impressum

Herausgeber: Der Rektor der Technischen Universität Ilmenau
Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c.
Peter Scharff

Redaktion: Referat Marketing
Andrea Schneider

Fakultät für Elektrotechnik und Informationstechnik
Univ.-Prof. Dr.-Ing. Frank Berger

Redaktionsschluss: 17. August 2009

Technische Realisierung (USB-Flash-Ausgabe):
Institut für Medientechnik an der TU Ilmenau
Dipl.-Ing. Christian Weigel
Dipl.-Ing. Helge Drumm

Technische Realisierung (Online-Ausgabe):
Universitätsbibliothek Ilmenau
ilmedia
Postfach 10 05 65
98684 Ilmenau

Verlag:
Verlag ISLE, Betriebsstätte des ISLE e.V.
Werner-von-Siemens-Str. 16
98693 Ilmenau

ISBN (USB-Flash-Ausgabe): 978-3-938843-45-1
ISBN (Druckausgabe der Kurzfassungen): 978-3-938843-44-4

Startseite / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=14089

INTERACTIVE 3D VIDEO FOR MULTIMEDIA APPLICATIONS - CONCEPTS AND SYSTEM

Christian Weigel, Sebastian Schwarz, Torsten Korn, Martin Wallbohr

Institute for Media Technology, Ilmenau University of Technology
Postfach 10 05 65, 98684, Ilmenau, Germany
email: christian.weigel@tu-ilmenau.de

ABSTRACT

In this paper we present an approach for interactive free viewpoint video (also called 3D video objects) which, in its most simple form, only relies on two cameras to synthesize novel views. The algorithms we use for analysis and synthesis are implemented using a customizable high performance software system. Due to a modular approach we can implement a number of multimedia applications. We show the possibilities of the system with an example where a 3D video object is embedded into a synthetic scene.

Index Terms— Free Viewpoint Video, 3D Video, 3DTV

1. INTRODUCTION

At the moment 3D video is a widely discussed topic in the research community. Besides stereoscopic presentation which is already entering the movie and TV market so called free viewpoint video gains a lot of interest as well. The idea behind free viewpoint video is to record a scene with a number of cameras in order to get a depth representation of the scene which is used to synthesize novel views of it. A number of common extraction techniques for 3D objects do have a computer graphic mesh as output [1] [2] [3]. These meshes are rendered using dynamic texturing with the RGB-images from the cameras. To obtain good result with this method the input must consist of images from at least 8 cameras. In contrast we only use pixel data to synthesize novel views of the scenery.

In [4] we presented an algorithm for rendering free viewpoint video that uses information from a stereo camera pair only. Compared to the model-based approaches we used an image-based approach. Although not sufficient for covering the whole viewing volume around the object we could achieve good results using extrapolation based on the trifocal transfer firstly introduced in [5] while there was much less effort for capturing the object compared to other techniques. In section 2 of this paper we review the preprocessing and synthesis steps. The extension of the algorithm to handle additional stereo pairs and the decision scheme which stereo pair is used to synthesize the virtual view is presented in section 3. The interactive viewing as well as the combination of the free viewpoint object with a synthetic scene is described in section 4 followed by some result given in section 5.

2. IMAGE-BASED FREE VIEWPOINT VIDEO FROM STEREO

Obviously using the information from only two cameras is not sufficient to get full virtual view latitude around the object. But they already provide enough input data to achieve fair virtual view latitude at positions close to the stereo pair. For some application this is an adequate compromise. To obtain the representation we need for interactive rendering form a stereo setup the following steps are necessary: 1. stereo camera calibration, 2. image rectification, 3.disparity estimation.

The camera calibration step has two purposes, the estimation of the rectification matrices for a stereo image and the estimation of external and internal camera parameters used for interactive rendering respectively.

Having only a stereo pair as input, we need to arrange the cameras in a convergent setup to obtain more image information. In this setup the optical axes of the cameras approximately intersect at a point within the object that is recorded. In order to simplify the subsequent step of finding corresponding pixels across both of the images they need to be rectified. Briefly summarized the rectification transforms the images such as that they appear to be recorded with cameras which are in parallel alignment. In the ideal case after rectification all pixels in the left and right image that correspond to the same point in the 3D space rest on the same line (y-coordinate) of the image.

In a last step for all pixels we estimate the offset between the pixel in the left image and the corresponding pixel in the right image being the projections of the same 3D point. The estimation heavily depends on the algorithm used and the quality of the rectification process. The disparity estimation algorithm we currently use is based on the dissimilarity measure as described by Birchfield and Tomasi [6]. Finally we filter the dis-

parity image with a Median filter to reduce outliers. We are aware that there are much advanced algorithms for disparity estimation available [7]. However we recognized that most of them focus on baseline interpolation. While they perform very well for this purpose these algorithms are not always the best choice for a synthesis based on trifocal transfer. Wrong disparities, especially in homogeneous image regions, do not always produce visible visual artifacts in baseline interpolation but can cause strong visual artifacts when using trifocal transfer. Nonetheless it is possible to choose different algorithms for disparity estimation. The disparities are usually stored as 8 bit grayscale values, the so called disparity map.

As input representation for interactive rendering we use the original left and right images and the rectified disparity map. Figure 1 illustrates all steps.

A novel view is synthesized using the texture information of one reference view, one associated disparity map, the rotation and translation between the two reference cameras and a specified position and orientation of the virtual camera in relation to the left camera. Using the method of the trifocal transfer a novel view can be obtained by solving an over determined system of equations [5] which is called forward mapping. A detailed description of how we obtain the trifocal tensor can be found in [4].

The forward mapping causes a number of missing pixel assignments. One reason for this effect is occlusions in one of the reference images and thus missing disparities which is a well known problem. The second reason is missing image information, e.g. when placing the virtual camera very close to the captured object. We solve this problem by a simple neighborhood filling algorithm which achieves good results with low computational complexity.

3. ADDING STEREO PAIRS

The algorithm in the previous section yields a respectable synthesis result. Yet the synthesis with just one reference view has still some flaws. The two major problems to be solved are occlusions and the latitude of the virtual view.

There are many solutions for implementing additional reference views to solve these problems. E.g. using the quadrifocal tensor [8], [9] or sampling density maps [10]. In this section we present a new method, combining some of those results for a high-quality real-time synthesis.

Basically only two reference views are needed to generate a virtual view. We ascertained a rotation of 10 degrees on the x-axis for the best offset between the reference views. This relatively small distance results in very few forward-mapping errors and barley partial occlusions, but still gives a comparatively good result off the baseline. We call this setup a *stereo pair*.

By using several stereo pairs in one scene we can dramatically increase the virtual view latitude. But combining the image information from all stereo pairs would require a number of view synthesis performed in parallel. With today's computers this is too time consuming to achieve real-time rates. Therefore we only use one pair as source for a one virtual view at a time. The decision about which stereo pair S_i we use is primarily based on the euclidean distance ν between the camera pair position $S_{pair} = [x_{pair}, y_{pair}, z_{pair}]^T$ and the virtual view position $S_{virt} = [x_{virt}, y_{virt}, z_{virt}]^T$. The closest *stereo pair* (S_a for virtal view **A** and **B** in the example case shown in fig. 2) mostly delivers the best result.

$$\nu_{pair} = \sqrt{\begin{array}{c}(x_{virt} - x_{pair})^2 + (y_{virt} - y_{pair})^2 + \\ (z_{virt} - z_{pair})^2\end{array}} \quad (1)$$

In some situation this might not be true. Therefore we introduce an "online" quality check algorithm based on the sampling density. We calculate the displacement ρ between the morphed pixel $p'_i = [x'_i, y'_i]^T$ and its origin $p_i = [x_i, y_i]^T$.

$$\rho(S_i) = \frac{\sum_{i=0}^{N} \sqrt{(x'_i - x_i)^2 + (y'_i - y_i)^2}}{N} \quad (2)$$

N is the number of pixels per frame. The average displacement ρ is a degree for the quality of the view synthesis result [10]. Within a given radius σ (blue circle in fig.2) around the stereo pair S_a this pair is considered "perfect" and no decision is required. Outside this threshold ReVOGS performs a trifocal transfer for S_a and the next closest pair S_b and decides based on the sampling displacement ρ.

$$S_{act} = \begin{cases} S_a \; \forall \; S_{virt} \mid \nu(S_a) \leq \sigma \\ S_a \; \forall \; S_{virt} \mid \nu(S_a) > \sigma \; and \; \rho(S_a) \leq \rho(S_b) \\ S_b \; \forall \; S_{virt} \mid \nu(S_a) > \sigma \; and \; \rho(S_a) > \rho(S_b) \end{cases} \quad (3)$$

Once this decision is made only one transfer step is necessary unless the virtual view jumps or moves away from the actual stereo pair. Then a new decision is made. By reducing the necessary transfer steps to a minimum the system obtains its real-time qualities. Fig. 2 shows a simple camera setup with two *stereo pairs*.

The stereo pair concept leads to a highly scalable, yet still real-time FVV-system with very little restrictions. The setup between the individual stereo pairs is not constricted and can be modified to match any application.

4. INTERACTIVE RENDERING

The Realistic Video Object Generation System briefly ReVOGS is a modular constructed imaging software

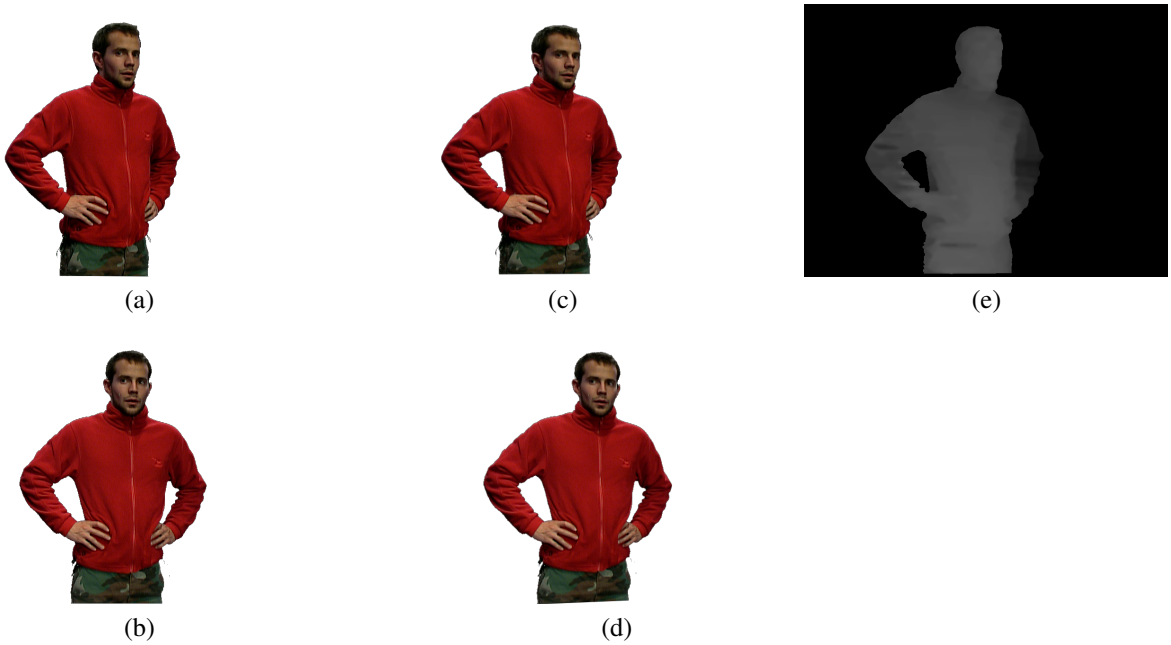

Fig. 1. Images a and b: Left and right view captured by a convergent setup. Images c and d: Rectfied left and right view. The images are also horizontally shifted in order to obtain always positive disparity values. Image e: The disparity map from left to right image (gamma level increased for illustration purposes).

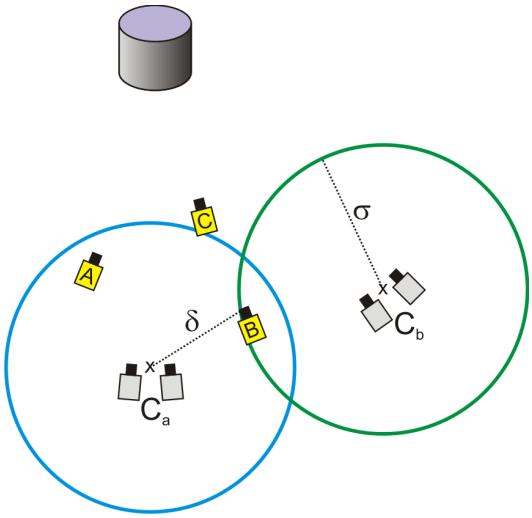

Fig. 2. Camera setup example with two stereo pairs. Only virtual view **C** yields a qualitiy decision. View **A** is within $\sigma(S_a)$ and view **B** within $\sigma(S_a)$ and $\sigma(S_b)$ but closer to S_a.

for view synthesis and quality assessment of free viewpoint video (i.e. 3D video objects - 3DVO). The system uses specific image processing modules which are combined in a workflow specified by an XML file. The workflows of which each constitutes a different algorithm chain are intuitively adaptable by a GUI or on the text level. In our first implementation as described in section 2 we implemented a virtual camera path with different viewing angles of the 3DVO by manually editing a XML file. This limits the path to be fixed during viewing. Since rendering on the CPU is fast enough already we have developed a new control module providing an interactive rendering of 3DVOs. Additionally, we combined the 3DVO with a synthetic OpenGL scene which can be explored by an interactively moving camera. We will briefly explain the two viewing modes in the following sections.

4.1. Object observation

We call the basic display mode “observe mode”. The mode directly displays the result of the trifocal transfer which is a physically correct view of the 3DVO. We implemented an interactive control module to provide an intuitive environment. This can be used, for example, for quality assessment and error analysis. The virtual camera can be moved in each direction and rotated around the 3DVO via mouse and keyboard input. According to the number of stereo pairs in use and to prevent loss of orientation we set adjustable borders which restrict the rotation angles around the 3DVO. Within

these borders the user can view the 3DVO from any point of view.

4.2. Embedding into synthetic scenes

Many of today's multimedia applications such as games or virtual meeting rooms rely on synthetic computer graphic scenes. Although the level of realism of interactive computer graphics has been increased dramatically in the last few years it is still clearly visually distinguishable from captured natural scenes. Our aim is to increase the level of realism of synthetic scenes by adding 3DVO to it. At a first glance using an image-based rendering approach as described in the previous sections seems to have disadvantages compared to a model-based approach. In the model based approach the scene representation (i.e. the mesh) is already the same as for the synthetic scene thus combining the two is an easy task. Anyway we still find that the models often look "computer graphics-like". Also the usage of a mesh-based representation is not as flexible as our method in terms of the number of cameras used to get the representation.

We found a simple convincing way to integrate views synthesized by the trifocal transfer into the synthetic scene. We call this "scene mode" within our system. When the pixel positions of the virtual view have been sampled we simply project the result as texture onto a billboard. The billboard is nothing else than a simple plane in the OpenGL 3D space that always faces towards the viewer.

The synthesis algorithm also provides an alpha mask as output which we also apply to the billboard to make only the part of the plane visible that is covered by the object. The texture of the billboard is updated with the output of the synthesis algorithm each time the viewing position is changed thus providing the impression of moving around the object.

One problem raised by this method was not yet discussed. The perspective projection rendering of the synthetic scene causes the billboard and its texture to be scaled in size when moving the virtual camera towards or away from it respectively. Since the virtual camera position is also an input to the synthesis algorithm the texture would get scaled two times - once by the scene projection and once by the synthesis algorithm. Fortunately we can easily avoid this effect by passing only the rotational components from the virtual camera to the synthesis algorithm while discarding any translation. This can be done without changing the virtual geometry of the object since both the synthesis algorithm and the synthetic scene rendering use the same projection model which is a simple pin-hole camera. Just the hole filling technique described in section 2 (when applied to holes caused by up scaling) is substituted by the interpolation method provided by the graphics hardware.

Although it is not completely physically correct using the billboard to display the 3DVO yields a very good 3D impression in most use cases. Certainly there are cases where the method does not work. Synthetics object cannot be covered partially (e.g. by an arm) from the 3DVO when it is very close to it for example. But for most of our use cases such close interactions are not required and the 3D impression is kept.

Finally we implemented a similar interactive control like in the "observe mode". The user has the possibility to fly through the scene and to examine all objects, comparable to modern computer games.

5. RESULTS

The procedure described in the previous sections is quite simple but provides satisfactory results. We can easily change the viewing latitude by adding or removing the data of additional stereo pairs respectively. The synthetic scene is easily exchangeable, the implementation of the billboard is anything but difficult and this simplification is barely noticeable. The viewing modes demonstrate a sample application for different use cases like virtual 3D worlds or subjective quality assessment for free viewpoint video. Fig. 3 shows some screenshots. Although using only some simple synthetic primitives for demonstration purposes it gives a good impression of how 3DVO and synthetic scenes can work together. ReVOGS renders the virtual view obtained from SD television input data (resolution 768x576) at 10 frames per second using a non-optimized brute-force implementation of the view synthesis algorithm on a standard PC (Pentium 4 3.2GHz, 2GiB RAM).

6. REFERENCES

[1] Joel Carranza, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel, "Free-viewpoint video of human actors," *ACM Trans. Graph.*, 2003.

[2] A. Smolic, K. Müller, P. Merkle, T. Rein, M. Kautzner, P. Eisert, and T. Wiegand, "Free viewpoint video extraction, representation, coding, and rendering," in *ICIP*, 2004, pp. 3287–3290.

[3] E. de Aguiar, C. Stoll, N. Ahmed, and H.-P. Seidel, "Performance capture from sparse multi-view video," in *Proc. of ACM SIGGRAPH 2008*, 2008.

[4] C. Weigel and L. Kreibich, "Advanced 3d video object synthesis based on trilinear tensors," in *Proc. ISCE*, 2006.

[5] S. Avidan and A. Shashua, "Novel View Synthesis by Cascading Trilinear Tensors," *IEEE Trans-*

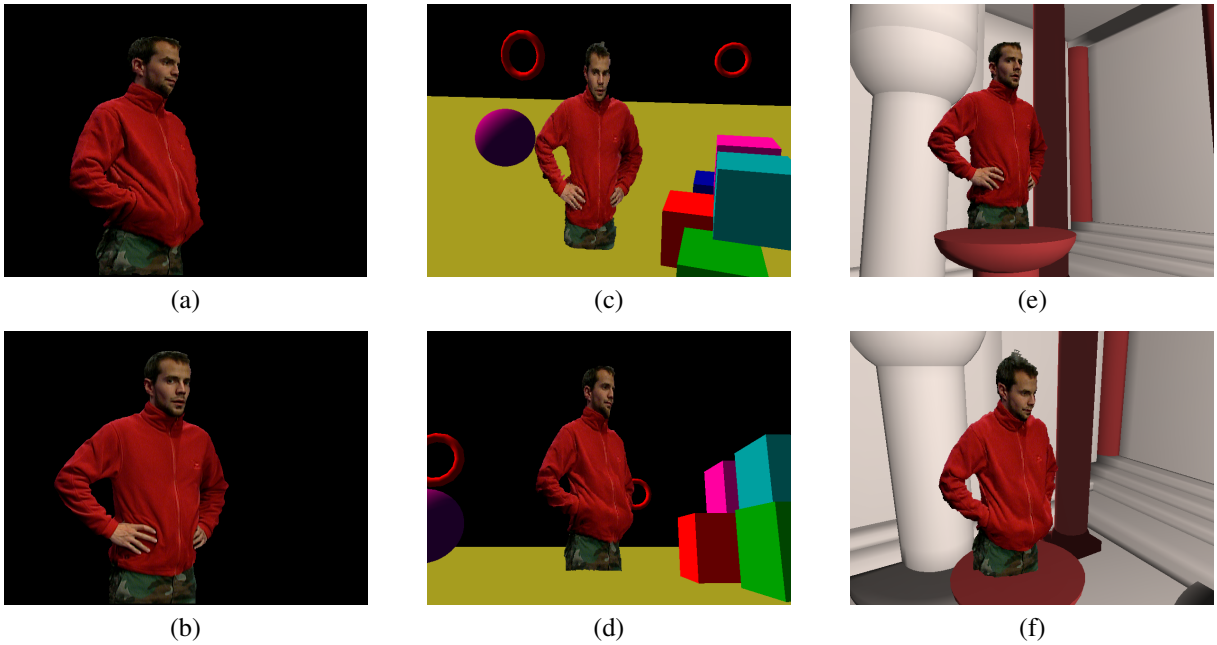

Fig. 3. The images show from left to right: The 3DVO in the observer mode (a,b). The 3DVO embedded in two synthetic scenes (c-f).

actions on Visualization and Computer Graphics, vol. 4, no. 4, pp. 293–306, 1998.

[6] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Trans.on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 401–406, 1998.

[7] R. Szeliski D. Scharstein, "Middlebury Stereo Evaluation Web Page," 2009.

[8] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Springer, Cambridge, 2 edition, 2004.

[9] R. Hartley and F. Schaffalitzky, "Reconstruction from projections using grassman tensors," in *Proceedings ECCV*, Prague, 2004.

[10] E. Cooke and N. OConnel, "Multiple image view synthesis for free viewpoint," in *Proceedings ICIP*, Genoa, Italy, 2005.