DATA FORMAT AND CODING FOR FREE VIEWPOINT VIDEO

P. Kauff, A. Smolic, P. Eisert, C. Fehn. K. Müller, R. Schäfer

Fraunhofer Institute for Telecommunications,
Heinrich-Hertz-Institut (FhG/HHI), Berlin, Germany

ABSTRACT

In this paper we will discuss data representation formats and coding aspects of a new 3D functionality called *Free Viewpoint Video (FVV).* Similar to *Computer Graphics (CG)* or *VRML* applications, FVV allows for navigating with a virtual camera in 3D scenes. However, in contrast to graphical scenes in CG and VRML, FVV is based on natural scenes consisting of 3D video objects that provide a higher level of realism and allow a 3D reproduction of real events. Against this background the paper will review different approaches on FVV and different 3D representation formats. It will discuss the acquisition of required representation parameters and related rendering aspects. Furthermore it will present results of coding experiments for two particular 3D representation formats.

INTRODUCTION

Interactivity is an important key feature of new and emerging audio-visual media where the user has the opportunity to be active in some way instead of just being a passive consumer. One kind of interactivity is the ability to look around within an audio-visual scene by freely choosing viewpoint and viewing direction. The first media that provided such functionality were based on shaded or textured 3D mesh models, as they are well known from computer graphics, computer games and virtual reality. For representation and exchange of such interactive virtual worlds, ISO/IEC has standardized a special language called Virtual Reality Modelling Language (VRML) that is widely used in the Web. However, although VRML still holds merit, it is getting dated due to the rapid progress in multimedia in general and virtual reality in particular. VRML is invariably graphics-oriented and scene realism is limited therefore. Most of the scenes are either purely computer generated or contain static 2D views of real word objects represented by still pictures or moving textures. Hence, as a complement to VRML, new standardization efforts have been launched to advance realism and functionality of 3D representations in interactive audio-visual media. The X3D standard of the Web3D consortium or the 3DAV activities of MPEG-4 are best examples for this evolution (1), (2).

Against this background, we can observe a new development for 3D applications and functionalities called *Free Viewpoint Video (FVV)*. It allows free navigation within dynamic real-world scenes by choosing arbitrary virtual viewpoints and viewing directions. FVV can be considered as an extension of classical computer graphics towards the natural representation of motion and appearance of real world objects. The 3D acquisition typically relies on a multi-camera setup as shown in Figure 1. An event is captured by a given number of real cameras (illustrated in dark grey). The images of such a multi-view capturing unit are then converted into a suitable 3D data representation format, which supports the functionality to interpolate intermediate views at arbitrary camera positions (illustrated in light grey). Hence, as known from VRML applications, a user can freely navigate with a virtual camera in a 3D scene, but in contrast to VRML applications the FVV scene consists of real audio-visual video objects providing a high level of realism.

From literature we know a lot of suitable representation formats for 3D scene description. Most approaches for modelling and rendering real world scenes are currently investigated in the 3DAV group of MPEG-4 (3). This includes video-based rendering as well as 3D reconstruction. The next section will discuss the two most promising approaches for FVV: imaged based rendering using multi-view disparity estimation and reconstruction of 3D models by shape from silhouette. Then, the following sections will concentrate on specific aspects of coding and rendering for each of these formats.

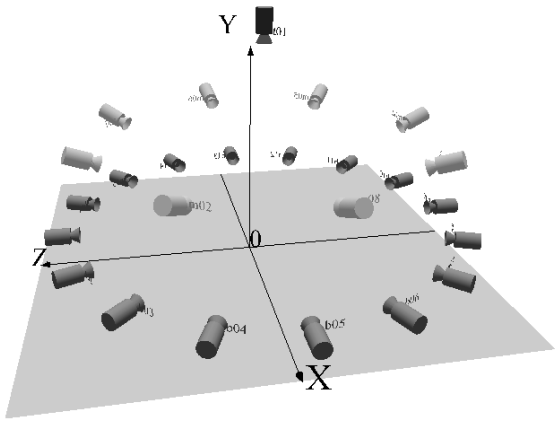


Figure 1 – FVV acquisition and functionality

3D REPRESENTATION FORMATS FOR FREE VIEWPOINT VIDEO

The methods for 3D scene or object representation are often classified as a continuum in between two extremes as illustrated in Figure 2 (4). One extreme is represented by classical 3D computer graphics. This approach can also be called geometry- or physics-based modelling. In most cases scene geometry is described on the basis of 3D wire-frames and meshes. Real world objects are reproduced using geometric 3D surfaces with an associated texture mapped onto them. More sophisticated attributes can be assigned as well. For instance, appearance properties (opacity, reflectance, specular lights, etc.) can enhance the realism of the models significantly. A drawback to this approach, however, is the high costs for content creation. Aiming at photo-realism, 3D scene and object modelling is complex and time consuming, and it becomes even more complex in the case of modelling a dynamic environment with moving objects and complex motion. Furthermore, an automatic or semi-automatic 3D object and scene reconstruction implies an estimation of camera geometry, depth structures and 3D shapes. Inherently, all these processes tend to produce occasional errors.

The other extreme is given by scene representations that do not use any 3D geometry at all. It is usually called image- or appearance-based modelling. In this case virtual intermediate views are generated from available real views by interpolation. The main advantages are a high quality of virtual view synthesis and an avoidance of a complex 3D scene reconstruction. However, these benefits have to be paid by dense sampling of the real world with plenty of reference views. In general, the synthesis quality increases with the number of available views. Hence, a large amount of cameras has to be set up to achieve high-performance rendering, and plenty of image data needs to be processed therefore. Contrariwise, if the number of used cameras is too low, occlusions of important scene elements will accumulate in all available views and interpolation algorithms are going to need additional depth information. As a consequence, interpolation and occlusion artefacts might appear in the synthesized image, possibly affecting the quality.

In between these two extremes of physics- and appearance-based representations there exists a continuum of methods that make more or less use of both approaches and combine the advantages of both in a particular manner. For instance, the well-known Lumigraph uses a similar representation as a light field but adds a rough 3D model (5). This provides some coarse information on the depth structure of the scene and therefore allows for reducing the number of required cameras. Other representations do not use an explicit 3D models but depth or disparity maps. Such maps assign a depth value to each pixel of an image.

Together with the original 2D image the depth map form a 3D-like representation. This can be extended to Layered Depth Images where multiple colour and depth values are stored in consecutively ordered depth layers (6). Closer to the physics-based end of the spectrum we can find methods that use view-dependent geometry and/or view dependent texture (7), (8). Surface light-fields combine the idea of light-fields with an explicit 3D model (9). In the following we will briefly review the two most promising techniques for FVV applications.

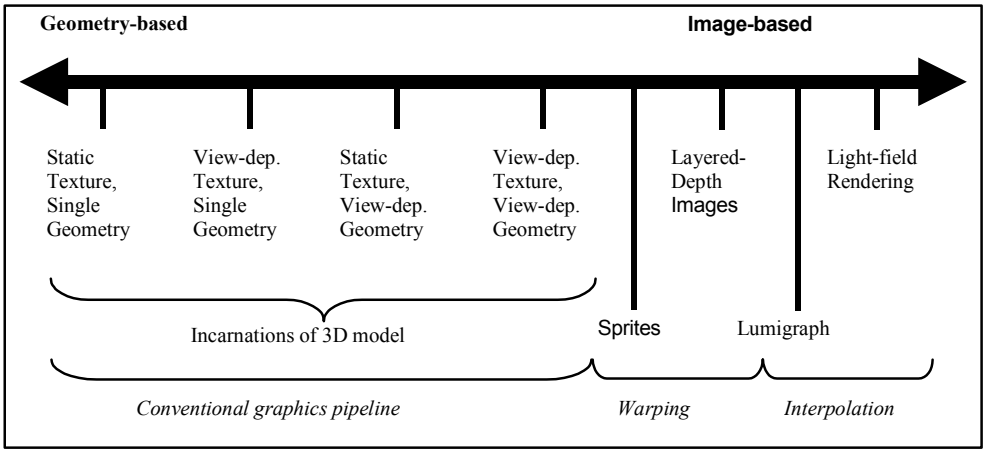
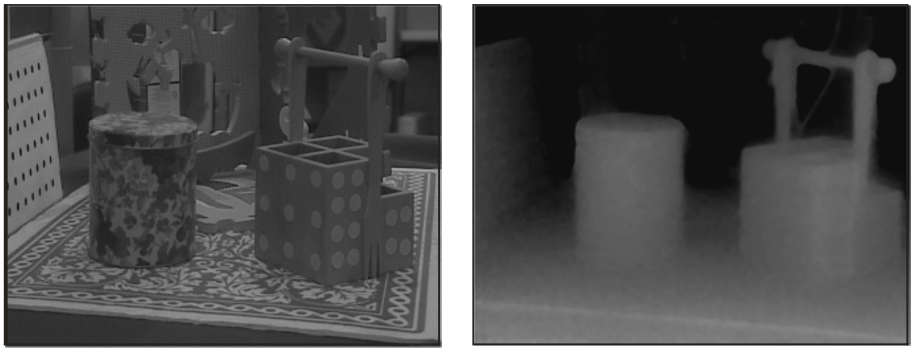


Figure 2 - Categorization of scene representations

Depth Image Based Rendering (DIBR)

In this case video scenes are represented by a number of views with associated depth maps. As shown in Figure 3, these maps define a depth value for each pixel of the 2D images. Usually depth maps are captured by Z-sensors or they are calculated from disparity estimation. Supposing that the capturing cameras are fully calibrated und their 3D geometry is known therefore, results of disparity estimation can directly be converted to corresponding depth values. Efficient disparity estimation algorithms that are suitable for multi-view stereo, partly even in real-time, can be found under (10), (11) and (12).

Together with appropriate scaling and information from camera calibration it is possible to render virtual intermediate views between the real cameras (see Figures 4 and 5). Techniques based on tri-linear warping or tri-focal tensors are used in this context (13). Self-occlusions appearing during the movement of the virtual camera are handled by an occlusion-compatible ordering (14). A good overview on algorithmic details and related applications can be found under (15).



(a) (b)

Figure 3 – Depth Image Based Rendering; (a) RGB image (b) Depth Map

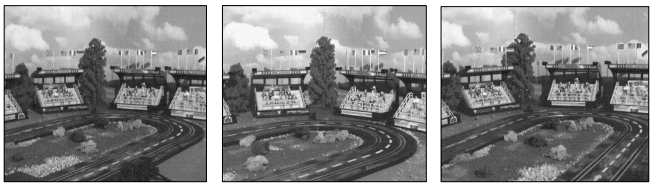


Figure 4 – Three views of a scene captured by real cameras

Figure 5 – Synthesised views from a virtual camera moving through the scene

Volumetric Models with View-Dependent Texture

Following the categorization from Figure 2, DIBR is clearly appearance- or image-based. In contrast to that, volumetric models are more physics- or geometry-based. As shown in Figure 6, the depth structure is represented by geometric 3D-models in this case. Different data representation and rendering formats have been proposed for this purpose. A straightforward solution is to use dynamic 3D meshes for geometry and to combine this with view-dependent texture mapping (16). Alternative representation formats are image-based visual hulls (17), point-based (18), or volumetric representations (19). However, as dynamic 3D meshes are widely used in computer graphics and are very efficiently supported by hardware and standard software APIs, it seems reasonable to concentrate on this format in practice.

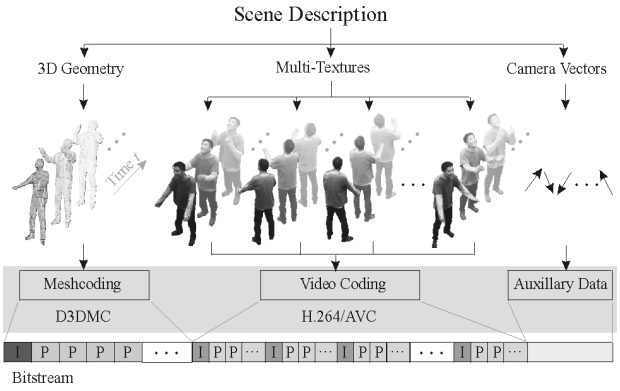


Figure 6 – 3D meshes with multiple textures

For photo-realistic rendering, the existing reference videos are then mapped onto the dynamic 3D meshes. Depending on reflectance properties and lighting conditions, natural materials may appear very differently from diverse viewing directions. Therefore static texturing, as we know it from CG applications, often leads to artificially looking rendering results. In contrast, FVV applications will use multi-textures as shown in Figure 6. In this case all existing reference views have to be available for rendering. The corresponding textures can then be mapped onto the 3D mesh model and weighted in relation to given geometric criteria. The virtual view is then generated by a weighted interpolation between multiple re-projections using the textures from all available reference views (20). This weighting ensures a seamless fading between the object appearances from different viewpoints, resulting in almost natural rendering results when navigating through the scene (see Figure 7).

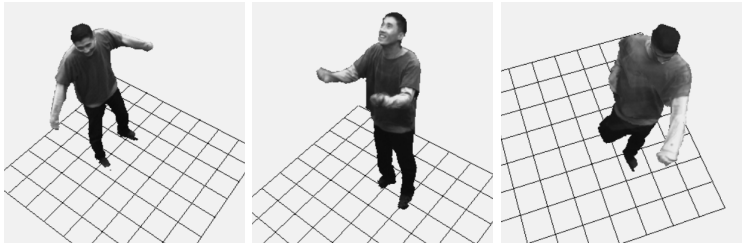


Figure 7 – Rendering with View-Dependent Textures

CODING OF DEPTH MAPS IN DIBR APPLICATIONS

In DIBR the 3D geometry is represented through depth maps that are available for each reference view. From the coding point of view it is therefore of high interest how this additional side information affects the overall coding efficiency. Related in-depth studies have carried out in the European research project ATTEST, dealing with various aspects of 3D-TV applications (21). To test depth-image compression performance of different MPEG technologies, four video codecs were evaluated in a comparative coding experiment. The test group consisted of the MPEG-2 reference model codec (TM-5), the MPEG-4 Visual reference model codec (MS-Ref), a high-quality rate-distortion optimized MPEG-4 Visual codec developed at FhG/HHI (R/D opt.) and the R/D optimized H.264/AVC reference model codec from MPEG-4 (v6.1a). The compression results for typical broadcast encoder settings are shown in Figure 8. The rate-distortion curve plots the image quality after decoding, measured in PSNR (Peak-Signal-to-Noise Ratio), against the required bit-rate.

The results, first of all, indicate that AVC as well as MPEG-4 Visual are very well suited for the coding of depth maps (with AVC being more efficient). The very particular statistics of depth maps (mainly the strong spatial and temporal correlation) enable very high compression ratios. For example, H.264/AVC compression of the given test sequence at 105 kbit/s results in a PSNR of 46.29 dB. At this high level, coding artefacts in the depth maps have no impact on the quality of the synthesised images. The threshold where artefacts caused by depth map coding become visible in the synthesised images is much lower. Thus, we can conclude that the side-information for DIBR application can be coded with 100 kbit/s are even less. This refers to standard TV resolution. Assuming that the bit-rate for the regular H.264/AVC video signal is at least 1 Mbit/s in this case, the resulting overhead for the additional depth information is below 10 %.

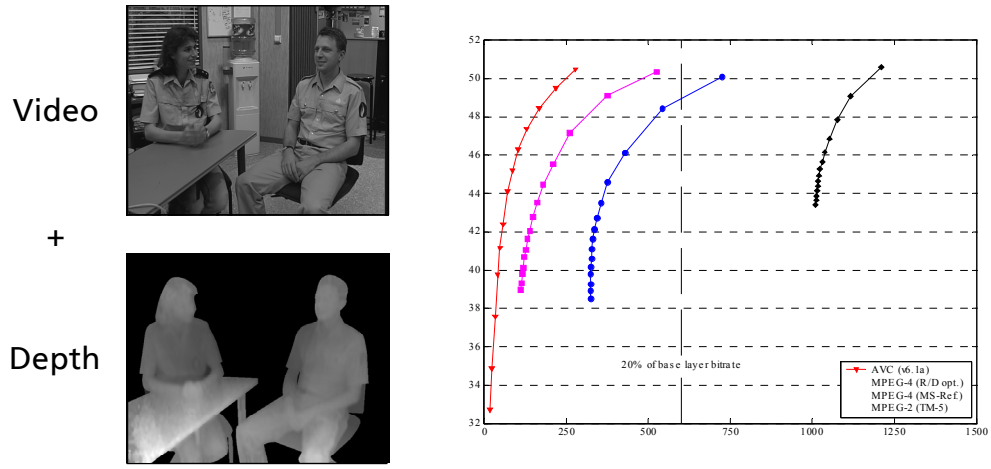


Figure 8 – Results on Depth Map Coding (Test Sequence: Interview)

CODING OF DYNAMIC 3D MESHES

The MPEG-4 standard already provides a compression tool for static 3D meshes, the so-called 3DMC (3D Mesh Coding) tool. Based on this tool, FhG/HHI has developed a novel compression scheme for time-consistent dynamic 3D meshes with common connectivity. It extends 3DMC towards a new predictive mode using a classical DPCM-loop with arithmetic coding of the residuals (20). Similar to MPEG video codecs that are organised in group-of-pictures (GOP), this differential 3DMC (D3DMC) system is based on group-of-meshes (GOM). Each GOM starts with the 3DMC tool for intra-coding of the initial 3D mesh model, followed by a DPCM structure resulting in 3D motion vectors for each vertex. Obviously, this presupposes the same number of vertices and the same connectivity in all frames of a

particular GOM. If this precondition is not fulfilled, the current GOP is closed and a new one is opend. In the predictive mode the 3D motion vectors are clustered in an oct-tree structure and the residual are compressed by using context-adaptive binary arithmetic coding.

As an example Figure 9 shows results for the synthetic “Chicken Crossing” sequence, which provides a sequence of 400 time-consistent meshes (GOM400) with 3030 vertices. It is compared with a reference method, called AFX-IC, for two different key-frame numbers (GOM length). AFX-IC is based on a tool from the Animation Framework Extension (AFX) of MPEG-4 - the Interpolation Compression (IC) tool that already allows predictive coding in combination with 3DMC. The results show that D3DMC clearly outperforms AFX-IC, especially for low bit-rates. Moreover, the results indicate that a good reconstruction quality can be ensured for 100 kbit/s or even below.

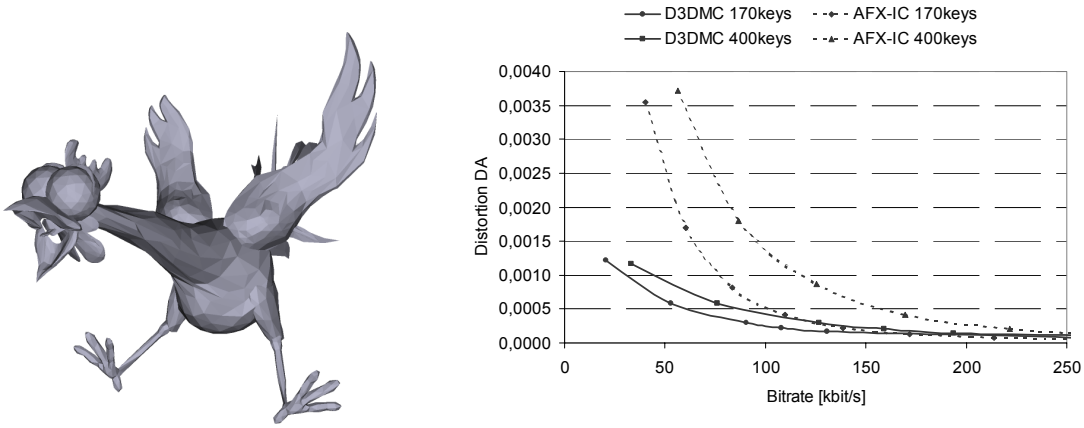


Figure 9 – Results on Dynamic Mesh Coding (Test Sequence: Chicken Cross)

CODING OF MULTI-VIEW TEXTURES

Both, depth image based rendering and 3D meshes with view-dependent texture mapping use multi-view sequences for representing the texture information in FVV applications. ISO MPEG and ITU SG16 offer a lot of video codecs that are suitable for data compression in this case. As long as the multi-view texture sequences consist of regular video frames, as it is the case in the DIBR example from Figure 4, it is obviously most efficient to use H.264/AVC, because it represents the state-of-the-art video coding standard with a clear performance gain over MPEG-4, H.263 or MPEG-2.

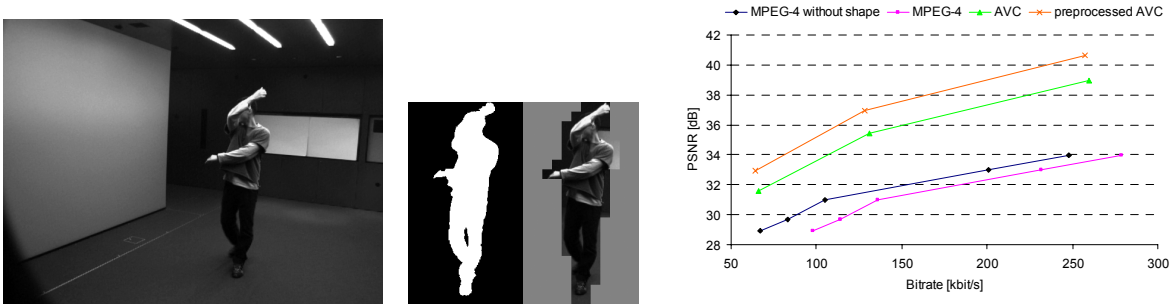


Figure 10 – Results on Texture Coding (bounding box is used for preprocessed AVC)

However, the situation is not that clear in the case of shaped video textures that are used for 3D meshes in Figure 6 and which might be used for some DIBR applications as well. To investigate this aspect, two most promising codecs have been compared in an experimental study: MPEG-4 Core Profile and H.264/AVC Main Profile. The latter does not support variable shape coding as the first one does. However, for texture mapping and rendering of

virtual views at the decoder, it is not necessary to transmit the complete rectangular video. Only the area covered by the object of interest needs to be encoded. Therefore, in case of H.264/AVC, the video is preprocessed prior to encoding as illustrated in Figure 10. For that, the bounding box that completely contains the dynamic object has been extracted from segmentation results. The width and height of the bounding box fits to the macro-block structure of H.264/AVC and all empty macro-blocks within the bounding box have been set to a constant value of 128. Note that the segmentation mask has not to be transmitted because it can be derived from the 3D mesh model at the decoder. Finally, the video is encoded using standard H.264/AVC syntax. The results from Figure 10 show that in all experiments H.264/AVC Main Profile clearly outperforms MPEG-4 Core Profile. The PSNR gain at same bit-rate is in a range of several dB, even if the bits used for shape by MPEG-4 Core Profile are not taken into account.

CONCLUSION

We have presented two promising approaches for data representation and 3D video coding in the application framework of *Free Viewpoint Video*: Depth Based Image Rendering and 3D Meshes Using View-Dependent Texture Mapping. We have shown that both approaches can efficiently be built on basic MPEG-4 tools, such as H.264/AVC Core Profile or 3D Mesh Coding. The additional bit-rate that is needed for coding the 3D structure can be kept in an acceptable range of 100 kbit/s.

Both approaches use the same multi-view paradigm for 3D acquisition and capturing of the required texture information, but have different advantages and drawbacks. Depth image based rendering is more suitable for large-scale scene covering the whole video frame. The estimation of depth maps may benefit from segmentation results, but this segmentation is not necessarily object-oriented. In contrast, the generation of suitable 3D meshes supposes a reliable object-oriented segmentation results, and it provides a 3D representation for one particular scene object, but not for entire large-scale scenes. Instead, it allows a higher degree of navigation with the virtual camera around this object and offers a better synthesis quality in this case.

Due to these conclusions and the fact that both approaches can be applied to the same multi-view set-up, it seems to be reasonable for future work to investigate a hybrid solution that benefits from both methods. Depth image based rendering can be used for representing the global environment of a scene (e.g. a soccer stadium) while 3D meshes and view-dependent texture mapping can be used for a high-quality representation of specific scene objects that are of particular interest (e.g. soccer players). Such a hybrid approach would fit into a joint MPEG-4 syntax and would only require a limited overhead bit-rate for the 3D data. In this context it should also be investigated how interrelations and correlations between the textures of different views can be used for further bit-rate savings.

REFERENCES

1. WEB 3D Consortium, 2005. Open Standards for Real-Time 3D Communication. www.web3d.com. July 2005.
2. Smolic, A. and McCutchen, D., 2004. 3DAV Exploration of Video-Based Rendering Technology in MPEG. IEEE Trans. on CSVT. March 2004, vol. 14, no. 9, pp. 348-356.
3. Smolic, A. and Kauff, P., 2004. Interactive 3D Video Representation and Coding Technologies. Proc. of the IEEE. Special Issue on Advances in Video Coding and Delivery. January 2005, vol. 93, no.1.
4. Kang, S.B. and Szeliski, R. and Anandan, P., 2000. The Geometry-Image Representation Tradeoff for Rendering. Proc. of IEEE International Conference on

Image Processing, ICIP. September 2000, Vancouver, Canada.
5. Gortler, S.J. and Grzesczuk, R. and Szeliski, R. and Cohen, M.F., 1996. The Lumigraph. Proc. of ACM SIGGRAPH 96. August 1996, pp.43-54.
6. Shade, J. and Gortler, S. and He, L.W. and Szeliski, R., 1998. Layered Depth Images. Proc. Of SIGGRAPH 98. July 1998, Orlando, FL, USA.
7. Debevec, P. and Taylor, C. and Malik, J., 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image based approach. Proceedings of SIGGRAPH 96. 1996, pp. 11-20.
8. Müller, K. and Smolic, A. and Droese, M. and Voigt, P. and Wiegand, T., 2003. Multi-Texture Modelling of 3D Traffic Scenes, Proc. Of IEEE International Conference on Multimedia & Expo, ICME 03. July 6.-9. 2003, Baltimore, MD, USA.
9. Chen, W.-C. and Bouguet, J.-Y. and Chu, M.H., and Grzeszczuk, R., 2002. Light Field Mapping: Efficient Representation and Hardware Rendering of Surface Light Fields. Proc. Of ACM Transactions on Graphics. 2002, 21 (3), pp. 447-456.
10. Atzpadin, N. and Askar, S. and Kauff, P. and Schreer, O., 2003. New Concept For Joint Disparity Estimation and Segmentation for Real-Time Video Processing. Proc. of 23rd Int. Picture Coding Symposium (PCS). April 2003, pp.155-160, Saint-Malo, France.
11. Mulligan, J. and Isler, V. and Daniilidis, K., 2002. Trinocular stereo: a real-time algorithm and its evaluation. International Journal of Computer Vision. 2002, 47(1/2/3):51—61.
12. Zitnick, C.L. and Kang, S.B. and Uyttendaele, M. and Winder, S. and Szeliski, R., 2004. High-quality video view interpolation using layered representation. Proceedings of SIGGRAPH 04. 2004, pp. ?.
13. Avidan, S. and Shashua, A., 1998. Novel View Synthesis by Cascading Trilinear Tensors. IEEE Transactions on Visualisation and Computer Graphics. 4(4); October-December,1998. pp. 293 to 306.
14. Oliveira, M., Bishop, G. and McAllister, D., 2000. Relief Texture Mapping. In Proceedings of SIGGRAPH. July, 2000. pp. 259 to 268
15. Fehn, C. and Kauff, P, 2002. Interactive Virtual View Video (IVVV) – The Bridge Between 3D-TV and Immersive TV. Proceedings of SPIE Three-Dimensional TV, Video and Display. July 2004, pp 14-25, Boston, MA, USA.
16. Matsuyama, T. and Wu, X. and Takai, T. and Wada, T., 2004. Real-Time Dynamic 3-D Object Shape Reconstruction and High-Fidelity Texture Mapping for 3-D Video. IEEE Trans. on CSVT. March 2005, Vol. 14, No. 3, pp. 357-369.
17. Matusik, W. and Buehler, C. and Raskar, R. and Gortler, S.J. and McMillan, L., 2000. Image-Based Visual Hulls. Proc. of SIGGRAPH 2000. 2000, pp. 369–374.
18. Goldlücke, B. and Magnor, M., 2003. Real-time Microfacet Billboarding for Free-viewpoint Video Rendering. Proc. of ICIP2003 Sept. 2003, Barcelona, Spain.
19. Würmlin, S. and Lamboray, E. and Gross, M., 2004. 3D video fragments: dynamic point samples for real-time free-viewpoint video. Computers and Graphics. 2004, vol. 28 (1), pp. 3-14, Elsevier Ltd.
20. Mueller, K. and Smolic, A. and Merkle, P. and Kautzner, M. and Wiegand, T., 2004. Coding of 3D Meshes and Video textures for 3D Video Objects. Proc. of Picture Coding Symposium, PCS 04. Dec. 2004, San Francisco, CA, USA.
21. Fehn, C. and Hopf, K. and Quante, B., 2004. Key Technologies for an Advanced 3D-TV System. Proceedings of SPIE Three-Dimensional TV, Video and Display. October 2004, vol. III, pp 66-80, Philadephia, PA, USA.