3D VIDEO FORMATS AND CODING METHODS

Karsten Müller[1], *Philipp Merkle*[1], *Gerhard Tech*[1], *and Thomas Wiegand*[1,2]

[1]Image Processing Department
Fraunhofer Institute for Telecommunications -
Heinrich Hertz Institute (HHI)
Einsteinufer 37, 10587 Berlin, Germany
{kmueller/smolic/dix/merkle/wiegand}@hhi.de

[2]Image Communication Chair
Department of Telecommunication Systems
School of Electrical Engineering and Computer Sciences
Technical University of Berlin
Einsteinufer 17, 10587 Berlin, Germany

ABSTRACT

The introduction of first 3D systems for digital cinema and home entertainment is based on stereo technology. For efficiently supporting new display types, depth-enhanced formats and coding technology is required, as introduced in this overview paper. First, we discuss the necessity for a generic 3D video format, as the current state-of-the-art in multi-view video coding cannot support different types of multi-view displays at the same time. Therefore, a generic depth-enhanced 3D format is developed, where any number of views can be generated from one bit stream. This, however, requires a complex framework for 3D video, where not only the 3D format and new coding methods are investigated, but also view synthesis and the provision of high-quality depth maps, e.g. via depth estimation. We present this framework and discuss the interdependencies between the different modules.

Index Terms— Depth estimation, View synthesis, MVD, 3D video, MPEG, video coding, MVC.

1. INTRODUCTION

The existing 3D video systems are based on stereoscopic technology. This includes 3D cinemas showing stereo-based 3D films and first mobile devices with stereo displays. The stereo video technology can be seen as the first generation of 3D video applications, where stereo video is recorded, coded and transmitted, and displayed. For this first generation, multi-view video coding (MVC) [2][3] was adopted for the compression of conventional stereo on the 3D Blu Ray Disc format.

Currently, a second-generation 3D video is being developed [4]. The second generation attempts to overcome one of the disadvantages of conventional stereo video, which is its restriction to two views at fixed spatial positions. Besides stereoscopic displays, also a variety of multi-view displays are being offered with different number of views. Therefore, a generic and flexible 3D video solution is required, where the display format is decoupled from the transmission and production format. With this format, only one 3D video bit stream is required for any multi-view display.

A complete 3D video coding framework that targets a generic 3D video format and associated efficient compression is shown in Fig. 1. For broad multi-view display support, depth data is estimated at the sender side for a limited number of e.g. 2-3 input views, giving a generic multi-view video plus depth format (MVD) for transmission. At the receiver side, the video and depth data are decoded and the view synthesis is used to generate as many additional views as required by the display. Since 3D video discussed here uses parallel camera setups, the view synthesis can be carried out using horizontal sample displacements of the original camera views towards the new spatial positions in the intermediate views. These shift values are derived from the depth data.

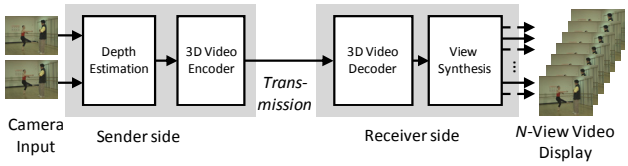

Fig. 1: Overview of 3D Video System.

The paper is organized as follows. Section 2 gives an overview on current MVC coding and its restrictions to 3D video. In section 3, the requirements and coding techniques for the 3D video format are introduced.

2. MULTI-VIEW VIDEO CODING: FEATURES AND RESTRICTIONS

The underlying technology of today's coding method for multiple camera views, is the H.264/AVC video coding standard [2][9]. Usually, the camera signals of a multi-view array share common scene content, such that a coding gain is achievable by exploiting existing statistical dependencies, especially in spatially neighboring views. Therefore, multiple view compression was investigated and the corresponding standardization activity led to the MVC extension [2][3] of H.264/AVC.

An MVC coder basically consists of *N* parallelized single-view coders. Each of them uses temporal prediction structures, were a sequence of successive pictures is coded as intra (I), predictive (P) or bi-predictive (B) pictures. One approach for further improving coding efficiency is the use of hierarchical B pictures [8], where a B picture hierarchy is created by a dyadic cascade of B pictures that are references for other B pictures.

For MVC, the single-view concepts are extended, so that a current picture in the coding process can have temporal as

well as inter-view reference pictures for prediction [6]. An example for an MVC coding structure with five linearly arranged cameras and a GOP size of 8 is shown in Fig. 2. For the base view (Cam 1 in Fig. 1), the prediction structure is identical to single view coding and for the remaining views, inter-view reference pictures are additionally used for prediction (red arrows). Another advantage of MVC is its backward compatibility, as the base view (Cam 1) is decodable by a legacy single view H.264/AVC decoder. Thus, MVC represents the current state-of-the art coding for multiple cameras and was adopted to the 3D Blu Ray specification for coding 2-view stereo in 2009.

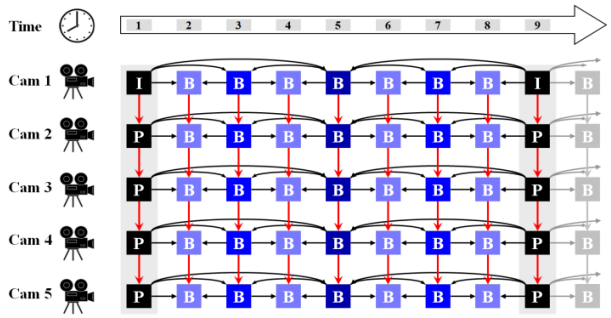

Fig. 2: MPEG-4 MVC coding structure for a linear 5 camera setup and GOP size of 8 pictures.

Since MVC was optimized for multi-view video signals with a given number of views, some restrictions occur, if other formats are investigated. As shown in section 3, advanced formats for 3D video coding require geometry data, e.g. in the form of depth maps. These depth maps have different statistical properties than video signal. Especially sharp edges in depth data need to be preserved.

But, MVC is optimized for the statistics of video signals, where sharp edges are typically smoothed when coding at low bit rates. For multi-view displays with different number of views, additional views have to be synthesized by depth-image-base rendering techniques, where the depth data is used to shift the color data to the correct position in the intermediate view. Therefore, errors in the depth data lead to geometric errors in the form of wrong video sample displacement.

Another restriction regarding the applicability of MVC to a higher number of views is the linear dependency of the coded data rate from the number of cameras. As shown in [6], the data rate of MVC-compressed multi-view video increases with each camera. Therefore, a reasonable data rate for 9 or even 50 views is not achievable with this concept. Here, advanced approaches are required, which decouple the number of views for coding and transmission from the number of required output views. Also, stereo repurposing with correct baseline change becomes possible, in contrast to color-only video formats.

3. DEPTH-ENHANCED 3D VIDEO

A generalized 3D video framework requires new functionality over current state-of-the-art MVC and transmission. The framework as shown in Fig. 1 targets realistic multi-view scene capturing by few cameras, e.g. 2 or 3. For these views, scene geometry has to be provided in order to enable synthesis of additional intermediate views.

The coded bit stream for transmission contains video and geometry data from few views and thus the overall bit rate is limited. A high-quality view synthesis will generate any number N of views for different displays within a restricted range around the original cameras at the receiver side. The 3D video bit stream can be used by any stereo or multi-view display, as the required number and spatial position of views is synthesized individually for each display.

3.1. 3D Video Format

One of the classical signal processing problems in multi-view video is the determination of the number and position of original cameras, which are required to reconstruct a 3D scene from arbitrary view points. For this, the original 3D scene can be considered as a continuous signal, which is recorded by a limited number of cameras at discrete positions. Thus, the recorded original continuous 3D signal is sampled by the discrete sampling grid of horizontal and vertical sensor elements in each camera, as well as by the discrete camera positions, creating a sampled light field. This "plenoptic sampling" was investigated by Chai *et al* in [1] as well as the conditions, under which intermediate views can be reconstructed without aliasing. As a result, the following sampling condition was derived for linear camera arrangements

$$\frac{2\pi}{\Delta s} \geq f\Omega_{u,\max} \underbrace{\left| \frac{1}{z_{\min}} - \frac{1}{z_{\max}} \right|}_{\text{depthrange}}. \qquad (1)$$

Here, Δs is the camera distance, f the camera focal length, $\Omega_{u.\max}$ the maximum horizontal frequency of an image and the depth range with minimum and maximum depth value z of the recorded scene. Equation (1) states, that the required camera distance is inversely proportinal to the depth range of the recorded 3D scene. Thus, if a scene has a large depth range, a small camera distance is necessary for alias-free intermediate view synthesis from the color data only. Note, that the highest horizontal frequency is usually fixed by the number of pixels of a given camera sensor.

Furthermore, it was shown in [1], that the use of additional geometry data, such as depth maps allows for larger camera distances Δs. In (1), it is assumed that a view synthesis is carried out, using only one (optimal) depth value z_{opt} for fulfilling the sampling condition. If more depth values of a scene are available, the depth range in (1) is split into sub ranges around each single depth value. Thus, the depth range is split into a number of smaller depth ranges

and the camera distance Δs can become larger than the camera distance for formats without depth data. Such a depth-enhanced 2-view format is shown in Fig. 3 with video and per sample depth data.

Fig. 3: Example for depth enhanced format: 2 view plus depth format for the *Breakdancers* set.

Such depth values can be obtained in different ways. One approach is to estimate the depth data based on the acquired pictures. Usually, methods for analyzing the displacement or disparity between identical scene objects in different camera views are applied. Although depth estimation algorithms have been improved considerably in recent years, they can still be erroneous in some cases due to mismatches, especially for partially occluded image content that is only visible in one view. Another method is the use of special sensors, like time-of-flight cameras, which record low-resolution depth maps. Here, post processing is required for interpolating depth for each color sample. Such sensors also lack depth accuracy for larger distances and have to be placed at slightly different positions than the video camera. It could be envisioned that in the future a recording device would capture high-precision depth for each color sample directly on the sensor. For synthetic sequences, such as computer generated scene content and animated films, scene geometry information is inherently available, e.g. in the form of wireframe models or 3D point coordinates. From this, depth data can be extracted as the distance between a selected camera position and the given geometry information.

Although depth data could also be generated at the receiver side using decoded video signals, we believe that depth data will be provided at the sender side and transmitted within the 3D video format. The advantage is that producers of 3D video content will have control over the resulting multi-view display output across different display types. Furthermore, different methods for depth provision can be used at the sender side, as described above and additional data can be incorporated for high-quality depth data provision, which is not part of the transmitted 3D video format.

The depth data is usually stored as inverted real-world depth data

$$stored\_depth = 255 \frac{\frac{1}{z_{\max}} - \frac{1}{z}}{\frac{1}{z_{\max}} - \frac{1}{z_{\min}}}. \quad (2)$$

Here, an 8 bit representation with values between 0 and 255 is assumed. This method of depth storage has the following advantages: Since depth values are inverted, a high depth resolution of nearby objects is achieved, while farther objects only receive coarse depth resolution. This also aligns with the human perception of stereopsis, where a depth impression is derived from the shift between left and right eye view. Thus, the stored depth values are quantized similar to these shift or disparity values.

However, the inverse quantized depth values are not identical to disparity values: The major difference is that disparity values depend on the camera distance or baseline, in contrast to depth values. This is of special importance since depth values are independent especially from neighboring cameras, as well as different camera sensor and image resolutions. Therefore, the stored depth representation in eq. (2) combines the advantages of inverse quantization for more natural depth capturing with the independency of depth data from camera baselines and image resolutions. For retrieving the depth values z from the depth maps, the following equation is applied, which is now widely used in synthesis scenarios:

$$z = \frac{1.0}{\frac{stored\_depth}{255.0} \cdot \left(\frac{1.0}{z_{\min}} - \frac{1.0}{z_{\max}} \right) + \frac{1.0}{z_{\max}}}. \quad (3)$$

For this, the original minimum and maximum depth values $z_{\min}$ and $z_{\max}$ are required, which have to be signaled with the 3D Video format for a correct geometric displacement in synthesized intermediate views.

3.2. 3D Video Coding Considerations

In section 2, it was discussed that current state-of-the-art coding approaches, such as MVC are only optimized for video signals. Therefore, new coding methods need to be developed, especially for the geometry or depth data. One initial approach, that considers better preservation of edges in depth maps, was introduced in [5]. Here, platelets were used as modeling functions for blocks in depth maps, for achieving a better representation of depth edges. As a result, the intermediate view quality could be improved, especially along foreground/background boundaries, as shown in Fig. 4. The synthesized result from uncoded depth and color data is shown in Fig. 4 *left*. Note, that even in this case, corona artifacts are visible, which can be removed by advanced synthesis algorithms [7]. If depth date is coded using MVC, color displacement artifacts around foreground objects due to depth edge smoothing become visible in Fig. 4 *middle*. In contrast, the foreground boundaries are much better preserved by the platelet depth coding approach, as

shown in Fig. 4 *right*. Only in cases with rather complex depth edge structures, some color displacement occurs.

Fig. 4: Impact of depth coding artifacts on view synthesis rendering for *Ballet* sequence with uncoded color data: *left*: original uncoded depth, *middle*: H.264/MVC coded depth, and *right* Platelet coded depth.

With the provision of per-sample depth data, any number of views within a given range can be synthesized from a few input views. Based on the principles of projective geometry, arbitrary intermediate views are generated via 3D projection or 2D warping from original camera views. This is typically referenced as depth-image-based rendering (DIBR). For currently investigated 3D video solutions, the camera views are rectified in a preprocessing step. Thus, the complex process of general DIBR can be simplified to horizontal sample shifting from original into newly rendered views.

One important aspect for the design of new 3D video coding methods is the quality optimization for all synthesized views. In classical 2D video coding, a decoded picture can always be compared against the uncoded reference and the quality be evaluated by distance measure, such as sample-wise mean squared error for PSNR. For the new 3D video format, however, a good picture quality has to be guaranteed for a continuous viewing range. That means, views are synthesized at new spatial positions, where no original reference image is available. For objective evaluation methods, however, some form of reference is required. For this, high quality depth data as well as a robust view synthesis are required, in order to generate a synthesized uncoded reference, as shown in Fig. 4 left. This reference should ideally be indistinguishable from original views. Then, comparing any decoded synthesized view with its uncoded version can objectively assess coding approaches.

4. SUMMARY AND CONCLUSIONS

This paper gave an overview on 3D video formats and coding. First, state-of-the-art MVC was introduced for coding of *N* color-only video sequences captured by camera arrays. However, the bit rate of MVC is dependent on the number of views, and the MVC format does not allow for synthesis of new views. Therefore, new depth-enhanced formats were introduced. Adding scene geometry e.g. in the form of quantized per-sample depth maps, allows alias-free view synthesis for much wider camera distances and thus reduces the number of necessary input views. Additionally, the restriction to few input views is feasible for real-world capturing of natural content. Depth-enhanced formats therefore enable to decouple the transmission format from the display format and provide the same bit stream to any 3D display, independent of the number and spatial positions of views. The new 3D video format also requires new coding approaches, since depth data has different statistics than color data. Especially sharp depth edges need to be preserved to avoid color displacement errors in synthesized views. Finally, a robust high-quality view synthesis is required in order to provide any number of display views.

ACKNOWLEDGMENT

We would like to thank the Interactive Visual Media Group of Microsoft Research for providing the *Ballet* data set.

REFERENCES

[1] J.-X. Chai, X. Tong, S.-C. Chan, H.-Y. Shum, “Plenoptic sampling”, Proc. SIGGRAPH, 2000.

[2] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audiovisual services," ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 1: May 2003, Version 2: May 2004, Version 3: Mar. 2005 (including FRExt extension), Version 4: Sep. 2005, Version 5 and Version 6: June 2006, Version 7: Apr. 2007, Version 8: July 2007 (including SVC extension), Version 9: July 2009 (including MVC extension).

[3] ISO/IEC JTC1/SC29/WG11, “Text of ISO/IEC 14496-10:200X/FDAM 1 Multiview Video Coding”, Doc. N9978, Hannover, Germany, July 2008.

[4] ISO/IEC JTC1/SC29/WG11, “Vision on 3D Video “, Doc. N10357, Lausanne, CH, February 2009.

[5] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P.H.N. de With, and T. Wiegand, “The Effects of Multiview Depth Video Compression on Multiview Rendering”, Signal Processing: Image Communication, vol. 24, is. 1+2, pp. 73-88, January 2009.

[6] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, “Efficient Prediction Structures for Multiview Video Coding”, invited paper, IEEE TCSVT, Vol. 17, No. 11, November 2007.

[7] K. Müller, A. Smolic, K. Dix, P. Merkle, and T. Wiegand, “Coding and Intermediate View Synthesis of Multi-View Video plus Depth”, Proc. IEEE International Conference on Image Processing (ICIP'09), Cairo, Egypt, pp. 741-744, Nov. 2009.

[8] H. Schwarz, D. Marpe, and T. Wiegand, “Analysis of hierarchical B pictures and MCTF”, ICME 2006, IEEE International Conference on Multimedia and Expo, Toronto, Ontario, Canada, July 2006.

[9] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. on Circuits and Systems for Video Tech., vol. 13, no. 7, pp. 560-576, July 2003.

[10] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-Quality Video View Interpolation Using a Layered Representation”, *ACM SIGGRAPH and ACM Trans. on Graphics*, Los Angeles, CA, USA, August 2004.