MULTI-VIEW VIDEO CODING VIA VIRTUAL VIEW GENERATION

E. Ekmekcioglu, S.T. Worrall, A.M. Kondoz
{E.Ekmekcioglu, S. Worrall, A. Kondoz}@surrey.ac.uk
Centre for Communication Systems Research, University of Surrey,
GU2 7XH Guildford, Surrey, United Kingdom

ABSTRACT

In this paper, a multi-view video coding method via generation of virtual picture sequences is proposed. Pictures are synthesized for the sake of better exploitation of the redundancies between neighbouring views in a multi-view sequence. Pictures are synthesized through a 3D warping method to estimate certain views in a multi-view set. Depth map and associated colour video sequences are used for view generation and tests. H.264/AVC coding standard based MVC draft software is used for coding colour videos and depth maps as well as certain views which are predicted from the virtually generated views. Results for coding these views with the proposed method are compared against the reference H.264/AVC simulcast method under some low delay coding scenarios. The rate-distortion performance of the proposed method outperforms that of the reference method at all bit-rates.

Index Terms—*MVC, video coding, virtual view synthesis.*

1. INTRODUCTION

The issue of multi-view video coding has grown in significance following recent advances in 3D capture and display technologies, making the applications like 3D TV [1] and Free viewpoint TV (FTV) [2] possible. However, in contrast with traditional single viewpoint videos, multi-viewpoint videos require a much higher bandwidth for transmission. Hence, it is critical that advanced compression techniques are used for coding of multi-view video data in terms of exploitation of redundancies in temporal domain as well as exploitation of inter-view redundancies. In order to make researchers concentrate on this way, MPEG has initiated a working group for the standardization of MVC and issued a Call for Proposals on MVC [3].

Certain types of proposed multi-view video coding algorithms include motion compensation based prediction from inter-view and temporal references in a multi-view video set. The already built-in motion estimation and motion compensation functions of the modern video codecs, especially of H.264/AVC, are used to remove temporal redundancies inside a certain view itself as well as to remove spatial redundancies between neighbouring views after slight modifications. The MVC method proposed in [4], based on *hierarchical B frame prediction* of in both temporal and spatial dimensions using H.264/AVC gives the best compression performance so far among other reference frame based proposals.

However, the motion compensation methods used for removing temporal redundancies are not enough to remove inter-view redundancies. The motion search range for the detection of motion in time can be kept within a limit but the motion in space is more complicated in the sense that it is both dependent on the distance between cameras and on the scene geometry. Therefore, exploitation of the scene geometry through the use of the camera parameters and depth maps is essential for achieving better compression efficiency. In [5], one solution explicitly exploiting the scene geometry is proposed, while in [6], a solution is proposed using the scene geometry implicitly without using depth maps. In [7], several prediction methods are analyzed. Moreover, most future applications requiring MVC already necessitate the usage of the scene geometry information [8, 9]. So, it is beneficial to use this information during the

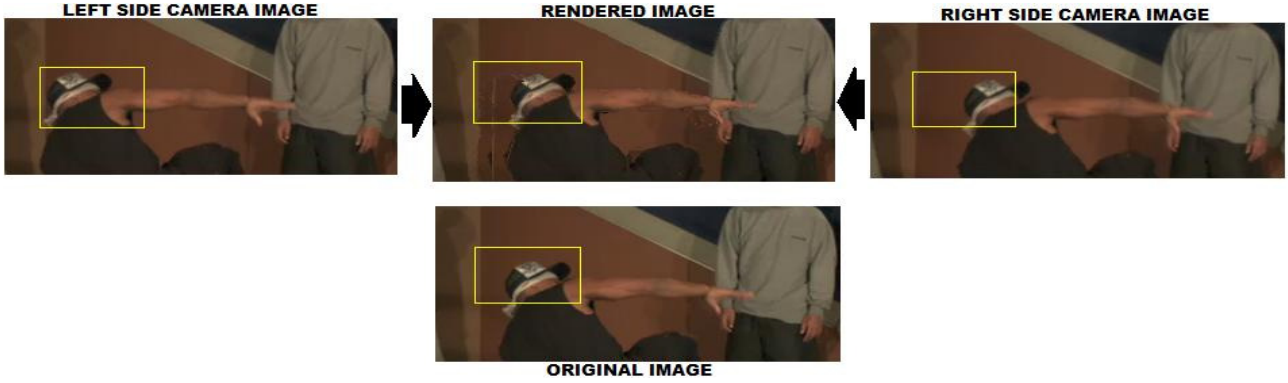

Figure 1: *Final rendered image (in the middle) constructed from the left and right rendered pictures.*

compression stage.
In the following sections, a multi-view coding method solely based on scene geometry exploitation and prediction from virtual sequences is explained in detail. Section 2 describes the virtual sequence generation stage. In section 3, the prediction method of the proposed MVC is explained in detail. Section 4 shows the simulation results and finally, section 5 gives the concluding remarks with an analysis of the produced performance graphs.

2. GENERATION OF VIRTUAL SEQUENCES

In order to be able to remove the spatial redundancy among neighbouring views in a multi-view set, caused by the 3D structure of the views, virtual sequences are rendered from already encoded frames of certain views. These views will be called "base view" in the rest of the paper. The rendered frames are then used as alternative predictions for the according frames to be predicted in certain views. These views will be called "intermediate view" or equivalently "b view" throughout the rest of the paper.
The virtual views are rendered through the unstructured lumigraph rendering technique explained in [10]. In our case, this method uses an already encoded picture of the base view, which is projected first to a 3-D world with the pinhole camera model and then projected back to the image coordinates of the intermediate view, taking into account the camera parameters of both the base view and the intermediate view. The pixel in base view image coordinate, (x,y) is projected to 3-D world coordinates using

$$[u, v, w] = R(c) \cdot A^{-1}(c) \cdot [x, y, 1] \cdot D[c, t, x, y] + T(c) \quad (1)$$

where [u, v, w] is the world coordinate. Here, c defines the base views camera. R, T, A define the 3x3 rotation matrix, the 3x1 translation vector and the 3x3 intrinsic matrix of the base view camera, respectively and D[c,t,x,y] is the distance of the corresponding pixel (x,y) from the base view camera at time t [10]. The world coordinates are mapped back to intermediate view image coordinate system using

$$[x', y', z'] = A(c') \cdot R^{-1}(c') \cdot \{[u, v, w] - T(c')\} \quad (2)$$

where [(x'/z'), (y'/z')] is the corresponding point on the intermediate views image coordinate system [10].
The matrices in the equations (1) and (2), i.e. R, T and A, and the corresponding depth images of the base views are provided by Microsoft Research for the multi-view *Breakdancer* sequence [8]. The camera parameters should be supplied to the image renderer, but the depth maps can be generated in the renderer, by using several techniques. A depth map extraction algorithm based on disparity matching between two views can be found in [9]. In our experiment, we use the depth maps supplied by Microsoft Research. In the proposed method, 3-D warping procedure is carried out pixel by pixel. However, care should be taken to avoid several visual artefacts.
First, some pixels in the reference picture may be mapped to the same pixel location in the target picture. In that case, a depth sorting algorithm for the pixels falling on the same point in the target picture is applied. The pixel, closest to the camera is displayed.
Second, not every pixel may fall on integer pixel locations. The exact locations should be rounded to fit to the nearby integer pixel locations in the target image. This makes many small visual holes appear on the rendered image. The estimates for empty pixels are found by extrapolating the nearby filled pixels, which is a valid estimation for holes with radius smaller than 10 pixels.
Third, some pre-processing for depth maps need to be carried out. Unlike the case of stereoscopic view generation, where it is very beneficial to smooth the depth maps [11], it is useful to make the depth transitions in MVC sharper especially between foreground and background objects. Unless the pre-processing is done, the pixel extrapolation process is misguided and foreground pixels in the vicinity of holes are treated as background pixels resulting in poor hole filling performance.
For every intermediate view, the two neighbouring base views are warped separately into the intermediate view image coordinate system. One of the resulting views yielding the best objective quality measurement is chosen for the prediction. For better prediction quality and better usage of the scene geometry, the formerly occluded regions in the final prediction view are compared with the corresponding pixels in the other warped image. *Fig. 1* shows a sample final rendered image segment which is formed from two side camera images.

3. MVC PREDICTION METHOD

Rendered views are considers as alternative prediction sources, in addition to temporal and inter-view predictions. There are two motivations for including the virtual images as predictions for intermediate view images. One of them is that since the 3-D scene geometry is utilized, inter-view redundancies are best removed by this way. However, since the camera calibration parameters cannot be known accurately, prediction quality is not perfect. Another motivation is that for future video applications, particularly for FTV, transportation of depth information will already be essential [12]. So, exploiting that information to improve compression of certain views (intermediate views in this case) would be quite reasonable. Besides the rendered references, it is important to not remove temporal references

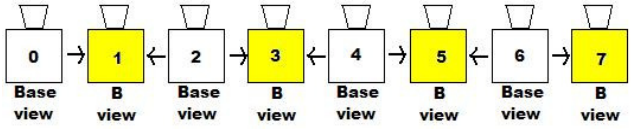

Figure 2. *Camera arrangement and view assignment*

from the prediction list, since the temporal references occupy the highest percentage of the references used for prediction in *hierarchical B frame prediction* [4]. In our test, other means of inter-view references are removed in order to be able to see the extent to which the proposed method outperforms conventional temporal predictive coding techniques. *Fig. 2* shows the camera arrangement and view assignment for the *Breakdancer* sequence. The view assignment is flexible. There are two reasons for such an assignment although it may not be the optimum assignment in a sense to minimize the overhead caused by depth map coding. One reason is that, for any intermediate view, the two closest base views are used for 3-D warping making the most likely prediction frame be rendered. Another reason is to be able to show the effects of using prediction frames rendered using just one base view. In our case, virtual prediction frames for the coding of intermediate view 7 are rendered using only base view 6.

The H.264/AVC based MVC draft software, Joint Multi-view Video Model (JMVM) [13], is used for the proposed multi-view video coding scenario. Both the colour videos and depth maps of the base views are encoded in H.264 simulcast mode (no inter-view prediction). However, the original depth maps are downscaled to their half resolution prior to encoding. The fact that depth maps don't need to include full depth information to be useful for stereoscopic video applications [11], motivated us to use downscaled versions of depth maps containing more sparse depth information. In the experiments, use of reduced resolution depth maps affected the reconstruction quality at the decoder negligibly, even for very low bit-rates. The PSNR of decoded and up-sampled depth maps changed between roughly 33 dB and 34.5 dB. *Table 1* shows the coding conditions for base views and depth maps.

Following the coding of base views with their depth maps, intermediate views are coded using the rendered virtual sequences as inter-view references. In this case, the original frames at I-frame and P-frame positions are coded using the corresponding virtual frame references. At P-frame locations, temporal referencing is still enabled. A lower quantization parameter is used for coding intermediate views. The prediction structure for intermediate view coding is illustrated schematically in *Fig. 3*. One reason for such a prediction structure is that we wanted to explore the coding performance of the proposed scheme for low delay coding scenarios. Besides, as the GOP size increases, where the coding performance of temporal prediction is maximized, the effect of the proposed method on the overall coding efficiency becomes less visible. It was observed in experiments, for GOP size of 12, that the proposed technique had no gain compared to the reference technique (H.264 based simulcast method).

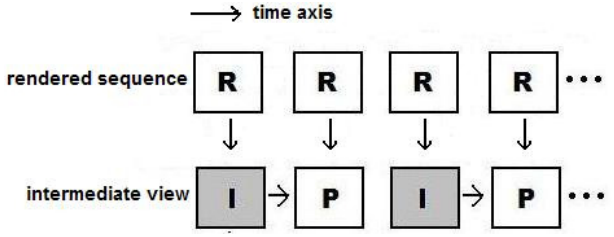

Figure 3: *Prediction structure of intermediate views.*

Table 1: *Codec configuration.*

Software	*JMVM 2.1*
Symbol mode	*CABAC*
Loop filter	*On (colour video), Off (depth maps)*
Search range	*96*
Prediction structure	*I P I P … (low delay, open GOP)*
Random access	*0.08 second (25 fps video)*

4. SIMULATION RESULTS

Fig. 4 (a)-(b) show the performance comparison of the proposed MVC method with H.264 based simulcast coding. Due to the reason discussed in *section 3*, the overhead caused by coding of depth maps is not taken into account when forming the performance graphs. Nevertheless, the coding bit-rate of the depth map didn't exceed 20% of the coding bit-rate of the associated colour video. *Fig. 4 (c)-(d)* show the performance comparisons between the proposed method and the reference method, where all frames in base views are intra coded and intermediate views are predicted only from rendered virtual sequences. *Fig. 4 (e)-(f)* show the results for *Ballet* test sequence, supplied by Microsoft Research with their per-pixel depth maps.

5. CONCLUSION

According to *Fig. 4*, the coding performance is improved in comparison to combined I and P prediction. The difference in gain between *Fig. 4 (a)* and *(c)* shows us that the proposed method has a considerable gain over intra coded pictures but also that the temporal references should be kept as prediction candidates to achieve optimum coding performance. Similar results are observed in *Fig. 4 (b)* and *(d),* where the performance of the proposed method is analysed for intermediate view 7. The proposed method still outperforms the reference coding method and the gain over intra coding is significant. The overall decrease in average coding gains when compared to that of the intermediate views 1, 3 and 5, shows that virtual sequences, rendered using two base views, can predict the original view better than the virtual sequences rendered using only one base view. Similar results are obtained for *Ballet* sequence as can be seen in *Fig. 4 (e)* and *(f)*. The subjective evaluation of the proposed method was satisfactory. Accordingly, the proposed method is suitable for use in multi-view applications under low-delay constraints.

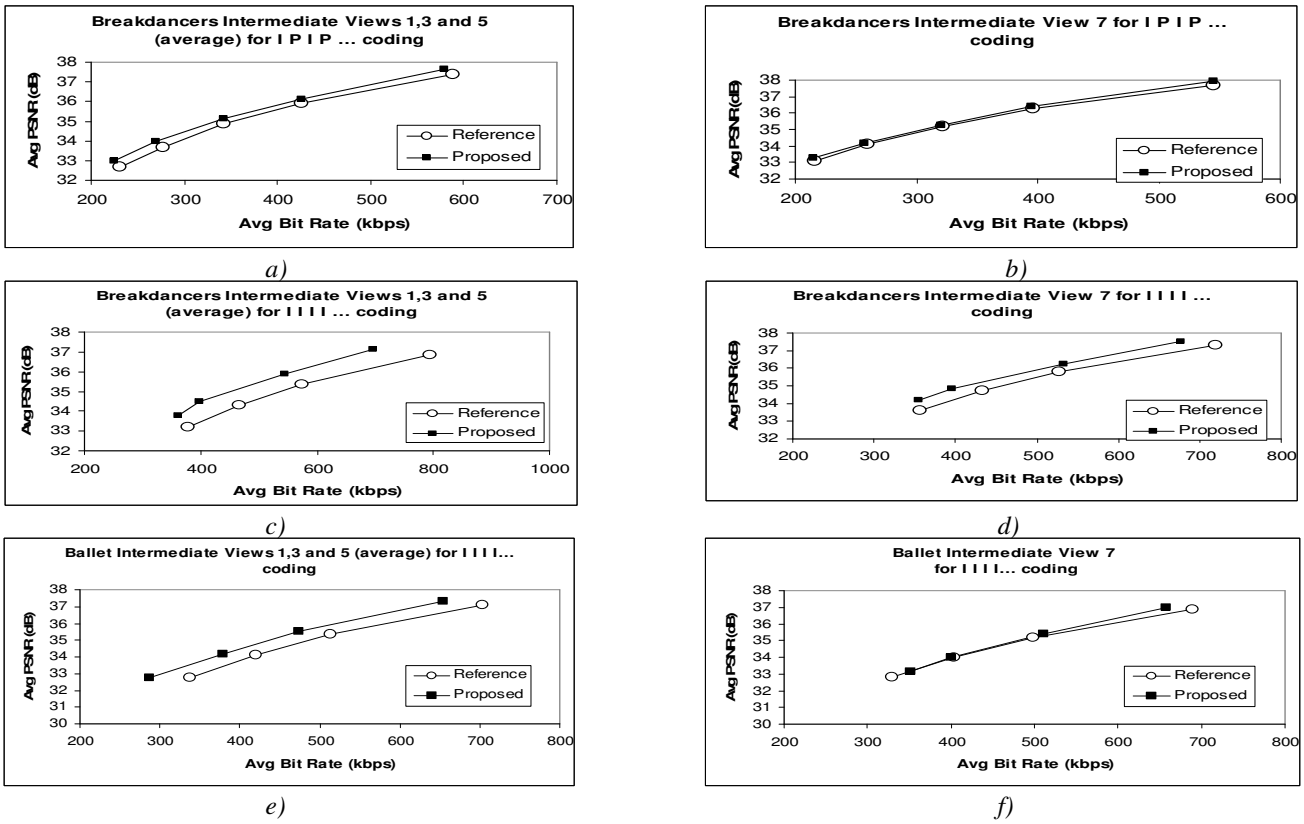

Figure 4: *Rate-distortion performance of proposed and reference schemes.*

6. ACKNOWLEDGEMENTS

The work presented was developed within VISNET II, a European Network of Excellence (http://www.visnetnoe.org), funded under the European Commission IST FP6 programme.

7. REFERENCES

[1] L. Onural, "Television in 3-D: What are the prospects?", Proceedings of IEEE, Vol. 95, No. 6, June 2007.

[2] M. Tanimoto, "Overview of free viewpoint television", Signal Processing: Image communication 21, pp. 454-461, 2006.

[3] ISO/IEC JTC1/SC29/WG11. Updated call for proposal on multi-view video coding, 2005.

[4] K. Müller and et. al., "Multi-view Video Coding Based on H.264/AVC Using Hierarchical B-Frames", PCS 2006, China, 2006

[5] E. Martinian and et. al., "View Synthesis for Multi-view Video Compression", Proc. PCS 2006, Beijing, China, April 2006.

[6] M. Kitahara and et. al., "Multi-view Video Coding using View Interpolation and Reference Picture Selection", IEEE International Conference on Multimedia and Exposition, Toronto, Ontario, Canada, July 2006.

[7] E. Martinian and et. al., "Extensions of H.264/AVC for Multiview Video Compression", IEEE ICIP, ISSN: 1522-4880, pp. 2981-2984, October 2006.

[8] C. L. Zitnick and et. al., "High-quality video view interpolation using a layered representation," ACM Siggraph and ACM Trans. on Graphics Aug. 2004.

[9] P. Kauff and et. al., "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability", Signal Processing: Image Communication 22, pp. 217-234, February 2007.

[10] Sehoon Yea and et. al., "Report on Core Experiment CE3 of Multiview Coding", JVT-T123, Klagenfurt, Austria, July 2006.

[11] W. J. Tam and L. Zhang, "Depth map preprocessing and minimal content for 3D-TV based on DIBR", JVT-W095, San Jose, California, USA, April 2007.

[12] M. Tanimoto and et. al., "Proposal on Requirements for FTV", JVT-W127, San Jose, California, USA, April 2007.

[13] A. Vetro, Y. Su, H. Kimata, and A. Smolic, "Joint multiview video model JMVM 2.0," ITU-T and ISO/IEC Joint Video Team, Document JVT-U207, November 2006.