Unpaired Multiview Video Plus Depth Compression

Payman Aflaki, Moncef Gabbouj
Department of Signal Processing
Tampere University of Technology
Tampere, Finland
(payman.aflaki, moncef.gabbouj)@tut.fi

Dmytro Rusanovskyy, Miska M. Hannuksela
Nokia Research Center
Nokia Corp.,
Tampere, Finland
(dmytro.rusanovskyy, miska.hannuksela)@nokia.com

***Abstract*— Recent developments of three-dimensional (3D) video coding greatly rely on the use of Multiview Video plus Depth (MVD) data format for representing a 3D scene. This type of data can be coded with conventional video compression schemes and can enable advanced 3D video functionality, such as rendering of virtual views at the decoder side. The MVD represents a 3D scene from different viewing angles as video and pixel-wise associated depth data. In this paper we consider the redundancy of depth data in the MVD representation and propose a novel scheme, called Unpaired MVD (UP-MVD) format to be used in 3D video applications. Being a subset of the MVD this new format assumes that a reduced number of depth views compared to the number of texture views can reduce bitrate as well as the encoding/decoding complexity while still providing the 3DV functionality in many scenarios. The simulation results show that the proposed unpaired MVD format outperforms the MVD format on average from 0.9% to 6.95% of Bjontegaard delta bitrate (dBR) for the baseline disparity adjustments from 50% to 100% of the coded baseline, respectively. Moreover, UP-MVD provides equal or better rate-distortion results for all test sequences for up to 20% view separation adjustment, and in five out of seven sequences a better rate-distortion performance is observed when 50% view separation adjustment is applied.**

Keywords-3DV; MVD; View synthesis.

I. INTRODUCTION

The presence of depth data at the decoder side enables a more flexible display of three-dimensional (3D) video compared to conventional stereoscopic and multiview video coding. While coding of two texture views provides a basic 3D perception on stereoscopic displays, it has been discovered that disparity adjustment between views is needed for adapting the content on different displays and viewing conditions as well as for individual preferences [1]. Moreover, in autostereoscopic displays a large number of views is typically required to be displayed simultaneously. However, it is impossible or impractical to transmit a large number of views through today's networks, such as the Internet, using existing video compression standards, such as the Multiview Video Coding (MVC) extension of the Advanced Video Coding (H.264/AVC) standard [2]. Therefore, the required views have to be generated in the playback device from the received views. These needs can be addressed by representing a 3D scene with a multiview video plus depth (MVD) format [3] and using the decoded MVD data as source for depth image-based rendering (DIBR) [4]. In the MVD format each video data pixel is associated with a corresponding depth map value from which new views can be synthesized using any DIBR algorithm in the post processing stage before displaying the content.

The Moving Picture Experts Group (MPEG) issued a Call for Proposals (CfP) on 3D video coding technology in March 2011 [5]. The CfP aimed at starting the standardization of a coding format that supports advanced stereoscopic display processing and improved support for auto-stereoscopic multiview displays. As a result of the CfP evaluation [6], the MPEG and, since July 2012, the Joint Collaborative Team on 3D Video Coding (JCT-3V) [7] have initiated the development of an MVC extension to include depth maps [2], abbreviated as MVC+D, and to specify the encapsulation of coded MVD data into a single bitstream [8]. According to this specification, MVC coding [2] is applied independently to both texture and depth components of MVD, and the texture views of MVC+D bitstreams can be decoded with a conventional MVC decoder. A reference test model of MVC+D is implemented in 3DV-ATM reference software [9] and it was used in our work.

DIBR enables the projection of a texture view to a virtual viewing position. However, DIBR has intrinsic limitations of being unable to render samples in areas that become uncovered in synthesized views (termed "dis-occlusions" or "holes") in the capturing process. This mainly happens while extrapolating one view, i.e., while rendering a view in a specific location from information of a single view. To overcome this, view interpolation can be performed in DIBR algorithms, i.e., a middle view can be rendered by exploiting information of the left- and right-side views while performing projection from different directions to fill the dis-occluded parts. The dis-occlusion problem of DIBR is due to areas covered by objects in the reference view which appear in the synthesized view. Such holes and dis-occluded areas have neither a certain depth or texture attribute, nor a correspondence in the reference views. In DIBR, these holes need to be filled properly otherwise annoying artifacts will appear in the dis-occluded regions. To solve the problem of dis-occlusions in DIBR, several algorithms have been developed. There has been an extensive research proposing different hole-filling methods [10] using simple and sophisticated image processing techniques. Most of these conventional methods exploit neighbor pixel values to fill-in the holes by extrapolation, linear interpolation, or diffusion techniques.

In our work, we assumed that the conventional MVD representation with pixel-wise correspondence between texture and depth can be redundant for some 3D scenes and for some of use cases. It is assumed and proved in this paper that in many application scenarios, the amount of depth map data describing the 3D scene can be reduced without a significant impact on DIBR performance. Analogously to [11], [12] and [13], where it was shown that depth information can be spatially decimated within a single view improving the performance of DIBR, we consider that depth data for some views can be completely ignored since it does not present a considerable amount of information in addition to other presented views.

As a result of our study, we propose a novel Unpaired MVD (UP-MVD) format to be used in 3D video applications. Being a subset of the MVD data format, UP-MVD reduces the number of depth views compared to texture views while maintaining a sufficient view synthesis capability for typical 3DV applications and reducing the encoding/decoding complexity. The process of removing the redundant depth components within MVD data can be conducted at the post-production stage of 3D video content capturing or at the encoder side of the 3DV coding chain.

The rest of paper is organized as follows. Section II describes the proposed unpaired MVD scheme. The test material and simulation results are presented in Section III, while Section IV concludes the paper.

II. PROPOSED UNPAIRED MVD SCHEME

A. Motivation

MVD represents a 3D scene from different viewing angles as video and pixel-wise associated depth data. However, as demonstrated in many of the responses to the 3DV CfP [5], coding of depth map data at a reduced resolution is a viable solution for improving the rate-distortion performance of the complete 3D video coding system. As a result, for most of available MVD content, e.g. MPEG 3DV Test Set [14], the depth component can be spatially decimated within each view without a significant impact on the performance of DIBR-based 3D Video applications. As a follow up of this concept, we considered view-level decimation of the depth data, and studied the redundancy within the currently available MVD content confirming that the number of depth data compared to texture views can be reduced without sacrificing the quality of synthesized views. As a result of the analysis of practical scenarios of 3D scene capturing and 3D video applications, a few examples where a complete MVD representation is unnecessary and/or redundant are presented below.

A typical 3D video scene capturing process nowadays features a stereo camera and depth information is derived through disparity estimation process having a stereoscopic video as input. A disparity search is performed in a one-directional manner, i.e. for each sample in the first view component (e.g. a picture of the left view), the corresponding sample in the second view component (e.g. a picture of the right view) is searched. The resulting disparity picture can be converted to a depth component of the first view. The depth information for the second view can be produced by inverting the disparity vectors computed for the first view. Thus, there is no need to produce and code depth component for the second view, since this information would be completely redundant.

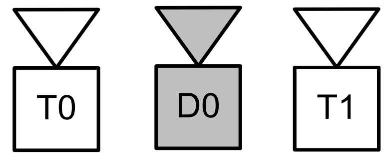

Figure 1. Stereo camera (T0 and T1) with a single ranging device (D0)

Alternatively, depth information may be generated using a specific depth sensor [15] [16] rather than generated in a per-pixel depth estimation process based on texture views. In such a camera setup, it is typical that a depth sensor is not collocated with any of the utilized image sensors. A visualization of this concept is shown in Fig. 1 where two cameras are accompanied by a single ranging device which is not located in the same place as any of the image sensors.

In some practical 3D Video applications, a stereo baseline adjustment would be required in relatively close proximity to one decoded texture view, whereas another decoded texture view would be displayed as it is. Therefore, a depth data associated with the first view can be sufficient as the input for DIBR in extrapolation mode and no need to process depth data associated with the second view. This example is illustrated in Fig. 2 where four stereo baselines can be achieved depending on the use of L, L1, L2 or L3 as the left view. Hence, even if a depth view is available for encoding for each texture view, this complete MVD representation for 3D scene may be unnecessary for enabling many 3D video applications and the encoder side may be

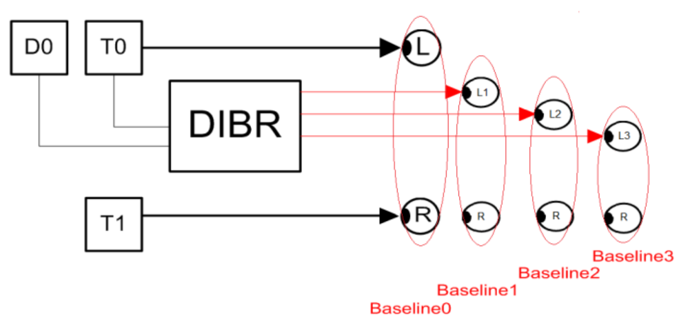

Figure 2. Visualization of depth perception adjustment by view synthesis in close proximity from one decoded view
T0 and T1 are input texture views while D0 is the only input depth view
L1, L2, and L3 present the synthesized views

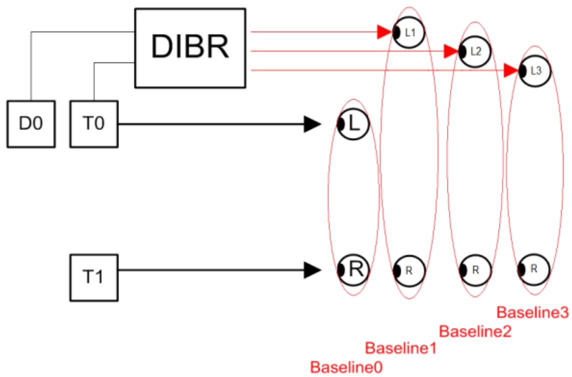

Figure 3. Visualization of depth perception adjustment by view synthesis to an extrapolated view
T0 and T1 are input texture views while D0 is the only input depth view
L1, L2, and L3 present the synthesized views

adjusted to limit the number of depth views to be encoded.

The need for synthesizing only one view from stereoscopic video also applies for use cases when the extrapolation of virtual views rather than interpolation is required. For example, an optimal disparity for a two-view autostereoscopic display for handheld use is typically wider than that for a living-room polarized or a shutter-glass display. Depth-enhanced stereoscopic content for handheld devices could therefore have two texture views and only one depth view, as the extrapolation quality would remain the same as that for two texture and depth views. This is depicted in Fig. 3.

As stated in Section I, DIBR does not guarantee a full and correct projection of one view to the target location, because some parts of the projected view may be dis-occluded or contain holes. The percentage of hole pixels in a rendered view depends on many factors e.g. scene characteristics and the view separation between the original and projected views. In the following paragraph, we evaluate the forward projection process on the first frame of depth views of the 2-view sequences used in this paper in order to demonstrate that the two depth views are correlated

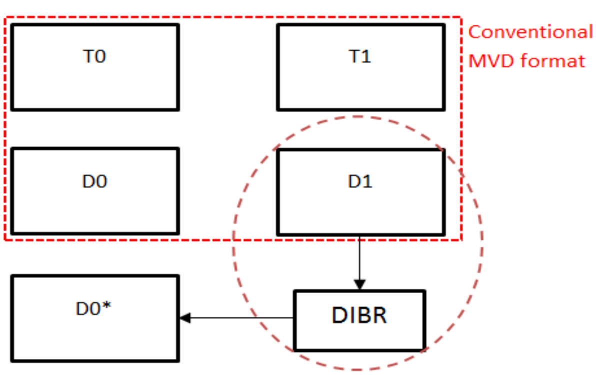

Figure 4. Forward projection of depth maps to calculate percentage of hole pixels in the rendered view

and that the second depth view contains only a moderate proportion of pixels that could not be projected from the first depth view.

For simplicity, the case with two texture views and their associated depth maps is considered (2-view MVD data). However, this approach can be extended to be applied for MVD content with more than two views. Considering Fig. 4, D1 is projected forward to the position of view 0 to produce D0*. The missing information in D0* results either from a different field of view compared to D0 or from the inaccuracy of the depth estimation and the depth projection algorithms. The location of the missing information is marked as a "hole". Table I shows the proportions of pixel locations with such missing information normalized by the number of pixels in a depth image D0 for the first frame of each sequence. The estimates have been evaluated with the MPEG 3DV test set [14]. As reported in Table I, on average more than 95% of the pixel locations in D0* contain projected pixels from D1 and hence, D0* can be assumed to provide a proper estimation of D0. This is further studied in Section III with a series of simulations and reported objective results.

Moreover, the encoder and the decoder complexity are reduced as a direct result of the view reduction introduced in proposed MVD format compared to the conventional MVD format.

B. *Proposed schemes*

In this sub-section, we describe a novel 3D video processing scheme that is based on the unpaired MVD format. The input of the proposed scheme is a conventional MVD format with a reduced number of coded depth views.

After removing some of the depth views, the full set of texture views and a subset of the depth views is coded with a modified MVC+D codec [9], resulting in a smaller bitrate than that of the respective complete MVD data coded with MVC+D. At the decoder side, the UP-MVD data is used for rendering virtual views with DIBR and the quality of these views is used for the performance evaluation. In our work we studied the impact of the use of unpaired MVD with two schemes, the details of which are given below.

Scheme 1 targets low-complexity encoder and decoder operations with reduced memory requirements, whereas Scheme

TABLE I. PERCENTAGE OF DISOCCLUDED AND UN-PROJECTED PIXELS IN D0*

Sequence	
Poznan Hall2[17]	3.8%
Poznan Street	2.2%
Undo Dancer	3.4%
Ghost Town Fly	4.3%
Kendo	11.4%
Balloons	5.4%
Newspaper	3.5%
Average	**4.9%**

2 assumes a more advanced decoder providing a higher subjective and objective quality.

Scheme 1:

A flowchart of this scheme is depicted in Fig. 5. First, two texture views and one depth view are encoded and decoded. Then, rendered views are created with an extrapolation of the base view (T0 and D0). The stereoscopic image-pair is made of one coded texture view (T1) and for the other view based on the desired baseline, either the other coded view (T0) or one of the synthesized views (L1, L2, L3) is used. This approach has a low computational complexity and memory requirement due to the fast extrapolation from one view. However, the drawback of this scheme is the omission of T1 in the rendering process where it can significantly improve the quality of the synthesized views.

Scheme 2:

This scheme is similar to Scheme 1 but for the rendering process, both texture views are utilized. The corresponding flowchart of Scheme 2 is shown in Fig. 6. In this scheme, D1* is produced from the available D0 through a projection to view location 1 followed by basic hole filling [10]. The desired views (L1, L2, and L3) are then obtained by interpolation using T0 and D0 as well as T1 and D1*. This enables exploiting the texture information of T1 associated with D1* in the rendering process, and therefore, improves the quality of view synthesis compared to that achievable with Scheme 1.

III. SIMULATION RESULTS

The simulation results run under the specifications of C2 scenario of 3DV Common Test Conditions (CTC) [14] and the complete set of MPEG 3DV test sequences was utilized. In this scenario two depth-enhanced texture views are encoded and then several possible intermediate views are synthesized in-between to be exploited in stereoscopic image-pair creation. 3DV-ATM software configured in MVC+D was utilized for coding UP-MVD, and 3DV VSRS [18] was used for the rendering of virtual the depth and texture views.

In our experiments, stereo baseline adjustment was enabled in the proximity of one of the decoded texture view, whereas the other decoded texture view was used as the second view of the displayed stereoscopic image-pair. Rendered views were located at 0.1, 0.2, and 0.5 of the baseline named L1, L2, and L3, respectively, as it is shown in Fig. 5. The specific views used at the input and output of this experiment are listed in Table II.

The quality of the stereoscopic video was compared to the anchor case where both views were coded with their associated depth map. To evaluate the performance objectively, the Peak

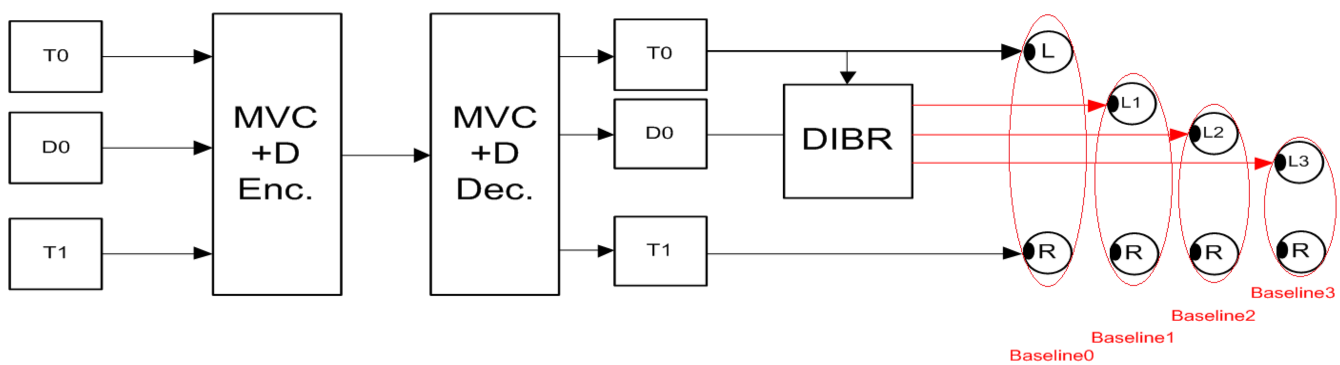

Figure 5. Proposed Scheme 1 for Unpaired MVD

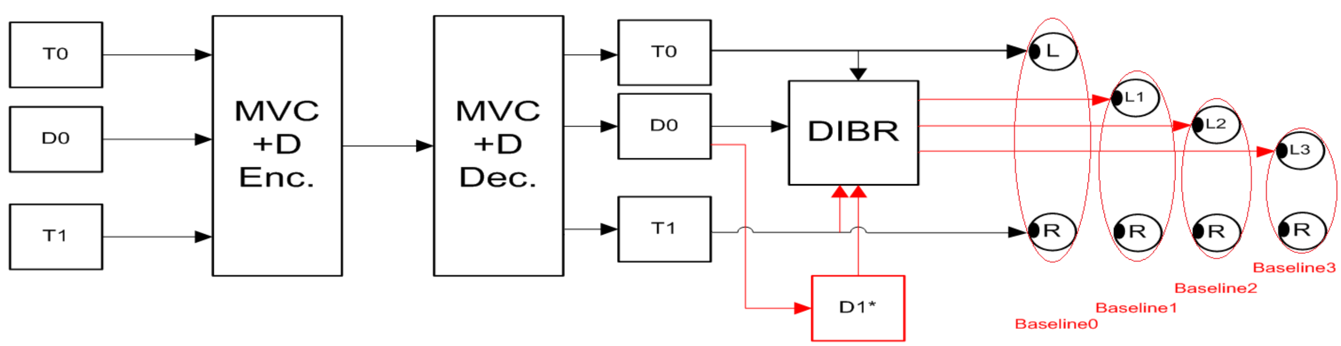

Figure 6. Proposed Scheme 2 for Unpaired MVD

TABLE II. 3DV TEST SEQUENCES, INPUT AND SYNTHESIZED VIEWS

Sequence	Input views	Synthesized views
PoznanHall2	7-6	7.5 – 7.8 – 7.9
Poznan Street	5-4	5.5 – 5.8 – 5.9
Undo Dancer	1-5	3 – 4.2 – 4.6
Ghost Town fly	9-5	7 – 5.8 – 5.4
Kendo	1-3	2 – 3.6 – 3.8
Balloons	1-3	2 – 3.6 – 3.8
Newspaper	2-4	3 – 4.6 – 4.8

Signal-to-Noise Ratio (PSNR) of both views in the stereoscopic image pairs (including one coded view and one synthesized view) was calculated. The PSNR of the synthesized views was calculated against the reference synthesized views created from the conventional MVD format including the original uncompressed texture and depth views as specified in MPEG CTC [14]. The objective results are presented using Bjontegaard delta bitrate (dBR) and delta PSNR [19]. The delta bitrate (dBR) is presented for the coded views and the stereoscopic image pairs created with one decoded view and one synthesized view. The results for the first and the second proposed schemes are provided in Table III and IV, respectively. Moreover rate-distortion curves for the largest baseline stereoscopic image pair (base + 0.9) for Scheme 2 is depicted in Fig. 7. The results show that a bitrate reduction, compared to the anchor, of 6.95% for coded views is achieved due to the removal of one depth view. Moreover, in Scheme 1, a larger baseline (i.e. base + 0.9) provides 4% dBR gain with a visible degradation in the performance when decreasing the baseline. However, this problem is addressed in Scheme 2 where a higher average performance compared to the anchor is always achieved. As reported in Table IV, the smallest baseline achieves 0.9% of dBR compared to the anchor. This gain increases up to 5.44% for the largest baseline achieved from one decoded and one synthesized view. It can also be concluded that using Scheme 2, up to 20% view separation adjustment can be achieved while the proposed UP-MVD format always outperforms the anchor conventional MVD format. Moreover, in five out of seven test sequences in the smallest baseline configuration, the anchor is outperformed by the proposed method.

IV. CONCLUSIONS

In this paper we studied the objective quality of encoded and synthesized views from an MVD data format. Our assumption was that the number of depth views can be smaller than the number of texture views. Hence, the proposed UP-MVD format as a subset of MVD data format was introduced, where the number of texture views differs from the number of depth views, e.g. two texture views are accompanied with only one depth view. The proposed data format succeeded to outperform the conventional MVD data format on average by 0.9% to 6.95% of dBR when changing the view separation from 50% to 100% of the coded baseline, respectively. Moreover, it was confirmed that the proposed UP-MVD enables up to 20% view separation adjustment while outperforming the anchor MVD format in all test sequences. Increasing the camera separation adjustment to 50%, still five out of seven sequences encoded with the proposed UP-MVD format outperformed the MVD anchor bitstreams in rate-distortion performance. As a future trend to further accomplish this research, the useof UP-MVD in multiview video

TABLE III. PERFORMANCE OF UP-MVD FORMAT WITH PROPOSED **SCHEME 1** AGAINST ANCHOR

	Total (Coded PSNR)		Stereo pair (Base + 0.9)		Stereo pair (Base + 0.8)		Stereo pair (Base + 0.5)	
	dBR, %	dPSNR, dB	dBR, %	dPSNR, dB	dBR, %	dPSNR, dB	dBR, %	dPSNR, dB
Poznan Hall2	-5.90	0.22	-3.15	0.10	3.55	-0.14	32.59	-1.22
Poznan Street	-4.97	0.16	-5.37	0.18	-2.70	0.09	14.88	-0.49
Undo Dancer	-2.28	0.08	-4.05	0.15	-0.55	0.02	24.09	-0.74
Ghost Town Fly	-3.18	0.13	-5.22	0.20	-4.45	0.17	4.96	-0.19
Kendo	-14.10	0.79	-5.13	0.22	-0.98	-0.01	19.75	-1.05
Balloons	-9.06	0.50	-4.76	0.23	0.17	-0.05	20.52	-1.09
Newspaper	-9.14	0.42	-0.38	-0.01	12.22	-0.53	55.54	-1.80
Average	**-6.95**	**0.33**	**-4.01**	**0.15**	**1.04**	**-0.07**	**24.62**	**-0.94**

TABLE IV. PERFORMANCE OF UP-MVD FORMAT WITH PROPOSED **SCHEME 2** AGAINST ANCHOR

	Total (Coded PSNR)		Stereo pair (Base + 0.9)		Stereo pair (Base + 0.8)		Stereo pair (Base + 0.5)	
	dBR, %	dPSNR, dB	dBR, %	dPSNR, dB	dBR, %	dPSNR, dB	dBR, %	dPSNR, dB
Poznan Hall2	-5.90	0.22	-5.51	0.18	-3.59	0.12	-16.07	0.60
Poznan Street	-4.97	0.16	-4.65	0.15	-4.18	0.14	-2.39	0.08
Undo Dancer	-2.28	0.08	-2.75	0.10	0.34	-0.01	11.08	-0.36
Ghost Town Fly	-3.18	0.13	-3.19	0.12	-2.96	0.11	-1.50	0.06
Kendo	-14.10	0.79	-11.58	0.59	-10.48	0.52	-7.26	0.33
Balloons	-9.06	0.50	-7.77	0.41	-6.58	0.34	-3.25	0.14
Newspaper	-9.14	0.42	-2.62	0.09	0.62	-0.06	13.09	-0.58
Average	**-6.95**	**0.33**	**-5.44**	**0.23**	**-3.83**	**0.16**	**-0.90**	**0.04**

with three or more views may be considered.

Acknowledgment

The authors would like to thank Professor M. Domañski and his co-authors for providing Poznan sequences and their Camera Parameters.

References

[1] T. Shibata, J. Kim, D. M. Hoffman, and M. S. Banks, "The zone of comfort: predicting visual discomfort with stereo displays," Journal of Vision, vol. 11, no. 8, July 2011

[2] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services," Jan. 2012

[3] A. Smolic, K. Müller, P. Merkle, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, R. Tanger, P. Kauff, T. Wiegand, T. Balogh, Z. Megyesi, and A. Barsi, "Multi-view video plus depth (MVD) format for advanced 3D video systems," Joint Video Team, document JVT-W100, Apr. 2007

[4] C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV," Proc. of SPIE stereoscopic displays and virtual reality systems XI, pp. 93–104, Jan. 2004

[5] "Call for proposals on 3D video coding technology," Moving Picture Experts Group, document N12036, Mar. 2011. Available: http://mpeg.chiariglione.org/working_documents/explorations/3dav/3dvcfp.zip

[6] http://mpeg.chiariglione.org/working_documents/explorations/3dav/3d-test-report.zip

[7] T. Suzuki, M. M. Hannuksela, Y. Chen, S. Hattori, and G. J. Sullivan (ed.), "MVC extension for inclusion of depth maps draft text 4," Joint Collaborative Team on 3D Video Coding Extension Development, document JCT3V-A1001, July 2012

[8] M. M. Hannuksela, Y. Chen, and T. Suzuki (ed.), "3D-AVC draft text 3," Joint Collaborative Team on 3D Video Coding Extension Development, document JCT3V-A1002, Sep. 2012

[9] D. Rusanovskyy and M. M. Hannuksela, "Description of 3D video coding technology proposal by Nokia," Moving Picture Experts Group, document M22552, Nov. 2011

[10] C. Vazquez, W. J. Tam and F. Speranza, "Sterescopic Imaging: Filling Disoccluded Areas in Depth Image-Based Rendering," Proc. Of SPIE, Vol. 6392, 2006.

[11] P. Aflaki, M. M. Hannuksela, D. Rusanovskyy, and M. Gabbouj, "Non-Linear Depth Map Resampling for Depth-Enhanced 3D Video Coding, " IEEE Signal Processing Letters, Vol. 20, issue 1, pp. 87-90, Jan. 2013

[12] M. O. Wildeboer, T. Yendo, M. Panahpour Tehrani, T. Fujii, and M. Tanimoto, "Color based depth upsampling for depth compression," in Proc. Picture Coding Symposiom, Dec. 2010, pp. 170–173.

[13] K.-J. Oh, S. Yea, A. Vetro, and Y.-S. Ho, B, "Depth reconstruction filter and down/up sampling for depth coding in 3-D video," IEEE Signal Process. Letters, vol. 16, no. 9, pp. 747–750, Sep. 2009.

[14] "Common test conditions for 3DV experimentation," ISO/IEC JTC1/SC29/WG11 MPEG2012/N12560, Feb. 2012.

[15] T. Leyvand, C. Meekhof, W. Yi-Chen Wei; Jian Sun; Baining Guo, "Kinect Identity: Technology and Experience, " Computer, p.94-96, 2011

[16] J. Zhu, L. Wang, R. Yang, J. Davis, Fusion of Time-of-Flight Depth and Stereo for High Accuracy Depth Maps, Proceedings of CVPR, pp. 1-8, 2008.

[17] M. Domañski, T. Grajek, K. Klimaszewski, M. Kurc, O. Stankiewicz, J. Stankowski, and K. Wegner, "Poznan multiview video test sequences and camera parameters," ISO/IEC JTC1/SC29/WG11 MPEG2009/M17050, Oct. 2009.

[18] H.Schwarz, et al., "Description of 3D Video Technology Proposal by Fraunhofer HHI (MVC compatible)," ISO/IEC JTC1/SC29/WG11 MPEG2011/M22569, Nov, 2011.

[19] G. Bjøntegaard, "Calculation of average PSNR differences between RD-Curves," ITU-T SG16 Q.6 document VCEG-M33, April 2001.

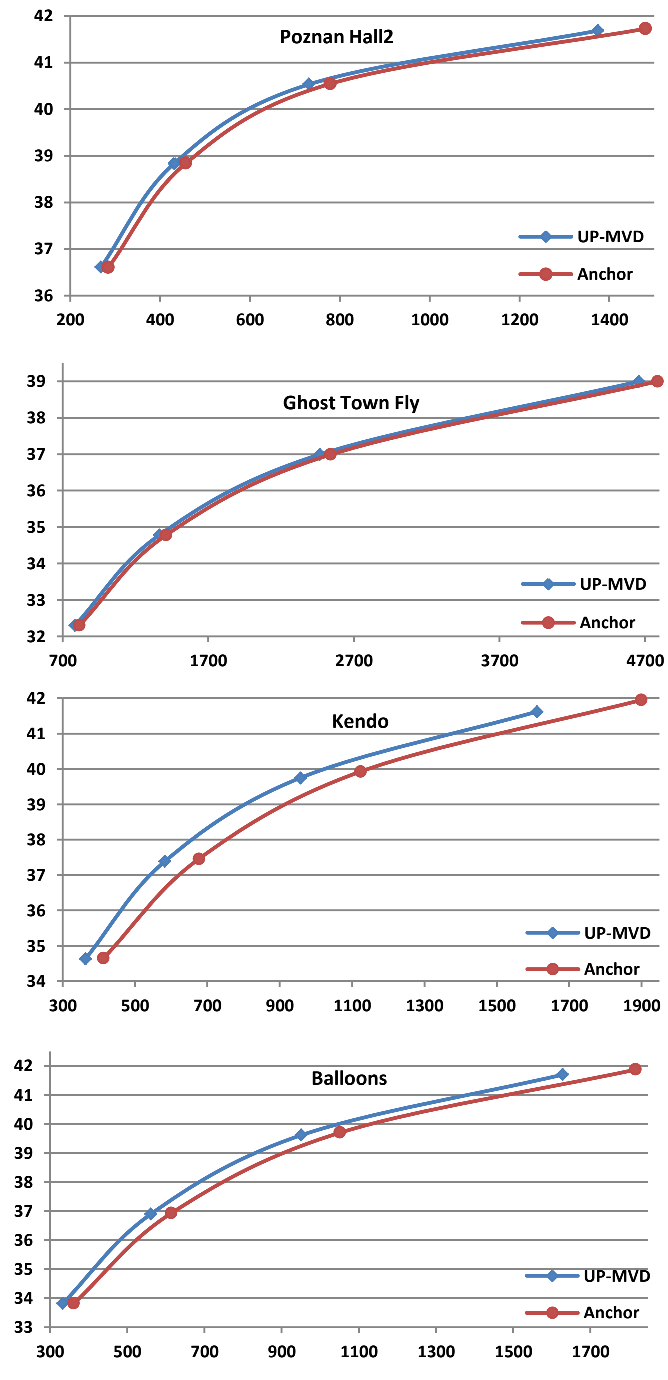

Figure 7. Rate distortion curves of four sequences with largest baseline stereopair (Base + 0.9) for proposed scheme 2