Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 940737, 7 pages
http://dx.doi.org/10.1155/2014/940737

Research Article

Rendering Distortion Estimation Model for 3D High Efficiency Depth Coding

Qiuwen Zhang,[1] Liang Tian,[2] Lixun Huang,[1] Xiaobing Wang,[3] and Haodong Zhu[1]

[1] *College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China*
[2] *College of Computer and Information Engineering, Xinxiang College, Xinxiang 453003, China*
[3] *Technology & Information Center, CPI Henan Power Limited Company, Zhengzhou 450001, China*

Correspondence should be addressed to Qiuwen Zhang; zhangqwen@126.com

Received 23 September 2013; Accepted 23 December 2013; Published 16 January 2014

Academic Editor: Chung-Hao Chen

A depth map represents three-dimensional (3D) scene geometry information and is used for depth image based rendering (DIBR) to synthesize arbitrary virtual views. Since the depth map is only used to synthesize virtual views and is not displayed directly, the depth map needs to be compressed in a certain way that can minimize distortions in the rendered views. In this paper, a modified distortion estimation model is proposed based on view rendering distortion instead of depth map distortion itself and can be applied to the high efficiency video coding (HEVC) rate distortion cost function process for rendering view quality optimization. Experimental results on various 3D video sequences show that the proposed algorithm provides about 31% BD-rate savings in comparison with HEVC simulcast and 1.3 dB BD-PSNR coding gain for the rendered view.

1. Introduction

3D video has gained increasing interest recently. It provides viewers with the illusion of 3D depth perception. The typical 3D video is represented using the multiview video plus depth (MVD) format [1, 2], in which few captured texture videos as well as associated depth maps are used. The depth maps provide per-pixel with depth corresponding to the texture video that can be used to render arbitrary virtual views by using depth image based rendering (DIBR) [3, 4]. For such depth enhanced 3D formats, high efficiency 3D video coding solutions are currently being developed in joint collaborative team on 3D video coding extension development (JCT-3V).

Since depth enhanced 3D video MVD representation causes huge amount of data to be stored or transmitted, it is essential to develop efficient 3D video coding techniques. The most straightforward approach to compress 3D video is using conventional video compression algorithms. The next generation video compression standard HEVC developed by Joint video team (JVT) is joined by both ISO/IEC motion picture experts (MPEG) and ITU-T video coding experts group (VCEG). It provides about 50% bit rate reduction as compared to H.264/AVC achieving the same subjective video quality [5]. Therefore even simulcast HEVC compression of multiview video is more efficient than multiview video coding (MVC). For 3D video coding, a simple extension is to apply two HEVC codecs: one for all texture videos and the other for all depth maps. However, compared to conventional 2D video images, depth maps have very different characteristics. One of the major differences is that the depth maps are only used to render virtual views but not directly used to display, so depth map coding errors cause distortions in synthesized virtual views. That is to say, in conventional texture video coding, we try to improve coding performance of video data; however, in depth maps coding, we focus more on better rendering quality rather than on depth quality. Thus, use of existing HEVC codecs to compress depth maps will introduce distortions into the novel virtual views.

To solve this problem, several approaches have been proposed to enhance the depth coding performance of synthesized view quality. Kim et al. [6] analyzed the geometry error in synthesized view due to depth map lossy coding, Oh et al. [7] introduced a novel distortion function to measure block distortion for synthesized view, and De Silva and Fernando [8] optimized intramode selection for depth map coding to

minimize rendering distortions only for a single viewpoint. Although these methods have better performance than the MVC depth coding methods, they are not compatible with the HEVC standards. In order to meet the compatibility of the HEVC standard, depth maps coding algorithms have moved interest on reducing compression artifacts that may exist in depth maps which are encoded by HEVC. A new distortion metric [9] base on HEVC was used for depth map coding to replace the conventional distortion function for 3D video coding. However, the method could not completely reflect the virtual view rendering process, so that the depth coding performances are not satisfactory.

In order to optimally solve the rendering distortion model problem in MVD coding with high accuracy, a novel distortion model is proposed to precisely estimate distortion in view rendering in this paper. Compared with the previous work for view rendering distortion model, the proposed distortion model highly improves the coding efficiency with carefully considering the depth error sensitivity and occlusion handling with low complexity. The modified distortion model gives more optimal decisions for tree block rate distortion optimization with regard to rendering view quality, based on HEVC technology. Experimental results are given to demonstrate the higher performance of the proposed new rendering distortion estimation algorithm.

2. Proposed Rendering Distortion Estimation

In this section, we derive a relationship between coding errors in the depth map and geometry errors in view rendering and propose a distortion estimation model for view synthesis caused by depth map compression.

Depth image based rendering (DIBR) is the process of synthesizing virtual views of a scene from reference color images and associated per-pixel depth information:

$$\begin{aligned}&\left[x', y', z'\right]^T \\ &\quad = A_v \cdot R_v^{-1} \cdot \left\{R_r \cdot A_r^{-1} \cdot \left[x, y, 1\right]^T \cdot Z_r\left(x_r, y_r\right) + T_r - T_v\right\},\end{aligned} \tag{1}$$

where A_r, R_r, and T_r, respectively, represent the reference camera parameters of the intrinsic matrices, rotation matrix, and translation vector, respectively. $Z_r(x, y)$ is the depth value associated with (x_r, y_r), the subscript v refers to the virtual view, and the subscript r indicates the reference view. The corresponding pixel located in the rendered image of the virtual view is $(x_v, y_v) = (x'/z', y'/z')$. According to (1), an arbitrary virtual view can be generated, when the depth value $Z_r(x_r, y_r)$ is known for every pixel in the reference image and the camera parameters are available.

In 3D video applications, efficient compression of texture video and depth map are necessary. Due to the strict limitation of data rate in 3D video broadcast, only the lossy compression of the texture video and depth map can meet the bandwidth requirement. During HEVC coding, texture video and depth map values are subject to coding errors, such that their reconstructed values differ from the original. While texture video errors only change the interpolation value, according to (1), per-pixel depth value determines how much the corresponding color pixel needs to be shifted when the virtual views are rendered. The depth map error will lead to a geometric error in the interpolation, which will cause view synthesis artifacts. Coding errors in depth map cause artifacts in rendering views, as explained in more detail below.

For 3D video systems with a horizontal camera arrangement, the view synthesis can be carried out using displacement of the original camera views towards the new spatial positions in the intermediate views. These shift values are derived from the depth data. The pixel value in depth map v represents depth Z at pixel location (x, y), and we have

$$v\left(x, y\right) = 255 \cdot \frac{1/Z\left(x, y\right) - 1/Z_{\text{far}}}{1/Z_{\text{near}} - 1/Z_{\text{far}}}, \tag{2}$$

where Z_{near} and Z_{far} are the nearest and farthest depth values in the scene, which correspond to values 255 and 0 in the depth map v.

For a horizontal camera arrangement, the depth values in different camera coordinates will be approximately equal to the depth values in the world coordinate Z. Thus the view warping in (1) can be simplified as

$$\begin{aligned}\left[x_v, y_v, 1\right]^T &= \left[\frac{x'}{z'}, \frac{y'}{z'}, \frac{z'}{z'}\right]^T \\ &= A_v \cdot R_v^{-1} \cdot R_r \cdot A_r^{-1} \cdot \left[x_r, y_r, 1\right]^T \\ &\quad + \frac{1}{Z_r\left(x_r, y_r\right)} \cdot A_v \cdot R_v^{-1} \cdot \left\{T_p - T_{p'}\right\}.\end{aligned} \tag{3}$$

In depth map coding, the quantization brings the depth map distortion. To examine the influence of depth map compression on synthesis quality, we approximate the coding effect of depth map by an additive $\Delta Z_r(x_r, y_r)$, which can be represented as

$$\widehat{Z}_r\left(x_r, y_r\right) = Z_r\left(x_r, y_r\right) + \Delta Z_r\left(x_r, y_r\right), \tag{4}$$

where $\widehat{Z}_r(x_r, y_r)$ is the compressed depth map.

Through the 3D warping, the depth distortion further results in warping error in the synthesized view image. The depth error $\Delta Z_r(x_r, y_r)$ causes the projection of the pixel p moving from p' to $\widehat{p'}$ and results in geometry distortion:

$$\begin{aligned}\widehat{p'} &= p' + \Delta p' = \left[x_v + \Delta x_v, y_v + \Delta y_v, 1\right]^T \\ &= \left[\frac{x' + \Delta x'}{z' + \Delta z'}, \frac{y' + \Delta y'}{z' + \Delta z'}, 1\right]^T.\end{aligned} \tag{5}$$

With (4) and (5), the virtual view synthesis after depth coding can be represented as

$$\begin{aligned}&[x_v + \Delta x_v, y_v + \Delta y_v, 1]^T \\ &= \left[\frac{x' + \Delta x'}{z' + \Delta z'}, \frac{y' + \Delta y'}{z' + \Delta z'}, 1\right]^T \\ &= A_v \cdot R_v^{-1} \cdot R_r \cdot A_r^{-1} \cdot [x, y, 1]^T \\ &\quad + \frac{1}{Z_r(x_r, y_r) + \Delta Z_r(x_r, y_r)} \cdot A_v \cdot R_v^{-1} \cdot \{T_r - T_v\}.\end{aligned} \tag{6}$$

The rendering position error can be calculated by subtracting (6) from (3) as

$$\begin{aligned}&[\Delta x_v, \Delta y_v, 1]^T \\ &= \left(\frac{1}{Z_r(x_r, y_r) + \Delta Z_r(x_r, y_r)} - \frac{1}{Z_r(x_r, y_r)}\right) \\ &\quad \cdot A_v \cdot R_v^{-1} \cdot \{T_r - T_v\}.\end{aligned} \tag{7}$$

The per-pixel depth value $Z_r(x_r, y_r)$ can be calculated from $v_r(x_r, y_r)$, using (2):

$$\frac{1}{Z_r(x_r, y_r)} = \frac{v_r(x_r, y_r)}{255} \cdot \left(\frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}}\right) + \frac{1}{Z_{\text{far}}}. \tag{8}$$

The derivative of (9) can be calculated by combining (7) and (8):

$$\begin{aligned}&[\Delta x_v, \Delta y_v, 1]^T \\ &= \frac{\Delta v_r(x_r, y_r)}{255} \cdot \left(\frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}}\right) \cdot A_v \cdot R_v^{-1} \cdot \{T_r - T_v\}.\end{aligned} \tag{9}$$

The linear relationship between the depth map distortion Δv and the rendering position error $\Delta p'$ in the rendered view can be represented as

$$\Delta p'(\Delta x_v, \Delta y_v) = k \cdot \Delta v_r(x_r, y_r) \tag{10}$$

$$k = \frac{1}{255} \cdot \left(\frac{1}{Z_{\text{near}}} - \frac{1}{Z_{\text{far}}}\right) \cdot A_v \cdot R_v^{-1} \cdot \{T_r - T_v\}, \tag{11}$$

where Δx_v is the horizontal error, Δy_v the vertical error, $\Delta v_r(x_r, y_r)$ is the depth map distortion at the reference camera position p, and k is the scale factor determined by the camera parameters and the depth ranges as shown in (11). We set up a horizontal camera arrangement, which can be used for a horizontal arc camera system. Since 3D video uses parallel camera setups, the view synthesis can be carried out only using horizontal displacement of the reference camera views towards the new positions in the intermediate views. The disparity or the relative shift generated by the DIBR algorithm is only in the horizontal direction; hence the calculation in vertical direction can be omitted. Thus, (10) can be simplified when Δy_v equals 0:

$$\Delta p'(\Delta x_v) = k \cdot \Delta v_r(x_r, y_r). \tag{12}$$

For example, if the distance between cameras T_r and T_v is large or the camera captures a near object, $1/Z_{\text{near}}$ becomes large; that is, k will be large, so that the geometry error will increase. This indicates that dense camera setting and farther scene composition are more robust to depth coding distortion in the rendering process.

Since HEVC encoding algorithms operate block-based structure, the calculation of depth map distortion to the view rendering distortions must be block-based as well. Therefore, it needs to be extended to calculate the exact view rendering distortions for a region based distortion estimation, which will be studied in more detail below.

In DIBR, complex textured images without depth discontinuities and object boundaries are very sensitive regions, and images with less textures or depth discontinuities are less sensitive. Impact of geometry error caused by depth error on view synthesis distortion depends on local characteristics of the images. To more precisely estimate the view synthesis distortion caused by the depth compression error, we propose to use the reference texture image that belongs to the same viewpoint as the depth map. Geometry error will have minimal impact on region with less textures. On the other hand, in region with object boundaries and complex texture, small changes in position can lead to significant changes in view synthesis. Thus, we classify a reference video image into several regions according to the local video characteristics. To define the area of supporting for each view synthesis distortion modeling function, we employ a quadtree decomposition to divide the reference video image into blocks of variable size. In each region, the depth values of all pixels are almost the same. The geometry errors of all pixels are almost constant in one region. Therefore, each region of the video image can be approximated by one view synthesis distortion modeling function. Due to the similarity between warping error and motion vector, the spectrum distortion analysis approach proposed in [10] is adopted to calculate the geometry error induced distortion D_{R_k} of each region R_k, which is expressed as

$$D_{R_k} = \left\|\Delta p'\right\|^2 \cdot \psi_{R_k} = \left\|k \cdot \Delta v_r(x_r, y_r)\right\|^2 \cdot \psi_{R_k}, \tag{13}$$

where $\Delta p'$ is the rendering position error calculated in (10). ψ_{R_k} represents the motion sensitivity factor of the region R_k in reference video image, which is computed as

$$\psi_{R_k} = \frac{1}{2 \cdot (2\pi)^2} \cdot \iint_{(-\pi,\pi]} S_x(\omega_1, \omega_2) \cdot \left(\omega_1^2 + \omega_2^2\right) d\omega_1 d\omega_2, \tag{14}$$

where $S_x(\omega_1, \omega_2)$ denotes the energy density of the region R_k in warping reference frame and (ω_1, ω_2) are the two-dimensional frequency vectors. Since the rendering position error is linear to the depth coding error, the synthesis distortion of region R_k can be computed as (13).

Besides, geometry occlusion and disocclusion varying in warping process also cause a significant variance in view rendering distortion. In DIBR, to inpaint the occlusion regions, the disocclusion pixels located near to the occlusion pixels are used for inpainting, which also contributes to the view rendering distortion. Though the holes due to occlusion regions are not very big (when current multiview camera array is set with a very small baseline, the occlusion regions are very small and the occlusion distortion can be tiny), the view rendering distortion caused by occlusion cannot be neglected. For the view rendering distortion in (13) did not consider holes or occlusion regions in warping process, we add the occlusion handling to improve the accuracy of the view rending distortion estimation.

If a pixel will be occluded after warping with its neighboring view image, the depth value of this pixel would not be as important as the pixels in disocclusion regions, and according to this condition we could adjust the proposed view rending distortion estimation model. Then, (13) can be rewritten as follows:

$$D_{R_k} = \begin{cases} (1-\alpha)^2 \cdot \left\| k_L \cdot \Delta v_L\left(x_L, y_L\right)\right\|^2 \cdot \psi_L \\ \quad +\alpha^2 \cdot \left\| k_R \cdot \Delta v_R\left(x_R, y_R\right)\right\|^2 \cdot \psi_R, \\ \qquad \text{if } R_k \text{ is visible in both cameras } L \text{ and } R, \\ \left\| k_L \cdot \Delta v_L\left(x_L, y_L\right)\right\|^2 \cdot \psi_L, \\ \qquad \text{if } R_k \text{ is only visible in camera } L, \\ \left\| k_R \cdot \Delta v_R\left(x_R, y_R\right)\right\|^2 \cdot \psi_R, \\ \qquad \text{if } R_k \text{ is only visible in camera } R, \\ D_o, \\ \qquad \text{otherwise}, \end{cases} \tag{15}$$

$$\alpha = \frac{\left|T_V - T_L\right|}{\left|T_V - T_L\right| + \left|T_V - T_R\right|}, \tag{16}$$

where D_o represents the rendering distortion induced by inpainting on the occlusion regions of the rendered virtual view image. The subscript L refers to the left camera and the subscript R indicates the right camera. α is a weighting coefficient defined in (16). T_L, T_R, and T_V are the translation vectors of the left camera, right camera, and virtual camera, respectively.

In occlusion regions, disocclusion pixels located near to the occlusion pixels are used for inpainting. Because the original obtained pixel is not available, we assume that the distribution of pixel i_o of occlusion regions is D_{i_o}. The occlusion rendering distortion D_{i_o} is estimated by the disocclusion pixels located near to i_o, and the pixel i_o will be filled with value $H(i_o)$, which is determined by the hole filling method employed. Accordingly, D_{i_o} can be calculated as follows:

$$D_{i_o} = \left|V\left(i_o\right) - H\left(i_o\right)\right|, \tag{17}$$

where $V(i_o)$ represents the luma value at pixel i_o in virtual texture image. Consequently, the distortion D_o caused by inpainting on the occlusion regions can be calculated as follows:

$$D_o = \sum_{i_o} \left|V\left(i_o\right) - H\left(i_o\right)\right|. \tag{18}$$

Finally, the overall view rendering distortion D_{R_k} can be estimated by (15) and (18). This model provides a pixel wise approximation to rendering errors, and these errors need to be minimized in HEVC depth coding.

3. Rate Distortion Optimization for HEVC Encoder

To enable rate distortion (RD) optimization using the proposed rendering distortion estimation, the described rendering distortion model is integrated in HEVC depth maps coding. As a 2D representation of the 3D scene surface, depth maps are utilized for rendering virtual views, and they will not be directly displayed. The quality of decoded depth map has only limited practical meaning. Therefore, the impact of HEVC depth coding artifacts needs to be evaluated further with respect to the rendering quality of virtual views. For this, the HEVC distortion computation carried out is replaced with the proposed rendering distortion estimation in all distortion computation steps. HEVC encoding mode decision is taken in a way that minimizes the errors in the image rendered by the depth map; in contrast to minimizing errors in the depth map itself, the computation of RD cost J has been modified to

$$J = D_v + \varphi \cdot \lambda \cdot R_d, \tag{19}$$

where D_v denotes the virtual view synthesis distortion caused by the HEVC depth compression, as provided by the proposed rendering distortion estimation in Section 2, φ is the constant scaling factor, R_d is the rate of the encoding depth map, and λ is the Lagrange multiplier in the HEVC encoder. We apply the new distortion D_v to the RD optimized mode selection process to decide whether the proposed prediction mode is used for current tree block. That is, when the Lagrange cost is calculated in the HEVC encoder, the estimated distortion in the rendered view is used in depth map coding.

4. Experimental Results

In order to verify the effectiveness of the proposed rendering distortion estimation algorithm, we implemented it into 3DV HEVC test model version 3.0. For the experiments, 8 MVD test sequences of various resolutions with different signal characteristics are used. Four of them were in the 1024 × 768 resolution (Kendo [11], Balloons [11], Lovebird1 [12], and Newspaper [13]) with 30 fps. Other 4 test sequences were in HD resolution of 1920 × 1088 (Undo-Dancer [14], GT-Fly [15], Poznan-Street [16], and Poznan-Hall2 [16]) with 25 fps. The detailed information of the test sequences is provided in Table 1. All 8 test sequences were evaluated in the two view cases. Depth maps were encoded using context-based

Table 1: Test sequence information.

Sequence	Resolution	Frame rate	GOP	Views
Kendo	1024 × 768	30	15	1-3-5
Balloons	1024 × 768	30	15	1-3-5
Lovebird1	1024 × 768	30	15	4-6-8
Newspaper	1024 × 768	30	15	2-4-6
Undo_Dancer	1920 × 1088	25	12	1-5-9
GT_Fly	1920 × 1088	25	12	9-5-1
Poznan_Street	1920 × 1088	25	12	5-4-3
Poznan_Hall2	1920 × 1088	25	12	7-6-5

Table 2: Encoder settings for test.

Codec	3D-HTM ver. 3.0
Number of frames	Full length
Interview prediction	P-I-P
Motion search range	64
MaxCU size	64 × 64
Depth QP	25, 30, 35, and 40
Configuration	Random access
View synthesis	VSRS 3.5

adaptive binary arithmetic entropy coding (CABAC) entropy coding and temporal prediction structures with hierarchical B-frames with GOP of 12 for 1024 × 768 resolution test sequences and 15 for 1024 × 768 resolution test sequences. The experimental results have been conducted using the test conditions of the MPEG 3DV standardization [17]. The more detailed encoder setting is provided in Table 2.

The proposed rendering distortion estimation algorithm is compared with the newly adopted 3D HEVC test model (3D-HTM ver.3.0) [18], the HEVC extension to multiple views (MV-HEVC ver.3.0) [18], and the HEVC simulcast (HM ver.6.0) [19] in terms of average PSNR and bit rate savings. The PSNR is calculated for the virtual views between the decoded synthesized views and the synthesized virtual views using uncompressed texture video and depth map. The reconstructed depth map and texture video (texture videos were not encoded) were used as inputs of view synthesis performed by using MPEG view synthesis reference software (VSRS) [20].

The rate distortion performance comparison of the proposed algorithm compared with 3D-HTM, MV-HEVC, and HEVC simulcast algorithm is shown in Figure 1. The horizontal and vertical axes represent the depth map bit rate and the quality of the rendering virtual view, respectively. From Figure 1, we can see that the proposed rendering distortion estimation algorithm is more effective than the distortion estimation algorithm in 3D-HTM, MV-HEVC, and HEVC simulcast.

Table 3 gives the coding performance of the proposed algorithm compared with 3D-HTM, MV-HEVC, and HEVC simulcast algorithm. Bitrate (BDBR) [21] represents the improvement of total bitrates for depth map coding; Bjontegaard Delta PSNR (BD-PSNR) represents the average PSNR gain over all coded virtual views rendering. From Table 3, we can observe that, compared with HEVC simulcast, a maximum BD-rate of 48.71% can be achieved for the depth map of "GT_Fly" and the average BD-rate brought by the proposed method is 31.20%, while the average BD-PSNR increase is 1.337 dB. Compared with MV-HEVC, the proposed algorithm performs better on all the sequences and achieves more than 20.66% coding bitrate saving, with a maximum of 24.5% in "GT_Fly" and a minimum of 14.6% in "Undo_Dancer." Meanwhile, the average PSNR increase for all the test sequences is 0.846 dB. Moreover, the proposed algorithm shows better performance with about 12.90% BD-rate gain and 0.517 dB BD-PSNR increasing than those achieved by 3D-HTM algorithm. Comparing the previous algorithms, the proposed distortion model algorithm fully mimics the warping view rendering process by even considering the depth sensitivity and occlusion process. Furthermore, the proposed RD cost function gives more optimal decision for tree block mode selection with regard to rendering view quality. Experimental results are given to demonstrate the significantly superior performance of the proposed new distortion model algorithm.

5. Conclusion

This paper presented a new distortion estimation model for 3D depth map compression based on the HEVC. The new distortion estimation model provides an exact measure that can be used in tree block based HEVC code. Compared with the previous work for view rendering distortion estimation, the proposed model significantly improves the coding efficiency with carefully considering the depth sensitivity and low complexity occlusion handling. Experimental results demonstrate that the derived view rendering distortion estimation is accurate and the proposed algorithm provides about 31% BD-rate savings in comparison with HEVC simulcast and 1.3 dB BD-PSNR coding gain for the rendered view.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the editors and anonymous reviewers for their valuable comments. The authors would also like to thank Microsoft Research, Nagoya University, Fraunhofer HHI, GIST, and ETRI, Poznan University, and Nokia, for providing their 3D video sequences and their valuable work on 3D video coding. This work was supported in part by the National Natural Science Foundation of China, under Grants nos. 61302118, 61374014, and 61201447 and

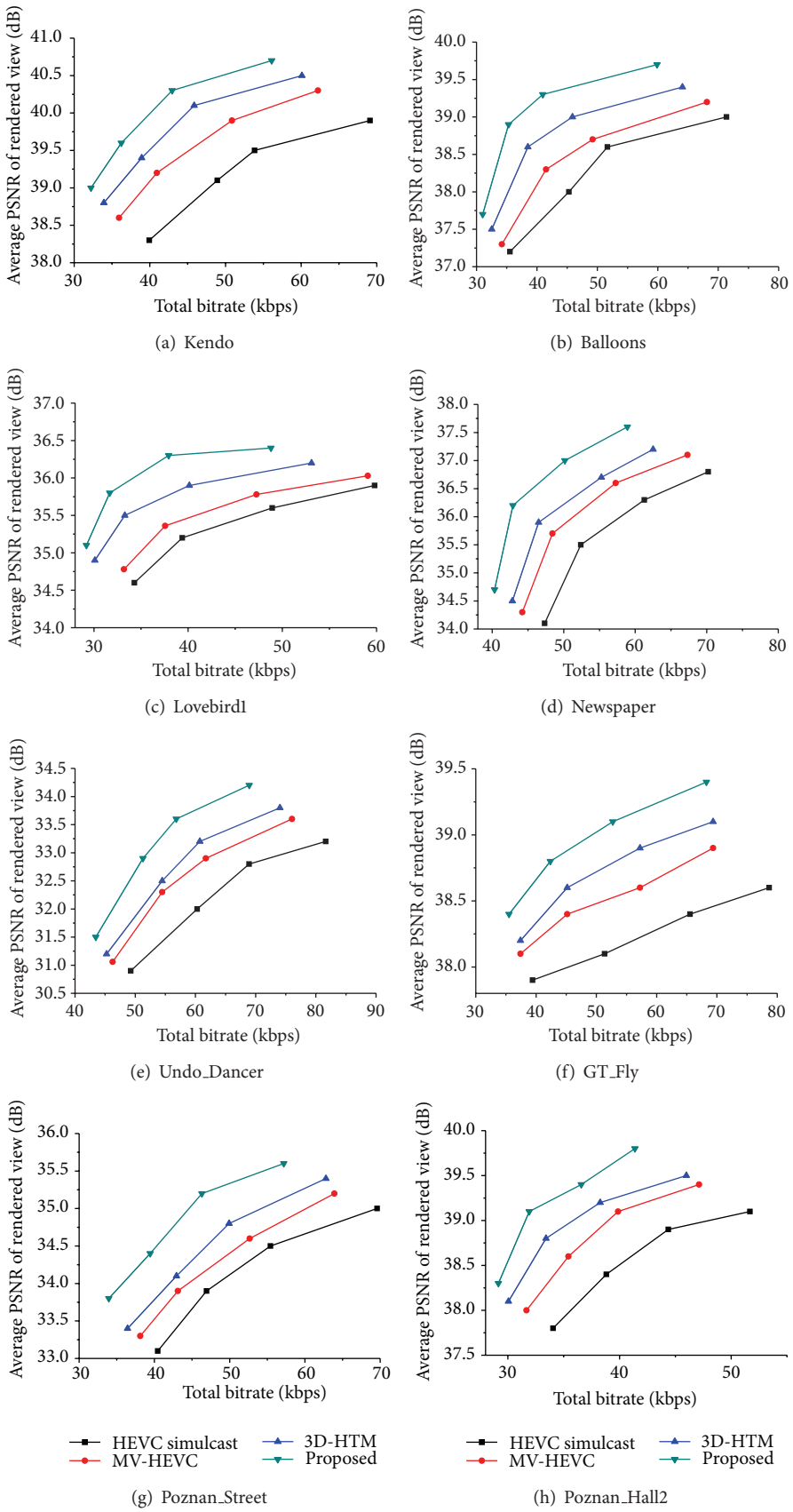

Figure 1: Rate distortion curves comparison.

Table 3: Bjontegaard delta results for proposed algorithm relative to 3D-HTM, MV-HEVC, and HEVC simulcast, comparing the decoded rendered virtual views from original uncompressed texture video and depth map.

Sequence	BD-rate (%)			BD-PSNR (dB)		
	3D-HEVC	MV-HEVC	HEVC simulcast	3D-HEVC	MV-HEVC	HEVC simulcast
Kendo	−11.955	−22.423	−35.135	0.402	0.853	1.537
Balloons	−13.231	−24.406	−31.894	0.428	0.844	1.176
Lovebird1	−8.468	−19.346	−24.397	0.444	0.806	1.052
Newspaper	−11.122	−17.092	−25.150	0.754	1.097	1.646
Undo_Dancer	−11.236	−14.624	−24.499	0.738	1.028	1.818
GT_Fly	−18.345	−24.548	−48.707	0.303	0.447	0.932
Poznan_Street	−15.651	−22.292	−28.287	0.656	0.966	1.306
Poznan_Hall2	−13.222	−20.565	−31.562	0.409	0.730	1.231
Average	−12.904	−20.662	−31.204	0.517	0.846	1.337

in part by the Doctorate Research Funding of Zhengzhou University of Light Industry, under Grant no. 2013BSJJ047.

References

[1] K. Müller, H. Schwarz, D. Marpe, and et al., "3D high-efficiency video coding for multi-view video and depth data," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3366–3378, 2013.

[2] K. Muller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643–656, 2011.

[3] P. Kauff, N. Atzpadin, C. Fehn et al., "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Processing: Image Communication*, vol. 22, no. 2, pp. 217–234, 2007.

[4] C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV," in *11th Stereoscopic Displays and Virtual Reality Systems*, vol. 5291 of *Proceedings of the SPIE*, pp. 93–104, May 2004.

[5] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards— Including high efficiency video coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669–1684, 2012.

[6] W. S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map coding with distortion estimation of rendered view," in *Visual Information Processing and Communication*, vol. 7543 of *Proceeding of the SPIE*, pp. 75430B-1–75430B-10, San Jose, Calif, USA, January 2010.

[7] B. T. Oh, J. Lee, and D.-S. Park, "Depth map coding based on synthesized view distortion function," *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1344–1352, 2011.

[8] D. V. S. X. De Silva and W. A. C. Fernando, "Intra mode selection for depth map coding to minimize rendering distortions in 3D video," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2385–2393, 2009.

[9] G. Tech, H. Schwarz, K. M. üller, and T. Wiegand, "3D video coding using the synthesized view distortion change," in *Proceedings of the Picture Coding Symposium (PCS '12)*, pp. 25–28, Krakow, Poland, May 2012.

[10] A. Secker and D. Taubman, "Highly scalable video compression with scalable motion coding," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1029–1041, 2004.

[11] ISO/IEC JTC1/SC29/WG11, "Moving multiview camera test sequences for MPEG-FTV," Document M16922, ISO, Xian, China, 2009.

[12] ISO/IEC JTC1/SC29/WG11, "3D video test material of outdoor scene," Document M15371, ISO, Archamps, France, 2008.

[13] ISO/IEC JTC1/SC29/WG11, "Multiview video test sequence and camera parameters," Document M15419, ISO, Archamps, France, 2008.

[14] ISO/IEC JTC1/SC29/WG11, "Undo Dancer 3DV sequence for purposes of 3DV standardization," Document M20028, ISO, Geneva, Switzerland, 2011.

[15] ISO/IEC JTC1/SC29/WG11, "Ghost Town Fly 3DV sequence for purposes of 3DV standardization," Document M20027, ISO, Geneva, Switzerland, 2011.

[16] ISO/IEC JTC1/SC29/WG11, "Poznań multiview video test sequences and camera parameters," Document M17050, ISO, Xian, China, 2009.

[17] ISO/IEC JTC1/SC29/WG11, "Common test conditions for 3DV experimentation," Document N12745, ISO, Geneva, Switzerland, 2012.

[18] G. Tech, K. Wegner, Y. Chen, and S. Yea, *3D-HEVC Test Model*, Document JCT3V-C1005, Geneva, Switzerland, 2013.

[19] HM6: High efficiency video coding HEVC reference software version, https://hevc.hhi.fraunhofer.de/svn/svnHEVCSoftware/tags/HM-6.0/.

[20] ISO/IEC JTC1/SC29/WG11, *View Synthesis Algorithm in View Synthesis Reference Software 3.0 (VSRS3.0)*, Document M16090, 2009.

[21] G. Bjontegaard, *Calculation of Average PSNR Differences between RD-Curves*, Document VCEG-M33, 2001.