# Compression-Induced Rendering Distortion Analysis for Texture/Depth Rate Allocation in 3D Video Compression

Yanwei Liu*†, Siwei Ma‡, Qingming Huang*†, Debin Zhao§,
Wen Gao‡, Nan Zhang‡
*Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
†Graduate University of Chinese Academy of Sciences, Beijing, China
‡Peking University, Beijing, China
§Harbin Institute of Technology, Harbin, China
Email:{ywliu,swma,qmhuang,dbzhao,wgao,nzhang}@jdl.ac.cn

## Abstract

*In 3D video applications, the virtual view is generally rendered by the compressed texture and depth. The texture and depth compression with different bit-rate overheads can lead to different virtual view rendering qualities. In this paper, we analyze the compression-induced rendering distortion for the virtual view. Based on the 3D warping principle, we first address how the texture and depth compression affects the virtual view quality, and then derive an upper bound for the compression-induced rendering distortion. The derived distortion bound depends on the compression-induced depth error and texture intensity error. Simulation results demonstrate that the theoretical upper bound is an approximate indication of the rendering quality and can be used to guide sequence-level texture/depth rate allocation for 3D video compression.*

## 1. Introduction

With the development of imaging and 3D displaying technologies, 3D video is an increasing interesting technology for home user living room applications [1]. 3D video provides the depth-enhanced viewing experience so that it is the natural extension to the 2D video. As an emerging new media format, it naturally revolutionizes the visual media services by enabling the 3D-TV and free view-point 3D video [2].

3D video renders the real world by multiview imaging technologies. In order to be compatible with the existed 2D video processing framework, the 2D texture plus depth format is used to represent the 3D video [1]. This kind of representation can render the high quality virtual view with very low cost so that it is suitable for 3D video communication [3]. Though the generation or capture of depth is somewhat complicated, the texture plus depth offers the flexibility to deliver a high-quality 3D viewing experience.

Due to the great amount of data, 3D video with the texture and depth representation is usually to be compressed. Fig. 1 shows a general compression framework for 3D video with texture and depth representation. Note that the virtual view rendering plays a key role for the final visual quality and it can guide the texture/depth rate allocation to guarantee the optimal visual quality. For multiview texture video, a lot of multiview video

1068-0314/09 $25.00 © 2009 IEEE
DOI 10.1109/DCC.2009.27

coding (MVC) algorithms are proposed to exploit the inter-view redundancy between the multiview videos [4, 5]. For depth map, it can be separately encoded with conventional coding techniques, such as H.264/AVC, or jointly coded with MVC [6], and it also can be compressed by some new methods with considering the special characteristics of depth map, such as the method by exploiting depth smooth properties [7].

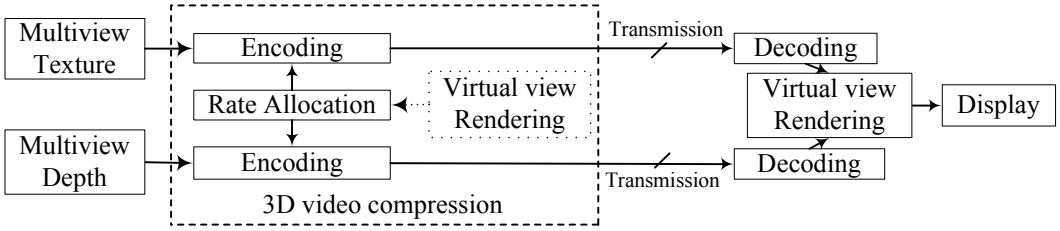



Fig. 1 The compression framework of the texture/depth based 3D video

For depth-image-based rendering, Nguyen et al. [8] present the theoretical analysis of the rendering error. The compression of texture and depth can cause the rendering error so that it has a great effect on the rendering quality [9]. However, how the compression of texture and depth affects the rendering quality has not been quantitative analyzed in the literature before. In this paper, we investigate the depth and texture compression effects on the virtual view rendering quality and give a theoretical analysis of the compression-induced rendering distortion. Based on the mathematical derivation, we provide a theoretical upper bound for the rendering distortion.

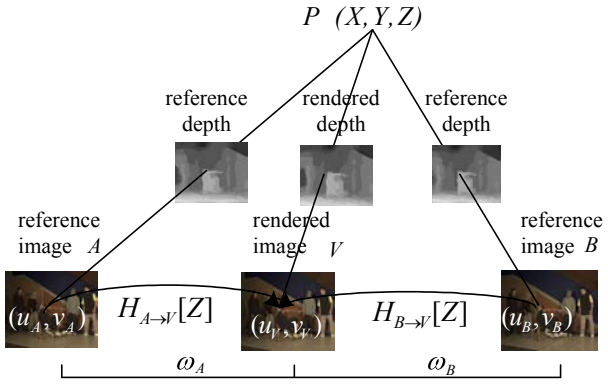



Fig. 2 Depth-image-based view rendering

## 2. Virtual view rendering

### 2.1. Rendering algorithm

Depth-image-based view rendering is usually performed as 3D warping [10]. The virtual view rendering at the middle of two captured views is shown in Fig. 2. According to the accurate camera parameters, the homography matrices from the adjacent source view to

the virtual view can be obtained at different depth values and further the pixels in virtual view image can be warped from those of the adjacent views. The homography matrix is usually computed by matching the correspondence points between the two view images integrated with the pinhole camera model [11].

In Fig. 2, $\mathbf{H}_{A\to V}[z]$and $\mathbf{H}_{B\to V}[z]$ are homography matrices at depth $z$ from view *A* and view *B* to the virtual view, respectively. The relation among $(u_V, v_V)$, $(u_A, v_A)$ and$(u_B, v_B)$ is described by

$$(u_V, v_V, 1) = \mathbf{H}_{A\to V}[z](u_A, v_A, 1)^T = \mathbf{H}_{B\to V}[z](u_B, v_B, 1)^T. \tag{1}$$

In consideration of the occlusion effect, the virtual view synthesis can be expressed as

$$I_V(u_V, v_V) = \begin{cases} w_A I_A(u_A, v_A) + w_B I_B(u_B, v_B), & \text{if } (u_V, v_V) \text{ is visible in view } A \text{ and } B \\ I_A(u_A, v_A), & \text{if } (u_V, v_V) \text{ is only visible in view } A \\ I_B(u_B, v_B), & \text{if } (u_V, v_V) \text{ is only visible in view } B \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $I_V(u_V, v_V)$ , $I_A(u_A, v_A)$ and $I_B(u_B, v_B)$ are the pixel intensity values of the matching points in different views, and that $w_A$ and $w_B$ are distance-dependent blending weights with $w_A + w_B = 1$. Because of the occlusion and pixel mapping uncertainty, some pixels in the virtual view have no matching points in the references view *A* and *B*, and they will be inpainted by the adjacent pixels which have been warped from the source reference views. In the pixel mapping, the mapped pixel sometimes will not locate at an integer position, and it will be rounded to the nearest integer position.

## 2.2. The effect of compression on virtual view rendering

In 3D video applications, efficient compression of texture and depth is necessary. Due to the strict limitation of data rate in 3D video broadcast, only the lossy compression of the texture and depth can meet the bandwidth requirement. Moreover, the bit rate of depth is generally required to be less than twenty percent of the texture bit rate. Therefore, the quantization in compression introduces the great loss of the depth information. Since the depth is a kind of range data [12], the depth loss can cause the warping error in the rendered view image.

Fig. 3 shows the relationship between depth loss and warping error. The left is the source reference view and the right is the virtual view to be synthesized. $C_A$ and $C_V$ are the project centers for view *A* and *V*, respectively. The pixel $(u_A, v_A)$ in view *A* corresponds to the 3D world point *P* with depth *z*, and it is re-projected to the pixel $(u_V, v_V)$ in the virtual view according to $(u_V, v_V, 1)^T = \mathbf{H}[z]\mathbf{m}$, where $\mathbf{m} = (u_A, v_A, 1)^T$ and $\mathbf{H}[z]$ is the homography matrix at depth $z$ from the left view to the virtual view. Due to the quantization of depth map, $P$ loses $\delta z$ and changes into $P'$. $P'$ is re-projected to $(u'_V, v'_V)$ by $(u'_V, v'_V, 1)^T = \mathbf{H}[z - \delta z]\mathbf{m}$. Then the warping error $\delta\mathbf{n}$ is computed by $(\delta\mathbf{n}, 1)^T = (\mathbf{H}[z] - \mathbf{H}[z - \delta z])\mathbf{m}$. For each pixel *i* in the rendered view image, the warping error due to depth loss is expressed as $\delta\mathbf{n}_i = (\delta x_i, \delta y_i)^T$ with the horizontal error $\delta x_i$ and the vertical error $\delta y_i$. The camera setup in Fig. 3 is parallel. When the camera setup is convergent, the relation between depth loss and the warping error is the same as that in Fig. 3.

Besides the pixel warping error introduced by depth quantization, the texture compression also brings the intensity loss which contributes to the rendering distortion. Let $I_A(u_A, v_A)$ denote the original pixel intensity, and $\hat{I}_A(u_A, v_A)$ be the reconstructed pixel intensity. Then,

$$I_A(u_A, v_A) = \hat{I}_A(u_A, v_A) + e_A, \tag{3}$$

where $e_A$ is compression-introduced intensity loss. Therefore, the pixel $(u_A, v_A)$ with intensity value $\hat{I}_A(u_A, v_A)$, which corresponds to the actual surface point *P*, will be projected to the $(u'_V, v'_V)$.

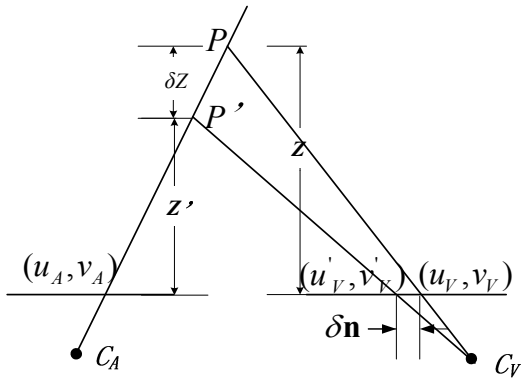



Fig. 3 Relationship between the warping error and depth loss (parallel camera setup)

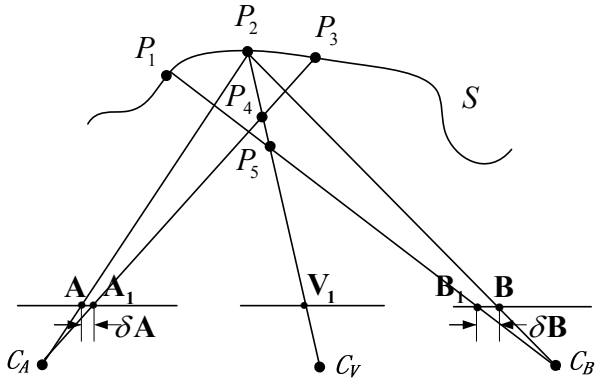



Fig. 4 The pixel mapping with compressed texture and depth (parallel camera setup)

Actually, the depth loss leads to certain confusion of the pixel mapping. Fig. 4 gives the actual warping process with the compressed texture and depth. The point $P_1$, $P_2$ and $P_3$ are the actual points in surface *S*. Both the pixel at position **A** of view *A* and the pixel at position **B** of view *B* correspond to the 3D world point $P_2$. The point $P_2$ is projected to the pixel position $\mathbf{V}_1$ in the virtual view *V* and so the pixel at position $\mathbf{V}_1$ is interpolated by the pixels at **A** and **B** when the depth does not lose any information.

In the actual warping, the 3D world point $P_3$, which corresponds to pixel at $\mathbf{A}_1$ in view *A*, loses some depth and then changes into $P_4$. The $P_4$ will project to the position $\mathbf{V}_1$ in the virtual view. Likewise, the 3D world point $P_1$, which corresponds to pixel at $\mathbf{B}_1$

in view *B*, loses some depth and then changes into $P_5$. The point $P_5$ will also project to the position $\mathbf{V}_1$ in the virtual view. Therefore, in the actual warping, the pixel at position $\mathbf{V}_1$ is interpolated by the pixel at $\mathbf{A}_1$ and the pixel at $\mathbf{B}_1$. In Fig. 4, $\mathbf{A}_1 = \mathbf{A} + \delta\mathbf{A}$ and $\mathbf{B}_1 = \mathbf{B} + \delta\mathbf{B}$. Hence, the interpolation is expressed as

$$\hat{I}_V(\mathbf{V}_1) = \begin{cases} w_A\hat{I}_A(\mathbf{A}+\delta\mathbf{A}) + w_B\hat{I}_B(\mathbf{B}+\delta\mathbf{B}), & \text{if } V_1 \text{ is visible in both view } A \text{ and } B \\ \hat{I}_A(\mathbf{A}+\delta\mathbf{A}), & \text{if } V_1 \text{ is only visible in view } A \\ \hat{I}_B(\mathbf{B},+\delta\mathbf{B}) & \text{if } V_1 \text{ is only visible in view } B \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where $\mathbf{A}$ and $\mathbf{B}$ are position vectors with $\mathbf{A} = (x_A, y_A)$ and $\mathbf{B} = (x_B, y_B)$. In (4), $\delta\mathbf{A}$ and $\delta\mathbf{B}$ are warping error vectors with $\delta\mathbf{A} = (\delta x_A, \delta y_A)$ and $\delta\mathbf{B} = (\delta x_B, \delta y_B)$.

## 3. Theoretical upper bound of compression-induced rendering distortion

According to the actual rendering process, this section provides the derivation of the theoretical upper bound for the compression-induced rendering distortion. For simplicity, we assume that the occlusion can be regarded as a kind of noise in the rendering. In other words, it is assumed that the scene in the virtual view is visible in both view *A* and view *B*. Therefore, the rendering is translated into a linear interpolation problem. Let $\mathbf{I}_A$, $\mathbf{I}_V$ and $\mathbf{I}_B$ denote the image plane for view *A*, view *V* and view *B*, respectively. Accordingly, the images of view *A*, view *V* and view *B* are mathematically described as $f_A(\mathbf{X})|_{\mathbf{X}\in I_A}$, $f_V(\mathbf{X})|_{\mathbf{X}\in I_V}$ and $f_B(\mathbf{X})|_{\mathbf{X}\in I_B}$. Let $f_V(\mathbf{X})$ be the interpolated pixel intensity at position $\mathbf{X}$ using uncompressed texture and depth. Because the rendering makes use of the original texture and depth, it does not introduce any warping error. Then the ideal rendering is described as

$$\begin{aligned} f_V(\mathbf{X}) &= w_A f_A(\mathbf{A}) + w_B f_B(\mathbf{B}) + n_o \\ &= w_A(\hat{f}_A(\mathbf{A}) + e_A) + w_B(\hat{f}_B(\mathbf{B}) + e_B) + n_o, \end{aligned} \tag{5}$$

where $n_o$ denote the noise effects in the rendering with original texture and depth, and $e_A$ and $e_B$ are the intensity errors of texture pixel at $\mathbf{A}$ and $\mathbf{B}$, respectively. The $\hat{f}_A(\mathbf{A})$ and $\hat{f}_B(\mathbf{B})$ in (5) denote the reconstructed pixel intensities at $\mathbf{A}$ and $\mathbf{B}$, respectively.

By comparison, the interpolated pixel $\hat{f}_V(\mathbf{X})$ using compressed texture and depth is mathematically expressed as

$$\hat{f}_V(\mathbf{X}) = w_A\hat{f}_A(\mathbf{A} + \delta\mathbf{A}) + w_B\hat{f}_B(\mathbf{B} + \delta\mathbf{B}) + n_c, \tag{6}$$

where $n_c$ also denotes the noise effects. In the video capturing, the actual surface is assumed to be Lambertian and all non-Lambertian view-dependent effect, such as geometry errors and illumination difference, are modeled as noise.

In order to evaluate the rendering quality, the view rendering distortion is generally to be computed. Since the rendering is mathematically characterized as the linear interpolation problem, there exists a theoretical upper bound for the rendering distortion. We first define several notions and then provide the theoretical bound.

**Definition 1.** *The $L_\infty$ norm of gradient $\nabla f(x,y)$ is defined as*

$$\|\nabla f(x,y)\|_\infty = \sup_{(x,y)\in\Omega}\{\|\nabla f(x,y)\|_2\},$$

*where $\Omega$ denotes the local pixel set which covers $(x,y)$.*

**Definition 2.** *The $i_{th}$ pixel is at position $\mathbf{X}_i$ in the virtual view image. The warped pixel which corresponds to $\mathbf{X}_i$ in view A image is at $\mathbf{A}_i$ , and the warped pixel which corresponds to $\mathbf{X}_i$ in view B image is at $\mathbf{B}_i$. The absolute intensity error for the texture pixel at $\mathbf{A}_i$ is expressed as $|e_{A,i}|$ and the absolute intensity error for the texture pixel at $\mathbf{B}_i$ is expressed as $|e_{B,i}|$. The absolute warping error for pixel at $\mathbf{A}_i$ is expressed as $|\delta\mathbf{A}_i|$ and the absolute warping error for pixel at $\mathbf{B}_i$ is expressed as $|\delta\mathbf{B}_i|$.*

**Theorem 1.** *The virtual view rendering distortion $D_V$ is bounded by,*

$$\begin{aligned} D_V \;\leq\; & \frac{1}{M\cdot N}\sum_{i=1}^{M\cdot N}(max\{|e_{A,i}|,|e_{B,i}|\} \\ & +max\{|\delta\mathbf{A}_i|\cdot\|\nabla\hat{f}_{A,i}\|_\infty, |\delta\mathbf{B}_i|\cdot\|\nabla\hat{f}_{B,i}\|_\infty\}+|n_{c,i}-n_{o,i}|)^2, \end{aligned} \tag{7}$$

*where $M\cdot N$ is the virtual view resolution and the distortion is characterized by mean-squared error (MSE).*

***Proof.*** According to the two-value Taylor expansion, we obtain

$$\hat{f}_A(\mathbf{A}+\delta\mathbf{A}) = \hat{f}_A(\mathbf{A}) + \delta\mathbf{A}\cdot\nabla\hat{f}_A(\xi_A) \text{ and}$$
$$\hat{f}_B(\mathbf{B}+\delta\mathbf{B}) = \hat{f}_B(\mathbf{B}) + \delta\mathbf{B}\cdot\nabla\hat{f}_B(\xi_B)$$

From (5) and (6), we can further obtain

$$\begin{aligned} \hat{f}_V(\mathbf{X}) - f_V(\mathbf{X}) =& w_A\hat{f}_A(\mathbf{A}+\delta\mathbf{A}) + w_B\hat{f}_B(\mathbf{B}+\delta\mathbf{B}) + n_c \\ & -(w_A(\hat{f}_A(\mathbf{A})+e_A) + w_B(\hat{f}_B(\mathbf{B})+e_B) + n_o) \\ =& w_A(\hat{f}_A(\mathbf{A}+\delta\mathbf{A}) - \hat{f}_A(\mathbf{A})) + w_B(\hat{f}_B(\mathbf{B}+\delta\mathbf{B}) - \hat{f}_B(\mathbf{B})) \\ & -(w_Ae_A + w_Be_B) + (n_c - n_o) \\ =& w_A\delta\mathbf{A}\cdot\nabla\hat{f}_A(\xi_A) + w_B\delta\mathbf{B}\cdot\nabla\hat{f}_B(\xi_B) - (w_Ae_A + w_Be_B) + (n_c - n_o) \end{aligned}$$

Taking the absolute values of the two sides, and because $w_A + w_B = 1$, we have

$$\begin{aligned} |\hat{f}_V(\mathbf{X}) - f_V(\mathbf{X})| \;=\; & |w_A\delta\mathbf{A}\cdot\nabla\hat{f}_A(\xi_A) + w_B\delta\mathbf{B}\cdot\nabla\hat{f}_B(\xi_B) \\ & -(w_Ae_A + w_Be_B) + (n_c - n_o)| \\ \leq\; & |w_A\delta\mathbf{A}\cdot\nabla\hat{f}_A(\xi_A) + w_B\delta\mathbf{B}\cdot\nabla\hat{f}_B(\xi_B)| \\ & +|(w_Ae_A + w_Be_B)| + |(n_c - n_o)| \\ \leq\; & max\{|\delta\mathbf{A}|\cdot\|\nabla\hat{f}_A\|_\infty, |\delta\mathbf{B}|\cdot\|\nabla\hat{f}_B\|_\infty\} \\ & +|(w_Ae_A + w_Be_B)| + |(n_c - n_o)| \\ \leq\; & max\{|\delta\mathbf{A}|\cdot\|\nabla\hat{f}_A\|_\infty, |\delta\mathbf{B}|\cdot\|\nabla\hat{f}_B\|_\infty\} \\ & +max|e_A|, |e_B| + |(n_c - n_o)| \end{aligned}$$

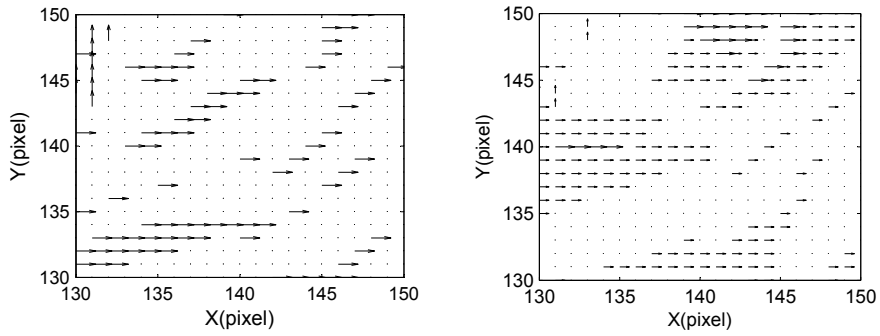
Therefore,

$$\begin{aligned} D_V &= \frac{1}{M \cdot N} \sum_{i=1}^{M \cdot N} |\hat{f}_V(\mathbf{X}_i) - f_V(\mathbf{X}_i)|^2 \\ &\leq \frac{1}{M \cdot N} \sum_{i=1}^{M \cdot N} (max\{|e_{A,i}|, |e_{B,i}|\} \\ &\quad + max\{|\delta \mathbf{A}_i| \cdot \|\nabla \hat{f}_{A,i}\|_\infty, |\delta \mathbf{B}_i| \cdot \|\nabla \hat{f}_{B,i}\|_\infty\} + |n_{c,i} - n_{o,i}|)^2. \end{aligned}$$

Apparently, Theorem 1 indicates that the compression-introduced rendering distortion bound is related with compression-induced depth error, compression-induced texture intensity error and the gradients of the source reference images.

## 4. Simulation results

In this section, we evaluate the effect of compression on rendering quality and verify the distortion bound. We performed several experiments with Breakdancer and Ballet (1024×768) sequences. These test sequences are captured at 15 fps with eight cameras arranged along a horizontal arc [13]. The provided depth map sequences are computed from stereo method. The experiments are performed to synthesize view1 using view0 and view2. The texture and depth data are coded by MVC software JMVM6.0 and the virtual view is synthesized by the compressed texture and depth at different bit-rate points.



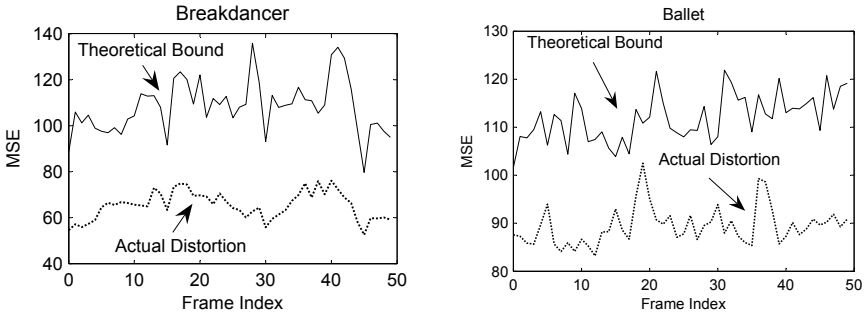

(a) Depth rate at 320kbps (b) Depth rate at 158kbps

Fig. 5 The warping error vector map for the first frame of Breakdancer sequence

In Theorem 1, the depth loss introduced warping error vector $\delta\mathbf{A}$ and $\delta\mathbf{B}$ can be obtained by twice warping using the original depth and compressed depth. Fig. 5 shows the distribution of warping error vectors from the left source reference image to the virtual image. The warpings are performed at different depth rates. Note that most of the warping error vectors are not very large and more warping error vectors are appeared at the reduced depth rate. In Theorem 1, The warping error induced distortion also depends on the $L_\infty$ norm $\|\nabla f_i\|_\infty$. The $L_\infty$ norm $\|\nabla f_i\|_\infty$ is obtained via Definition 1 in a local rectangle area defined by the warping error vector.

In the rendering, there are two main sources for generating noise. One is disocclusion processing which inpaints the occluded holes in the rendered view. The second is the pixel-rounding effect of the rendered image. In the warping, the pixel in the source

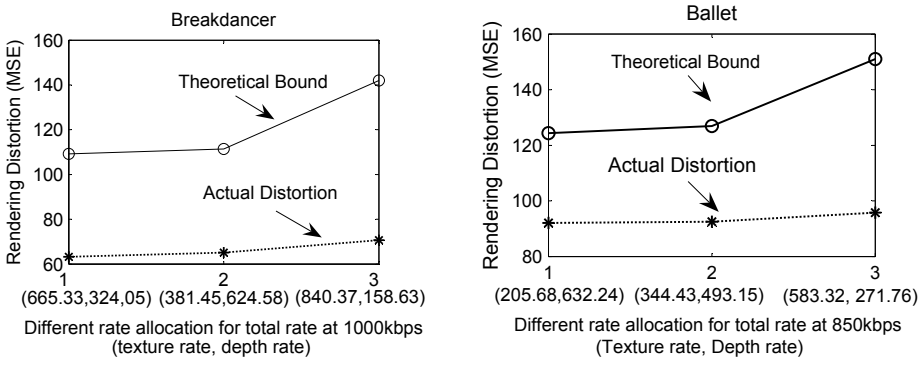
reference view may not always be at an integer position, and when the mapped pixel position is not an integer position, it will be round to the nearest integer position. The two parts of noise are both related with the texture video of source reference view. Since the rendering distortion is sensitive with the overall level of noise, but not with the actual shape of noise, we model the noise with the same shape with intensity loss of the source reference texture video. The noise term in Theorem 1 is related by equation $|n_{c,i} - n_{o,i}| = \alpha \cdot max\{|e_{A,i}|, |e_{B,i}|\}$, where $\alpha$ is the ratio of noise to reference video intensity loss.




(a) (b)

Fig. 6 The comparison between the theoretical bound and the actual distortion

Fig. 6 shows the comparison between the theoretical bound and the actual distortion. The actual distortion is MSE between the rendered view1 and the original view1. In Fig. 6, the texture is compressed at 194kbps and the depth is coded at 330kbps. It can be observed that the theoretical bound takes the similar trend as the actual distortion. In the simulation, $\alpha$ for different sequence is differently assigned. In Fig. 6, we set $\alpha = 0.02$ for Breakdancer and $\alpha = 0.4$ for Ballet. Because there exist different self-occlusions in different scenes, the disocclusion processing noise needs to be differently characterized. At different depth rate, the pixel-rounding noise has different effect on the rendering distortion. In the simulation, we set different $\alpha$ value at different depth rate to describe the pixel-rounding noise effect.

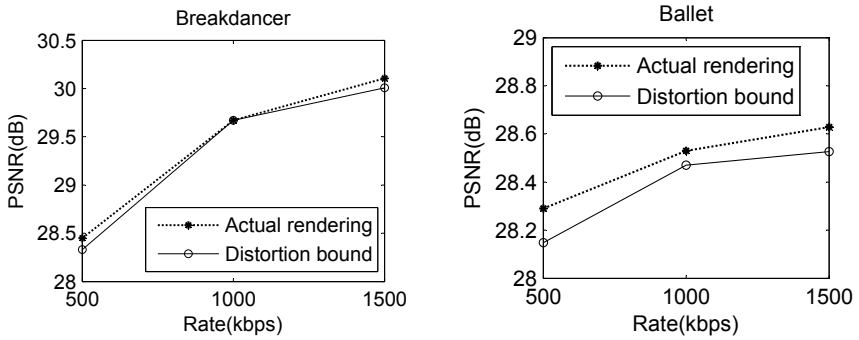



(a) (b)

Fig. 7 The comparison between the actual distortion and theoretical bound for different texture/depth rate allocation

In 3D video compression, the good rendering quality is necessary for the end user. Under the channel rate constraint, the proper rate allocation between the texture and depth can guarantee the optimal rendering quality. Fig. 7 shows the comparison between the actual distortion and theoretical distortion bound for different texture/depth rate combination under the same total bit-rate. The comparison is performed at the sequence-level. It can be observed that, under the total rate constraint, the different rate allocation between texture and depth has different distortion performance. In the figure, the actual rendering distortions for the three texture/depth rate combinations are sorted in ascending order. The curve of theoretical bound keeps the same monotonous ascending shape as actual distortion curve. It shows that the theoretical bound can differentiate the optimal rendering performance from the different texture/depth rate combinations.

Fig. 8 shows the sequence-level rate allocation performance at different rate points. It can be seen that the distortion bound can obtain the similar rate allocation performance with the actual rendering method. In some rate points, the distortion bound does not accurately reflect the actual rendering distortion due to the noise effect so that it sometimes only finds the suboptimal texture/depth rate combination.




(a) (b)

Fig. 8 Rate allocation performance at different rate points

The computation of theoretical distortion bound only involves warping and image gradient operation. Its computational complexity is less than that of the complete rendering. When the total rate is at 1000kbps, It can save the computational complexity about 27% and 22% for Breakdancer and Ballet, respectively. The complete rendering method can accurately measure the rendered view quality through computing the MSE between the rendered view and the original view. However, in practical 3D video applications, the original captured view of the virtual view position is not always available to be used as reference for MSE computing. The proposed theoretical distortion bound does not require the existence of original view so that it is more suitable for the practical 3D video applications.

## 5. Conclusion

This paper first analyzes how the depth and texture compression affect the rendering quality and then proposes a theoretical upper bound for compression-introduced rendering distortion. The distortion bound depends on the compression-induced depth error and

texture intensity error. Simulation results suggest that the derived distortion bound is an approximate indication for evaluating the virtual view rendering quality and it can be used to guide the sequence-level rate allocation between texture and depth for 3D video compression.

The proposed distortion bound mainly reflects the compression effect on the rendering. Because there are many noise factors which affect the total rendering distortion, the accuracy of the proposed distortion bound needs to be further improved to increase the rate allocation reliability.

## Acknowledgment

The work was supported in part by National Natural Science Foundation Research Program of China under Grant No.60672088 and No.60736043, Major State Basic Research Development Program of China (973 Program, 2009CB320905), Chinese youth scientific funds (No.60803069), and Co-building Program of Beijing Municipal Education Commission. The authors would like to thank Microsoft research for providing the video sequences of Ballet and Breakdancer.

## References

[1] A. Vetro, S. Yea, A. Smolic, Towards a 3D video format for auto-stereoscopic displays, Proceedings of the SPIE: Applications of Digital Image Processing XXXI, San Diego, CA, USA, 2008.

[2] Introduction to 3D Video, ISO/IEC JTC1/SC29/WG11, Doc.N9784, Archamps, France, May 2008.

[3] 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.

[4] P. Merkle, A. Smolic, K. Mller, and T. Wiegand, Efficient Prediction Structures for Multiview Video Coding, IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Multi-view Video Coding and 3DTV, 17(11) (2007) 1461-1473.

[5] W. Yang, Y. Lu, F. Wu, J. Cai, K. N. Ngan, and S. Li, 4-D Wavelet-Based Multiview Video Coding, IEEE Trans. on Circuits and Systems for Video Technology, 16(11):1385-1396, November 2006.

[6] P. Merkle, A. Smolic, K. Mller, and T. Wiegand, Multi-view Video plus Depth Representation and Coding, In: Proceedings of ICIP 2007, pp. 201-204.

[7] Y. Morvan, D. Farin and P. H.N. de With, Depth-Image Compression based on an R-D Optimized Quadtree Decomposition for the Transmission of Multiview Images, In: Proceedings of ICIP 2007, pp. 105-108.

[8] H. T. Nguyen and M. N. Do, Error analysis for image-based rendering with depth information, in Proc. IEEE Int. Conf. on Image Processing, October 2006, pp. 381-384.

[9] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Mueller, P. H.N. de With, T. Wiegand, The Effect of Depth Compression on Multiview Rendering Quality, in IEEE Conference on 3D-TV, vol. p. 245-248, May 2008, Istanbul, Turkey.

[10] M. Tanimoto, T. Fujii and K. Suzuki, Experiment of view synthesis using multi-view depth, ISO/IEC JTC1/SC29/WG11, Doc. M14889, October 2007.

[11] R. Hartley, A. Zisserman, Multiple view geometry in Computer Vision, Cambridge University Press, UK, 2003.

[12] H. G. Lalgudi, M. W. Marcellin, A. Bilgin, and M. S. Nadar, Lifting-based View Compensated Compression of Volume Rendered Images for Efficient Remote Visualization, in Proceedings of 2008 Data Compression Conference.

[13] http://research.microsoft.com/IVM/3DVideoDownload/.