AN ERROR ROBUST DISTORTION MODEL FOR DEPTH MAP CODING IN ERROR PRONE NETWORK

Min Gao, Xiaopeng Fan, Tao Zhang, Debin Zhao and Wen Gao

Department of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
{mgao, fxp, tzhang, dbzhao}@hit.edu.cn, wgao@jdl.ac.cn

ABSTRACT

Error robustness becomes an important issue when the compressed depth map is transmitted over error prone network. There have been some algorithms to improve the error robustness of color video in the past years. However, their extensions to depth map are not reasonable due to the difference between depth map and color video. In this paper, a novel error robust distortion model is proposed to enhance the error robustness of depth map, in which the end-to-end distortion of virtual view is estimated. More specifically, the proposed distortion model recursively computes the expected decoded depth for each pixel by considering the channel condition and error concealment method, and then the expected decoded depth is used to estimate the end-to-end distortion of virtual view. Experimental results show that the proposed distortion model consistently outperforms the conventional distortion model and outperforms the random intra updating algorithm in most cases.

Index Terms— Depth map coding, error robustness, distortion model, end-to-end distortion

1. INTRODUCTION

In 3DTV system, the depth map is commonly compressed by using conventional video compressing standards, such as H.264/AVC [1] and HEVC [2]; and the compressed depth map is commonly transmitted through packet loss networks.

While the compressed depth map is transmitted through error prone networks, error robustness to packet loss is a crucial requirement since the conventional video coding standard is based on the hybrid coding framework, in which the prediction loop propagates errors and causes substantial distortion of received depth map.

To deal with the error propagation problem, several error resilience coding techniques have been proposed. Insertion of intra coded blocks is a widely used and standard compliant type, sine the intra coded blocks can switch off the inter-frame prediction loop and the reproduced blocks are no more dependent on the previously decoded frames. However, the intra coded blocks generally require more bits. Therefore, a careful selection of intra coded block in terms of rate and distortion is necessary.

Some algorithms have been proposed to insert intra coded blocks for color video. The early algorithms have been proposed to insert intra coded blocks randomly [3] or periodically [4]. To improve the coding performance further, several rate distortion optimized methods [5]-[9] have been proposed to optimally choose the number and position of the intra coded blocks. In [5], a generalized framework for joint rate control and error control is proposed, in which the intra refresh rate is determined prior to the coding of a frame. Some end-to-end distortion models have been proposed in [6]-[9], in which the end-to-end distortion of the color video was estimated in different manners. In [6], a recursive optimal per-pixel estimate (ROPE) algorithm has been proposed to estimate the end-to-end distortion at pixel level by recursively computing the first and second moments of the reconstructed pixel value. However, it is assumed that the integer motion compensated prediction is used rather than sub-pixel prediction in ROPE to avoid the intensive computing and storage required in pixel averaging operation. To deal with this problem, two model-based cross-correlation approximation methods were proposed in [7], in which the cross-correlation of two pixels were approximated as a function of their available first and second marginal moments. The H.264/AVC test model adopted an error-resilient rate distortion model proposed in [8], in which the expected end-to-end distortion is estimated in a manner of independently simulating K copies of channel behaviors at the encoder. The end-to-end distortion model in [9] takes the overall distortion as the sum of several separate distortion items.

The above methods can enhance error robustness of color video to packet loss. However, their extension to depth map is not reasonable, because it is not depth map, but virtual view to be displayed to users. In error free environment, some virtual view distortion models have been proposed to improve the coding efficiency of source coding for depth map. In [12], a distortion model taking the global video characteristic into account was proposed to estimate the distortion of virtual view. In [13], Kim et al. extends the method by considering the local characteristic

of video. To make the distortion model more accurate, the disparity rounding problem was considered in [11]. In [14], an alternative model has been proposed by mimicking the view rendering process. However, these distortion models are not suitable for depth map coding in error-prone environments, since the channel condition and error concealment method are not considered.

In this paper, we proposed an error robust distortion model for depth map coding in error-prone environments. In the proposed model, the end-to-end distortion of virtual view should be estimated. Toward this goal, we firstly estimate the expected decoded depth for each pixel, and then use the distortion model in [11] to estimate the end-to-end distortion of virtual view.

The rest of the paper is organized as follows. In Section 2, the depth image based rendering process (DIBR) and the virtual view distortion model in [11] are described. Section 3 described the proposed error-robust distortion model for depth map coding. The experimental results were presented in Section 4. Section 5 concludes the paper.

2. VIRTUAL VIEW DISTORTION MODEL FOR DEPTH MAP CODING

In this section, we firstly describe the rendering process of the virtual view and then introduce the virtual view distortion model for depth map coding in [11].

2.1. Rendering process of virtual view

The virtual view can be synthesized by using the reference video and corresponding depth map through depth image based rendering (DIBR) process. Simply speaking, DIBR process projects the pixel value at reference video to the corresponding position at the virtual view.

In the following description, we assume that (x_r, y_r) denotes pixel position in reference video and (x_v, y_v) denotes the corresponding pixel position in virtual view. The subscript r indicates reference view, and the subscript v indicates the virtual view. Therefore, the DIBR process can be described by the following two steps.

First, position (x_r, y_r) is mapped into the real world coordinate (u, v, w) with the depth map by

$$[u,v,w]^T = R_r \cdot A_r^{-1} \cdot [x_r, y_r, 1] \cdot Z_r(x_r, y_r) + T_r \quad (1)$$

where R_r is rotation matrix, A_r is intrinsic camera parameter matrix, $Z_r(x_r, y_r)$ is real depth value at position (x_r, y_r) and T_r is translation vector. The real depth value $Z_r(x_r, y_r)$ can be calculated according to depth map $D_r(x_r, y_r)$ throuth

$$\frac{1}{Z_r(x_r,y_r)} = \frac{D_r(x_r,y_r)}{255}\left(\frac{1}{Z_{near}} - \frac{1}{Z_{far}}\right) + \frac{1}{Z_{far}} \quad (2)$$

where Z_{near} and Z_{far} represent the nearest and farthest depth value in the scene, respectively. If the depth value is represented by 8 bits, they correspond to 0 and 255, respectively.

Second, the real world coordinate (u, v, w) is projected into the position (x', y', z') in the virtual view.

$$[x', y', z']^T = A_v \cdot R_v^{-1} \cdot \{[u,v,w]^T - T_v\} \quad (3)$$

where A_v, R_v and T_v are the intrinsic camera parameter matrix, rotation matrix and translation vector of virtual view, respectively.

The corresponding pixel position (x_v, y_v) in virtual view can be calculated by

$$[x_v, y_v] = \left[\frac{x'}{z'}, \frac{y'}{z'}\right] \quad (4)$$

However, for multi-view video system with parallel camera arrangement, the virtual view can be simply synthesized by the following equation.

$$\begin{aligned}[x_v, y_v] &= \left[\frac{x_r \cdot Z_r(x_r,y_r) + f \cdot \Delta_x}{Z_r(x_r,y_r)}, y_r\right] \\ &= \left[x_r + \frac{f \cdot \Delta_x}{Z_r(x_r,y_r)}, y_r\right]\end{aligned} \quad (5)$$

where f denotes focal length, and Δ_x is the distance between two cameras.

The term $\frac{f \cdot \Delta_x}{Z_r(x_r,y_r)}$ in equation (5) is also called disparity in horizontal direction and denoted as d. In most cases, the disparity is not integer and it needs to be rounded to integer in rendering process. If the rendering process enables a pixel to be rounded to 1/M sub-pixel in the virtual view, the disparity λ-rounding function [10] can be written as

$$round(d) = \frac{\lceil (d-\lambda) \cdot M \rceil}{M} \quad (6)$$

where λ is usually set $0.5/M$.

2.2. Virtual view distortion model

Assume the original depth map is D(x, y) and its reconstruction value is $\hat{D}$(x, y) after encoding. Their associated disparities calculated according to equation (5) are d and $\hat{d}$, respectively. The rendering position error Δx caused by depth distortion can be calculated by

$$\begin{aligned}\Delta x &= (x + round(\hat{d})) - (x + round(d)) \\ &= round(\hat{d}) - round(d)\end{aligned} \quad (7)$$

According to [11], the distortion $D_v(x)$ in virtual view caused by Δx can be approximated as

$$D_v(x) = 2 \cdot \delta^2 \cdot (1 - \rho^{|\Delta x|}) \quad (8)$$

where δ^2 is the variance of the video block collocated with depth map block and ρ represents the video block correlation when translated by one pixel.

Since Δx is not very large, the distortion of virtual view can be approximated by

$$D_v(x) = 2 \cdot \delta^2 \cdot (1 - \rho) \cdot |\Delta x| \quad (9)$$

Therefore the distortion of virtual view caused by the current depth map block is

$$D = \sum_{x=1}^{N} 2 \cdot \delta^2 \cdot (1 - \rho) \cdot |\Delta x| = 2 \cdot \delta^2 \cdot (1 - \rho) \cdot \sum_{x=1}^{N} |\Delta x| \quad (10)$$

where *N* is the size of the current depth map block.

3. THE PROPOSED ERROR-ROBUST DISTORTION MODEL FOR DEPTH MAP

From the above description, we can see that the distortion of virtual view can be estimated if the reconstructed depth map is obtained. However, the decoded depth map is unavailable at the encoder due to the random channel behavior when the depth map is transmitted over the error-prone network. So the decoded depth map should be estimated at the encoder. In the following, we firstly analyze the reconstruction process of depth by considering two cases depending on whether the pixel belongs to intra coded block or inter coded block, and then propose a method to estimate the reconstructed depth map at decoder.

3.1. Analysis of reconstruction process of depth at decoder

We assume that D_n^i be the original depth of pixel *i* in frame *n*, and $\hat{D}_n^i$ denote its reconstruction at encoder. Let $\tilde{D}_n^i$ be its reconstructed depth at decoder, which is potentially different from the reconstruction depth at encoder $\hat{D}_n^i$ due to the packet loss. Assume that *p* is the packet loss rate.

3.1.1. Reconstruction process of depth in intra coded block

If the packet containing intra coded block to which pixel i belongs is correctly received, then $\tilde{D}_n^i = \hat{D}_n^i$ and the probability of this event is *1- p*. If the packet is lost, the depth of pixel *k* in the previous frame *n-1* is used to estimate the pixel *i* in the current frame, then $\tilde{D}_n^i = \tilde{D}_{n-1}^k$ and the probability of this event is *p*. Therefore, the reconstructed depth for intra coded pixel at decoder can be represented as

$$\tilde{D}_n^i = \begin{cases} \hat{D}_n^i & w.p.\ 1-p \\ \tilde{D}_{n-1}^k & w.p. \quad p \end{cases} \quad (11)$$

3.1.2. Reconstruction process of depth in inter coded block

Let us assume that pixel *i* in the current frame *n* is predicted from pixel *j* in the reference frame *ref* and the encoder prediction is $\hat{D}_{ref}^j$. We denote the encoder quantization residue as $\hat{e}_n^i$, which means that the reconstruction depth of this pixel at encoder, $\hat{D}_n^i$, is equal to $\hat{D}_{ref}^j + \hat{e}_n^i$.

If the packet consisting of quantization residue and motion vector is correctly received, the decoder reconstruction of depth of pixel *i* in frame *n* is given by $\tilde{D}_n^i = \hat{e}_n^i + \tilde{D}_{ref}^j$, which is potentially different from the encoder reconstruction $\hat{D}_n^i$ due to the difference between $\tilde{D}_{ref}^j$ and $\hat{D}_{ref}^j$. The probability of this event is 1-*p*. If the packet is lost, the reconstruction depth of pixel *k* in the previous frame *n-1* at decoder is used to estimate the reconstruction of pixel *i* in the current frame, that is $\tilde{D}_n^i = \tilde{D}_{n-1}^k$ and the probability of this event is *p*. So the reconstructed depth for inter coded pixel at decoder can be represented as follows.

$$\tilde{D}_n^i = \begin{cases} \tilde{D}_{ref}^j + \hat{e}_n^i & w.p.\ 1-p \\ \tilde{D}_{n-1}^k & w.p. \quad p \end{cases} \quad (12)$$

3.2. Estimation of reconstructed depth at decoder

According to the above description, we can use the expectation of the decoded depth map to be the estimation of the reconstructed depth map at the decoder.

For the pixel in the intra coded block, the expectation of the reconstructed depth at decoder can be calculated by the following equation.

$$E\{\tilde{D}_n^i\}(I) = (1-p) \cdot \hat{D}_n^i + p \cdot E\{\tilde{D}_{n-1}^k\} \quad (13)$$

For the pixel in the inter coded block, the expectation of the reconstructed depth at decoder can be calculated by the following equation.

$$E\{\tilde{D}_n^i\}(P) = (1-p) \cdot (E\{\tilde{D}_{ref}^j\} + \hat{e}_n^i) + p \cdot E\{\tilde{D}_{n-1}^k\} \quad (14)$$

where *I* and *P* indicate the current pixel is intra coded or inter coded, respectively.

Note that the expectation of the decoded depth of the first frame can be obtained directly without error propagation because it is coded as intra frame. Hence, the expectation of the decoded depth of the following frames can be recursively calculated frame by frame.

3.3. Estimation of end-to-end distortion of virtual view

According to the above description, we can calculate the expectation of the decoded depth for each pixel. Given the expectation of the decoded depth $E\{\widetilde{D}(x,y)\}$ and the original depth $D(x,y)$, the corresponding rounding disparities $d_{E\{\widetilde{D}\}}$ and d_D can be computed according to equations (5) and (6). Hence, the rendering position error is obtained by $\Delta x = d_{E\{\widetilde{D}\}} - d_D$. Therefore, the end-to-end distortion for virtual view can be estimated for the current depth map block according to equation (10).

4. EXPERIMENT AND RESULTS

To evaluate the accuracy of the proposed distortion model, it is integrated into the RDO framework to choose optimally the number and position of intra coded blocks. The competing methods consist of random intra updating algorithm ("RU") and conventional RDO algorithm ("Con-RD"). In random intra updating algorithm, given packet loss rate p, a fraction p of macro-blocks (MB) in each frame is coded as intra. In conventional RDO algorithm, the packet loss rate is ignored. In the two methods, the distortion model of virtual view described in Section 2 is adopted. In the disparity rounding part of the distortion model in Section 2, we set $M = 1$, $\lambda = 0.5$ in this experiment.

The JM 15.1 H.264/AVC codec was employed. In the experiment, we employed CABAC for entropy coding; a single reference frame is used in motion estimation and sub-pixel motion estimation is disabled. The de-blocking filter is enabled and inter pixels are not used for intra macro-block prediction. Each row of macro-blocks in each frame composes a slice and is packed in a separate packet. The original rate control algorithm from JM codec is used and the temporal-replacement is employed for error concealment.

In the experiment, the depth maps from different views are compressed independently under the same bit rate, and the texture videos are not compressed. The first frame of the sequence is coded as I-frame and all the rest frames are coded as P-frame. The reconstructed depth map and the texture videos are used to synthesize the virtual view by view synthesis reference software (VSRS) version 3.5. The virtual view synthesized using original depth map and texture video is used as reference to compute PSNR. At the decoder, 200 randomly generated packet loss patterns are used, and the average luminance PSNR of the virtual view is calculated to evaluate the performance.

As a test set, we selected four multi-view sequences *Balloons*, *Newspaper*, *Lovebird1* and *BookArrival*. The resolution of these sequences is 1024x768. The detailed test setting is shown in Table 1.

At the encoder, four bit streams are generated in terms of each algorithm; at the decoder, these bit streams are decoded after simulating packet loss rate 5%, 10%, 15% and 20%, respectively. It is assumed that the packet containing the parameter set and packets containing the first frame are correctly received at the decoder. The performance comparison for different sequences is presented in Table 2.

Table 1. Test setting for each sequence

	Balloons	Newspaper	Lovebird1	BookArrival
Bit rate(*kbps*)	688.57	788.96	528.49	518.57
Frames	150	150	200	100
View-number	1-3	4-6	4-6	8-10
Frame rate(*fps*)	30	30	30	16.67

Table 2. Performance comparisons of average PSNR [dB] for virtual view at different packet loss rate

sequence	scheme	PSNR of different packet loss rate			
		5%	10%	15%	20%
Balloons	Proposed	43.83	43.13	42.55	42.07
	RU	43.66	42.83	42.15	41.60
	Con-RD	43.25	41.94	40.87	39.96
Newspaper	Proposed	41.64	40.97	40.39	39.84
	RU	41.32	40.18	39.31	38.47
	Con-RD	40.75	39.35	38.29	37.42
Lovebird1	Proposed	42.71	42.23	41.82	41.39
	RU	43.32	41.34	40.14	38.65
	Con-RD	42.27	41.41	40.68	40.04
BookArrival	Proposed	41.32	40.66	40.00	39.40
	RU	41.48	40.59	39.87	39.29
	Con-RD	40.93	39.72	38.72	37.87

From the Table 2, we can see that the proposed method outperforms the conventional RDO algorithm in all cases, and outperforms the random intra updating algorithm in most cases.

In particular, it outperforms the conventional RDO algorithm about 2.4dB and the random intra updating algorithm 1.4dB for *Newspaper* sequence at the packet loss rate 20%. Such gain in PSNR also means the visual quality improvement. The visual quality comparison of the frame 138 in the virtual view between the conventional RDO algorithm, random intra updating algorithm and the proposed method is shown in Fig.2. It can be seen that the proposed method achieves the best visual quality.

At the packet loss rate 5%, the random intra updating algorithm achieves better performance than the proposed method for *Lovebird1* and *BookArrival*. That is because the random intra updating algorithm inserts more intra coded blocks in the beginning frames of the sequence and the quality of these frames is higher than that of the frames encoded by the proposed method. However, there are more

bits required to encode these intra coded blocks in the random intra updating algorithm, so the bits left to encode the following frames is fewer than that left in the proposed method. Therefore the quality of the following frames in the proposed method will be higher than that of the frames in the random intra updating algorithm. This is can be shown in Fig.1, in which we illustrate the PSNR of the frames from 176 to 199 in the virtual view of *Lovebird1* that is rendered using the depth map encoded by the proposed method and the random intra updating algorithm at the packet loss rate 5%.

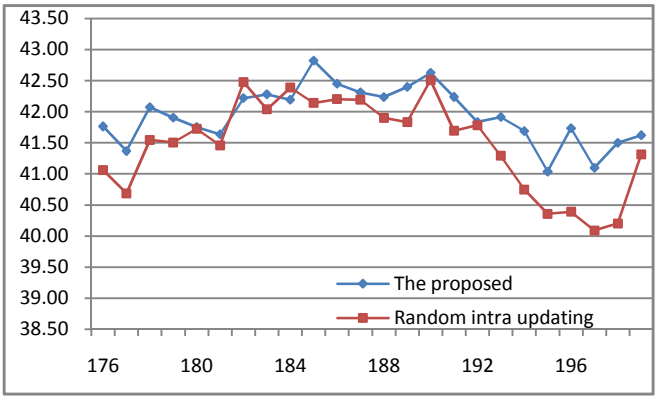

Fig.1. PSNR of frames from 176 to 199 in the virtual view for *Lovebird1* rendered using depth map encoded by two methods

At packet loss rate 10%, 15% and 20%, the random intra updating algorithm achieves less performance than the convention RDO algorithm for *Lovebird1*. In my opinion, this is caused by the characteristic of the sequence. The depth of the foreground and background are nearly not changed in sequence *Lovebird1*; based on the assumption that the first frame is correctly received, for some regions, it is not necessary to code them as intra, since the depth after error concealment approximates to the decoded one. Therefore, compared with the random intra-updating algorithm, the conventional rate-distortion algorithm can allocate more bits to the regions which have more influence on the quality of virtual view. So the performance of the conventional RDO algorithm is better than that of the random intra updating algorithm.

5. CONCLUSION

This paper proposes an error robust distortion model for depth map coding to enhance the error robustness of depth map coder to packet loss. Different from the existing virtual view distortion functions used in the source coding of depth map in error-free environment, the channel condition and the error concealment algorithm used in the decoder are taken into account. To demonstrate the accuracy of the proposed distortion model, we incorporate the distortion model within the RDO framework to choose the optimal number and position of the intra coded blocks. The experimental results show that the proposed method achieves better performance than the conventional RDO algorithm in all cases and outperforms the random intra updating algorithm in most cases. In the future work, we will analyze this distortion model and improve its performance.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation of China (NSFC) under grants 61272386 and 61100095, and the Program for New Century Excellent Talents in University (NCET) of China (NCET-11-0797), and the Fundamental Research Funds for the Central Universities (Grant No. HIT.BRETIII.201221).

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra,"Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, pp. 560-576, July 2003

[2] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 8," ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-J1003, July 2012.

[3] G. Cote and F. Kossentini, "Optimal intra coding of blocks for robust video communication over the internet," Signal Process.: Image Commun., vol. 15, pp. 25–34, Sep. 1999

[4] Q. F. Zhu and L. Kerofsky, "Joint source coding, transport processing and error concealment for H.323-based packet video," in Proc. SPIE VCIP'99, San Jose, CA, Jan. 1999, vol. 3653, pp. 52–62.

[5] Z. H. He, J. F. Cai, and C. W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," IEEE Transactions on Circuits and Systems for Video Technology., vol. 12, pp.511–523, Jun. 2002

[6] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal intra/inter mode switching for packet loss resilience," IEEE Journal on Selected Areas in Communications, vol. 18, no. 6, pp. 966-76, June 2000.

[7] H. Yang and K. Rose, "Recursive end-to-end distortion estimation with model-based cross-correlation approximation," in Proc. IEEE ICIP'03, Sep. 2003, vol. 3, pp. 469–472.

[8] T. Stockhammer, D. Kontopodis, and T.Wiegand, "Rate-distortion optimization for JVT/H.26L coding in packet loss environment," in Proc. Packet Video Workshop, Pittsburgh, PA, Apr. 2002

[9] Y. Zhang, W. Gao, Y. Lu, Q. Huang and D. Zhao, "Joint Source-Channel Rate-Distortion Optimization for H.264 Video Coding Over Error-Prone Networks,"

IEEE Transactions on Multimedia, vol 9, pp 445-454, April 2007

[10] Y. Zhao, C. Zhu, Z. Chen, L. Yu, “Depth no-synthesis-error model for view synthesis in 3D video,” IEEE Transactions on Image Processing, vol. 2, no. 8, pp.151–172, Aug. 2011.

[11] T. Zhang, X. Fan, D. Zhao and W. Gao, “New Distortion Model for Depth Coding in 3DVC,” in Proc. IEEE VCIP, 2012

[12] W. S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, “Depth map distortion analysis for view rendering and depth coding,” in Proc. IEEE ICIP’09, Nov, 2009, pp 721-724.

[13] W. S. Kim, A. Ortega, P. Lai, D. Tian, C. Gomila, “Depth Map Coding with Distortion Estimation of Rendered View,” in Proc. SPIE VCIP, 2010.

[14] Byung Tae Oh,Jaejoon Lee and Du-Sik Park, "Depth Map Coding Based on Synthesized View Distortion Function," IEEE Journal of Selected Topics in Signal Processing., vol. 5, pp. 1344-1352, Nov. 2011.

(a) Virtual view rendered by original depth map

(b) Virtual view rendered by depth map encoded by Con-RD

(c) Virtual view rendered by depth map encoded by RU

(d) Virtual view rendered by depth map encoded by the proposed method

Fig. 2. Visual comparison for *Newspaper* at packet loss rate 20%