A SCALABLE DEPTH CODING WITH ARC BREAKPOINTS BASED SYNTHESIS IN 3-D VIDEO

Xiaopeng Zhang[1], *Junni Zou*[2], *Xiao Gu*[1], *Hongkai Xiong*[1]

[1]Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
[2]School of Communication and Information Engineering, Shanghai University, Shanghai 200072, China

ABSTRACT

Depth map represents three-dimensional information and is used for depth image-based rendering to support 3-D video applications. Conventional video coding standards may cause serious coding artifacts along the depth discontinuities, which ultimately affect the synthesized view quality. This paper proposes an efficient technique to compress the depth map by encoding a reduced resolution of depth map and using special upsampling method to reconstruct the original depth map. Unlike previous upsampling schemes with the guidance of corresponding color video, the proposed method upsamples the depth map using the breakpoints which are generated at the decoder side with only a few breakpoints. Instead of the edge representation of the depth map, the breakpoints would substantially take less bits and low errors with the regularity of the contour. Experimental results show that the proposed technique significantly reduces bit rate while achieving a better quality of the synthesized view in terms of subjective and objective measures.

Index Terms— depth coding, depth down/upsampling, curve completion, arc breakpoints

1. INTRODUCTION

With recent development of 3-D multimedia technologies, 3-D video has played an emerging role in entertainment industry. Multiview video representations enable such applications, where the real word is captured by multiple synchronized cameras. Since the transmitted data is linearly proportional to the number of views in multi-view video coding [1] , a new 3-D coding framework aims at fewer (2 or 3) views at the encoder side and corresponding view synthesis procedure is used to generate virtual views. The depth map which represents a relative distance between the camera center and objects is utilized to synthesize the virtual views at the receiver side. Since the quality of synthesized views mainly depends on the precision of the transmitted depth map, the depth coding would guarantee the high-quality of synthesized virtual views at a lower bit rate cost. In this sense, the performance is determined by the quality of synthesized view, not the depth map itself.

Unlike the natural image full of complex texture, depth map exists large smooth changes and abrupt signal changes around object boundaries. Reconstruction errors especially around the boundaries in depth map would lead to wrong sample displacement in synthesized views. Thus, to preserve the boundary of depth map is of vital importance for depth map coding. Traditional transformations to decorrelate the data, e.g. discrete cosine transform (DCT) and discrete wavelet transform (DWT), perform well in smooth regions. However, they loose their decorrelation power at the vicinity of discontinuities. Thus, it would cause serious artifacts along the depth discontinuities to code depth map by existing video coding standards (e.g. H.264/AVC), which ultimately affect the synthesized view quality.

Morvan *et al.* have ever proposed platelet-based coding to preserve the depth discontinuities [2], which models depth map by piecewise-linear functions and divides the depth map into blocks of variable size by a quad-tree decomposition where each block is approximated by one polynomial function. It could achieve a higher rendering quality with an inferior depth map to H.264/AVC coding [3]. In 2010, Maitre and Do developed a depth compression algorithm based on shape-adaptive wavelet transform [4]. It makes use of the edge information to prevent wavelet bases from crossing the edges, which leads to small wavelet coefficients along depth edges. Obviously, they are difficult to be extended into video domain for making use of temporal redundancies. An alternative solution for depth coding is to downsample depth map prior to MVC coding [5][6], and recover the original depth map by special upsampling technique after decoding. In this aspect, Ekmekcioglu *et al.* suggested to enhance the temporal and inter-view depth maps by post-processing [7]. In order to meet the compatibility of H264/AVC, extra intra prediction modes were adopted to encode depth maps with the wedgelet and contour partition [8].

This paper proposes an independent depth coding algorithm without the corresponding color video. Since the depth values vary smoothly except object boundaries, we downsample the depth map, transmit sampled key points to the decoder, and complete the missing parts of the edges. Inspired

The work was supported in part by the NSFC, under grants 61221001.

by [9], the completed curves would generate breakpoints between neighboring points. The breakpoints would provide the guidance when upsampling the reduced resolution depth map. Instead of the edge representation of the depth map, the breakpoints would substantially take less bits and low errors with the regularity of the contour. Both subjective and objective experimental results demonstrate that the proposed scheme can significantly improve the quality of rendered image.

The rest of this paper is organized as follows: Section 2 introduces the background of curve completion using the breakpoints. In section 3, the proposed scheme is presented, including the downsampling, the breakpoints guided upsampling, and the edge preservation smoothing. The experiments are evaluated in Section 4, and a conclusion is drawn in Section 5.

2. BACKGROUND OF CURVE COMPLETION

Visual curve completion is a fundamental perceptual issue which fills up the missing parts of the observed object contours. Studies on curve completion usually assume that the completed curve is induced by two oriented line segments at the point of occlusions. The rationality of curve completion is that the curve of the missing part should comply with the trends of the known induced point information. To address the characterization and reconstruction of the shape of the curve between the given two induced line segments, it can be formulated as the following problem:

Curve Completion Problem: Give the position and orientation of two inducers $p_0 = [x_0, y_0, \theta_0]$ and $p_1 = [x_1, y_1, \theta_1]$ in the image plane, find the shape of the "correct perceptual curve that passes between the inducers.

A lot of completion methods have been developed to solve the problem depending on the restricted conditions during completion. These constrained conditions include isotropy, smoothness, and total minimum curvature [10]. The elastica model takes a shape in which its total bending energy is minimal [11], and the Euler spiral model assumes that the curvature changes linearly with the arc length. To some extent, all of them can get a perceptual plausible completed curve. Recently, Ben-Yosef [12] proposed a new completion method in a three dimensional space $T(I) : R^2 \times S^1$, where R^2 represents the image plane I, and S^1 represents the curvature space. It is inspired from the observation that curve completion is regarded as an early visual process which relates with the orientation selective cells in primary visual cortex. It can be formalized as minimizing the following problem:

Tangent Bundle Curve Completion Problem: Given two endpoints $p_0 = [x_0, y_0, \theta_0]$ and $p_1 = [x_1, y_1, \theta_1]$ in space $T(I)$, find the curve $\beta(t) = [x(t), y(t), \theta(t)]$ that minimizes the function

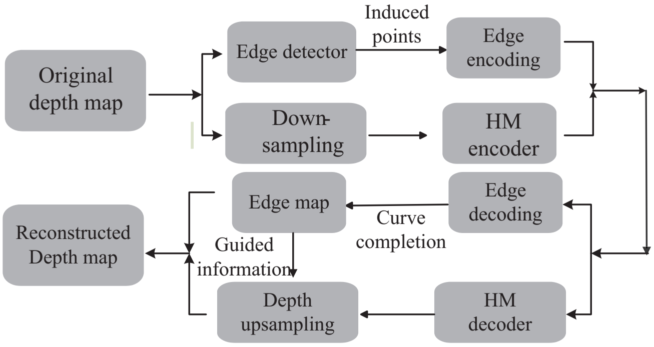


Fig. 1. The proposed scheme framework

$$\begin{aligned} &\zeta = \int_{t_0}^{t_1} \sqrt{\dot{x}^2 + \dot{y}^2 + \hbar^2 \dot{\theta}^2} dt \\ &s.t. \quad \beta(t_0) = p_0, \quad \beta(t_1) = p_1 \qquad (1) \\ &\tan\theta(t) = \frac{\dot{y}(t)}{\dot{x}(t)}, \quad where\; \dot{x}(t) = \frac{dx}{dt}, \; \dot{y}(t) = \frac{dy}{dt} \end{aligned}$$

where $\hbar$ is a weighted factor which balancing the total length in the image plane and the total curvature in the curvature plane. The problem can be converted to an ordinary differential equation and being solved via Euler method. In most scenarios, the completed curve of the missing part is perceptually plausible and better than existing completion models.

3. PROPOSED DEPTH CODING METHODS

The original depth is downsampled to obtain the reduced resolution depth map. Also, the edge map is extracted from the original depth map and some breakpoints would be sampled as key points at the encoder. The breakpoints shall be transmitted to the decoder for generating the entire breakpoint set. It would favor the upsampling process to generate the original resolution depth map followed by an edge preserving filter. The procedure is shown in Fig.1.

3.1. depth downsampling

Traditional downsampling filter consisting of low-pass filter and interpolation filter, will smooth the sharp edges in the depth map. In this paper, we employ a simple direct downsampling filter as

$$D_{down}(x, y) = D_{org}(\frac{x}{s}, \frac{x}{s}) \qquad (2)$$

where s is the downsampling factor and is chosen as 2 here. The direct downsampling filter reserves the sth pixels both in horizontal and vertical directions to get the reduced depth map.

3.2. edge detection and coding

From the edge map, the explicit breakpoints are extracted and transmitted to the decoder. We modify the popular Canny detector for depth edge detection, where the edge locations are attained by [4]. It defines edges at fractional locations between pixels. i.e., edges around pixel (i, j) occur at $(i, j \pm 0.5)$ and $(i \pm 0.5, j)$. First, we get depth map derivatives δ^h and δ^v as

$$\begin{aligned} \delta^h_{s+\frac{1}{2},t} &= x_{s+1,t} - x_{s,t} \\ \delta^v_{s,t+\frac{1}{2}} &= x_{s,t+1} - x_{s,t} \end{aligned} \tag{3}$$

The edge elements whose gradient magnitudes are not local maximal are discarded. For simplify, the non-maximality is defined by comparing the derivative of each edge element with those of its two neighbors which have the same directions:

$$\begin{aligned} \delta^{h2}_{s+\frac{1}{2},t} &\geq max(\delta^h_{s+\frac{1}{2},t}\delta^h_{s-\frac{1}{2},t}, \delta^h_{s+\frac{1}{2},t}\delta^h_{s+\frac{3}{2},t}) \\ \delta^{v2}_{s,t+\frac{1}{2}} &\geq max(\delta^v_{s,t+\frac{1}{2}}\delta^v_{s,t-\frac{1}{2}}, \delta^v_{s,t+\frac{1}{2}}\delta^v_{s,t+\frac{3}{2}}) \end{aligned} \tag{4}$$

The edge elements which satisfy Eq. (4) are set to 1, otherwise 0. Thus, we get the horizontal and vertical edge maps $e^h_{s+\frac{1}{2},t}$ and $e^v_{s,t+\frac{1}{2}}$. The edge elements are encoded using a differential Freeman chain code because it is accurate for preserving the edge information.

It is worth mentioning that partial edge elements would be sampled as explicit breakpoints and generate other breakpoints via curve completion at the decoder side. Through scanning the half-integer locations of the edge map, the edge chains are extracted. The rationality is derived from two observations:

• The edge elements describe the contour of the natural objects, so should exhibit regular shapes.

• The completed contours can tolerate an error of $(-0.5, 0.5)$ at each edge element position, thus reduces the false generated arc breakpoints.

For simplify, two explicit breakpoints are extracted to complete fixed-length chains (every two explicit breakpoints are triggered to complete 50 edge pixels here). Noted that the values of θ_0 and θ_1 of Eq. (1) in curve completion are determined by the neighboring points of p_0 and p_1. For example, finding its neighboring two points of p_0, fitting a circular arc to the three points, then θ_0 is chosen as the slant angle of the tangent line of the circle at p_0. The explicit breakpoints will be encoded by arithmetic coding. The Freeman chain code and the generated breakpoints via curve completion can be illustrated in Fig. 2.

3.3. Depth upsampling using the completed curve

Once the missing contours between the two neighboring explicit breakpoints are completed via the constrained condition Eq. (1), the completed curves would help generate breakpoints among pixels. If the completed curves intersect with the arc which connects two neighboring integer pixels, then there is an arc breakpoint between the two pixels. The generated breakpoints can favor guiding the upsampling procedure. In turn, the reconstructed downsampled depth map is independently upsampled in the horizontal, vertical, and diagonal direction. Take the horizontal upsampling process as an example, the unknown entries are interpolated via the following principle:

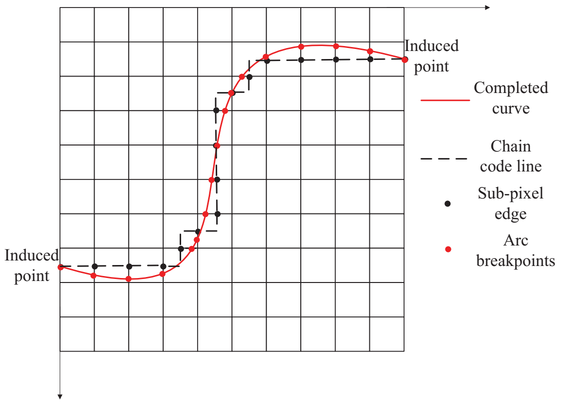


Fig. 2. Example results of the generated arc breakpoints via curve completion and the differential Freeman chain code

$$D_{rec}(2i+1, 2j) = \Lambda(\hat{D}_{down}(i, j), \hat{D}_{down}(i+1, j)) \tag{5}$$

where $\hat{D}_{down}$ is the reconstructed depth map, Λ is a breakpoint dependent prediction operator which switches between linear interpolation and zero-order-hold (ZOH) depending on the location of arc breakpoints. The operator can be represented as follows:

$$\Lambda = \begin{cases} \frac{\hat{D}_{down}(i,j)+\hat{D}_{down}(i+1,j)}{2} & A(2i+1, 2j) = 0 \\ \hat{D}_{down}(i+1, j) & A(2i+1, 2j) = -1 \\ \hat{D}_{down}(i, j) & A(2i+1, 2j) = 1 \end{cases} \tag{6}$$

where $A(2i + 1, 2j) = 0$ means there are no breakpoints in the interval $(2i, 2j) \sim (2i + 2, 2j)$, while the values -1 and 1 correspond to breakpoint on the left half and right half of the interval, respectively. Similar interpolation procedures are performed in the vertical and diagonal directions.

3.4. Edge preserving filtering

Ideal depth map should have clear edges on the boundaries and locally uniform values inside objects. Although the upsampled depth map gets clearer edges around boundaries, it might be still not extremely flat in the flat regions due to coding and upsampling errors. Thus, we apply an edge preserving filter [13] to the reconstructed depth map. It is a local linear filter between the guided image I, the input image p, and the filtered image q, which can be expressed as follows:

$$
\begin{aligned}
q(i) &= \sum_j W_{ij}(I) p_j \\
W_{ij}(I) &= \frac{1}{|\omega|^2} \sum_{k:(i,j)\in\omega_k} \left(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}\right)
\end{aligned}
\tag{7}
$$

where i, j are the pixel indexes. The filter kernel W_{ij} is a function of the guidance image I and independent of p. ω_k is a square window, u_k and σ_k^2 are the mean and variance of I in window ω_k, $|\omega|$ is the number of pixels in ω_k and the parameter ϵ is a regularization parameter. The edge preserving filter chooses the depth image itself as a guided image (I and p are identical in Eq. (7)). It could further reduce artifacts in the synthesized view.

4. EXPERIMENTAL RESULTS

We evaluate the proposed method on *Breakdancers* and *Ballet* test sequences with 8 views and resolution of 1024×768 [14]. In the experiments, the input depth maps are downsampled by a factor of 2 in both horizontal and vertical directions. The reduced resolution depth maps are encoded by the HM 9.0 reference software. The pixel size is chosen as 0.01 in the curve completion process.

The synthesized view is used to measure the performance instead of the depth map itself. View 3 and view 5 are selected as reference views and a virtual view 4 is synthesized using the View Synthesis Reference Software (VSRS) 2.3. For an objective comparison, the PSNR of the synthesized view from the compressed depth map is calculated with respect to the one synthesized by the original depth map. In Fig. 3, The rate-distortion curves are depicted by total bit-rate required to encode the depth maps of both the reference views. It can be seen that that the proposed down/upsampling approach outperforms the original depth coding over a wide scope of bitrate. The PSNR of the synthesized view raises up to 1.5dB by the proposed scheme than the original coding scheme without the guidance information of corresponding color video. Fig. 4 shows the subjective comparison of the synthesized view by the original depth coding scheme and the proposed depth coding scheme at similar bit-rate, where the proposed scheme suppressed coding artifacts along depth edges besides the P-SNR improvement.

5. CONCLUSION

In this paper, we propose a depth down/upsampling scheme for depth map in 3-D video coding. Unlike prior upsampling methods, it conducts upsampling without the corresponding color video, thus avoiding the inaccuracy of the compressed color video and reducing the storage space for color video. It upsamples the reduced resolution depth map with the guidance of breakpoints, which are generated via completed edge contours. Only a few explicit breakpoints are required to transmitted to the decoder, and other breakpoints would be generated via completed contours. The proposed scheme attains significant improvements in both rate-rendering distortion and subjective performance.

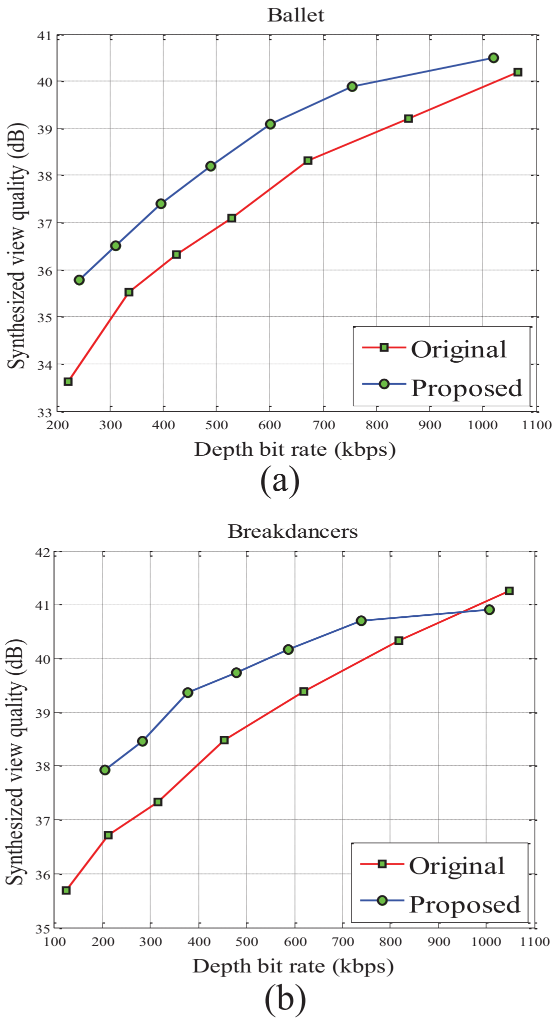
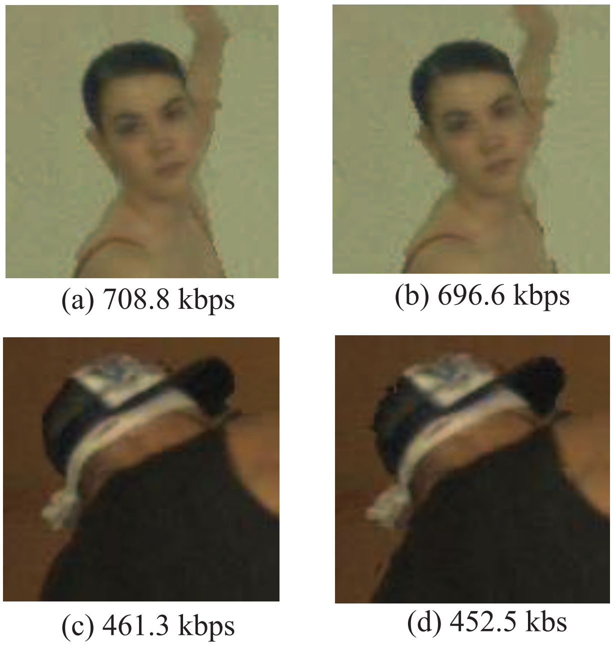


Fig. 3. R-D curves of both the original depth coding scheme and the proposed scheme. (a) *Ballet*. (b) *Breakdancers*.

6. REFERENCES

[1] K. Muller, P. Merkle, T. Wiegand, “3-d video representation using depth maps”, *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643-656, Mar. 2011.

[2] Y. Morvan, D. Farin, et al., “Platelet-based coding of depth maps for the transmission of Multiview images”,



(a) 708.8 kbps

(b) 696.6 kbps

(c) 461.3 kbps

(d) 452.5 kbs

Fig. 4. The sampled frames of the synthesized view by the proposed scheme ((a),(c)) and original depth coding ((b),(d)) at similar bit-rate for sequences *Ballet* and *Breakdancers*.

Proceedings of SPIE, Stereoscopic Displays and Applications, San Jose (CA), USA, vol. 6055, Jan. 2006.

[3] P. Merkle, Y. Morvan, A. Smolic, et al., "The effects of multiview depth video compression on multiview rendering", *Signal Processing: Image Communication*, vol. 24, no. 1, pp. 73-88, Jan. 2009.

[4] M. Maitre, M. Do, "Depth and depth-color coding using shape-adaptive wavelets", *Journal of Visual Communication and Image Representation*, vol. 21, no. 5, pp. 513-522, May 2010.

[5] K.J. Oh, S. Yea, A. Vetro, and Y.S. Ho, "Depth reconstruction filter and down/upsampling for depth coding in 3-d video", *IEEE Signal Processing Letter*, vol. 16, no. 9, pp. 747-750, Mar. 2009.

[6] V. Nguyen, D. Min, M. Do, "Efficient techniques for depth video compression using weighted mode filtering", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 189-202, Feb. 2013.

[7] E. Ekmekcioglu, V. velisavljevic, and S.T. Worrall, "Edge and motion-adaptive median filtering for multi-view depth map enhancement", *Proc. Picture Coding Symposium*, Chicago, Illinois, USA, May 2009.

[8] P. Merkle, C. bartnik, K. Muler, et al., "3d video: Depth coding based on inter-component prediction of block partitions", *Proc. Picture Coding Symposium*, Krakow, Poland, pp. 149-152, May 2012.

[9] R. Mthew, P. Zanuttigh, D. Taubman, "Highly scalable coding of depth maps with arc breakpoints", *Proc. IEEE Data Compression Conference*, Snowbird, Utah, USA, pp. 42-51, Mar. 2012.

[10] S. Ullman, "Filling-in the gaps: The shape of subjective contours and a model for their generation", *Biological Cybernetics*, vol. 25, no. 1, pp. 1-6, 1976.

[11] B. Horn, "The curve of least energy", *ACM Transactions on Mathematical Software*, vol. 9, no. 4, pp. 441-460, 1983.

[12] B.Y. Guy and B.S. Ohad, "A tangent bundle theory for visual curve completion", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1263-1280, 2012.

[13] K. He, J. Sun, X. Tang, "Guided image filtering", *Proc. European Conference on Computer Vision*, Grete, Greece, pp. 1-14, Mar. 2010.

[14] C.L. Zitnick, S.B. Kang, M. Uyttendaele, et al., "High-quality video view interpolation using a layered representation", *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 600-608, 2004.