Non-Linear Depth Map Resampling for Depth-Enhanced 3D Video Coding

Payman Aflaki[a], Miska M. Hannuksela[b], Dmytro Rusanovskyy[b], Moncef Gabbouj[a]

[a]Department of Signal Processing, Tampere University of Technology, Tampere, Finland;
[b]Nokia Research Center, Tampere, Finland;

Abstract— **Depth-enhanced 3D video coding includes coding of texture views and associated depth maps. It has been observed that coding of depth map at reduced resolution provides better rate-distortion performance on synthesized views comparing to utilization of full resolution (FR) depth maps in many coding scenarios based on the Advanced Video Coding (H.264/AVC) standard. Conventional techniques for down and upsampling do not take typical characteristics of depth maps, such as distinct edges and smooth regions within depth objects, into account. Hence, more efficient down and upsampling tools, capable of preserving edges better, are needed. In this letter, novel non-linear methods to down and upsample depth maps are presented. Bitrate comparison of synthesized views, including texture and depth map bitstreams, is presented against a conventional linear resampling algorithm. Objective results show an average bitrate reduction of 5.29% and 3.31% for the proposed down and upsampling methods with ratio ½, respectively, comparing to the anchor method. Moreover, a joint utilization of the proposed down and upsampling brings up to 20% and on average 7.35% bitrate reduction.**

Index Terms—**MVC, depth map, resampling, non-linear.**

I. INTRODUCTION

The multiview video plus depth (MVD) format [1], where each video data pixel is associated with a corresponding depth map value, is one of the most promising methods for providing 3D video services flexible for different types of multiview displays as well as user adaptation at disparity between rendered views. The MVD format allows reducing the input data for the 3DV systems significantly, since most of the views will be rendered from the available decoded views and depth maps using a Depth Image Based Rendering (DIBR) [2] algorithm.

3D video coding (3DV) standardization by the Moving Picture Experts Group (MPEG) is a recent activity targeting at enabling a variety of display types and preferences including varying baseline to adjust the depth perception. Another important target of the MPEG 3DV standardization is the support for multiview auto-stereoscopic displays, thus many high-quality views shall be available in decoder/display side prior to displaying. As the existing video compression standards were found to be sub-optimal to achieve these targets, MPEG issued a Call for Proposals for 3D video coding (hereafter referred to as the 3DV CfP) [3] to kick off the 3DV standardization activity targeting to provide 3D enhancement to the existing the Multiview Video Coding extension of the Advanced Video Coding standard, H.264/MVC [4], as well as to the ongoing High Efficiency Video Coding (HEVC) standardization. As one consequence of the 3DV CfP, a H.264/MVC-based test model [5] (hereafter referred to as 3DV-ATM) was chosen and has been further developed by MPEG as collaborative standardization effort. In addition to exploiting temporal and inter-view correlation among texture or depth views to achieve high coding efficiency, 3DV-ATM provides means to encode depth maps into the same bitstream with texture and enhances H.264/MVC with coding tools utilizing the correlation between depth and texture data.

In 3D video applications, depth maps are used for synthesizing new images but not to be directly viewed by end users. Thus, when coding depth maps, the goal is to maximize the perceived visual quality of the rendered virtual color views instead of the visual quality of the depth maps themselves [6]. Traditional video coding methods have been designed to operate through a Rate-Distortion Optimization (RDO) of coded data and a pixel-based distortion introduced by codec, e.g. Sum of Absolute Differences (SAD). However, coding distortions of a depth map typically have a non-linear impact on the visual quality of rendered views [7]. For example, errors in the depth map close to a sharp edge can result in severe rendering artifacts, while errors on a smooth area may have negligible subjective influence on the final quality. Therefore, utilization of traditional RDO for depth map compression may result in suboptimal performance of a 3D video coding system [7].

As demonstrated in many of the responses to the 3DV CfP and enabled in 3DV-ATM, coding of depth map data at a reduced resolution is a viable solution for improving the rate-distortion performance of the complete 3D video coding system. In such systems, depth map data is downsampled prior to the encoding and upsampled to the original FR after decoding. Obviously, downsampling of depth maps, which is performed in combination with low-pass filtering for aliasing suppression, may lead to smoothed edges and therefore to a significant distortion in rendered views.

In this letter, a novel algorithm for non-linear down and upsampling of depth map is presented. The proposed

algorithm preserves edges in processed depth map data and provides quality improvement in synthesized images.

The rest of the letter is organized as follows. Section II provides a review of depth map resampling methods. The proposed down and upsampling methods are introduced in section III while the simulation setup and results are presented in section IV. Finally, the letter concludes in section V.

II. DEPTH MAP RESAMPLING

Downsampling traditionally includes low pass filtering, which suppresses high frequency components in the depth map and therefore leads to over-smooth edges. The consequent quality reduction due to resampling causes significant visual artifacts in synthesized views particularly at object boundaries. Hence, edge-preserving downsampling for depth map should be considered even though traditional image downsampling techniques use linear filters not designed to preserve edges. For example, in [8] and [9], the median value of an N×N window was chosen as the most representative value to be used at reduced-resolution depth map (where factor N specifies the downsampling ratio).

Similarly to downsampling, upsampling should preserve depth edges. In various works, e.g. [9] and [10], cross-component bi-lateral filtering has been used for depth upsampling. In a cross-component bi-lateral filter, the similarity of co-located texture samples is used to derive filter weights for depth in addition to the conventional filtering window applied spatially for the depth samples.

Another approach for coded depth upsampling and restoration was used in [8] and [11]. Other than a depth resampling technique to improve the quality of rendered views, authors proposed that a decoded low resolution depth map image to be processed with a depth reconstruction filter. This filter consists of a novel frequent-low-high filter and a bilateral filter. Depth map is first upsampled using a nearest neighbor filter, which is followed by post-processing using a median filter, a frequent-low-high filter and a cross-component bi-lateral filter. The 2D median filter is used to smooth blocking artifacts caused by coding. The frequent-low-high filter is a non-linear filter used to recover object boundaries, which results into selecting either the most frequently occurring sample value below or above the median sample value within a filter window. The bilateral filter is used to eliminate the errors still present after both filtering procedures.

In [12] an edge adaptive upsampling method for better compression of depth maps is presented. In this work edge information is extracted from the high resolution reconstructed texture video by applying 3×3 Sobel filter operators. Gradients caused by texture transitions, rather than depth changes, are eliminated by considering the local depth intensity gradients. Then the linear interpolation filters are replaced with a locally adaptive filter. Test results reported in [12], show that the proposed technique outperformed linear MPEG upsampling filter [13] in terms of objective and subjective quality of synthesized views. However, the utilization of texture data in upsampling process of depth map can be considered a drawback of the proposed method due to a significant increase in the memory access bandwidth and computational complexity.

III. PROPOSED DOWN AND UP SAMPLING METHODS

The proposed down and upsampling method presented in this section can be applied directly to depth maps and do not need complementary information from the reconstructed or the decoded texture images. In following sub-sections a detailed description of the algorithms is presented.

A. Downsampling

To perform the proposed downsampling method, a block of pixels (BOP) will be determined based on the downsampling ratio. The FR image will be covered with the necessary number of non-overlapping BOPs and for each BOP a single value will be calculated to present it in the downsampled image. The size of the BOP is defined as the reciprocal of the downsampling ratio; e.g. if the image is downsampled with ratios 1/x and 1/y (both x and y are positive values equal or bigger than 1) along the horizontal and vertical direction where the size of the BOP is specified with x and y in width and height, respectively.

The proposed downsampling method utilizes a closeness-favored averaging algorithm as described in the following paragraphs. In the first step an average over the BOP will be calculated, as seen in (1).

$$Avg_{BOP} = \frac{\sum_{i=1}^{x} \sum_{j=1}^{y} BOP_{(i,j)}}{x \times y} \tag{1}$$

where $BOP_{(i,j)}$ presents a pixel value where *i* and *j* are the horizontal and vertical pixel indices within the BOP.

In the next step pixels of BOP are categorized into two sets as shown in (2).

$$BOP(i,j) \in \begin{cases} G_{high}, & if\ BOP(i,j) \geq Avg_{BOP} \\ G_{low}, & otherwise. \end{cases} \tag{2}$$

where $BOP(i,j)$ is the same as in equation (1).

If the number of pixels in G_{high} is equal to or greater than half of the number of pixels in the BOP, the Estimated Value (*EV*) of the associated BOP is an average over the pixel values of G_{high}. Otherwise, *EV* is set to Avg_{BOP}, as shown in (3) and (4).

$$Avg_{G_{high}} = \frac{\sum_{i=1}^{x} \sum_{j=1}^{y} BOP_{(i,j)}}{Count(G_{high})}\ , \qquad BOP(i,j) \in G_{high} \tag{3}$$

$$EV = \begin{cases} Avg_{G_{high}}, & Count(G_{high}) \geq \dfrac{Count(BOP)}{2} \\ Avg_{BoP}, & otherwise \end{cases} \tag{4}$$

where Count(X) counts the number of elements in X. The calculated EV is the value which represents the considered BOP in the downsampled image. As can be observed from the equations, if at least half of the pixels in a BOP are classified to belong to objects that are close-by, i.e. closer than the average depth value of the BOP, the method considers only those pixels in downsampling and hence attempts to preserve sharp boundaries of foreground objects. Since the entire FR image is processed with non-overlapped

BOPs, the calculated EVs form a downsampled version of the input image.

B. *Upsampling*

Considering Figure 1 pixel values {A, B, C, D, E, F, G, H, I} in the downsampled image are utilized to upsample pixel E and calculate values of pixels {a, b, c, d} in the associated BOP in the upsampled image. Afterwards, a, b, c, and d will be utilized to create possible remaining pixel values in the BOP of upsampled image.

Let us consider the pixel which needs to be upsampled (pixel E in Figure 1). To calculate the value of the top-left pixel in the BOP of the upsampled image (a in Figure 1), the pixel values on the left and top of E will be considered (D and B in Figure 1, respectively).

In the first step, the absolute differences of E with D and B are calculated. This is shown in (5) and (6).

$$diff_{EB} = |E - B| \tag{5}$$
$$diff_{ED} = |E - D| \tag{6}$$

The filter window (FW) by which the value of a will be calculated is defined as following. If both diff_{EB} and diff_{ED} are smaller than a threshold (th), then it is assumed that A, B, D, and E belong to the same depth region, and consequently the final FW contains pixels A, B, D, and E. Otherwise, the final FW is chosen to contain only A, B, and D, as shown in (7). This choice of the filter window attempts to restore the shape of a depth boundary between E and a depth object containing A, B, and D.

$$FW \in \begin{cases} \{A, B, D, E\}, & diff_{EB} < th \text{ and } diff_{ED} < th \\ \{A, B, D\}, & otherwise \end{cases} \tag{7}$$

In the next step, the average of pixel values in selected FW is calculated and utilized as a in the upsampled image (see Figure 1).This is presented in (8).

$$a = average\ (FW) \tag{8}$$

The complete procedure to calculate a from A, B, D, and E can be presented with function $upsampler$ as shown in (9).

$$a = upsampler(E, A, B, D) \tag{9}$$

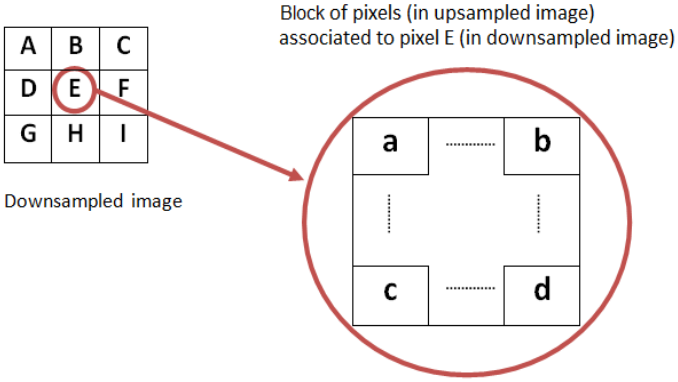

Figure 1. To be upsampled pixel value (E) and associated block of pixels in upsampled image

The same process is applied for the other three corner pixels, i.e. b, c, and d. The pixel values of b, c, and d in Figure 1 can be calculated using the equations shown in (10).

$$\begin{aligned} b &= upsampler(E, C, B, F) \\ c &= upsampler(E, G, D, H) \\ d &= upsampler(E, I, H, F) \end{aligned} \tag{10}$$

Finally, when a, b, c, and d in the BOP of the upsampled image are available, bi-linear interpolation is applied to obtain the remaining pixel values in the BOP.

IV. SIMULATIONS

A. *Simulation Setup*

3DV-ATM reference software [5] (hereafter referred to as reference software (RS)) was utilized for encoding the multiview plus depth (MVD) data. Simulations were conducted according to the MPEG 3DV Common Test Conditions (CTC) [14].

Depth maps for all resampling schemes were first downsampled to half resolution along each of coordinate axes prior to encoding and upsampled to the FR after decoding. The threshold utilized for proposed upsampling process was fixed to 16 for all sequences. The view synthesis was performed with VSRS software, version 3.5 [15] with configuration and camera parameters information provided with MPEG 3DV CTC [14]. In our experiment, we provide the results for C3 scenario, described in [14] where three evenly distributed intermediate views between each two input (coded) views were synthesized.

B. *Simulation Results*

The proposed algorithm was tested against the depth map resampling utilized in the RS, with 12-tap low pass filtering in downsampling according to Joint Scalable Video Model (JSVM) [16] and bi-linear upsampling for upsampling. The performance of the proposed down and upsampling methods was evaluated separately against the techniques utilized in the RS with the two following set of experiments:

- First experiment: A combination of the proposed downsampling and RS upsampling compared against RS used for both downsampling and upsampling
- Second experiment: A combination of the RS downsampling and the proposed upsampling compared against RS used for both downsampling and upsampling

In the third experiment the efficiency of the method in [11] and a joint utilization of the proposed down and upsampling was tested against the RS.

Simulation results for the first and second experiments using Bjontegaard delta bitrate and delta Peak Signal-to-Noise Ratio (PSNR) [17] are reported in Tables I while results of the third experiment are presented in table II. In these calculations the total bitrate of texture plus depth maps along with the average luma PSNR of all six synthesized views were considered.

Table I shows that the proposed downsampling method outperformed the anchor method of [14] by 5.29% of Bjontegaard delta bitrate reduction (dBR) and 3.31% dBR on average was achieved by the proposed upsampling algorithm. Results of the third experiment show that a joint utilization of both proposed methods provided 7.35% dBR comparing the RS. From our simulations, the algorithm presented in [11] performed worse than anchor objectively. However, based on our expert subjective viewing, the perceived quality of our proposed method and the algorithm presented in [11] outperformed that of RS. Moreover, in [11] it is claimed that by applying the proposed filter on depth maps a better compression for depth map and higher subjective quality for rendered views are achieved. Additionally, the decoder execution time of the proposed method was 85% of the RS on average, while the method presented in [11] has more computation operations per pixel than our proposed algorithm.

V. CONCLUSIONS

Due to the characteristics of depth maps, coding of depth maps at a lower spatial resolution than the resolution of luma texture pictures typically results into improved rate-distortion performance. However, traditional resampling algorithms which use linear filtering result to significant distortions introduced to rendered views. In this experiment, we improved the depth-enhanced 3D video coding through edge-preserving techniques for depth map resampling. Two novel algorithms for down and upsampling depth maps were presented in this letter. Results showed that proposed down and upsampling steps with ratio ½ outperform MPEG 3DV anchor resampling methods by 7.35% of dBR on average and up to 20.4%. In addition to this, the proposed implementation decreased the decoder execution time by 15% compared to the MPEG H.264/AVC-based 3DV reference software.

TABLE I. FIRST AND SECOND EXPERIMENTS: PERFORMANCE OF PROPOSED DOWN AND UP SAMPLING AGAINST ANCHOR

	Proposed downsampling against anchor		Proposed upsampling against anchor	
	dBR,%	dPSNR ,dB	dBR,%	dPSNR ,dB
Poznan Hall2	-2.47	0.08	-1.61	0.05
Poznan Street	-3.43	0.10	-1.93	0.05
Undo Dancer	-17.15	0.65	-9.87	0.33
Ghost Town	-5.69	0.21	-1.88	0.07
Kendo	-2.62	0.12	-2.62	0.12
Balloons	-1.07	0.05	-1.01	0.04
Newspaper	-4.57	0.17	-4.21	0.16
Average	**-5.29**	**0.20**	**-3.31**	**0.12**

TABLE II. THIRD EXPERIMENT: PERFORMANCE OF JOINT UTILIZATION OF PROPOSED DOWN/UP SAMPLING AGAINST RS

	Method presented in [11]		Proposed method	
	dBR,%	dPSNR ,dB	dBR,%	dPSNR ,dB
Poznan Hall2	0.22	-0.27	-4.34	0.15
Poznan Street	0.79	-0.02	-5.24	0.15
Undo Dancer	0.99	-0.04	-20.40	0.87
Ghost Town	1.91	-0.07	-8.54	0.33
Kendo	1.72	-0.07	-4.53	0.21
Balloons	1.92	-0.09	-1.19	0.05
Newspaper	2.94	-0.1	-7.19	0.29
Average	**1.50**	**-0.09**	**-7.35**	**0.29**

ACKNOWLEDGMENT

The authors would like to thank Prof. M. Domañski, et al. for providing Poznan sequences and Camera Parameters [18].

REFERENCES

[1] P. Merkle, A. Smolic, K. Müller, and T. Wiegand, "Multi-view video plus depth representation and coding," Proc. of International Conf. on Image Processing, vol. 1, pp. 201-204, Oct. 2007.

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

[3] "Call for Proposals on 3D Video Coding Technology," ISO/IEC JTC1/SC29/WG11 MPEG2011/N12036, March 2011.

[4] "Advanced video coding for generic audiovisual services," ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), 2012.

[5] "Test model for AVC based 3D video coding," ISO/IEC JTC1/SC29/WG11 MPEG2012/N12558, Feb. 2012.

[6] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P. H. N. de With, T. Wiegand, "The effects of multiview depth video compression on multiview rendering," Signal Processing: Image Communication, vol. 24, pp. 73–88, Jan. 2009.

[7] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map coding with distortion estimation of rendered view," Proc. of IS&T/SPIE Electronic Imaging, vol. 7543, pp. 75430B–75430B-10, Jan, 2010.

[8] D. Tian, D. Graziosi, Y. Wang, N. Cheung, A. Vetro, "Mitsubishi Response to MPEG Call for Proposal on 3D Video Coding Technology," ISO/IEC JTC1/SC29/WG11 MPEG2011/M22663, Nov. 2011.

[9] M. O. Wildeboer, T. Yendo, M. Panahpour Tehrani, T. Fujii, M.Tanimoto, "Color Based Depth Upsampling for Depth Compression," Proc. of Picture Coding Symposiom, pp. 170-173, Dec. 2010.

[10] A. K. Riemens, O. P. Gangwal, B. Barenbrug, R-P. M. Berretty, "Multi-step joint bilateral depth upsampling," Proc. of Visual Communications and Image Processing, vol. 7257, pp. 72570M-72570M-12 Jan. 2009.

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

[12] E. Ekmekcioglu, M. Mrak, S.Worrall, and A. M. Kondoz, "Utilisation of edge adaptive upsampling in compression of depth map videos for enhanced free-viewpoint rendering," Proc. of International Conf. on Image Processing, pp. 733–736, Nov. 2009.

[13] G. Sullivan and S. Sun, ,"Spatial Scalability Filters," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6/JVT-P007, July 2005

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

[15] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, and Y. Mori, "Reference softwares for depth estimation and view synthesis," ISO/IEC JTC1/SC29/WG11/M15377, Apr. 2008.

[16] JM reference software: http://iphome.hhi.de/suehring/tml/download

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

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