Joint Texture and Depth Coding using Cuboid Data Compression

Manoranjan Paul[*], Subrata Chakraborty[Ω], Manzur Murshed[₵] and Pallab Kanti Podder[*]
[*]School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW-2795, Australia
[Ω]School of Management and Enterprise, University of Southern Queensland, Queensland, Australia
[₵]School of Information Technology, Federation University,VIC-3842, Australia
{mpaul ; ppodder}@csu.edu.au, subrata.chakraborty@usq.edu.au, manzur.murshed@federation.edu.au

***Abstract*— The latest *multiview video coding* (MVC) standards such as 3D-HEVC and H.264/MVC normally encodes texture and depth videos separately. Significant amount of rate-distortion performance and computational performance are sacrificed due to separate encoding due to the lack of exploitation of joint information. Obviously, separate encoding also creates synchronization issue for 3D scene formation in the decoder. Moreover, the hierarchical frame referencing architecture in the MVC creates random access frame delay. In this paper we develop an encoder and decoder framework where we can encode texture and depth video jointly by forming and encoding 3D cuboid using high dimensional entropy coding. The results from our experiments show that our proposed framework outperforms the 3D-HEVC in rate-distortion performance and reduces the computational time significantly by reducing random access frame delay.**

Keywords—video coding; cuboid; McFIS; dynamic background; mutiview video

I. INTRODUCTION

Providing the necessary interactivity in the three-dimensional (3D) space to satisfy end-users' desire to observe objects and actions from different viewpoints, a scene is captured as a normal RGB video (i.e., texture video) and a depth video (consists of relative distance of scene content from the camera) by multiple cameras with different angles. To generate a 3D scene we need texture video as well as depth video of a number of viewing angles. Obviously using more texture and depth videos in 3D scene formation provides more realistic 3D scene. Considering the significant overlapping of the views and, more importantly, the availability of a rich set of relations on the geometric properties of a pair of views from camera properties, known as the *epipolar geometry*, joint encoding/decoding of views can achieve significant compression by exploiting inter-view correlation, in addition to the traditional intra-view correlation. According to earlier studies, the H.264/MVC [1]-[3] introduces a frame referencing mechanism among the texture views (*S*) and the temporal (*T*) images. In the *multiview video coding* (MVC) reference architecture, the hierarchical B-picture prediction format [4] is applied for both the intra-view and the inter-view. This approach encodes the current frame by exploiting the redundancies from the neighbouring encoded texture frames as references from both the inter-view and the intra-view. Research shows that for inter-view coding path is currently selected for only 10–30% of blocks [22], although their inter-view spatial overlapping area is more than 90% [4]. More recently, 3D-HEVC [22] provides a conceptual framework for joint texture and depth video encoding to exploit not only inter and intra-view texture video redundancy but also texture-depth video inter-relationship. Although, a number of techniques, for example [23], are proposed to encode texture and depth videos jointly, a standard frame-referencing framework for join-encoding is not recommended yet for 3D-HEVC. An example of jointly encoding block diagram is shown in Fig. 1 [24]. The basic structure of the 3D-HEVC video codec is similar as for H.264/MVC, all video pictures and depth maps that represent the video scene at the same time instant build an access unit and the access units of the input texture and depth signals are coded consecutively. Inside an access unit, the video picture of the so-called independent view is transmitted first directly followed by the associated depth map.

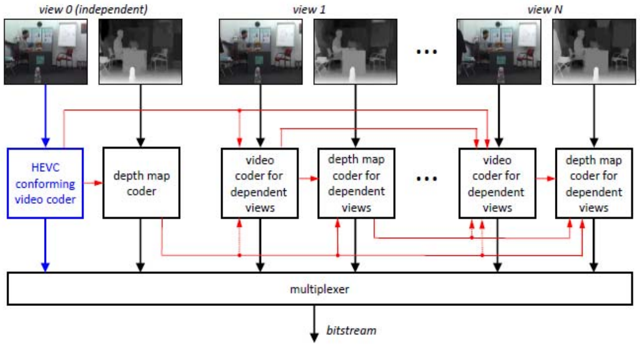

view 0 (independent)
view 1
view N
...
HEVC conforming video coder
depth map coder
video coder for dependent views
depth map coder for dependent views
...
video coder for dependent views
depth map coder for dependent views
multiplexer
bitstream

Fig. 1: Block diagram of 3D-HEVC Video Codec [24].

Fig. 2 shows depicts the recommended frame prediction architecture by the H.264/MVC standard for texture video coding where five views are used with *group of picture* (GOP) size of eight. As indicated by this architecture, a frame can use up to 4 reference frames from the inter-view and the intra-views. Under this technique, to decode a current frame, we need to decode a set of frames earlier, which introduces the *random access frame delay* (RAFD) problem

978-1-4673-9930-2/15/$31.00 ©2015 IEEE

and restricts the interactivity. The RAFD is a measured indicates the maximum number of frames that requires decoding to gain access to a B-frame in the structure. The random access delay for the topmost hierarchical order can be defined as: $F_{\max} = 3M_{\max} + 2\lfloor (N-1)/2 \rfloor$, where M_{max} is denoted as the highest order and N indicates the number of views [3]. As an example, to gain access to a B-frame in the 4th order (b4-frames shown in Fig. 2), we need to first decode 18 frames. The RAFD problem poses challenges for some applications e.g., real-time communication in an interactive manner may not be practical by applying the existing prediction architecture. Moreover, when we use joint texture and depth coding, using the above mentioned hierarchical-frame referencing, we introduce more RAFD for decoding.

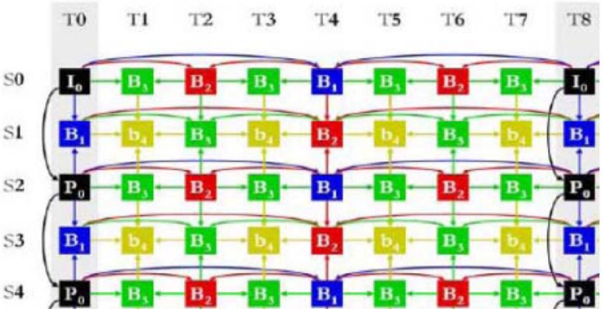

T0 T1 T2 T3 T4 T5 T6 T7 T8
S0
S1
S2
S3
S4
I_0 B_3 B_2 B_3 B_1 B_3 B_2 B_3 I_0
B_1 b_4 B_3 b_4 B_2 b_4 B_3 b_4 B_1
P_0 B_3 B_2 B_3 B_1 B_3 B_2 B_3 P_0
B_1 b_4 B_3 b_4 B_2 b_4 B_3 b_4 B_1
P_0 B_3 B_2 B_3 B_1 B_3 B_2 B_3 P_0

Fig. 2: H.264/MVC recommend prediction architecture for different texture views (S) and temporal (T) images.

The basic foundation of video codec of 3D-HEVC is HEVC which provides 50% better compression with respect to the H.264/AVC for the same perceptual video quality using 4 times more computational time [25]-[26] to encode a single video. A 3D-HEVC video coder requires huge computational time as it needs to encode multiple texture and depth videos in the same time. Thus, reducing computational time by keeping the same *rate-distortion* (RD) performance is always a welcome issue especially for low-powered devices.

In this study we propose a novel cuboid data compression technique for joint texture and depth video coding to improve the rate-distortion performance by reducing RAFD problem and computational time. Moreover, a joint encoding framework reduces the synchronization issue between texture and depth videos for 3D scene formation. In our proposed technique, we form a 3D frame using the texture and depth frames from the same temporal slice (i.e., i_{th}) of a number of views. The 3D *motion estimation* (ME) is then conducted for the current 3D *coding unit* (CU) using the immediate predecessor as a reference frame. The predecessor is developed by using the (*i*-1)th frame of a set of views. Then, 3D zigzag scan, 3D quantization, and 3D coding have been used for better compression. Since the intra-view images show higher correlations among them compared to the inter-view images, our proposed cuboid technique maintains comparable RD performance, but achieves significant reduction in the overall computational time and the RAFD problem compared to 3D-HEVC, thus enabling more interactive communications in real-time mode.

In recent studies dynamic backgrounds are used in video coding techniques [4], [7]-[8]. Paul *et al.* [4], [7] utilised the concept of the *most common frame in a scene* (McFIS) applying the Gaussian mixture [9]-[11] based dynamic background modelling for video coding. The McFIS is considered as an additional reference frame while encoding the present frame. The motion aspects of the present frame is assumed to be referenced from the immediate predecessor frame and from the use of the static background. McFIS is used as reference for the uncovered background aspect. The final reference is detected using the Lagrangian multiplier at block and sub-block levels [6]. In this paper, we also propose another technique known as Cuboid-McFIS. In this technique an additional 3D reference matrix is developed which includes McFISes from a number of views which is then used for 3D motion estimation. Results from our experiments show that our proposed Cuboid technique provides comparable RD performance and significantly reduces computational time along with reducing the RAFD problem as compared to the 3D-HEVC. Moreover, the proposed Cuboid-McFIS technique outperforms the 3D-HEVC by improving RD performance and reducing computational time and RAFD problem.

II. PROPOSED JOINT CODING FOR CUBOID FRAMEWORK

Multiview capturing systems are expected to deliver high quality interactivity in the 3D space so that the end users can view quality video contents from different angles and depths. Current MVC technologies do not have sufficient support for interactivity, computational time and RD performance. Literature suggests that texture and depth motion for a CU may vary [19]-[20]. We assume that at finer level (block and sub-block) the texture and depth motion may vary however due to the fact the motion belongs to the same object there will be similarity between the texture and depth motion. In this study we exploit these similarities to achieve better coding performance by the proposed joint coding scheme. If we intend to encode texture and depth separately we need to use two separate coders with the traditional 3D-HEVC or H.264/MVC frame-referencing architecture (see Fig. 1 or Fig. 2) or somehow we need to combine both texture and depth videos in the architecture. However, with our joint coding approach using the cuboid architecture we can achieve this with a single coder.

After formation of 3D frames, we estimate motion for the current 3D CU using the variable size blocks (e.g., 32×32×8, 32×16×8, 16×16×8, 16×8×8, 8×16×8, 8×8×8, etc. where last dimension is for the number of views) using the previous 3D reference frame. Then, we encode 3D CU using 3D coding. Details are described in the sub-sections.

A. Forming 3D Frames

About 70~90% references in the 3D-HEVC or H.264/MVC scheme are coming from intra-view [22]. We can achieve better interactivity and computational time that enhance the scope of the MVC by using 3D formation and 3D ME and sacrificing some RD performance. We can form 3D frame in different ways. In this paper achieve this by stacking the same temporal positioned texture and depth frames of a number of views. Paul *et al.* [12]-[14] first saw the benefit of 3D ME using the first approach of 3D formation. Fig. 3 shows the 3D formation technique using *n*-texture views and n-depth views in the proposed cuboid data compression architecture.

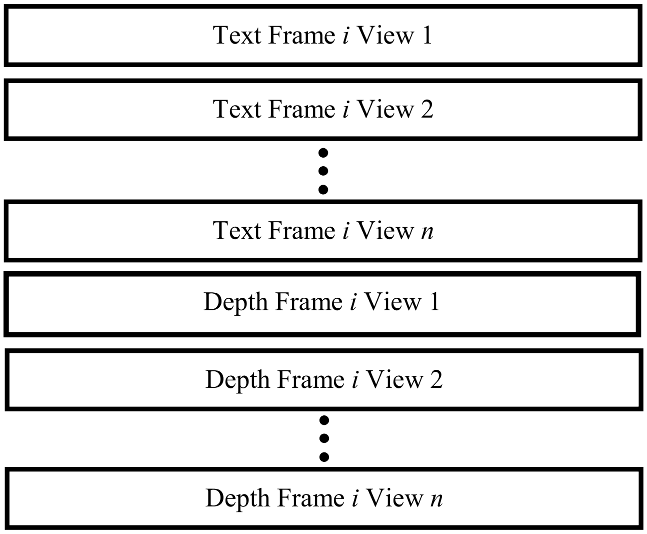


Fig. 3: 3D frame formation for texture and depth coding in the cuboid architecture.

B. 3D Motion Estimation

In the proposed joint texture and depth coding technique, we form a 3D frame comprising *i*th frames (texture and depth) of a number of views. ME can be carried out for a 3D CU and its sub-blocking where the reference 3D frame is developed with the immediate predecessor (i.e., *i*-1th) frames of the same number of views. We do not utilise the inter-view redundancy explicitly in our proposed 3D ME technique, for following reasons: (a) the correlation among the intra-view images is found to be higher than that of among the inter-view images [1]-[3], (b) to eliminate the RAFD issues, and (c) to minimize the time for computation. Our proposed technique requires just one ME as opposed to multiple ME required for each reference frame (e.g., b4 frame of view S3 at T3 in Fig. 2 requires 4 separate ME utilising 4 reference frames),. The proposed technique is capable of reducing a large amount of computational time as it does not require any disparity estimation and ME for multiple reference frames during actual coding process. The proposed method has reduced RAFD problem as all frames of a number of views in the same temporal position are encoded at the same time which is an additional benefit of the proposed technique.

C. 3D Coding

Research suggest the possibility of achieving huge computational by transform coding (without ME) compared to the ME-compensation-transform coding while a 3D-block is formed with temporal images and 3D-DCT [17] is applied on 3D-block [15][16]. As the proposed technique forms the 3D-CU utilises the frames from the same temporal slice of a number of different views and the ME for each frame uses the reference frame of the same view, we can exploit almost all intra-view temporal redundancy. Moreover, relation motion of different views also guide each other to provide average motion vector of 3D block. Applying 3D-DCT can concentrate the image energy in the upper-top-left areas in 3D-block more perfectively compared to multiple 2D-DCT so that 3D zigzag [18] scanning has been applied. For zigzag (i.e., conversion of 3D matrix to 1D vector) we have applied *reshape* Matlab function which rearranges the elements in column wise. Further investigation is needed to find optimal zigzag scan order. After 3D-DCT, the distributions of a majority of the significant AC coefficients can be modelled by the Gamma distribution and the distribution of the DC coefficient can be approximated by a Gaussian distribution in most cases. This knowledge can enable the design of optimal quantization values for 3D-DCT coefficients that produce minimum distortion and thus achieve close to optimal compression efficiency [16]. The proposed technique uses following quantization $q(\vec{k}) = \lfloor Q/4(1+k_1^p+k_2^p+k_3^p) \rfloor$ where $q(k)$ is the quantization value at position k , Q is the *quantization parameter*, and the value of p should be 0 to 1 where '0' provides same quantization for all coefficients and '1' provides coarse quantization for high frequency components. In our implementation we have used 0.5. As the full list of CAVLC/CABAC codes are not available for 3D DCT coefficient, we have generated VLC codes by dividing all coefficients into allowable number of coefficients by HEVC.

III. PROPOSED CUBOID-MCFIS CODING FRAMEWORK

Despite the proposed technique successfully addresses three identified limitations of standard method including computational time, frame delay, and texture-depth synchronization for 3D scene formation, in its current form could not outperform the 3D-HEVC in RD performance in most videos. Particularly it underperforms slightly for motion-active multiview video sequences because we do not fully exploit inter-view redundancy which contributes around 15% references. In the proposed cuboid-McFIS scheme, we generate a McFIS for each view and forms 3D-McFIS in the similar fashion of the cuboid architecture and use it as an additional reference frame for ME and compensation. The 3D-McFIS is used as a reference areas for the normal static areas and uncovered static areas. The McFIS is formed using the *Gaussian Mixture* modelling. As the McFIS is used for the static areas, we may use relatively small motion search compared to the other reference frame to reduce the computational time.

IV. EXPERIMENTAL RESULTS

Algorithms for the proposed cuboid and cuboid-McFIS techniques are implemented according to the 3D-HEVC recommendations such as 25 Hz, search length of ±31 and quarter-pel accuracy, GOP size 16. In our proposed techniques, we have used the IBP prediction format as compared to the hierarchical B picture predication architecture used in 3D-HEVC. We consider symmetric and asymmetric block portioning scheme and we set CU size as 32×32. Thus, we use all inter-modes from 32×32 to 8×8. We use dual reference frames for the proposed schemes. For the proposed cuboid scheme we use two immediate previously coded frames as the reference frames, on the other hand, for the Cuboid-McFIS scheme, we use the immediate previously coded frame and the McFIS as the reference frames. The experimental results are produced using 3 views in the cuboid and the 3D-HEVC hierarchical B-structure. We compared the RD performance for the proposed schemes against the 3D-HEVC for Texture coding, depth coding and combined texture and depth coding using four standard multiview video sequences (Ballet, Break dancer, Car Park, and street). When we encode both texture and depth jointly in the proposed cuboid and cuboid-McFIS schemes, we use two texture views and a depth view. To compare the results we calculate the average bits and PSNR of the 3D-HEVC after encoding texture and depth separately.

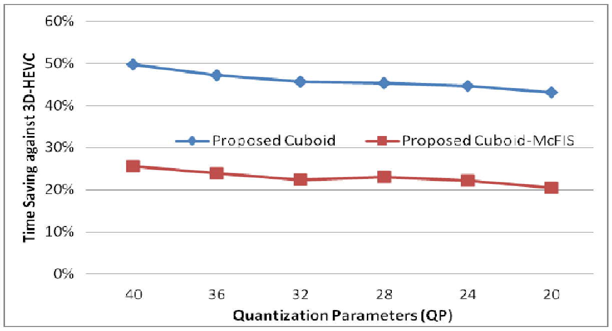
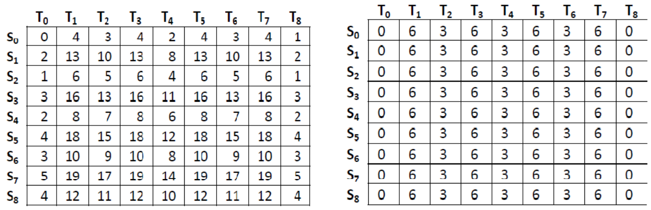


Fig. 4: Average computational time saving by the proposed algorithms against 3D-HEVC.

Fig. 4 shows computational performance comparison of the proposed coding schemes against 3D-HEVC scheme. The figure shows superior performances in computational complexity by the proposed schemes compared to 3D-HEVC. As shown in the figure, the cuboid scheme reduces up to 50% and the cuboid-McFIS scheme reduces up to 25% computational time compared to 3D-HEVC. The cuboid-McFIS requires more computational time compared to the proposed Cuboid scheme because the former requires some computational time to generate McFIS frames using background modelling.



	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
S_0	0	4	3	4	2	4	3	4	1
S_1	2	13	10	13	8	13	10	13	2
S_2	1	6	5	6	4	6	5	6	1
S_3	3	16	13	16	11	16	13	16	3
S_4	2	8	7	8	6	8	7	8	2
S_5	4	18	15	18	12	18	15	18	4
S_6	3	10	9	10	8	10	9	10	3
S_7	5	19	17	19	14	19	17	19	5
S_8	4	12	11	12	10	12	11	12	4

	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
S_0	0	6	3	6	3	6	3	6	0
S_1	0	6	3	6	3	6	3	6	0
S_2	0	6	3	6	3	6	3	6	0
S_3	0	6	3	6	3	6	3	6	0
S_4	0	6	3	6	3	6	3	6	0
S_5	0	6	3	6	3	6	3	6	0
S_6	0	6	3	6	3	6	3	6	0
S_7	0	6	3	6	3	6	3	6	0
S_8	0	6	3	6	3	6	3	6	0

Fig. 5: A number of frame-delay of the 3D-HEVC or H.264/MVC hierarchical B-prediction structure (left) and the proposed cuboid structure (right).

Fig. 5 shows the number of frames required to decode before watching a particular frame in both schemes using 9 views and 8 GOP. Obviously if we increase the number of views and GOP size, the frame delay amount will be increased according to the above mention equation (see F_{max} formulation) for the 3D-HEVC or H.264/MVC structure, which is obviously higher for the proposed scheme. According to the figure, the proposed cuboid structure requires 3.67 frame delay whereas the H.264/AVC or 3D-HEVC structure requires 8.99 frame delay for 9 views and 8 GOP size. Thus, the proposed scheme can reduce the RAFD problem significantly.

RD performance comparison among the proposed schemes and 3D-HEVC is shown in Fig. 6. The results are for *only texture videos*, *only depth videos*, and *texture plus depth videos*. We observe that the cuboid and the cuboid-McFIS schemes outperform the 3D-HEVC by up to 1.5dB in 3 video sequences with the exception of Break dancer video where 3D-HEVC scheme performs slightly better than the proposed schemes for texture videos. The figure also shows the depth coding comparison with new schemes showing superior performances. The joint texture and depth coding performances are shown in the same figure. We observe that the proposed cuboid-McFIS is the best performer followed by proposed cuboid scheme for 3 of the video sequences. The cuboid-McFIS scheme achieves up to 1.2dB performance gain over the standard 3D-HEVC. We also observe under performance of the proposed schemes for the Break dancer video sequence. The main cause of the under performance of the proposed scheme for the Break dancer sequence compared to 3D-HEVC would be complex motion which cannot be captured in the cuboid data compression structure. However, the proposed cuboid data compression framework can reduce computational time and RAFD problem significantly for all video sequences.

V. CONCLUSIONS

In this paper, we proposed new joint texture and depth coding scheme using the cuboid architecture. The new cuboid scheme reduces the interactivity challenge in the standard 3D-HEVC scheme by the use of cuboid coding. The new schemes reduces the computational time by up to 50% compared to the standard 3D-HEVC scheme. We also proposed the Cuboid-McFIS scheme where an extra 3D reference frame is utilised along with the immediate predecessor 3D frame. The additional 3D frame is generated utilising dynamic nature of the background frames of each view known as McFIS using Gaussian mixture modelling. The results from our experiments show that the cuboid and cuboid-McFIS outperform the 3D-HEVC by improving up to 1.2 dB PSNR compared to the 3D-HEVC and reducing random access frame delay significantly.

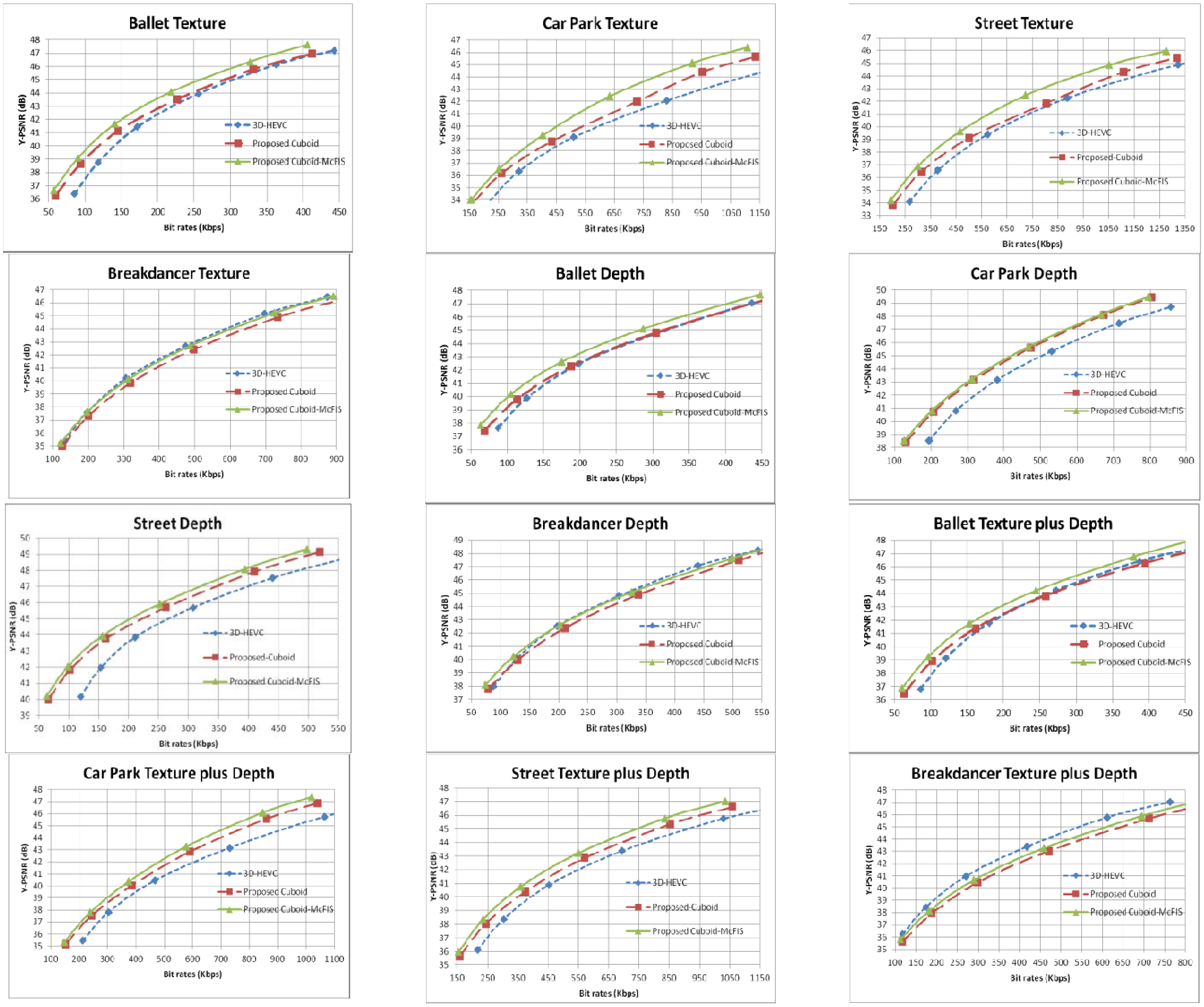


Fig. 6: Performance for Rate-distortion using four standard multiview video sequences by the 3D-HEVC, proposed cuboid and cuboid-McFIS algorithms for only texture, depth, and texture plus depth.

REFERENCES

[1] A. Vetro, T. Wiegand, and G. J. Sullivan, “Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC standard,” Proceedings of the IEEE, 99(4), 626 - 642, 2011.

[2] P. Pandit, A. Vetro, Y. Chen, “Joint Multiview Video Model (JMVM) 7 Reference Software,” N9579, MPEG of ISO/IEC JTC1/SC29/WG11, Antalya, Jan. 2008.

[3] M. Talebpourazad, “3D-TV content generation and multi-view video coding, PhD thesis, 2010.

[4] M. Paul, W. Lin, C. T. Lau, and B. –S. Lee, "McFIS in hierarchical bipredictive picture-based video coding for referencing the stable area in a scene," *IEEE International conference on Image Processing* (IEEE ICIP-11), 2011.

[5] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, “Overview of the H.264/AVC Video Coding Standard,” *IEEE Transaction on Circuits and Systems for Video Technology*, 13(7), pp. 560-576, 2003.

[6] M. Paul, W. Lin, C. T. Lau, and B. –S. Lee, "Direct Inter-Mode Selection for H.264 Video Coding using Phase Correlation," *IEEE Transactions on Image Processing*, 20(2), pp. 461 – 473, 2011.

[7] M. Paul, W. Lin, C. T. Lau, and B. –S. Lee “Explore and model better I-frame for video coding,” *IEEE Transaction on Circuits and Systems for Video Technology*, 21(9), pp. 1242-1254, 2011.

[8] M. paul, W. Lin, C. T. Lau, and B. S. Lee, “Pattern based video coding using dynamic background modelling,” *EURASIP Journal on Advances in Signal Processing*, December 2013.

[9] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” IEEE CVPR, vol. 2, 246–252, 1999.

[10] X. Li, D. Zhao, S. Ma, and W. Gao, “Fast disparity and motion estimation based on correlations for multi-view video coding,” *IEEE Transactions on Consumer Electronics*, 54(4), pp. 2037-2044, 2008.

[11] S. Zhang, K. Wei, H. Jia, X. Xie, and W. Gao, “An efficient foreground-based surveillance video coding scheme in low bit-rate compression,” *IEEE Visual Comm. and Image Proc*., pp. 1-6, 2012.

[12] M. Paul, J. Gao, and M. Antolovich, “3D motion estimation for 3D video coding,” *IEEE Int. Conference on Acoustics, Speech, and Signal Processing* (IEEE- ICASSP), pp. 1189-1192, 2012.

[13] M. Paul, “Efficient multi-view video coding using 3D motion estimation and virtual frame,” *Elsevier Journal on Neurocomputing*, online published, 2015, doi: 10.1016/j.neucom.2015.10.094.

[14] M. paul, C. Evans, M. Murshed, “Disparity-adjusted 3D multi-view video coding with dynamic background modelling,” *IEEE*

International conference on Image Processing (IEEE ICIP 2013), pp. 1719-1723, September 2013.

[15] A. Burg, R. Keller, J. Wassner, N. Felber, and W. Fichtner, "A 3DDCT real-time video compression system for low complexity singlechip VLSI implementation," MoMuC, 2000.

[16] M. Bhaskaranand and J. D. Gibson, “Distributions of 3D DCT coefficients for video,” IEEE ICASSP, 2009.

[17] N. Bozinovic and J. Konrad, “Motion analysis in 3D DCT domain and its application to video coding,” Signal Processing: Image Communication, 20, 2005.

[18] B.L. Yeo and B. Liu, “Volume rendering of DCT-based compressed 3D scalar data,” *IEEE Trans. Virtual. & Comp. Graph*., 1(1), 1995.

[19] S. Shahriyar, M. Murshed, M. Ali and M. Paul, “Inherently edge-preserving depth-map coding without explicit edge detection and approximation,” *IEEE Conference on Multimedia and Expo Workshops* (ICMEW), 2014.

[20] S. Shahriyar, M. Ali, M. Murshed, and M. Paul, “Efficient Depth Coding By Exploiting Temporal Correlations in Depth Maps,” *IEEE International conference on Digital Image Computing: Techniques and Applications* (IEEE DICTA-14), 2014.

[21] A. Kaup and U. Fecker, “Analysis of multi-reference block matching for multi-view video coding,” Proc. 7th *Workshop Digital Broadcasting*, 33-39, 2006.

[22] K. Müller, H. Schwarz, D. Marpe, C. Bartnik, et al., “3D high-efficiency video coding for multi-view video and depth data,” *IEEE Transactions on Image Processing*, 22(9), 2013.

[23] S. Bosse, H. Schwarz, T. Hinz, T. Wiegand, "Encoder Control for Renderable Regions in High Efficiency Multiview Video Plus Depth Coding", *Picture Coding Symposium*, May 2012.

[24] H. Schwarz and T. Wiegand, "Inter-View Prediction of Motion Data in Multiview Video Coding", *Picture Coding Symposium*, May 2012.

[25] P. Podder, M. Paul, and M. Murshed, "Efficient coding strategy for HEVC performance improvement by exploiting motion features," *40th IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), 2015.

[26] P. Podder, M. Paul, and M. Murshed, “A Novel Motion Classification Based Intermode Selection Strategy for HEVC Performance Improvement,” *Elsevier Journal on Neurocomputing*, online published, 2015, doi:10.1016/j.neucom.2015.08.079.