Fast Inter-Mode Decision Strategy for HEVC on Depth Videos

Pallab Kanti Podder[*], Manoranjan Paul[*], and Manzur Murshed[€]
[*]School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW-2795, Australia
[€]School of Information Technology, Federation University,VIC-3842, Australia
{ppodder ; mpaul}@csu.edu.au, manzur.murshed@federation.edu.au

§***Abstract*— Multiview video employs the utilization of both texture and depth video information from different angles to create a 3D video for more realistic view of a scene. Unlike texture, depth video is a gray scale map that represents the distance between the camera and 3D points in a scene. Existing *multiview video coding* (MVC) techniques including 3D- *High Efficiency Video Coding* (HEVC) standard encode both texture and depth videos jointly by exploiting texture video information for the corresponding *depth video coding* (DVC) to reduce computational time as the texture and depth videos have motion similarity in representing the same scene. This strategy has two limitations: (i) more bits and computational time might be required due to the large residuals for the misalignment between depth and texture edges and (ii) switching between different views may require more times due to the increased dependency between texture and depth. In this paper, we propose an independent DVC technique using HEVC (a video coding standard for single view) so that we can improve the *rate distortion* (RD) performance and reduce computational time by improving switching speed. For this, we use motion features to reduce a number of *motion estimation* (ME) and *motion compensation* (MC) modes in HEVC. As we use motion feature which is the underlying criteria for selecting different modes in the standard and then we select a subset of modes which can provide almost the same RD performance. Experimental outcomes reveal a reduction of 48% encoding time of HEVC encoder with similar RD performance and better interactivity.**

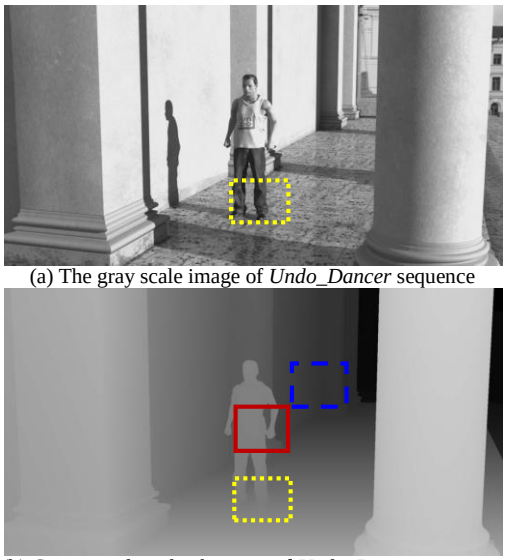
***Keywords—Depth video, HEVC, Intermode selection, Motion features*.**

I. INTRODUCTION

Highly advanced new multiview video coding technology offers more realistic 3-D scenes to the users by transmitting texture video and corresponding depth map from several viewpoints [1]-[2]. Depth map is a gray scale image that represents the distance between the camera and 3D points in the scene. Depth images reveal different characteristics from its corresponding texture images and they are distinguished by smooth regions with sharp edges [3]-[5]. Fig. 1 shows the disparity between a texture image and its corresponding depth image for the standard *Undo_Dancer* video sequence. The block marked by blue square indicates one depth and the block marked by red square indicates the block containing more than one depth which has different values as depicted in Fig. 1 (b).

§This work was supported in part by the Australian Research Council under Discovery Projects Grant DP130103670.

A number of electronic devices with limited processing and battery capacity could not use 3D video since large encoding time is required for 3D-HEVC due to the processing of a number of views. To reduce this computational time, there are a number of accessible techniques, including the 3D-HEVC texture-assisted depth coding strategy. Both intra and inter-prediction based depth coding are available in the literature. A number of intra prediction-based depth video coding techniques [6]-[10] are presented in the literature. Park [9] and Gu *et al*. [10] attempt to reduce computational time for intra depth coding by selecting a sub-set of available intra modes on the basis of edge type and smoothness of the block respectively. Normally intra prediction-based depth coding techniques require more bits compared to inter-prediction techniques.



(a) The gray scale image of *Undo_Dancer* sequence

(b) Corresponding depth image of *Undo_Dancer* sequence

Fig. 1. Distinction of a depth image from its gray scale presentation. In (b), the blue square indicates one depth and the red square indicates the block containing more than one depth.

The depth information based fast inter-mode decision algorithm for color plus depth-map, 3D video coding is presented by Lin *et al.* in [11]. Utilizing the depth information, they classify scenes according to the associated depth and assign the appropriate modes for texture videos. The time saving is about 60% with the bit rate increment of 3.84% and

quality loss of 0.18 dB compared with the conventional full search mode selection approach adopted in H.264 JM13.2 reference software. Li *et al.* in [12] attempt to perform a pixel-based motion estimation for a better predictor in the depth coding by exploiting the texture motion. Their proposed coding method achieves superior RD performance compared to JM 18.2 although average encoding time increases over 6.26%. Shen *et al.* [13] incorporate an adaptive motion search range determination method and a fast mode decision method based on the correlation of texture and depth video to reduce computational time of depth video coding. In comparison with the original H.264 JMVC encoder, they reduce on average 60% of encoding time with PSNR loss of 0.08 dB and the average bitrate increment of 0.6%.

To reduce computational time for depth coding, majority of existing inter-prediction techniques [12]-[13] use motion information of texture video to encode depth video. The texture-assisted depth coders are capable to save bits by avoiding transmission of motion vector as they predict motion vector from the corresponding texture frame. Moreover, it reduces computational time by reusing texture motion vector for the corresponding depth frame by completely/partially avoiding costly motion estimation process. However, these processes suffer from the following three limitations. Firstly, misalignment with large residuals that may occur between the depth and the texture edges when texture-assisted depth prediction strategy is applied [12]. The dotted yellow box in Fig. 1 (b) shows the misalignment between texture and depth video especially in the area of bottom edges compared to Fig. 1 (a). More bits and additional computational time might be required as an impact of this misalignment. Secondly, interactivity among different views would be reduced as switching between different views would require more times as the dependency between texture and depth increases. Thirdly, texture video's *Lagrangian multiplier* (LM) is used for the mode selection in depth coding, as to the best of our knowledge, there is no distinct LM developed for the depth coding. Therefore this sole parameter based mode decision would not provide the best RD performance at different operational coding points due to different characteristics of depth map compared to texture video.

To address the aforementioned limitations, the strategies applied by the proposed method include the following: (i) it independently encode the depth map regardless of considering its corresponding texture in order to provide more interactivity and avoid misalignment problem, and (ii) it uses a preprocessing motion criteria to select a subset of inter-modes by partially avoiding the dependency of existing LM to reduce computational time. For this, we use the latest HEVC [14] encoder that is designed for single view coding. Since motion feature is the underlying criteria for mode selection, in this work, phase correlation based three motion features are extracted for a subset of intermode selection. The final mode decision is taken only on the selected subset using the existing LM, thus, significant percentage of computational time can be saved. Moreover, the proposed technique improves the RD performance as it does not use any texture-assisted motion vector especially for those misaligned boundary areas. The proposed method also provides more interactivity since it can independently encode depth maps by exploiting HEVC coder.

The remainder of the paper is organized as follows: Section II illustrates the working mechanism of HEVC, Section III explains the key steps of the proposed technique; experimental results and analysis are detailed in Section IV, while Section V concludes the paper.

II. Working Mechanism of HEVC

The latest HEVC standard introduces a number of innovative and powerful coding tools to acquire better compression efficiency [15] for different video types. Since there is no dedicated encoder assigned for depth video coding, we employ the standalone HEVC encoder (HM 12.1) [16] for the selection of *motion estimation* (ME) and *motion compensation* (MC) modes in depth videos. In texture video, the compression efficiency is acquired at the cost of more than 4 times algorithmic complexity [17][18] due to the extended number of coding depth levels from 16×16 up to 64×64-pixel, and complex *coding unit* (CU) partitioning scheme compared to its predecessor H.264. HM eventually selects a particular motion prediction mode by checking the *Lagrangian cost function* (LCF) exhaustively using all modes in a single or more coding depth levels. The LCF, (T_z) for mode selection (T_z is the zth mode) is noted:

$$\Phi(T_z) = D(T_z) + \lambda \times R(T_z) \quad (1)$$

where $D(T_z)$ is the distortion, λ is the LM, and $R(T_z)$ is the resultant bit. HM therefore decides the best partitioning mode by exploring minimum 8, to maximum 24 interprediction modes which consumes a substantial percentage of encoding time. Moreover, due to some limitations of existing LM used by HM, the proposed DVC technique does not merely depend on the LCF, rather it executes some preprocessing stages to select a subset of modes in a flexible way to reduce encoding time. As the motion feature is the basic criteria for mode selection, therefore, due to more accurate decision on ME and MC modes for depth video, in this work, phase correlation based three motion features are extracted. These interactively selected weighted motion features are combined through a fusion process to determine the motion types and a subset of interprediction modes. ME and MC are performed only on the selected subset. Compared to the exhaustive mode selection approach in HM, the proposed technique endeavors computational time reduction by selecting motion feature based subset of appropriate block partitioning modes.

III. Proposed Technique

The phase-correlation provides us the relative displacement information between current block and the motion compensated block in the reference frame by *Fast Fourier Transformation* (FFT).

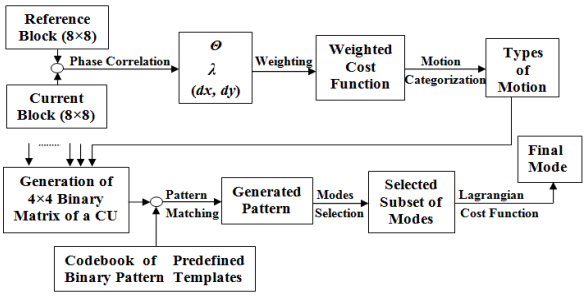


Fig. 2. Block diagram of proposed mode selection process for depth video

In this paper, phase-correlation is applied between the current and the reference block with 8×8 pixels size to extract three motion features including (i) *energy concentration ratio* (i.e., ECR (Θ)), (ii) predicted motion vector (dx,dy) and (iii) phase correlation peak (λ). Then we develop a cost function using weighted average of the normalized motion features to determine a unified motion feature for the current block which then converted into binary motion type by applying threshold. Thus $n \times n$ binary matrix can be obtained for a given CU. The best-matched pattern template is selected using a similarity metric where each of the templates corresponds to a sub-set of inter-prediction modes. Only the final mode is determined based on the lowest value of LCF. The whole process block diagram is shown in Fig. 2.

A. Motion Features Extraction

We calculate the phase correlation by applying the FFT and then *inverse* FFT (IFFT) of the current and reference blocks and eventually applying the FFTSHIFT function as-

$$\Omega = fftshift\left|ifft\left(e^{j(\angle F_r - \angle F_c)}\right)\right| \quad (2)$$

where F_c and F_r are the Fast Fourier transformed blocks of the current C and reference R blocks respectively and $\angle$ is the phase of the corresponding transformed block and Ω is a two dimensional (2-D) matrix. We evaluate the phase correlation peak (λ) from the position of ($dx + \beta/2 + 1$, $dy + \beta/2 + 1$) by-

$$\lambda = \Omega(dx + \beta/2 + 1, dy + \beta/2 + 1) \quad (3)$$

where the blocksize denoted by β is 8 as the 8×8-pixel block is used by the proposed approach to calculate phase correlation. In the matched block generation process, we use the phase of the current block and magnitude of the motion-compensated block in the reference frame. Then we calculate the matched reference block (B_{MR}) for the current block by:

$$B_{MR} = \left|ifft\left(|F_r|e^{j(\angle F_c)}\right)\right| \quad (4)$$

The displacement error (§) is calculated by: § = $C - B_{MR}$. We then apply the *discrete cosine transform* (DCT) to error § and calculate the ECR (i.e., Θ) as the ratio of low frequency component and the total energy of the error block by:

$$\Theta = (D_{EL} / D_{ET}) \quad (5)$$

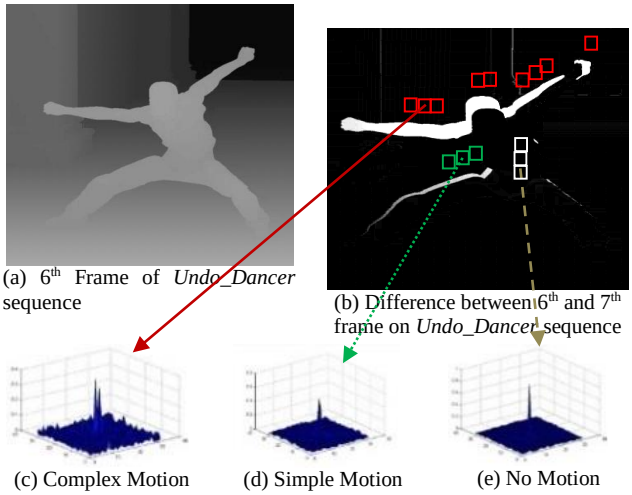
where D_{EL} and D_{ET} represent the top-left triangle energy and the whole area energy of a particular block. The two sides of the top-left triangle are considered 6-pixels in the proposed implementation as we regard the three-fourth of a blocksize.

B. Cost Function Based Motion Categorization

Once the phase correlation extracted motion features (i.e., Θ, λ and (dx, dy) are evaluated, a cost function $\epsilon(i,j)$ using these features for (i,j)th block is finally developed by

$$\epsilon(i,j) = \omega_1\Theta(i,j) + \omega_2(1-\lambda) + \omega_3\left(\frac{|dx|}{\delta} + \frac{|dy|}{\delta}\right) \quad (6)$$

where δ is the maximum block size, and ω_1 to ω_3 are the equal weights with $\sum_{i=1}^{3}\omega_i = 1$. If the cost function value (i.e., ϵ (i,j)) > predefined threshold (i.e., 0.55), the motion type is marked by 1, otherwise motion type is marked by 0, where 1 and 0 correspond to the presence and absence of motion respectively in the video frame. Fig. 3 demonstrates the phase correlation related quantitative magnitude of the motion accuracy (i.e., motion peak λ). Fig. 3 (a) is the original image taken from 6th frame of *Undo_Dancer* video, (b) presents the difference between 6th and 7th frame (multiplied by 6 for better visualization), (c-e) show the signal peak at coordinates corresponding to the shift between current and reference block. The associated blocks with red, green and white present the areas with complex, simple and no motion respectively.



(a) 6th Frame of *Undo_Dancer* sequence

(b) Difference between 6th and 7th frame on *Undo_Dancer* sequence

(c) Complex Motion (d) Simple Motion (e) No Motion

Fig. 3. Illustration of motion features generated at different CUs of 7th frame on *Undo_Dancer* video; (c-e) are the phase shifted plots for complex motion (0.8), simple motion (0.7) and no motion (0.3) respectively.

C. Intermode Selection

For the generation of binary matrix, we exploit each of the 8×8 pixel blocks from the 32×32 pixel blocks (i.e., CU) and produce a matrix of 4×4 binary values for each CU (applying threshold). The cost function generated 4×4 binary matrix is then compared with a codebook of predefined binary pattern templates to select a subset of modes (illustrated in Fig. 2). Each of the templates is constructed with a pattern of motion and no-motion block (1 and 0 respectively) focusing on the rectangular and regular object shapes of depth videos at 32×32 block level as shown in Fig. 4. Based on the similarity metric, we explore the best-matched *binary pattern templates* (BPTs) for a binary motion block of a CU. We use a simple similarity metric using the *sum of absolute difference* (SAD) between the binary matrix of a CU generated by phase correlation and the BPTs in Fig. 4.

We select the best-matched BPT that provides the minimum SAD for a CU. The SAD, D_n is determined as-

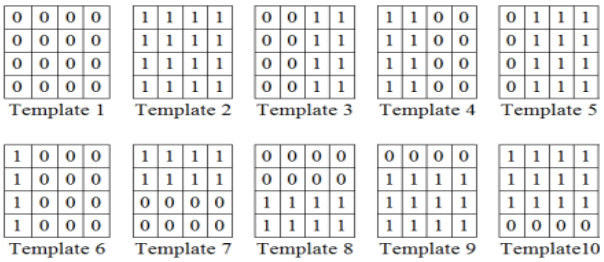


Fig. 4. Codebook of the proposed binary pattern templates for subset of inter-mode selection.

$$D_n(x,y) = \sum_{x=0}^{4}\sum_{y=0}^{4}\left|M(x,y) - P_n(x,y)\right|. \quad (7)$$

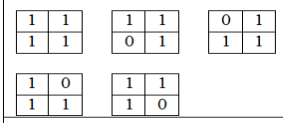
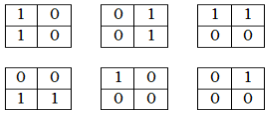
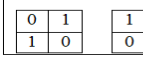
where M is the binary motion prediction matrix of a CU comprising 4×4 '1' or '0' combination and P_n is the n-th BPT. The best-matched j-th BPT is selected from all BPTs as-

$$P_j = \arg\min_{\forall P_n \in BPT}(D_n). \quad (8)$$

TABLE I. PROPOSED MODE SELECTION AT 32×32 BLOCK LEVEL USING THE CODEBOOK OF PREDEFINED BINARY MOTION PATTERN TEMPLATES

Templates Based on Motion Patterns at 32×32 Block Level	Selection of Modes at 32×32 Block Level
Template 1	Skip or Inter 32×32
Template 2	Intra 16×16 or Inter 16×16
Template 3 & 4	Inter {32×16 or 16×16}
Template 5	Inter {32×8 or 16×16}
Template 6	Inter {32×24 or 16×16}
Template 7 & 8	Inter {16×32 or 16×16}
Template 9	Inter {8×32 or 16×16}
Template 10	Inter {24×32 or 16×16}

TABLE II. SUBSET OF MODE SELECTION AT 16×16 BLOCK LEVEL BASED ON MOTION PATTERNS

Motion Patterns at 16×16 Block Level	Selected Subset of Modes
	Inter {16×8, 8×16 or 8×8}
	Inter {16×8 and 8×16}
	Inter {16×12, 16×4, 12×16 or 4×16}.

The mode selection process from the BPTs at 32×32 and 16×16 coding depth levels is illustrated in TABLE I and TABLE II respectively. Once a particular template selects a subset of candidate modes at 32×32 level, the final mode is decided by their lowest LCF. In addition, at 32×32 level, if any of the 16×16 coding depth level mode is selected, we further explore smaller modes at 16×16 level using the motion pattern (shown in TABLE II). Then the equation for the final mode (ξ) selection is given by:

$$\xi = \arg\min_{\forall T_z}(\Phi(T_z)) \quad (9)$$

where $\Phi(T_z)$ is the LCF for mode selection (where T_z is the zth mode).

D. Threshold Selection

We apply the static threshold value and fix it by 0.55 in order to elevate the threshold selection complexity although we consider both the homogeneous and heterogeneous motion regions in the CUs of all sequences. Because of improper distribution of ECR values, Podder *et al.* [19] (also mentioned different thresholds in [20]-[21]) use a range of thresholds from 0.37 to 0.52 for different bit-rates. Those thresholds could not perform well for depth videos as we do not employ the texture assisted depth coding. Almost in all cases, it exceeds the value of 0.55 for all sequences especially for the blocks having dominant motion. Therefore, in the proposed technique we use the fixed value of threshold (i.e., 0.55) for a wide range of bit-rates and different video resolutions.

IV. EXPERIMENTAL RESULTS

Performance evaluation results of the proposed technique are presented using nine widely used standard depth videos- *GT_Fly* (1088×1920), *Pozan_CarPark* (1088×1920), *Pozan_Street* (1088×1920), *Pozan_Hall* (1088×1920), *Undo_Dancer* (1088×1920), *Lovebird* (768×1024), *Newspaper* (768×1024), *Exit* (480×640), and *Ballroom* (480×640). These test videos are representative in the sense ranging from of low motion activity such as *Lovebird* to the high motion activity such as *Undo_Dancer*. The sequences are encoded with 25 frame rate and search length ±64.

The inter-prediction based depth coding techniques mentioned in section I are developed based on H.264 which is the previous standard of HEVC. Those techniques could not be straightforward applied for HEVC because of the extended number of modes, CU size extensions from 16×16 up to 64×64-pixels, symmetric/asymmetric CU partitioning patterns, coding length of motion vectors and other advanced parameter settings. Moreover, the algorithmic complexity of HEVC is more than 4 times compared to H.264 [17]-[18]. Therefore the distinct feature of the proposed method is that it uses HEVC reference encoder HM12.1 as a benchmark and achieves 48% computational time savings for that of the encoder with similar RD performance.

A. Set-up of the Experiment

The test platform used in the simulations is a dedicated desktop machine with the configuration of Intel Core i7-3.4 GHz, 16 GB RAM, 1TB HDD running 64 bit Windows operating system. The proposed technique is integrated into the reference encoder HM12.1 and evaluated with the simulation specifications provided in [16]. RD performance of both schemes are compared considering the maximum CU size of 32×32 by enabling both symmetric/asymmetric partitioning

block size of 32×32 to 8×8 depth levels. In the proposed technique, each CU is set as a 32×32 pixel block and we use 8×8 pixel blocks for binary matrix generation. The tested bit-rates used in the simulation are QP=20, 24, 28, 32 and 36. The calculation of PSNR and the bitrate difference were performed according to the procedures described in [22].

B. Results and Analysis

The experimental evaluation reveals that over nine sequences with different resolutions, and for a wide range of bit-rates, the proposed method reduces on average 48% (range: 39%-52%) computational time as shown in Fig. 5. The equation for the time savings (ΔT) is defined as:

$$\Delta T = \frac{(T_{HM} - T_p)}{T_{HM}} \times 100\% \qquad (10)$$

where T_{HM} and T_p denote the total encoding time consumed by HM and the proposed method respectively. For the sequences containing low motion activity such as *Lovebird*, the proposed algorithm reduces the depth coding time up to 70%. The reason is that sequences with homogeneous motion and texture generate large areas of flat regions in depth maps. In addition to bit-rate time analysis, we also demonstrate the encoding time analysis of both techniques by categorizing different resolutions of depth videos and notice that the proposed method achieves on average 46% encoding time savings compared to HM12.1 as shown in Fig. 6. The figure also reveals that the proposed technique saves more time for 768×1024 resolution-type videos compared to the resolutions of 1088×1920 or 480×640.

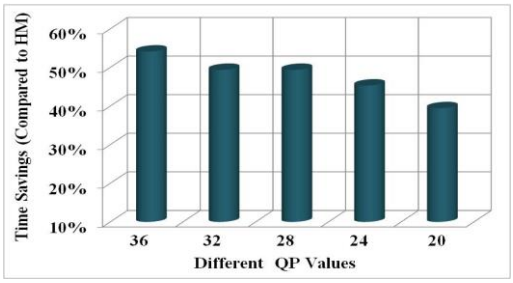


Fig. 5. Illustration of overall time savings by the proposed method against HM at different bit-rates.

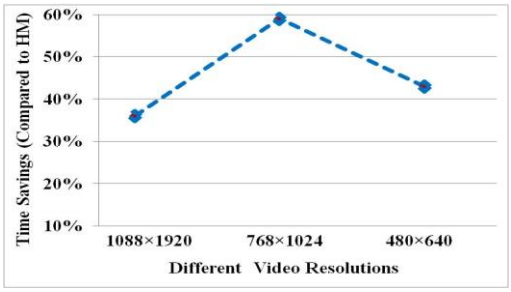


Fig. 6. Overall time savings by the proposed method against HM for different resolutions of depth videos.

Fig. 7 shows the average percentage of three different depth level modes for nine depth video sequences at QP=20 to 36. For the proposed method, the higher depth level modes (16×16 and 8×8) are exploited for the CUs with dominated motion due to acquire similar/improved RD performance.

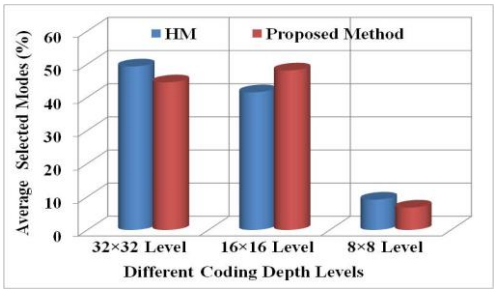


Fig. 7. HM and proposed method based average mode selection at three coding depth levels.

For RD performance evaluation of the proposed method, first we generate the RD test curves and compare against HM using three different sequences *Lovebird*, *Newspaper*, and *Undo_Dancer* for a wide range of bit-rates as demonstrated in Fig. 8. The outcomes of the figure reveal the similar RD performance of proposed technique with HM12.1. The reason is a special caring about the motion dominated CUs and partitioning them by smart selection of appropriate block partitioning modes. TABLE III presents the performance comparison results of the proposed method (against HM) for all the different sequences used in the experiment. The results show that compared to the mode selection approach in HM, the proposed technique achieves an almost similar BD-PSNR and BD-Bit Rate values (small average reduction of 0.08dB PSNR) with a negligible average bit-rate increment of 0.53%.

TABLE III. PERFORMANCE COMPARISON OF THE PROPOSED TECHNIQUE AGAINST HM USING BD-BIT RATE AND BD-PSNR

Sequence Resolutions	Name of the Sequences	BD-PSNR (dB)	BD-Bit Rate (%)
1088×1920	*Undo_Dancer*	-0.03	0.27
	Pozan_Hall	-0.15	0.93
	GT_Fly	-0.12	0.72
	Pozan_CarPark	-0.14	0.81
	Pozan_Street	-0.11	0.63
Average		-0.11	0.67
768×1024	*Newspaper*	-0.02	0.18
	Lovebird	-0.01	0.15
Average		-0.01	0.16
480×640	*Ballroom*	-0.11	0.63
	Exit	-0.07	0.51
Average		-0.09	0.57
Overall Average		-0.08	0.53

C. Subjective Quality Test

Fig. 9 (a) shows the original image of *Undo_Dancer* video taken for subjective quality assessment, while Fig. 9 (b) and (c) illustrate the reproduced images by HM and the proposed method respectively. To present the comparison of image quality, let us concentrate on the hands movement edges in the three images which are marked by the red, yellow, and blue ellipses respectively. It can be perceived that the three ellipse marked sections have almost similar image quality. It was presented in experimental discussion section that the proposed technique also saves encoding time. As a result, a number of electronic devices with limited processing and battery capacity can exploit 3D video facilities with better interactivity.

V. CONCLUSION

In this work an independent depth video coding framework is developed for computational time reduction and performance improvement of HEVC encoder. Existing texture video information based depth video coding techniques incur with the limitations of misalignment and switching, thereby, suffers from proper interactivity. To address these limitations, the proposed method independently encode the depth videos, regardless of considering its corresponding texture in order to avoid misalignment problem and provide more interactivity. For this, the proposed technique exploits preprocessing motion criteria to select a subset of inter-modes by using predefined binary motion pattern templates. From the selected subset, the final mode is decided based on their lowest Lagrangian cost function. Compared to HEVC encoder (HM12.1), the proposed strategy reduces on average 48% computational time (range: 39%-52%) while providing almost similar rate-distortion performance for a wide range of bit-rates and better interactivity.

REFERENCES

[1] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegard, "3d video and free viewpoint video- technologies applications and MPEG standards," *IEEE International Conference on Multimedia and Expo* (ICME), pp. 2161-2164, 2006.

[2] M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, "FTV for 3-D spatial communication," *Proceedings of the IEEE*, vol. 100, no. 4, pp. 905-917, April 2012.

[3] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Muller, P. With and T. Wiegand, "The effect of depth compression on multiview rendering quality," *3DTV Conference*: *The True Vision Capture, Transmission and Display of 3D Vide*, pp. 245-248, 2008.

[4] E. Bosc, M. Pressigout, and L. Morin, " Focus on visual rendering quality through content based depth map coding, " *Picture Coding Symphosium* (PCS), pp. 158-161, 2010.

[5] Y. Zhao, C. Zhu, Z. Z. Chen, D. Tian, and L. Yu, "Boundary artifact reduction in view synthesis of 3D video: From perspective of texture-depth alignment, " *IEEE Transactions on Broadcasting*, pp. 510-522, vol. 57, no. 2, June 2011.

[6] G. Shen, W. S. Kim, A. Ortega, J. Lee and H. Wey, "Edge aware intra prediction for depth map coding," *IEEE International Conference on Image Processing*, pp. 3393-3396, September 2010.

[7] 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.

[8] S. Shahriyar, M. Manzur, M. Ali, and M. Paul, "Cuboid Coding of Depth Motion Vectors Using Binary Tree Based Decomposition," *Data Compression Conference*, 2015.

[9] C. S. Park, "Edge-Based Intramode Selection for Depth-Map Coding in 3D-HEVC" *IEEE Transactions on Image Processing*, vol. 24, issue 1, pp. 155-162, January 2015.

[10] Z. Gu, J. Zheng, N. Ling, and P. Zhang, "Fast depth modelling mode selection for 3D HEVC depth intra coding," *IEEE International Conference on Multimedia and Expo Workshops*, pp. 1-4, July 2013.

[11] Y. H. Lin and J. Ling, "A depth information based fast mode decision algorithm for colour plus depth-map 3D videos," *IEEE transactions on Broadcasting*, pp. 542-550, vol.57, no. 2, June 2011.

[12] S. Li, J. Lei, C. Zhu, L. Yu, and C. Hou, "Pixel-based interprediction in coded texture assisted depth coding," *IEEE Signal Processing Letters*, vol. 21, no.1, January 2014.

[13] L. Shen, P. An, Z. Liu, and Z. Zhang, "Low complexity depth coding assisted by coding information from colour video," *IEEE Transactions on Broadcasting*, pp. 128-133, vol. 60, no.1, March 2014.

[14] High Efficiency Video Coding, document ITU-T Rec. H.265 and ISO/IEC 23008-2 (HEVC), ITU-T and ISO/IEC, April 2013.

[15] B. Bross, Han, W. J. Ohm, J.R. Sullivan, and G. J. Wiegand, "High Efficiency Video Coding Text Specification Draft 8," JTCVC- J1003, Sweden 2012.

[16] Joint Collaborative Team on Video Coding (JCT-VC), HM Software Manual, CVS server at: (http://hevc.kw.bbc.co.uk/svn/jctvc-hm/), 2013.

[17] Y. Lu "Real-Time CPU based H.265/HEVC Encoding Solution with Intel Platform Technology," Intel Corporation, Shanghai, PRC, December 2013.

[18] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC Complexity and Implementation Analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1684-1695, December 2012.

[19] P. Podder, M. Paul, M. Murshed, and S. Chakrabarty, "Fast Intermode Selection for HEVC Video Coding Using Phase Correlation," *IEEE International Conference on Digital Image Computing: Techniques and Applications* (DICTA), pp. 1-8, November 2014.

[20] 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*, vol. 20, no. 2, pp. 461 – 473, 2011.

[21] 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.

[22] G. Bjontegaard, "Calculation of Average PSNR Differences Between RD curves," ITU-T SC16/Q6, VCEG-M33, Austin, USA, 2001.

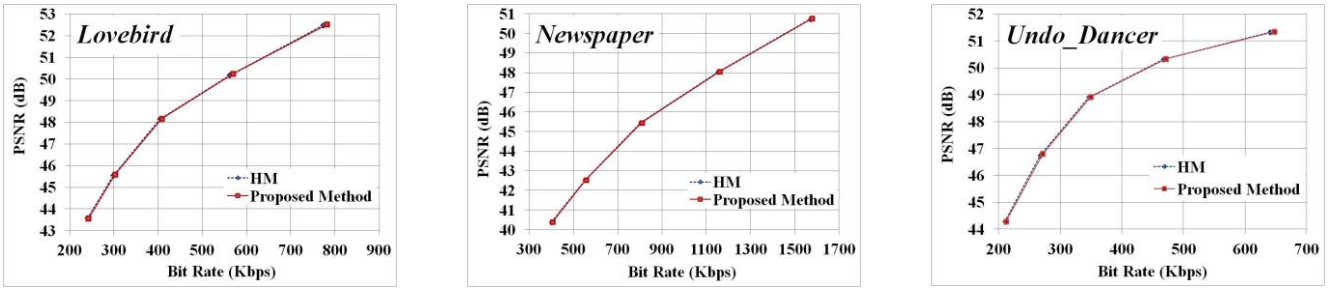
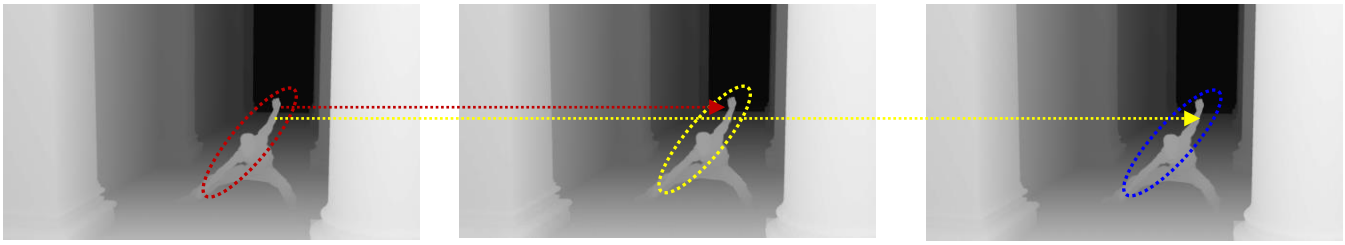


Fig. 8. Comparative study on RD performance by HM and the proposed method for a wide range of bit-rates.



(a) Original image of *Undo_Dancer* sequence (b) HM reproduced image (c) Proposed method reproduced image

Fig. 9. Subjective quality assessment for HM12.1 and the proposed method for *Undo_Dancer* video sequence. The figures are achieved from the 10th frame of the *Undo_Dancer* video at the same bit-rate.