A NOVEL DEPTH EDGE PRIORITIZATION BASED CODING TECHNIQUE TO BOOST-UP HEVC PERFORMANCE

Pallab 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
In addition to the texture, multiview video employees the utilization of depth coding for the reconstruction of 3D video and Free viewpoint video. Standing on some texture-depth correlations, a number of methods in literature reuses texture motion vector for the corresponding depth coding to reduce encoding time by avoiding costly motion estimation process. However, texture similarity metric is not always equivalent to the corresponding depth similarity metric especially at edge levels. Since their approaches could not explicitly detect and encode acute edge motions of depth objects, eventually, could not reach the similar or improved rate-distortion (RD) performance against High Efficiency Video Coding (HEVC) reference test model (HM). With a view to more accurate motion detection and modeling, the proposed technique exploits an extra Pattern Mode comprising a group of pattern templates (GPTs) with different rectangular and non-rectangular object shapes and edges compared to the existing HEVC block partitioning modes. Selecting an extra mode i.e., the Pattern mode, it can avoid further exploration of modes at next depth levels, thereby reducing encoding time by avoiding further branching. Moreover, the proposed Pattern Mode only encodes the motion areas and skips the background areas. The experimental results show that the proposed technique could save 30% encoding time and improve average 0.1dB Bjontegard Delta peak signal-to-noise ratio (BD-PSNR) compared to the HM.

Index Terms— HEVC, Depth edge, Motion estimation, Pattern mode, Intermode selection.

1. INTRODUCTION
Multiview video employs the utilization of both texture and depth video information from different angles to create a 3D video [1][2] and free viewpoint video (FVV) [24] which is gradually becoming more popular for its advanced visual experience with depth perception [3][4]. Unlike texture, depth video is a gray scale map that represents the distance between the camera and 3D points in a scene [21]-[23]. Compared to the traditional texture video coding, the depth video incur with extensive burden in terms of detecting and encoding its edges [5]. Compared to the reference texture image in Fig. 1 (a), the appeared depth image in Fig. 1 (b) demonstrates misalignment particularly in the edge like areas due to the occurrence of irregular motion patterns (e.g. red square). Therefore, different techniques which use the texture motion vector information for the corresponding depth video coding suffer from proper texture-depth misalignment issue that require more bits to compensate large residuals. This may also reduce interactivity among views and interrupts real-time operations. These constraints limit a number of electronic devices with limited processing capacity to use 3D video and FVV features.

![Gray scale image of Newspaper sequence](image1)

![Corresponding depth image of Newspaper sequence](image2)

Fig. 1. Distinction of a depth image from its gray scale presentation. In (b), the red square denotes an example of irregular motion patterns at edges.

Many researchers have introduced different approaches of depth video compression to fasten the encoding process [6]-[8]. To reduce computational time of depth video coding, Shen et al. [9] incorporate an adaptive motion search range determination and a fast mode decision algorithm using the prediction mode and the motion vector correlation of texture and depth video. Compared to the original H.264 JMVC encoder, they save 60% average encoding time by sacrificing 0.08dB PSNR with 0.6% bit-rate increment. To speed-up the mode decision process, Yeh et al. [10] attempt to reduce candidate modes by analyzing the RD cost of previously encoded view and determining a threshold for each of the modes in the current view. Compared to JMVC 4.0, their simulation results reveal a reduction of 76.65% average encoding time by sacrificing 0.07dB PSNR and 0.26% bit-rates. Lei et al. [11] propose a FMD method by evaluating inter-view and inter-component coding correlations and activating different early termination strategies for anchor and non-anchor frames. Although this process reduce 78.07% encoding time, simulation results also reveal that it incurs with the quality loss of 0.06 dB and bit-rate increment of 0.79% on average.

The above mentioned depth coding algorithms in the existing literature are developed for different test model versions of H.264 standard (i.e., JM). Compared to the state-of-the-art H.264 standard, the latest HEVC video coding...
standard [12][13] offers about double data compression ratio at the same level of video quality. However, its encoding complexity increases four times due to achieve this performance gain [14]. For the HEVC test model (HM), the motion prediction mode for a block is decided by exhaustively checking all the modes in one or more coding depth levels and minimizing their Lagrangian cost function (LCF) [15]. The equation for LCF \( j(m) \) is defined as
\[
j(m) = D(m) + \lambda \times R(m)
\]
where \( D(m) \) means the distortion, \( \lambda \) is the Lagrangian multiplier (LM) for the mode selection, \( R(m) \) denote the resultant bits required for encoding the block. Mode selection approach of HM consumes much higher encoding time as the depth level increases. Thus, the techniques that are developed based on different versions of JM, could not be straightforward applied to different versions of the HM due to (i) three times extended number of modes, (ii) coding unit (CU) size extension from 16x16 up to 64x64-pixel, (iii) complex block partitioning structures, and (iv) other advanced parameter settings in the HM.

Depth video compression by computational time reduction in the HEVC video coding standard is recently proposed by Podder et al. [16]. The prime limitation with this approach is the motion detection system which may be suitable for detecting rectangular object motions only. Since this detection technique fails to recognize the depth motions with rough and irregular edges, it sacrifices 0.08dB PSNR on average.

Since motion detection is the underlying criteria for mode selection, for more accurate motion detection and modeling, the proposed technique explicitly captures different aspects of object motions by exploiting the GPTs [17]. The pattern templates are designed considering a wide variety of rectangular and non-rectangular object shapes and edges, which are different from the existing HEVC block partitioning modes, to capture both regular and irregular object motions. Initial subset of modes is selected based on this motion criteria. From the selected subset, only the final mode is decided from the least value of the LCF. Since the patterns are designed to encode only the foreground by skipping background areas, it could also avoid the necessity of using extra bits. By introducing and selecting an extra mode i.e. the Pattern mode, the proposed technique can avoid further exploration of modes at next depth levels, thereby reducing encoding time by avoiding further branching. Moreover, adopting the strategy of encoding only the motion blocks, it could eventually reduce a substantial percentage of encoding time compared to the HM.

Since the Pattern mode is selected by detecting complex edge motions and beating all other modes in a block, the proposed technique could improve the RD performance. Moreover, while selecting the Pattern mode, as the larger blocks are represented by smaller blocks for finer level motion estimation and appropriate mode selection, it could also improve the RD performance compared to the HM.

Developing an independent depth coding framework (based on HEVC), it improves the interactivity within views by avoiding texture-depth misalignment issue. Therefore, all these features become more suitable for those low processing capacity based electronic devices to use 3D video and FVV.

### 2. PROPOSED TECHNIQUE

In the proposed coding technique, we use the CU size comprising with 64x64-pixels, and similar to the HM, we exhaustively encode all modes at that level using the LCF. Once any 32x32 level mode is selected, then we apply the proposed phase correlation technique due to reduce the computational time from that level to the next levels i.e., 16x16 and 8x8. The used phase correlation is a Fourier Transformation based approach to estimate the relative translational displacement between current block and the motion compensated block in the reference frame. Using this phase correlation based energy concentration ratio (ECR) feature, the proposed technique determines whether a block has no-motion, single-motion, or multiple-motions by applying a pre-defined thresholding procedure (discussed in Section 2.4). The phase correlation peak notifies us about the amplitudes of these motions. Mode selection is then performed based on the classified motion feature. The entire process is shown as a process diagram in Fig. 2.

![Fig. 2. Entire process diagram of the proposed mode selection technique](image)

#### 2.1. Coding with Pattern Templates

In the proposed coding technique, a real-time pattern selection algorithm is incorporated to select the best pattern for a depth image block from the codebook of predefined GPTs. The algorithm for template selection uses a piecewise pixel similarity measure and the selection of the best pattern for a block mainly focuses on jointly using the relevance and similarity metric. The relevance metric makes the selection process faster, whereas, the similarity metric targets to the image quality. The algorithm also uses a novel technique to control the size of the codebook within predefined bounds, to adapt computational complexity of the pattern selection process, thereby ensuring real-time operations. Further detail of the template selection could be found in [18]. Fig. 3 shows 32 different shaped patterns comprising with 64-pixels which are defined in 16x16 blocks. Thus a 32x32 block should have four 16x16 sub-blocks, thereby having 4 templates for motion detection. The white and black regions inside the templates denote the presence and absence of motions.
respectively. We exploit these diversified templates especially for more accurate motion detection and mode selection.

![GPTs of 32 different shaped, 64-pixel patterns](image)

**Fig. 3.** GPTs of 32 different shaped, 64-pixel patterns, defined in 16×16 blocks; while the term “P” stands for pattern.

### 2.2. Motion Calculation and Classification

Due to calculate the phase correlation, we first apply the *Fast Fourier Transform* (FFT) and then *inverse FFT* (IFFT) of the current and reference blocks and finally apply the FFTSHIFT function as follows:

$$P_c = F[1(e^{j\langle F_r, F_c \rangle})]$$  \hspace{1cm} (2)

where $F$ and $I$ means the FFTSHIFT and IFFT respectively, $F_r$ and $F_c$ are the Fast Fourier transformed blocks of the current block $C$ and reference block $R$ respectively. The *phase correlation peak* ($\Theta$) from the position of $(dx + \Omega/2 + 1, dy + \Omega/2 + 1)$ is calculated by:

$$\Theta = P_c(dx + \frac{\Omega}{2} + 1, dy + \frac{\Omega}{2} + 1)$$  \hspace{1cm} (3)

where the blocksize denoted by $\Omega$ is 32 since 32×32-pixel block is used by the proposed technique to calculate phase correlation. Using the phase of the current block and magnitude of the motion-compensated reference block, we calculate the matched reference block ($\mu$) by:

$$\mu = \left[ I \left( |F_r| e^{j\langle F_r, F_c \rangle} \right) \right]$$  \hspace{1cm} (4)

We then perform $\mu$-C process to calculate the displacement error ($\Phi$). Due to determine the ECR (i.e., $\alpha$), we finally apply the *discrete cosine transform* (DCT) to error $\Phi$, by calculating the ratio from the top-left triangle energy (i.e., $P_{\alpha}$) with respect to the total area energy (i.e., $P_{T}$) by: $\alpha = (P_{\alpha}/P_{T})$. If the calculated value of $\alpha$ is greater than the predefined Threshold1 (TH1), motion type is tagged by “complex-motion”, else if the value of $\alpha$ is greater than the predefined Threshold2 (TH2), motion type is tagged by “simple-motion”, otherwise motion type is tagged by “no-motion”.

Fig. 4 (a) shows the difference between 10th and 11th frame of *Newspaper* sequence where the blocks with Red, Purple, and Blue indicate different aspects of motions. From the whole frame, we just consider the Red, Purple and Blue blocks at (8, 10), (8, 7), and (4, 8) positions to exemplify the existence of complex, simple and no-motion respectively in those blocks. Experimentally obtained values of ECR for these blocks are presented in Fig. 4 (b) which shows the highest value for complex motion (0.89) and lowest for simple motion (0.27). To display how these motions look like, the Phase shifted plots of $\Theta$ for the complex, simple and no-motion of those positions are illustrated in (c), (d), and (e) of Fig. 4. It is clearly observed from the figure that the values of ‘$\alpha$’ has a positive correlation with motion, while the ‘$\Theta$’ has an inverse correlation with motion.

![Difference between 10th and 11th frame of Newspaper video](image)

**Fig. 4.** Illustration of different kinds of motion obtained at different blocks of 11th frame on *Newspaper* video; in (b), we plot the values of ECR for the blocks at (8, 10), (8, 7), and (4, 8) positions which indicate complex, simple, and no-motion blocks respectively; (c), (d), and (e) show the phase shifted plotted values for the respective blocks.

### 2.3. Intermode Selection

Like the HM, we perform exhaustive mode selection at 64×64 level. Once 32×32 level mode is selected, then we activate the proposed mode selection feature to determine a subset of inter-modes at 32×32 to 8×8 level. In the proposed scheme, the correlation between motion classification and a subset of mode selection is very high. Motion is classified by analyzing video contents, and since the mode selection process is executed from the categorized motion, thus, the confidence of selecting the best partitioning mode is also very high.

![Mode selection from Classified Motion](image)

**Fig. 5.** A sub-set of inter-mode selection by the proposed technique based on dissimilar motion types. $i$, $i$, and $P_i$ stand for Inter, intra, and Pattern mode respectively.
Both for 32×32 and 16×16 level, we ensure the use of pattern mode especially focusing on different shapes of depth objects edge motions. In the proposed coding structure, the LM is multiplied by 4 in the pattern mode selection process. The rationality of this multiplication is not only to adjust weight HM but also restrict the selection of Pattern mode for those blocks in which motions are not properly aligned with the pattern templates. This could also avoid the necessity of using extra bits for coding with the Pattern mode. From the selected sub-set of modes, the final mode is determined by using the lowest value of the LCF.

2.4. Threshold Specification

We observe the trend, if the bit-rate is increased and the thresholds are kept stationary, numerous regions in the moving object may not be classified as motion blocks. This trend motivates us to use dynamic thresholds. Therefore, we first derive thresholds against entire range of quantization parameters (QPs) used in the HEVC and then test the proposed technique for the whole range of QPs using these thresholds. We observe them to fit with the proposed algorithm properly and obtain improved RD performance. The range of QPs and their corresponding values of TH1 and TH2 are presented in Fig. 6 (a). From all these QPs, we just select six popularly used sample QPs and their corresponding threshold values which is presented in Fig. 6 (b). We approximate the evaluation TH1 and Th2 by non-linear functions where QPs are used as independent variables. The approximation of these thresholds are developed by:

\[
TH1 = 0.0445 \times \sqrt{QP} + 0.60 \quad (5)
\]

\[
TH2 = 0.0225 \times \sqrt{QP} + 0.45 \quad (6)
\]

3. EXPERIMENTAL RESULTS AND ANALYSIS

To evaluate the efficiency of the proposed coding technique, we perform experiments on six popularly used depth sequences. The sequences with resolutions 1920×1088 (W×H) are GT_Fly, Poznan_Street, Poznan_Hall, while, the sequences with resolutions 1024×768 (W×H) are Newspaper, Lovebird1, and Balloons. For performance evaluation, we first compare the proposed method results with the HM and then compare the produced results with existing five recent and most relevant state-of-the-art methods (shown in Table 2). The test platform used for the experiments is a 64-bit Microsoft Windows 7 operating system running on a dedicated desktop machine with Intel Core i7 CPU of 3.33 GHz and 32-GB RAM. The proposed scheme and the HEVC with exhaustive mode selection scheme are developed based on the HM12.1 [19] under the common test conditions. Using a wide range of QPs (i.e. 20, 24, 28, 32, 36, and 40), the tested sequences are encoded with 25 frame rate and search range ±64 (horizontal and vertical). The performance is evaluation is carried out based on the BD-PSNR, BD-bit Rate (BD-BR) (using the procedure in [20]), and encoding time saving (ΔT).

3.1. Block Partitioning Mode Analysis

Fig. 7 shows the distributions of the block partitioning modes for the HM and proposed technique for the 11th frame of Newspaper sequence at QP=32. We first consider the blocks at (4, 8), (8, 7), and (8, 10) positions which encompass with no-motion, simple-motion and complex-motion respectively as shown in Fig. 4. For both techniques, we observe similar partitioning patterns for the blocks at (4, 8) and (8, 7) positions (blue and pink squares respectively) in Fig. 7 (a–b). However, for the remaining block at (8, 10) position (red square), the HM selects 16x16 level modes for partitioning. However, the proposed technique on the other hand, partitions that of the block with both 16x16 level mode and the Pattern mode. The partitioning structures of these two blocks are highlighted in Fig. 8.
3.2. Encoding Time Analysis

We demonstrate the computational time analysis of the proposed technique against the HM both at QP and video resolutions basis. The equation for the calculation of computational time saving ($T_s$) is given by:

$$T_s = \frac{(T_{HM} - T_{PRO})}{T_{HM}} \times 100\%$$

(7)

where $T_{HM}$ and $T_{PRO}$ mean the encoding time consumed by the HM and the proposed method respectively.

![Time Saving](image)

(a) Average time saving at QPs (b) Average time saving at resolution

Fig. 9. Time saving by the proposed technique compared to HM

The experimental results reveal that for a wide range of QPs, the proposed technique reduces 31.24% encoding time on average as is shown in Fig. 9 (a). For further analysis, (b) shows average time saving by video resolution basis and the proposed technique saves 29.53% encoding time on average compared to the HM. Time saving is noticed higher for the 1024x768 resolution videos. The reason for this time saving is not only avoiding the unnecessary branching but also adopting the strategy of encoding only the motion blocks.

3.3. RD Performance Analysis

The performance is evaluated for the range of QPs used in this experiment and the RD test results of two sequences (one from each type of resolutions) are demonstrated in Fig. 10. Compared to the HM, the obtained [min–max] PSNR difference for Newspaper and GT_Fly are [0.03–0.19]dB and [0.04–0.28]dB respectively which indicate relatively improved RD performance of the proposed technique.

![RD Performance](image)

Fig. 10. performance comparison by the HM and the proposed technique

As an additional analysis, Table 1 presents the comparison results of the proposed technique against the HM in terms of BD-PSNR and BD-BR, where ‘+’ and ‘−’ sign indicate the increment and decrement respectively. Over six different sequences, the proposed technique reveal 0.10dB BD-PSNR gain, while, decreasing 0.60% BD-BR on average compared to the mode selection approach in the HM12.1.

<table>
<thead>
<tr>
<th>Sequences</th>
<th>Resolutions</th>
<th>BD-PSNR (dB)</th>
<th>BD-BR (%)</th>
<th>$\Delta T_s$ (%)</th>
<th>Videos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newspaper</td>
<td>1024x768</td>
<td>+0.12</td>
<td>-0.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lovebird1</td>
<td>1024x768</td>
<td>+0.09</td>
<td>-0.61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Balloons</td>
<td>1024x768</td>
<td>+0.09</td>
<td>-0.58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GT_Fly</td>
<td>1920x1088</td>
<td>+0.11</td>
<td>-0.65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poznan_Street</td>
<td>1920x1088</td>
<td>+0.08</td>
<td>-0.51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poznan_Hall</td>
<td>1920x1088</td>
<td>+0.09</td>
<td>-0.56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall-average</td>
<td></td>
<td>+0.10</td>
<td>-0.60</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2 presents the performance comparison results of the proposed technique against five recent and relevant depth coding techniques. It can be seen that Lei’s method provides the superior performance compared to the existing state-of-the-art methods (i.e., −0.06dB BD-PSNR and +0.38% BD-BR). They also save 78.07% computational time on average compared to one of the implementations of H.264 (i.e., JMVC 8.5). It is mentioned in the Introduction that the techniques developed based on different versions of JM (i.e., based on H.264) could not be straight-forward applied to different versions of the HM (i.e., based on HEVC) due to a number of constraints. Although the encoding time saving of the proposed technique is lower than the state-of-the-art methods presented in Table 2, however, the proposed technique outperforms all these methods in terms of both improving the BD-PSNR and reducing the BD-BR.

3.4. Subjective Quality Evaluation

We represent an example in Fig. 11 for subjective quality test, where (a) denotes the original image of Newspaper sequence which is considered as the benchmark for this test. In the same figure, (b) and (c) indicate the images reproduced by the HM and proposed method respectively. For assessing the image quality, first, let us concentrate at the top corner of the newspaper in three images which are marked with Red, Green, and Yellow squares in (a), (b), and (c) respectively. From the red square of the original image in (a), we notice the very top of the newspaper having a Π shape with it (a kind of complex edge motions) which could be distinguished from the rest of the part of it within the square. Once the same image is reproduced by the HM, due to imperfect detection of edge motion in numerous places like the one in the Green square of (b), it could not generate any Π type shape at the top of the newspaper. However, since the proposed technique mainly targets to the image quality improvement by accurate motion modeling, it could preserve the Π shape at the top of the newspaper as shown by the Yellow square in Fig. 11 (c).
Such distinguishing approaches could also reflect on improving the overall image quality.

![Image 1](http://hevc.kw.bbc.co.uk/svn/jctvc-hm/)

(a) Original image of 
Newspaper sequence
(b) Image reproduced 
by the HM12.1
(c) Image reproduced 
by the proposed technique

Fig. 11. An example of subjective test outcome of the HM and proposed technique. The images are obtained at QP=28.

4. CONCLUSION

Different texture assisted depth coding methods in the existing literature incur with the quality loss due to imperfect detection of depth edge motions and also suffer from proper interactivity for the texture-depth misalignment issue. To address these limitations, in addition to the surface level motion detection, the proposed technique also incorporates a group of pattern templates for more explicit motion estimation at depth edge levels and partitioning those edge blocks by Pattern mode. By avoiding unnecessary branching and selectively encoding only the motion blocks, it could save 30% (26%–33%) average encoding time. Developing an independent depth coding framework, it not only improves on average 0.1dB (0.04~0.37) BD-PSNR with HM but also improves interactivity within views by avoiding texture-depth misalignment issue.

12. REFERENCES


