Fast Mode Decision in the HEVC Video Coding Standard by Exploiting Region with Dominated Motion and Saliency Features

Pallab Kanti Podder*, Manoranjan Paul* and Manzur Murshed†

*School of Computing & Mathematics, Charles Sturt University, Bathurst, Australia
{ppodder;mpaul}@csu.edu.au
†School of Information Technology, Federation University, VIC-3842, Australia
manzur.murshed@federation.edu.au

Abstract

The emerging High Efficiency Video Coding (HEVC) standard introduces a number of innovative and powerful coding tools to acquire better compression efficiency compared to its predecessor H.264. The computing time complexities have also increased multiple times for this acquisition. This paper employs a novel coding strategy to reduce the time complexity in HEVC encoder by efficient selection of appropriate block-partitioning modes based on human visual features. The proposed technique exploits human visual attention modelling-based saliency feature and phase correlation-based motion features. The features are innovatively combined through a fusion process by developing a content-based adaptive weighted cost function to determine the region with dominated motion/saliency (RDMS)-based binary pattern for the current block. The generated binary pattern is then compared with a codebook of predefined binary pattern templates aligned to the HEVC recommended block-partitioning to estimate a subset of inter-prediction modes. Without exhaustive exploration of all modes available in the HEVC standard, only the selected subset of modes are motion estimated and motion compensated for a particular coding unit. The experimental evaluation reveals that the proposed technique notably down-scales the computational time of the latest HEVC reference encoder by 38% while providing similar rate-distortion (RD) performance.

Keywords: HEVC, Motion Features, Saliency Feature, RDMS, Intermode Selection.
1. Introduction

The prime goal of the latest HEVC [1] video coding standard is to provide the same perceptual image quality compared to its antecedent H.264 [2] at approximately 50% bit-rate reduction for efficient transmission and storage of high volume video data [3]. HEVC achieves this improved performance at the cost of more than 4 times algorithmic complexity due to the extended number of levels and complex coding unit (CU) partitioning scheme compared to its predecessor H.264 in a particular implementation [4,5]. For this reason, any electronic devices with limited processing capacity could not fully exploit HEVC encoding and decoding features. This motivated us to reduce the computational time of HEVC encoder by appropriate selection of inter-prediction modes. For this to happen, only the RDMS in a video is taken into account that comprise with visually attentive area based saliency feature and phase correlation based motion features.

Through the mode selection process, many researchers have contributed in different ways to abate computational complexity of HEVC reference test model in terms of both inter and intra coding. To alleviate the computational complexity of HEVC encoder, Shen et al. [6] propose fast CU size decision and intra-mode decision approach at each coding depth level. They adopt the technique of avoiding a number of prediction modes not often used in the parent CUs in upper depth levels or spatially nearby CUs. Experimentally they can save on average 21% computational complexity of HEVC encoder with negligible loss of coding efficiency. Recently, the authors in [7] propose an effective CU size decision algorithm for HEVC intra prediction. The algorithm is developed in two aspects: (i) the early CU size decision with adaptive thresholds by analysing texture homogeneity, and (ii) for the large CU size, adopting the technique of avoiding the in-depth intra prediction. The simulation results demonstrate that their technique obtain a complexity reduction up to 67% although it incurs with the loss of 0.06 dB peak signal-to-noise ratio (PSNR) or 1.08% bit-rate increment. A fast prediction unit (PU) skip and split termination algorithm for HEVC intra-prediction is proposed by Lim et al [8]. The early skip, PU skip and PU split algorithms performs- immediate skipping of large PUs, skipping of full RD cost computation and terminating further PU splitting using the RD cost. The experimental results reveal that the proposed algorithm achieves a 44.05% time savings on average with similar RD performance.

Several authors have proposed heuristics for simplifying the partitioning decision for CUs, PUs, and transform units (TUs), aiming at decreasing the computational complexity of HEVC encoder through inter-mode selection [9-13]. Recently Shen et al. [14] introduce a TU size decision based early termination algorithm for HEVC encoders by using the Bayesian decision theory and the correlation between the variance of residual coefficients to reduce the number of candidate transform sizes for a given block. The experimental results confirm that their proposed algorithm is capable of saving 30–46% of transform processing complexity with some losses in coding efficiency. To encode the current CU block, Ahn et al. [15] utilizes the spatiotemporal correlation of HEVC encoders, where they make use of the sample-adaptive-offset parameters as the spatial encoding parameter, while the motion vectors, TU size, and coded block flag
information as the temporal encoding parameters to estimate texture and temporal complexity of a CU. Experimental results show 46% encoding time reduction on average with the bit-rate increment of 1.2%. The authors in [16] propose an early skip mode decision technique to reduce time complexity of HEVC encoder. The skip mode for a prediction unit is determined by calculating the rate-distortion cost of $2N \times 2N$ merge mode based on the estimated threshold. Once a mode is decided as a merge mode do not undergo further partitioning. Experimental results show that proposed method alleviate on average 28% encoding complexity of HEVC encoder with virtually no coding loss.

The authors in [16] propose an early skip mode decision technique to reduce time complexity of HEVC encoder. The skip mode for a prediction unit is determined by calculating the rate-distortion cost of $2N \times 2N$ merge mode based on the estimated threshold. Once a mode is decided as a merge mode do not undergo further partitioning. Experimental results show that proposed method alleviates on average 28% encoding complexity of HEVC encoder with virtually no coding loss.

The technique presented in [17] by Shen et al. jointly utilizes the inter-level correlation of quadtree structure and the spatiotemporal correlation to fasten the intermode decision for HEVC. From the existing correlations of prediction mode, motion vector, and RD cost of coding depth levels, they analyse the prediction mode distribution and coding correlation of adjacent CUs. They eventually propose three adaptive inter-mode decision techniques that results in 49%-52% computational complexity reduction with negligible loss of coding efficiency.

Hou et al. [18] recommend a RD cost based threshold to explore modes only in the higher level that results in 30% time savings with 0.5% quality loss. Vanne et al. [19] propose an efficient inter-mode decision scheme by finding the candidate modes of symmetric and asymmetric motion partition. The tested results reveals the reduction of HEVC encoder complexity by 31%-51% at the cost of 0.2%-1.3% bit-rate increment. Pan et al. [20] introduce an early MERGE mode decision algorithm to reduce computational complexity of HEVC encoder. Based on all zero block and motion information, they first apply MERGE mode for the root CUs then for the children CUs by mode selection correlation. They achieve 35% time savings with the bit-rate increment of 0.32%, and quality loss of 0.11 dB peak signal to noise ratio (PSNR). Shen et al. [21] introduce checking criteria based early termination method which selects 36% and 14% of modes at depth level ‘0’ and ‘3’ respectively. This process incurs with quality loss especially for sequences containing a large area with high motion activities although their algorithm saves around 30% of computing time.

In addition to the above mentioned mode selection algorithms based on HEVC video coding standard, other fast mode selection algorithms based on H.264 video coding standard are also available in the literature [22-24]. Paul et al. [25] extract energy concentration ratio (ECR) from phase correlation and employ it for modes selection process to alleviate encoding time in H.264 standard. The approach used in [25] would not be straightforward applied in HEVC to select the direct mode or a subset of inter-modes due to the three times extended number of modes, double/quadruple size of CUs, and complex (i.e., symmetric/asymmetric) CU partitioning patterns compared to H.264. Moreover, only the ECR based mode selection would not provide expected compression results in HEVC as it indicates only the residual error between current block and the motion-compensated reference block. It also unnecessarily uses smaller block-partitions while a block does not have any translational motion or provides high accurate predicted motion.

For more accurate decision on motion estimation (ME) and motion compensation (MC) modes, in this paper, we extract three motion features including ECR, phase correlation peak and motion vector information from phase correlation to
overcome the problem in [25]. Other than motion, we also extract visually attentive areas by exploiting the saliency feature as human visual system is also sensitive for different colours, brightness, and contrast in static areas. Thus, the RDMS is comprised with the combination of three motion features and one saliency feature. We incorporate them by developing an adaptive weighted cost function to actuate RDMS based binary pattern for the current block. The resultant binary pattern is then compared against a codebook of predefined binary pattern templates aligned to the HEVC recommended block-partitioning to estimate a subset of inter-modes. From the selected subset of inter-modes, the final mode is determined based on their lowest Lagrangian cost function. The proposed method not only reduces the computational time by appropriate selection of RDMS based ME and MC modes but also demonstrates the similar RD performance.

The major contributions of this paper are summarized as follows: (i) For the first time we exploit three motion features of phase correlation and exploit them for fast mode selection process in HEVC standard, (ii) We include the saliency in our algorithm and effectively use this feature for RDMS categorization, (iii) For the RDMS determination, we adaptively design the binary pattern templates based on the CU partitioning patterns in HEVC, and (iv) We develop a content-based adaptive weighted cost function through features fusion and innovatively derive weights for each feature adaptively.

The rest of the paper is organized as follows: Section 2 articulates working mechanism of recent HEVC, Section 3 describes the key steps of the proposed method; experimental results and discussions are evaluated in section 4, while section 5 is the conclusion of the paper.

### 2. Recent HEVC Analysis

Compared to the state-of-the-art H.264 standard, HEVC introduces inventive approaches including the CU size extension from 16×16 up to 64×64-pixels, variable size PU and TU, and the symmetric/asymmetric block partitioning phenomenon. To select a particular motion prediction mode, HM checks the Lagrangian cost function (LCF) [26] exhaustively using all modes in each coding depth level (level 64×64, 32×32, 16×16 and 8×8 are denoted as depth level 0, 1, 2, 3 respectively). The LCF, \( J_n \) for the \( n \)th mode selection is defined by:

\[
J_n = D_n + \lambda \times R_n
\]  

(1)

where \( \lambda \) is the Lagrangian multiplier, \( D \) is the distortion, and \( R \) is the resultant bit, which are determined by modes for a CU. In order to select the best partitioning mode in a coding depth level, the HM checks minimum 8 (i.e., 64×64, 64×48, 48×64, 64×32, 32×64, 16×64, 64×16, and 32×32), and maximum 24 inter-prediction modes (i.e., similar partitioning with smaller blocks from 32×32 to 8×8) with lowest LCF. This process is extremely time consuming due to the exploration of all modes in one or more coding depth levels. Unlike HEVC test model (HM12.1) [27], in the proposed technique, the selected CUs with RDMS are motion estimated and motion compensated with modes in the higher depth levels, on the other hand, the CUs without RDMS are motion estimated and motion compensated with modes in lower depth levels. Thus, we can avoid...
3. Proposed Technique

In the proposed encoding method, we use 64×64 CU size and encode all inter-modes at level 0 exhaustively using LCF similar to the HM. If 32×32 mode is selected then we use phase correlation to reduce the computational time from that level to higher level i.e., level 1 to 3. As the probability of selecting a 64×64 partition size for the video sequences with mid to lower range resolution is below 10%, we do not apply the proposed phase correlation strategy for level 0. The phase-correlation provides us the relative displacement information between current block and the reference block by Fast Fourier Transformation (FFT). In order to calculate an approximation of the corresponding motion between two blocks from the current and reference images, we apply the phase correlation technique. In this paper, we exploit phase correlation based three motion features including: (i) energy concentration ratio \((\alpha)\), (ii) predicted motion vector \((dx,dy)\), and (iii) phase correlation peak \((\beta)\) that focus on three different aspects of motions. Content based visual attentive portions are captured by saliency feature \((\gamma)\) of graph based visual saliency (GBVS) modeling. This GBVS is applied on the current image blocks so that it could calture the best salient salient points based on human visual perception and attention. Then we develop a cost function through a fusion process based on the RDMS using weighted average of the saliency values and the normalized motion features to determine a unified RDMS feature for the current block. Finally we configure the binary RDMS pattern from the unified RDMS features by applying threshold. 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. The final mode is determined based on the lowest value of the Lagrangian cost function. The entire process is shown as a process diagram in Fig. 1.
3.1 Motion Features Extraction

We calculate the phase correlation by applying the FFT and then inverse FFT (IFFT) of the current and reference blocks and finally applying the FFTSHIFT function as follows:

\[ \Omega = \| \Gamma \left( \Lambda \left( e^{j(\angle F_c - \angle F_r)} \right) \right) \| \]  

where \( F_c \) and \( F_r \) are the Fast Fourier transformed blocks of the current \( C \) and reference \( R \) blocks respectively, \( \Gamma \) is the FFTSHIFT, \( \Lambda \) denote the IFFT and \( \angle \) is the phase of the corresponding transformed block. Note that \( \Omega \) is a two dimensional matrix. We evaluate the phase correlation peak (\( \beta \)) from the position of \((dx + \mu/2 + 1, dy + \mu/2 + 1)\) as follows:

\[ \beta = \Omega(dx + \mu/2 + 1, dy + \mu/2 + 1) \]  

where the blocksize denoted by \( \mu \) is 8 if 8×8-pixel block is used for phase correlation. Then we compute the predicted motion vector \((dx, dy)\) by subtracting \( \mu - 1 \) from the \((x, y)\) position of \( \Omega \) where we find the maximum value of \( \Omega \). 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 and finally calculate the matched reference block (\( \Psi \)) for the current block by:

\[ \Psi = \| \Lambda \left( \left( \left| F_r \right| e^{j(\angle F_c)} \right) \right) \| \]  

Now the displacement error (\( \xi \)) is enumerated by:

\[ \xi = C - \Psi \]  

We then apply the discrete cosine transform (DCT) to error \( \xi \) and calculate the ECR (i.e., \( \alpha \)) as the ratio of low frequency component and the total energy of the error block (i.e., ratio from the top-left triangle energy with respect to the whole area energy) by:

\[ \alpha = \left( \frac{D_{EL}}{D_{ET}} \right) \]  

where \( D_{EL} \) and \( D_{ET} \) represent the top-left triangle energy and the whole area energy of a particular block. Note that the two sides of the top-left triangle are three-fourth of the blocksize i.e., 6 in our implementation.
3.2. Saliency Feature Extraction

Salient areas may be considered in an image with high motion, high resolution or coloured region. As an example, the authors in [28] exploit superpixel-based spatiotemporal correlation among video frames and detect temporal saliency by incorporating distinct motion information of superpixels with a scheme of saliency prediction in the temporal level. Thus, saliency map gives us the prominent and interesting regions in an image based on human visual perception and attention (visual attention based rate control algorithm is detailed in [29]). However, the leading models of visual saliency may consist of (i) extracting feature vectors (ii) forming an activation map using the feature vectors and (iii) normalizing the activation map [30,31]. We exploit the saliency map as a tool in this paper (based on [30,32]) and incorporate into our coding architecture. The exploited GBVS technique gives us the variance map of RDMS based human visual features for an 8×8-pixel block consisting of the values ranging from 0 to 1. For a given 8×8 block, we extract a salience feature, γ, by averaging all values. In the proposed scheme, we incorporate this map into the cost function to determine the visually attentive areas with/without motion. Therefore, this feature would effectively capture the visually sensitive regions and our focus is to encode the RDMS based salient portions with more bits not only to achieve better compression efficiency but also for coding performance improvement.

3.3. RDMS Categorization through Feature Fusion

After evaluating the phase correlation extracted motion features (i.e., α, β and (dx, dy) and saliency extracted variance map (i.e., γ), we finally determine a cost function. The development of the content-based adaptive weighted cost function for a block is carried out through a feature fusion process and denoted as follows:

\[
\Psi = \omega_1 \alpha + \omega_2 (1-\beta) + \omega_3 \left( \frac{|dx|}{\delta} + \frac{|dy|}{\delta} \right) + \omega_4 (\gamma)
\]

(7)

where \(\delta\) be the maximum block size, and \(\omega_1\) to \(\omega_4\) are the weights with \(\sum_{i=1}^{4} \omega_i = 1\). We innovatively derive weights for each feature adaptively and do not consider all possible weight-combinations in this experiment. We only consider 0.50, 0.25, 0.125, and 0.125 weights based on the relative texture deviation of the current block against that of the whole frame. To calculate the deviation we apply Standard Deviation (STD) both on the current block and the current frame and use those weights for four attributes. First we sort four features based on their values and if the value of the STD of the block is smaller than the value of the current frame then the highest weight (i.e., 0.50) is applied to the feature 1 (i.e., sorted) and the lowest weight (i.e., 0.125) is applied to the feature 4 (according to sorted list); otherwise, inverse weighted order is applied. If the resultant value of the cost function (i.e., \(\Psi\)) is greater than a predefined threshold the block is tagged by ‘1’ otherwise tagged by ‘0’ where ‘1’ and ‘0’ corresponds to RDMS and non-RDMS respectively.
The rationality of the proposed weight selection strategy is that if the current block has higher texture variation compared to the current frame, the current block should be encoded with more bits compared to the rest of the blocks to achieve similar/improved RD performance. To ensure spending more bits we need to categorize the block as RDMS block which is done by our threshold selection strategy. Other weight selection approach might work better, however, the experimental results show that the proposed technique provides similar RD performance (Fig. 12 and Table 5).

Fig. 2 demonstrates the relationship of the quantitative motion and salience features with the human visual features. Fig. 2 (b-d) shows the categories of motion-peak ($\beta$) and their corresponding values provided by ECR (in Fig. 2 (e)) and saliency feature (in Fig. 2 (f)) for Tennis video. The applied GBVS technique produces the resultant cost function based actual salience maps. These maps are generated between 11th and 12th frame on Tennis video for CU at positions (3, 1), (3, 10) and (5, 7) respectively with its texture deviation as illustrated in Fig. 3.

We level the complex texture and smooth texture areas by reddish and bluish colour respectively while any other colour corresponds to simple texture areas. From the figure and the experimental results, we can easily observe that the features $\alpha$, ($dx$, $dy$), and $\gamma$ have positive correlation and $\beta$ has inverse correlation to indicate human visual features.
Fig. 3. Salience feature after applying GBVS on three different image blocks (GBVS technique applied on the current image blocks) with different motions and texture of a frame identified in Fig. 2; note that for clear visualization we use 32×32-pixel block.

Fig. 4 illustrates the impact of saliency feature and from Fig. 4 (a), we clearly depict the ellipse like red marked area as the table tennis court edge (obviously a visually attentive area) that is precisely identified by the saliency feature (red marked ellipse like area in Fig. 4 (c)) although three motion features of phase correlation could not grasp the court edge as it does not have any motion (white marked ellipse like area in Fig. 4 (b)). Thus, the proposed combined strategy improves the RD performance by recognizing not only the RDMS-based motion features of phase correlation but also the RDMS-based visually attentive areas in the videos.

3.4. Intermode Selection

In the proposed scheme we use 64×64 as a CU size. The proposed scheme selects the best inter-mode at level 0 using LCF without using any phase correlation-based pre-processing. If 32×32 size inter-mode is selected then we activate the proposed RDMS feature to select a subset of inter-modes at level 1 to level 3. For the generation of binary matrix, we exploit each of the 8×8 pixel blocks from the
32×32 block and produce a matrix of 4×4 binary values \( S(x, y) \) in equation (8)) for each 32×32 (applying threshold). 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 (see Fig. 5 and Table 1). Each of the templates is constructed with a pattern of RDMS and non-RDMS block (1 and 0 respectively) focusing on the rectangular and regular object shapes at 32×32 block level. In the proposed technique, these adaptively designed binary pattern templates (BPTs) are fully aligned to the HEVC recommended block-partitioning to cover all the modes at 32×32 level.

The best-matched \( j \)-th BPT is selected from all BPTs as follows:

\[
T_j = \arg \min_{\forall T_k \in \text{BPT}} \left( H_k \right).
\]

Table 1. Mode selection at 32×32 block level based on the codebook of predefined binary pattern templates.
The mode selection process from the BPTs at 32×32 and 16×16 coding depth levels is illustrated in Table 1 and Table 2 respectively. Once a particular template selects a subset of candidate modes at 32×32 level, only the final mode is decided by their lowest Lagrangian cost function.

Table 2. Subset of mode selection at 16×16 and 8×8 coding depth level based on the patterns of RDMS

<table>
<thead>
<tr>
<th>Patterns of RDMS at 16×16 and 8×8 Block Level</th>
<th>Selected Subset of Inter-modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Pattern Image]</td>
<td>16×8, 8×16 or 8×8</td>
</tr>
<tr>
<td>![Pattern Image]</td>
<td>16×8 and 8×16</td>
</tr>
<tr>
<td>![Pattern Image]</td>
<td>16×12, 16×4, 12×16 or 4×16</td>
</tr>
</tbody>
</table>

Again, at 32×32 level, if any of the 16×16 level mode is selected, we further explore smaller modes including 8×8 at 16×16 coding depth level based on different RDMS block patterns as demonstrated in Table 2 (i.e., presence or absence of binary 1 and 0 in Table 2). Then we calculate the minimum value of the Lagrangian cost function (demonstrated in Eq. 1) to determine the final mode.
3.5. Threshed Selection

In lieu of selecting thresholds dynamically, we apply the static threshold value and fix it by 0.15 in order to decrease the threshold selection complexity although we consider both the homogeneous and heterogeneous motion regions in the CUs of all sequences. Moreover, the Th (i.e., 0.15) used in the proposed experiment is properly fitted with a variety of test sequences and test conditions. To investigate the implication of the Th used in our algorithm, we apply it for the similar range of QPs recommended by JCT-VC (i.e., 22, 27, 32 and 37) for different sequences and notice that this Th value can appropriately capture the RDMS based regions with accuracy even for a wide range of video contents, different aspects of object motions, camera motion and resolutions. The acquired RDMS based blocks are then encoded with more bits in order to achieve improved quality. To illustrate the effect of using the Th value 0.15, consider the following Fig. 6. It is obvious from the figure that for Th=0.15, the proposed algorithm can detect motion in a more realistic way which is the underlying criteria for mode selection. The reddish blocks in Fig. 6 (b~d) present the motion and Fig. 6 (c) shows more real scenario of motion detection than Fig. 6 (b) or (d) in terms of alignment with the moving regions for the original image of Tennis sequence in Fig. 6 (a)). Since this approach of motion detection is observed for almost all sequences, we fix the value of Th by 0.15.

(a) Original image of Tennis sequence

(3b) When values of Th = 0.08 (c) When value of Th = 0.15 (d) When values of Th = 0.22

Fig. 6. The representation of motion map for various Th values at QP=28. The reddish blocks in (b~d) indicate the motion regions and the bluish blocks indicate the non-motion regions.

Due to the imbalanced distribution of ECR values Paul et al. [25] use a range of thresholds for different bit-rates. Those thresholds could not perform well in the proposed method as the distribution of cost function values is more compact and almost in all cases, it exceeds the value of 0.15 in all types of sequences for the blocks with motion and dominant salience. Therefore, in the proposed technique, we use the fixed value of threshold (i.e., 0.15) for a wide range of bit-rates.
4. Experimental Results and Analysis

To verify the performance of the proposed algorithm, experimental results are presented with twelve standard video sequences (SVS). These include: six standard definition (SD - with resolution (288×352)) videos, four high definition (HD - with resolution (1088×1920)) videos, and two multiview (MV - with resolution (480×640)) videos. The sequence names and their corresponding BD-PSNR and BD-Bit Rate (compared to HM) results are presented in Table 5. We also test other video coding experts' recommended standard class video sequences (SCVS) that include Class A, B, C, D, and E. The names and resolutions of these test sequences and their corresponding BD-PSNR and BD-Bit Rate (compared to HM) results are presented in Table 6. The sequences taken for the experiment are representative in the sense ranging from low motion activity to the high motion activity, representing a wide range of contents, different aspects of camera motions and resolutions. We compare the proposed method results with HM of HEVC standard as HM outperforms all the existing mode selection techniques stated in the literature.

In order to validate the effectiveness of the proposed fast coding scheme, we implement the proposed method into HM12.1 and test it under the common test conditions. RD performance of both schemes are compared considering the maximum CU size of 64×64 by enabling both symmetric and asymmetric partitioning block size of 64×64 to 8×8 depth levels. Each CU is set as a 64×64 pixel block and we only apply RDMS based technique when a 32×32 inter-mode is selected. We use 8×8 pixel blocks for binary matrix generation in the 32×32 block (i.e., a 32×32 block is comprised with 4×4 binary values) for a wide range of bit-rates (using QP=20, 24, 28, 32 and 36).

Table 3 shows the performance comparison results of five recent and most relevant fast mode selection algorithms where all the algorithms obtain significant computational time savings compared to different implementations of HEVC by increasing bit-rates and reducing PSNR. The Table confirms that the proposed technique improves the performance in terms of both reducing encoding time and improving rate-distortion performance compared to the existing state-of-the-art methods. Although the time savings for the proposed method is a bit lower than the method used in [9], the bit-rate reduction and PSNR gain are more significant. Compared to the method used in [15], the proposed technique shows the improved results in terms of time savings and PSNR reduction, however, it achieves the higher bit-rate savings. The overall results of Table 3 also reveals that the proposed method outperforms HM12.1 by saving 34% computational time on average with insignificant bit rate increment (i.e., 0.13%) and negligible PSNR reduction (i.e., 0.01dB). The terms BD-BR and BD-PSNR in Table 3 denote BD-Bit Rate and BD-PSNR respectively.

Table 3. Performance comparison of different fast mode selection algorithms compared to HEVC encoder in terms of BD-BR, BD-PSNR and computational time.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>BD-BR (%)</th>
<th>BD-PSNR (dB)</th>
<th>Average Time Savings (%)</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correa et</td>
<td>5.70</td>
<td>-0.80</td>
<td>50</td>
<td>03 videos</td>
</tr>
</tbody>
</table>
4.1. Experimental Setup

In this paper, the experiments are conducted by a dedicated desktop machine (with Intel core i7 3770 CPU @ 3.4 GHz, 16 GB RAM and 1TB HDD) running 64 bit Windows operating system. The proposed scheme and HEVC with exhaustive mode selection scheme are developed based on the reference software HM (version 12.1[27]). In order to validate the effectiveness of the proposed fast coding scheme, we implement the proposed method into HM12.1 and test it under the common test conditions. RD performance of both schemes are compared considering the maximum CU size of 64×64 by enabling both symmetric and asymmetric partitioning block size of 64×64 to 8×8 depth levels. Each CU is set as a 64×64 pixel block and we only apply FMSSR technique when a 32×32 intermode is selected. We use 8×8 pixel blocks for binary matrix generation in the 32×32 block (i.e., a 32×32 block is comprised with 4×4 binary values) for a wide range of bit-rates using QPs: 20, 24, 28, 32 and 36 (i.e., similar range of QPs recommended by JCT-VC). All the sequences are encoded with 25 frame rate and search length ±64 (horizontal and vertical). We use IPPP… format with Group of picture (GOP) 32 for both techniques and two reference frames to encode a P-frame for both techniques. The performance is evaluated based on the difference of coding time (ΔT_s), the BD-PSNR, and BD-BR. BD-PSNR and BD-BR are calculated according to [33].

4.2. Results and Discussions

If any method exhaustively checks all the options in a level to select a particular option, theoretically it should necessitate more computational time. This complexity increases multiple times if any technique has to explore all modes in more depth levels to select a particular mode. Therefore, for all sequences, we compare the average number of modes selected in each CU by HM12.1 and the proposed method. The results in Table 4 shows that HM checks more options in all cases and normally requires more computational time. From Table 4, the overall average percentage of encoding time savings for the SVS by the proposed method is 39.85, while for the SCVS, this percentage is 41.44. The reason behind this acquisition is the efficient subset of intermode selection with simple criteria. However, we cannot ignore the pre-processing stages of the proposed method and by calculation we find that over twelve sequences on average 6.71% extra
encoding time is required to execute phase correlation and saliency related pre-
processing overheads (see Fig. 1). Thus, theoretically we anticipate to acquire
33.14% and 34.73% computational time savings on average for SVS and SCVS
respectively.

<p>| Table 4. A theoretical analysis - percentage of time savings by the proposed method (against HM) for different sequence types based on average no. of Inter-modes selected per CU. |</p>
<table>
<thead>
<tr>
<th>Name of the Sequences</th>
<th>Average no. of inter-modes selected per CU by HM</th>
<th>Average no. of inter-modes selected per CU by proposed method</th>
<th>Average Percentage (%) of time saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tennis</td>
<td>20.15</td>
<td>11.07</td>
<td>45.06</td>
</tr>
<tr>
<td>Paris</td>
<td>18.05</td>
<td>10.26</td>
<td>43.15</td>
</tr>
<tr>
<td>Average time saving for SD type video</td>
<td></td>
<td></td>
<td>44.10</td>
</tr>
<tr>
<td>Pedestrian</td>
<td>20.68</td>
<td>12.34</td>
<td>40.32</td>
</tr>
<tr>
<td>Parkrun</td>
<td>19.21</td>
<td>11.39</td>
<td>40.70</td>
</tr>
<tr>
<td>Average time saving for HD type video</td>
<td></td>
<td></td>
<td>40.51</td>
</tr>
<tr>
<td>Exit</td>
<td>20.66</td>
<td>13.16</td>
<td>36.30</td>
</tr>
<tr>
<td>Ballroom</td>
<td>18.12</td>
<td>12.04</td>
<td>33.55</td>
</tr>
<tr>
<td>Average time saving for MV type video</td>
<td></td>
<td></td>
<td>34.92</td>
</tr>
<tr>
<td>Class-A</td>
<td>21.05</td>
<td>12.23</td>
<td>41.90</td>
</tr>
<tr>
<td>Class- B</td>
<td>18.56</td>
<td>12.18</td>
<td>34.37</td>
</tr>
<tr>
<td>Class-C</td>
<td>19.78</td>
<td>11.29</td>
<td>42.92</td>
</tr>
<tr>
<td>Class-D</td>
<td>18.96</td>
<td>10.13</td>
<td>46.57</td>
</tr>
<tr>
<td>Average time saving for Class type videos</td>
<td></td>
<td></td>
<td>41.44</td>
</tr>
</tbody>
</table>

Fig. 7. Illustration of overall time savings by the proposed method against HM at different bit-rates for SVS.

Fig. 8. Illustration of overall time savings by the proposed method.
The experimental evaluation reveals that over twelve different sequences, and for a wide range of bit-rates, the proposed method reduces on average 32.21% (range: 28%-38%) overall computational time for SVS as shown in Fig. 7. For the SCVS, we experimentally achieve 34.14% (range: 27%-39%) overall encoding time savings on average that is shown in in Fig. 8. The equation for the time savings \( \Delta T \) is defined as:

\[
\Delta T = \frac{(T_0 - T_p)}{T_0} \times 100\% \tag{10}
\]

where \( T_0 \) and \( T_p \) denote the total encoding time consumed by HM and the proposed method respectively. For comprehensive performance test, we execute the computational time analysis of both techniques based on video categories for SVS, and find that the proposed method achieves on average 34% encoding time savings compared to HM12.1 as shown in Fig. 9 (a). For the SCVS, this percentage is 35 which is shown in Fig. 9 (b). The figure also reveals that the proposed technique saves more time for SD video type compared to HD or MV.

![Fig. 9. Illustration of overall average time savings by the proposed method against HM based on different video categories. In (a), the numbers of videos used in HD, MV and SD are 4, 2 and 6 respectively, while in (b), the numbers of Class A, B, C, D and E are 1, 4, 2, 4, and 3 respectively.](image)

Fig. 10 shows the average percentage of three different depth level modes for twelve sequences at QP=20 to 36. For the proposed method, the higher depth level modes (16×16 and 8×8) are exploited for the RDMS based CUs due to acquire similar/improved RD performance. Fig. 11 reveals the block partitioning example for the 12th frame of Tennis video at QP=24 by both schemes. If we compare Fig. 11 (a) and (b) we observe an identical block partitioning framework for CU at (3, 1) position (dark block) and (3, 10) position (pink block) in both methods. Both techniques consider these blocks with no motion and simple motion respectively (justified by Fig. 2) and thus, partition accordingly. However, the partitioning evidence of yellow marked CUs are distinguishable from each other.
First, we consider the CU at (5, 7) position in Fig. 2 and observe the complex motion therein. HEVC could not grasp any motion in that block and keep it unpartitioned, while, the proposed technique considers it as a complex motion block (Fig. 2 (d)) with the the highest ECR and saliency values (Fig. 2 (e)-(f)). Therefore, the proposed technique partitions that block with higher level modes. This approach of RDMS based mode selection reflects on achieving the similar RD performance for a wide range of bit-rates. In Fig. 11, the blocks with red partitions indicate that at least one 8×8 level mode is selected for partitioning that CU.

Fig. 10. HM and proposed method based average mode selection at three coding depth levels

Fig. 11. Block partitioning for the 12th frame of the Tennis video at QP=24 with HM and the Proposed method.
To test the performance of the proposed method objectively, we first compare the RD performances against HM using three sequences (one SD, HD and MV) for a wide range of bit-rates as demonstrated in Fig. 12. In Fig. 12, the proposed method shows similar RD performance especially caring about the RDMS based CUs and partitioning those CUs with appropriate depth level modes. We also observe the similar results for other five class sequences (*Traffic*, *BQTerrace*, *BQMall*, *BlowingBubbles*, and *Vidyo4* - i.e., one from each class type). Table 5 and Table 6 present the overall performance comparison results of the proposed technique for twelve different SVS and fourteen different SCVS respectively. The results reveal that compared to the mode selection approach in HM12.1, the proposed technique achieves an almost similar RD performance (small average reduction of 0.01dB and 0.01 dB PSNR for SVS and SCVS respectively) with a negligible bit-rate increment of 0.13% and 0.10% for SVS and SCVS respectively.

Table 5. Performance comparison of proposed technique compared to HM12.1 using BD-bit rate and BD-PSNR for SVS

<table>
<thead>
<tr>
<th>Sequence Resolutions</th>
<th>Name of the Sequences</th>
<th>BD-PSNR (dB)</th>
<th>BD-BR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>288×352</td>
<td>Tennis</td>
<td>-0.01</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>Silent</td>
<td>-0.01</td>
<td>0.14</td>
</tr>
<tr>
<td></td>
<td>Waterfall</td>
<td>-0.01</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>Tempete</td>
<td>-0.01</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>Paris</td>
<td>-0.01</td>
<td>0.10</td>
</tr>
<tr>
<td>Class</td>
<td>Sequence Resolutions</td>
<td>Name of the Sequences</td>
<td>BD-PSNR (dB)</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------</td>
<td>------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>A</td>
<td>1600×2560</td>
<td>Traffic</td>
<td>-0.01</td>
</tr>
<tr>
<td>B</td>
<td>1080×1920</td>
<td>BasketballDrive</td>
<td>-0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>QTerrace</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cactus</td>
<td>-0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tennis</td>
<td>0.00</td>
</tr>
<tr>
<td>C</td>
<td>480×832</td>
<td>BasketballDrill</td>
<td>-0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BQMall</td>
<td>0.00</td>
</tr>
<tr>
<td>D</td>
<td>240×416</td>
<td>BasketballPass</td>
<td>-0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BlowingBubbles</td>
<td>-0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BQSquare</td>
<td>-0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Flowervase</td>
<td>0.00</td>
</tr>
<tr>
<td>E</td>
<td>720×1280</td>
<td>Vidyo1</td>
<td>-0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vidyo3</td>
<td>-0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vidyo4</td>
<td>-0.01</td>
</tr>
<tr>
<td>Overall Average</td>
<td></td>
<td></td>
<td>-0.01</td>
</tr>
</tbody>
</table>

Table 6. Performance comparison of proposed technique compared to HM12.1 using BD-bit rate and BD-PSNR for the class video sequences for SCVS.

(a) Original image of Tennis sequence
4.3. Subjective Quality Assessment

It is widely accepted that objective assessment based on PSNR does not always provide reliable assessments of video quality, since a higher PSNR may not always promise better video quality [34]. Therefore, it has become common practice in international coding-standard activities to combine both objective and subjective assessments in evaluating and comparing video coding algorithms. To compare the perceptual performance, we perform a subjective test using Double-Stimulus Continuous Quality Scale (DSCQS) assessment using the test conditions of [35]. We used a number of video sequences serially to accomplish this test. The viewers are asked to rate the quality of two video sequences known as “H” and “P” on a continuous scale ranging between “Excellent” and “Bad”. The options “H” and “P” are the reconstructed video sequences using HM12.1 and the proposed algorithm. The experimental results reveal that viewers recognize and mark them as very similar.

As an example, Fig. 13 (a) shows the original image of Tennis video taken for subjective quality test and Fig. 13 (b) and Fig. 13 (c) illustrate the reproduced images by HM and the proposed method respectively. To present the comparison in image quality let us concentrate on the cuff and sleeve sections of the shirt 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. The figures (Fig. 13 (b) and (c)) are achieved from the 20th frame of the Tennis video at the same QP (i.e., 24) and for both techniques, we notice the same bits per frame (52508) and PSNR (41.52) values. It was presented earlier in this manuscript that the proposed method requires less encoding time. Hence it can be concluded that the proposed technique shows significant computational time savings compared to HM12.1 with similar image quality for a wide range of bit-rates. Due to this phenomenon, the proposed implementation is expected to become more suitable for all real time video coding applications especially for a number of electronic devices with limited processing power and battery capacity.

5. Conclusion

In this work we present a novel coding framework for the performance improvement of HEVC encoder. This is implemented by exploiting RDMS based mode selection technique comprising phase correlation based three different motion features and human visual attention based saliency feature. The motion features focus on three different aspects of motions in the CUs and the saliency feature captures visual attentive areas. We formulate a content based adaptive
weighted cost function through feature fusion to determine a subset of inter-modes using predefined RDMS based binary pattern templates. The Lagrangian optimization criterion is set only on the selected subset to fix the final mode. Compared to the mode selection approach in HM12.1, the proposed scheme reduces on average 38% computational time (ranging 32%-42%) while providing similar RD performance (reduction of 0.01dB PSNR or, 0.13% bit-rate increment) for a wide range of bit-rates.

References


27. Joint Collaborative Team on Video Coding (JCT-VC), HM Software Manual, CVS server


