IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 20, 2018, accepted November 20, 2018, date of publication November 29, 2018,
date of current version December 27, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2883967

Efficient Video Coding Using Visual Sensitive Information for HEVC Coding Standard

PALLAB KANTI PODDER[1], (Student Member, IEEE),
MANORANJAN PAUL[2], (Senior Member, IEEE),
AND MANZUR MURSHED[3], (Senior Member, IEEE)

[1]Department of Information and Communication Engineering, Pabna University of Science and Technology, Pabna 6600, Bangladesh
[2]School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW 2795, Australia
[3]School of Science, Engineering, and Information Technology, Federation University, Churchill, VIC 3842, Australia

Corresponding author: Manoranjan Paul (mapul@csu.edu.au)

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

ABSTRACT The latest high efficiency video coding (HEVC) standard introduces a large number of inter-mode block partitioning modes. The HEVC reference test model (HM) uses partially exhaustive tree-structured mode selection, which still explores a large number of prediction unit (PU) modes for a coding unit (CU). This impacts on encoding time rise which deprives a number of electronic devices having limited processing resources to use various features of HEVC. By analyzing the homogeneity, residual, and different statistical correlation among modes, many researchers speed-up the encoding process through the number of PU mode reduction. However, these approaches could not demonstrate the similar rate-distortion (RD) performance with the HM due to their dependency on existing Lagrangian cost function (LCF) within the HEVC framework. In this paper, to avoid the complete dependency on LCF in the initial phase, we exploit visual sensitive foreground motion and spatial salient metric (FMSSM) in a block. To capture its motion and saliency features, we use the dynamic background and visual saliency modeling, respectively. According to the FMSSM values, a subset of PU modes is then explored for encoding the CU. This preprocessing phase is independent from the existing LCF. As the proposed coding technique further reduces the number of PU modes using two simple criteria (i.e., motion and saliency), it outperforms the HM in terms of encoding time reduction. As it also encodes the uncovered and static background areas using the dynamic background frame as a substituted reference frame, it does not sacrifice quality. Tested results reveal that the proposed method achieves 32% average encoding time reduction of the HM without any quality loss for a wide range of videos.

INDEX TERMS Background modeling, fast mode decision, FMSSM, foreground motion, HEVC, motion estimation, spatial saliency.

I. INTRODUCTION

The *High Efficiency Video Coding* (HEVC) standard [1]–[3] is the next-generation compression technology lauded as the enabler for a host of new services and capabilities. The ultimate goal of this standard is to ensure similar perceived video quality with its predecessor H.264 [4] at approximately 50% bit-rate decrease for the proficient broadcasting and storage of large volume video data. The important share of the coding performance improvement is the adoption of the large number of *motion estimation* (ME) and *motion compensation* (MC) inter-modes in the HEVC. The *HEVC reference test model* (HM), an implementation of HEVC recommendation, uses the tree-structured hierarchical mode selection approach. In **FIGURE 1**, let us first assume the block partitioning structure at coding depth levels 64×64, 32×32, 16×16 and 8×8 by the levels 0, 1, 2, and 3 respectively. Now we notice how the 8×8 mode selection is carried out by the HM at higher depth level. If 64×64 is the *coding unit* (CU), then the *prediction unit* (PU) modes at level-0 are explored. Once 32×32 is selected as the smallest PU mode from this level, it then further explores smaller modes at level-1. Once again, if 16×16 is selected as the smallest PU mode from this level, only then it could explore all the modes at level-2 to select the final 8×8 mode which is hierarchically shown in **FIGURE 1** by the golden borders and blue circles with its associated dotted lines. To pick the best partitioning

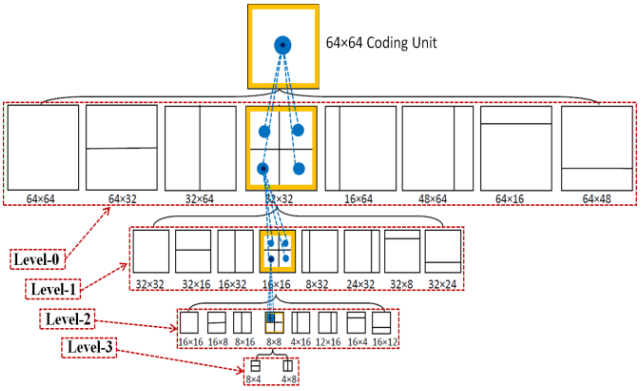


FIGURE 1. Hierarchical mode selection approach of the HM to select 8×8 mode as an example.

mode at any coding depth level, HM thus, tests out at least 8 (i.e. all modes at level-0), and at most 24 PU inter-modes (i.e. similar partitioning for smaller blocks from level-1 to level-3) using the least value of *Lagrangian cost function* (LCF). We denote this procedure as a partially-exhaustive mode selection approach of the HM which incurs several times computational complexity increment [5], [6] compared to the H.264. On the other hand, if any approach is fully-exhaustive in nature, for all CUs it should check all the 24 PU modes for the final mode selection. Although the HM mode selection is partially-exhaustive in nature, it still explores a large number of PU modes to select the final mode for a CU based on the LCF. The LCF creates a unique criterion by adding distortion with bit requirements by multiplying *Lagrangian Multiplier* (LM) [7]. According to **FIGURE 1**, to select the 8×8 mode using the existing LCF the HM requires maximum 22 times ME & MC for a block. This results in highly increased encoding time that affects a number of electronic devices with limited processing to utilize the HEVC encoding and decoding features in real-time.

To decrease the HM encoding time complexity through mode reduction, the existing mode decision algorithms [8]–[15] select a subset of inter-modes based on the properties of homogeneity, residual and statistical correlation among different coding depth level modes (to be detailed in Section II). Standing on the aforementioned analysis and mode selection strategy, the procedures in the literature mostly depend on the existing LCF within the HEVC framework. Therefore their approaches reduce encoding time, however, could not achieve similar or relatively improved *rate distortion* (RD) performance compared to the HM.

Merely the existing LCF dependent mode selection would not yield the best RD performance in terms of subjective and objective video quality for various operational coding points due to more complex CU partitioning patterns, extended number of modes, coding length of motion vectors and the inclusion of other advanced parameters in the HEVC standard. The LCF- $j(m)$ for mode selection is defined by:

$$j(m) = D(m) + \lambda \times R(m) \tag{1}$$

where D stands for the *sum of squared differences* between an original block and its reconstructed block which is gained through the original block coding using candidate mode m; λ is the LM for mode selection; $R(m)$ denote the number of bits required for encoding the block with m. Normally the equation for the LM is calculated with an empirical formula using the selected *quantization parameter* (QP) for each block by:

$$\lambda = \alpha \times W_k \times 2^{((QP-12)/3)} \tag{2}$$

here the values of α depends on the following parameters:

$$\alpha = \begin{cases} 1.0 - Clip3(0.0, 0.5, 0.05 * num\_of\_B\_frames \\ 1 \end{cases} \tag{3}$$

where the value 1 is applicable for the non-referenced pictures only, while the other parameter settings are employed for the referenced pictures and the *Clip3* function clips the number of B frames to the range (0.0, 0.5). The value of W_k is 0.57 for the I type Slice, while for the GBP type Slice its values are 0.44 and 0.57 for the *random access* (RA) and *low delay* (LD) respectively (more detail to be found in [17]). Thus the value of α in equation (3) depends on various number of factors and the calculated value of λ in equation (2) is highly related to a variety of parameters set-up. Therefore, the overall execution of $j(m)$ in equation (1) also completely depends upon the diversified parameter settings. As a result, design an optimal LCF is almost impossible for different resolutions, various video contents, motion vector coding length, and many other advanced configuration settings in HEVC for actual coding.

Paul *et al.* [18] observe that RD performance could be varied by using different LM values for which they modified the mode selection strategy [19] in the H.264 using the *energy concentration ratio* (ECR) of phase correlation. However, the best compression results could not be obtained using only the ECR feature as it completely depends on the residual error and unreasonably applies smaller block splitting although a block does not have any translational motion but have some textured residue due to quantization. It also requires a large number of extra bits to encode background areas of numerous non-motion blocks. Thus, the direct application of Paul's approach could not improve the overall RD performance of the HM.

The main hypothesis of the proposed coding technique is to divide the mode selection scheme into two distinguishing phases. In the first phase, a sub-set of PU modes are selected by using motion and saliency features without requiring the necessity of existing LCF. In the second phase, the LCF is only employed to determine the final mode from the initially selected subset of PU modes in the first phase.

As motion prediction is the underlying criterion of mode decision, in this test, we use *dynamic background modeling* (DBM- to be explained in Section III-A) for more explicit motion prediction in foreground. The prime limitation of the

traditional frame difference approach (TFDA) is its ability to capture only the object boundary motion areas. However, we observe that more explicit motion information can be obtained by subtracting the current frame from the background frame as this image difference more precisely intensifies the pixel locations that have been changed between two frames by exploring the uncovered areas. Moreover, there is no match of uncovered areas in the current frame with the previous frames but the *most common frame in a scene* (McFIS- obtained by using DBM) provides the exact match. Therefore, in this work, the uncovered background areas are better encoded by using the McFIS as a reference frame and a significant amount of residual errors could be reduced for those uncovered areas which eventually decreases the bit-rate. This could also save encoding time since it requires no motion estimation, thus, there is no latency issue for the encoder and decoder.

For better motion modeling, the proposed coding scheme exploits three motion features from phase correlation i.e. ECR, phase correlation peak, and the predicted motion vector. The phase correlation peak indicates how accurately we can estimate the motion from the reference block, the predicted motion vector indicates the translational displacement length and ECR indicates residual error. Thus, combining these three features can provide better motion information of a block. Other than motion, the human visual system is sensitive to contrast/brightness in static areas as well. To capture visual information, the proposed scheme exploits the *graph based visual saliency*(GBVS) modelling as a tool which could provide higher values for the blocks having dominant salience. Since the GBVS captures the salient information in spatial domain, it will be combined with the motion features in temporal domain by developing an adaptive weighted cost function to form a criterion termed as *foreground motion and spatial salient metric* (FMSSM). The FMSSM is used to categorize a block as a *visual attentive block* (VAB- assigned as '1') or *non-visual attentive block* (NAB- assigned as '0') based on a predefined threshold. A subset of inter-modes is selected by the arrangement of '1' and '0' blocks in a CU against predefined binary pattern templates. Since the features of proposed FMSSM are comprised of two most predictive parameters (i.e. motion and saliency) of eye movements [20] and responsive to the human visual system for quality assessment, our motivation is to encode the VABs (that have higher FMSSM value) with relatively higher level modes for better quality and the rest of the NABs with lower level modes for faster coding. This preprocessing phase for a subset of PU mode selection in the proposed technique is fully independent from the existing LCF. The LCF is employed to determine only the final mode from the selected subset of PU modes in the first phase (see **FIGURE 2** and Section III-E).

To summarize, the encoding time savings of the proposed method is due the following reasons: (i) it directly performs the mode selection process (to be explained in Section III-E); (ii) it checks/hits fewer number of modes compared to the HM to select the final mode in a block (to be explained Section IV-B); (iii) it uses simple FMSSM criteria to select a subset of PU modes. In contrast, as a byproduct, the proposed technique can achieve a minor improvement in RD performance compared to the HM due to the following reasons: (i) the strategy is to better encode the uncovered background areas by using the McFIS as a reference. Thus, lots of residual errors could be reduced for those uncovered areas; (ii) the DBM explored partitioning modes for uncovered motion block would be different from HM to obtain an improving RD performance; (iii) unlike HM, the proposed technique adopts the content aware FMSSM criteria for initial subset of PU mode selection by using the LCF independent preprocessing phase. This requirement of encoding time saving without affecting coding loss impact would be important for a number of electronic devices with limited processing and computational resources to use different features of the HEVC standard.

The major contributions of this work can be summarized below: (i) for detecting and employing the foreground motions from the video contents, the DBM technique is implemented; (ii) various aspects of motion are captured using three motion features of phase correlation; (iii) the saliency feature is incorporated as an additional mode selection criterion; (iv) to determine the proposed FMSSM, the binary pattern templates are adaptively designed which are also aligned to the HEVC recommended block partitioning; and (v) a content aware weighted cost function is developed by feature synthesis where the weights for each feature are derived adaptively.

The remainder of the paper is structured as follows: Section II reviews the background study; Section III illustrates the key steps of the proposed coding scheme; Section IV provides detail discussions of the experimental outcomes; while Section V concludes the paper.

II. BACKGROUND REVIEW

Several approaches have been proposed for simplifying the partitioning result of CUs,*prediction units*(PUs), and *transform units*(TUs) to decrease the HM encoding time that mainly fall into inter-coding and intra-coding. Many researchers in the literature try to alleviate the encoding complexity of the HM using intra-prediction based fast approaches [21]–[23]. Recently, Lim *et al.* [24] introduce a fast PU skip and split termination algorithm by developing the early skip, PU skip, and PU split algorithms. These perform instant skipping of RD cost computations for large PUs, skip full RD cost calculation, and terminate further PU splitting using the RD cost respectively. Experimentally they could save 44.05% average encoding time of HM with similar RD performance. By investigating the RD cost and the *sum of absolute differences* for a given QP, Tariq *et al.* [25] propose a model to determine the RD cost of 35 intra modes using the quadratic relation. Test results show 34.5% average encoding time saving by sacrificing 0.99% bit-rate increment. However, normally the intra-prediction based coding requires more bits than the inter-prediction based approaches

and its efficiency highly depends on user specified modeling parameters [26].

Much research have been conducted in the area of encoding complexity reduction of the HM and to fasten it by reducing a number of inter-modes [27]–[29]. Vanne *et al.* [8] recommend a proficient inter-mode selection method by discovering the PU modes of symmetric and asymmetric motion division. Experimentally their presented approach reduces 31%-51% HEVC encoder complexity by sacrificing 0.2%-1.3% bit-rate increase. The developed approach in [9] by Shen *et al.* uses inter-level correlation of quadtree structure and spatiotemporal correlation to make HM intermode selection faster. In general, they finally recommend three adaptive mode selection methods which result 49%-52% computing complexity decrement by slightly sacrificing coding efficiency. Pan *et al.* [10] initiate an early MERGE mode decision based procedure to lessen computation intricacy of the HM. Based on all zero block and motion information, they initially apply MERGE mode for the root CUs, then for the children CUs using the correlation of mode selection. This approach saves 35% time by sacrificing 0.32% bit-rate increment and 0.11dB quality loss. Xiong *et al.* [11] present a pyramid motion divergence-based CU selection algorithm to fasten the inter-prediction procedure. This approach could save 40% encoding time although 2.21% bit-rate increases on average. To encode the current CU, recently Ahn *et al.* [12] explore spatiotemporal correlation of the HEVC encoders. They exploit 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 approximate texture and temporal complexity of a CU. Simulation results depict average 46% encoding time saving with the bit-rate increase of 1.2%. Shen *et al.* [13] introduce a TU size decision based early termination algorithm for HEVC encoders. They use the Bayesian decision theory and the correlation between the variance of residual coefficients to reduce the number of candidate transform size for a given block. The experimental results confirm that their proposed algorithm is capable of saving 30–46% transform processing complexity with some losses in coding efficiency. Correa *et al.* [14] introduce a set of procedures that are based on decision trees acquired through data mining techniques in order to come to a decision whether the block partitioning optimization algorithm should be terminated early or run to the end by using the exhaustive search approach for the best configuration. They eventually generate and implement three sets of decision trees to skip running the rate distortion optimization algorithm to its full extent. Their experimental outcomes reveal an average computational complexity reduction of 57% with 0.96% bit-rate increment. Lee *et al.* [15] introduce an early skip mode decision to reduce the encoding complexity of the HM without sacrificing its coding quality. This technique decides the skip mode by calculating the rate-distortion cost of 2N×2N merge mode. The prediction units decided to be partitioned by the skip mode do not undergo remaining mode decision processes. They could directly determine the user friendly threshold for the early mode decision from the video data itself. Compared to the HM, the technique presented by them reduces 30.1% and 26.4% average encoding complexity in random access and low-delay condition respectively with virtually no coding loss.

The content property analysis based fast motion estimation was introduced by Pan *et al.* [16] in which they adopt the strategy of selecting the best motion vector correlation among different size PU modes. This process suffered from the computational complexity during estimating the PU modes at higher bit-rates, and also for the high definition videos having complex motion areas. Result shows that compared to HM12.0, they saved on average 12.29% encoding time while sacrificing 0.03dB BD-PSNR and 0.86% BD-BR increment for RA condition. This process saves an average of 15.04% encoding time, while sacrificing 0.02dB BD-PSNR and 0.55% BD-BR increment at LD test condition. As the above mentioned methods used existing LCF framework, they reduce computational time, however, they could not improve the RD performance compared to the HM.

III. PROPOSED TECHNIQUE

In the proposed coding scheme, like the HM, we use 64×64 CU size and encode all intermodes at level-0 using LCF. Once 32×32 level mode is selected, then the phase correlation based preprocessing is activated to reduce encoding time from that level to higher levels i.e. level-1 to 3. Since the likelihood of selecting a 64×64 partition size for the video sequences with mid to lower range resolution is below 10%, we skip the implementation of proposed phase correlation strategy for level-0. In this work, we apply the phase correlation technique to calculate motion approximation between two blocks of the current and reference images i.e. McFISes. We exploit three motion features from phase correlation including (i) predicted motion vector (dx, dy), phase correlation peak (β) and (iii) ECR ($\mathcal{R}$) that focus on three dissimilar aspects of motions.

To capture visual attentive portions of video contents, we use the saliency feature (σ) of GBVS modeling. These features are then innovatively synthesized by evolving an adaptive weighted cost function to determine the FMSSM based binary pattern for the current block. The generated patterns are then compared with the predefined templates aligned to the HEVC suggested block partitioning and the best fitted template is considered for a subset of PU mode decision. The LCF is applied only on selected subset to decide the final mode. The whole process is presented as a process diagram in **FIGURE 2**.

A. CONTENT BASED BACKGROUND MODELLING

To detect the background frame as a reference frame, we carry out the DBM technique that initially goes through a learning process (online) on a number of already encoded frames for collecting video content information where a pixel may be considered as a part of various objects and background over

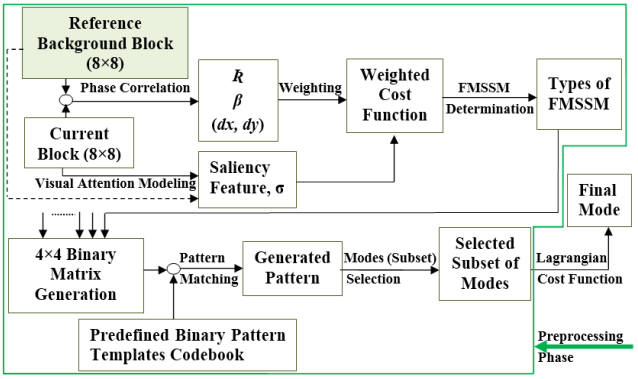
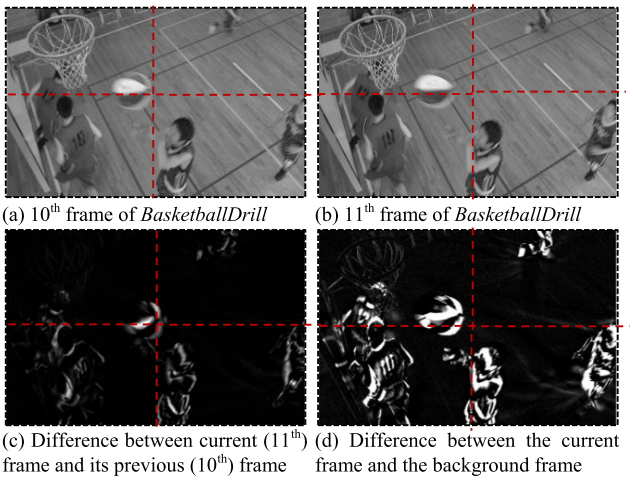


FIGURE 2. **Process diagram of the proposed mode selection technique.**



(a) 10th frame of *BasketballDrill* (b) 11th frame of *BasketballDrill*

(c) Difference between current (11th) frame and its previous (10th) frame (d) Difference between the current frame and the background frame

FIGURE 3. **Background modeling is more reactive to motion features compared to the traditional frame difference approach- (i.e. current frame and its previous one) the pictorial presentation. The image difference using *BasketballDrill* sequence presented in (c) and (d) is multiplied by six (6) for better visualization.**

time. To represent each part, a number of Gaussian models are developed to mould a pixel over time and each of the models is expressed by pixel intensity variance, weight and means [30]. A model having low variance and large weight is assumed to be the most stable background. The mean value of the best background model is taken as background pixel intensity. To speed up the learning rates where minimum number of frames are required for DBM, Haque *et al.* [31] use a parameter called the *recentVal* to store recent pixel intensity value with predefined condition. Paul *et al.* [32] argue that the intensities of *mean*and *recentVal* are two extreme factors to produce actual background intensity for efficient video coding. Therefore, in this work, we use a weighting factor between the *mean*and *recentVal*to decrease the delay response (because of the *mean*) and to fasten the learning rates (because of the *recentVal*) as recommended in [32]. More detail procedure of DBM is explained in [32] and [33].

FIGURE 3 shows the effectiveness of using the DBM for better motion modeling in which (a) and (b) present two successive frames of *BasketballDrill* sequence; (c) and (d) indicate the motion (i.e. whitish areas) captured

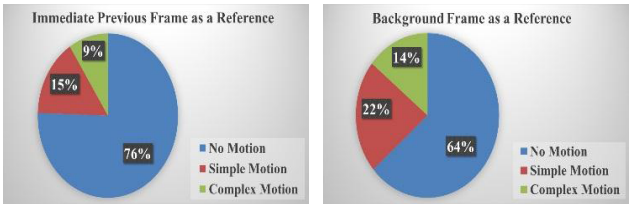


FIGURE 4. **Distribution of different aspects of motion for the 11th frame of *BasketballDrill* sequence. The higher percentage of motion can be captured by using the McFIS as a reference frame.**

by the TFDA and DBM respectively. It is therefore clear from **FIGURE 3** (d) that the uncovered areas (appeared with precise motion) can be better coded once the McFIS is used as a reference frame. Now **FIGURE 4** illustrates the motion distribution for the 11th frame of the *BasketballDrill* sequence that is used in **FIGURE 3**. Compared to the TFDA, 7% more simple-motion and 5% more complex-motion has been detected by using the McFIS as a reference. Thus, both the boundary and surface level motions of moving object can be captured eventually to improve the blockiness issue in the decoded image.

B. CALCULATION OF MOTION FEATURES

We figure out the phase correlation by applying the *Fast Fourier Transform* (FFT) and then *inverse* FFT (IFFT) of the current and reference blocks (from the instant McFIS) and finally applying the FFTSHIFT function as follows:

$$\Omega = \Gamma\left(\left|\emptyset\left(e^{j(\angle\eta - \angle\delta)}\right)\right|\right) \tag{4}$$

where Γ and $\emptyset$ denote the FFTSHIFT and IFFT respectively, δ and η present the Fast Fourier transformed blocks of the current C blocks and reference R blocks respectively and $\angle$ represents the phase of the equivalent transformed block. The calculated Ω is a two dimensional matrix. From the position of $(dx + \varphi/2 + 1, dy + \varphi/2 + 1)$, now we calculate the phase correlation peak (β) as follows:

$$\beta = \Omega(dx + \frac{\varphi}{2} + 1, dy + \frac{\varphi}{2} + 1) \tag{5}$$

the blocksize indicated by φ is 8 in the equation (5) since 8×8-pixel block is employed by the proposed algorithm to estimate the phase correlation. Then we compute the predicted motion vector (dx, dy) by subtracting φ-1 from the (x, y) position of Ω to detect the maximum value of Ω. Using phase of the current block and magnitude of the motion-compensated reference block, we finally calculate the matched reference block (μ) for the current block by:

$$\mu = \left|\emptyset\left(|\eta|\, e^{j(\angle\delta)}\right)\right|. \tag{6}$$

The displacement error (τ) is calculated by:

$$\tau = C - \mu. \tag{7}$$

The *discrete cosine transformation* (DCT) is then applied to error τ to evaluate the ECR (i.e. $\mathcal{R}$) using the ratio of low

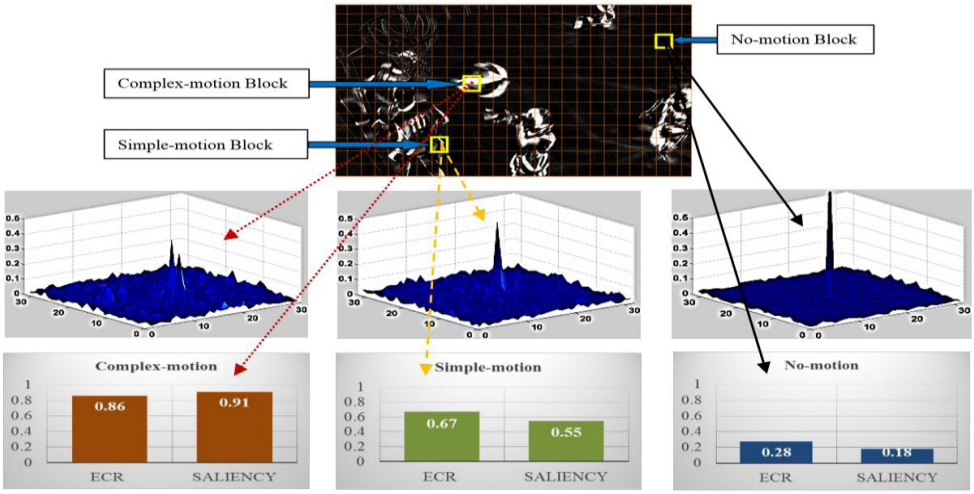


FIGURE 5. Illustration of motion and salient feature values obtained at different blocks of 11th frame for *BasketballDrill* sequence; in the first row, the current frame is subtracted from the background frame; in the second row, the phase shifted plots (i.e. β) for multiple/complex motion (0.21), simple motion (0.36), and no motion (0.62) are presented for the 32×32 blocks at positions (8,10), (13,8) and (4,24) respectively (yellow rectangles); for the same blocks, in the third row, the ECR (i.e. $\mathcal{R}$) and Saliency feature (i.e. σ) generated respective values are shown. The 32×32 block size is demonstrated for better visualization.

frequency component with respect to the whole energy of the error block (i.e. ratio of the amount of top-left triangle energy (i.e. ∇_L) and the amount of entire area energy (i.e. ∇_W)) by:

$$\mathcal{R} = (\nabla_L / \nabla_W). \tag{8}$$

The two sides of the top-left triangle are three-fourth of the blocksize which is 6-pixels in the proposed implementation.

C. CALCULATION OF VISUAL SALIENCY FEATURE

The saliency mapping could provide the significant and attractive regions in a video according to the human visual perception and concentration. The leading models of visual saliency may consist of (i) the extraction of feature vectors at locations over the image plane; (ii) structuring an activation map(s) by extracted feature vectors; and (iii) normalization of the activation map [34], [35]. For more appropriate salient location detection and visual attention modeling, in this work, the GBVS modeling is applied on the current frames of video streams. The GBVS is exploited to obtain the variance map for an 8×8 pixel block consisting of the values that range from 0 to 1 where '0' indicates no saliency and '1' indicates the highest saliency. Once the saliency mapping is carried out, the average of the saliency values for the current 8×8 block is used as a feature for FMSSM calculation. Since the GBVS captures salient information in spatial domain, it will be fused with the motion features in temporal domain. The rationale of selecting the GBVS modeling is its simplicity and having an appearance of ground truth using high saliency regions which are most likely to be found in a scene. More detail about the GBVS modeling could be found in [36].

D. FMSSM DETERMINATION BY FEATURE FUSION

FIGURE 5 exhibits the relationship between the quantitative motion as well as the salient features with the human visual features. The first row of **FIGURE 5** shows the difference between current frame (i.e. 10th) and the background frame for *BasketballDrill*, the second row shows the values of the motion peak (i.e. β), while the third row shows the values of the ECR (i.e. $\mathcal{R}$) and saliency (i.e. σ) for the blocks at (8,10), (13,8) and (4,24) positions respectively. It is clearly observed that the values of β are inversely proportional with motion i.e. it has high value for no/little motion block and small value for complex motion block, while values of $\mathcal{R}$ and σ are proportional to the motion. As the *dx*, *dy*,and β present motion displacement and ECR presents amount of residual error, the combination of these three should provide better motion classification compared to the ECR alone. Moreover, combining saliency feature with the motion feature provides better block categorization in terms of human visual attentive areas. We develop the content aware adaptive weighted cost function for a block by a feature fusion process as follows:

$$\mathfrak{L}_W = \omega_1 \mathcal{R} + \omega_2 (1 - \beta) + \omega_3 \left(\frac{|dx|}{\varphi} + \frac{|dy|}{\varphi} \right) + \omega_4(\sigma) \tag{9}$$

where ω_1 to ω_4 are the weights with $\sum_{i=1}^{4} \omega_i = 1$ and φ is the blocksize i.e. 8×8 in this experiment. We originate weights for each feature adaptively and take into account only four weight combinations: $\omega_{\{i=1,2,..,4\}} = \{0.50, 0.25, 0.125, \textit{and } 0.125\}$.

The weights are distributed considering the relative texture divergence of the current block against that of the entire frame. The deviation of both of the current block and the current frame is calculated by the *standard deviation* (STD)

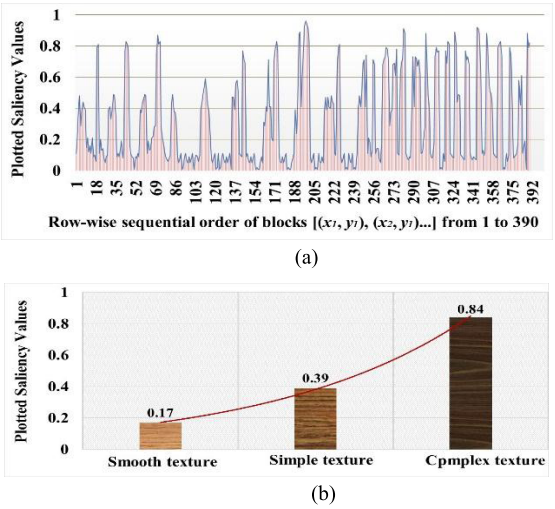


FIGURE 6. Plotted saliency values for all the blocks of 11th frame (i.e. 832×480 resolution and representing them with 32×32 blocks, it is calculated 26 columns × 15 rows = 390 blocks) of *BasketballDrill* and their average value for different textured contents. (a) Saliency plots for all the blocks of 11th frame using *BasketballDrill*. (b) Calculated average of saliency values for the Smooth, Simple, and Complex textured blocks of 11th frame using *BasketballDrill*.

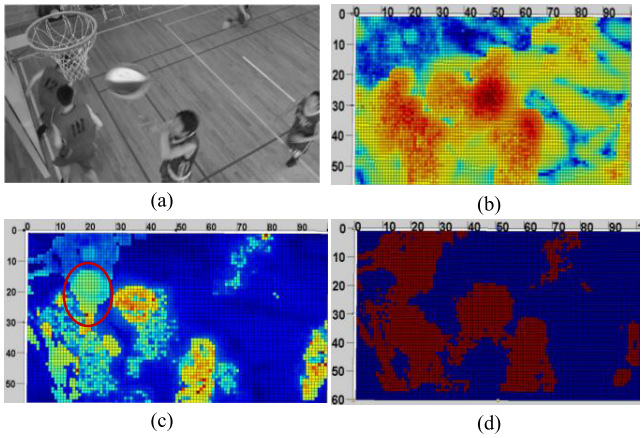


FIGURE 7. The VABs and NABs categorization using FMSSM. (a) Original image taken from 10th frame of *BasketballDrill* sequence. (b) Combined values of motion features using equal weight. (c) FMSSM values using motion and saliency features. (d) VABs (reddish) and NABs (bluish) after thresholding.

for those weights of four features. First we sort the features based on their values and if the STD value of a block becomes less compared to the current frame value, then the highest weight (i.e. 0.50) is applied to the first feature in sorted order and the lowest weight (i.e. 0.125) is applied to the last feature. Otherwise, inverse weighted order is applied. For a 8×8 block within a 32×32 block, '1' is assigned if the corresponding value of FMSSM i.e. $\mathfrak{L}_W$ is greater than the previously defined threshold θ_t; otherwise it is assigned '0' where '1' indicates the VABs and '0' indicates the NABs. The discussion of the threshold, θ_t and its implication is described in Section III- F.

The rationale of such weight distribution strategy is that if the current block has higher texture deviation than the current frame, the current block should be encoded with more bits compared to the rest of the blocks to obtain improved RD performance. To ensure spending few more bits we first categorize the blocks as VABs that have higher FMSSM values which is done by thresholding. Other weight selection procedures may perform better; however, the experimental outcomes reveal that the proposed approach did not sacrifice the RD performance as reported in **FIGURE 16** and TABLE 5.

FIGURE 6 (a) shows the GBVS applied saliency plots for all the blocks of 11th frame on *BasketballDrill* sequence. Since *BasketballDrill's* resolution is 832×480 and representing them with 32×32 blocks, it is calculated 26 columns ×15 rows = 390 blocks. The calculated average salient values for the blocks having smooth, simple and complex textured contents are presented in **FIGURE 6** (b) which clearly indicates that the GBVS value for the complex textured region is much higher compared to other textured regions. The graph also reports that saliency values are proportional to the texture complexity.

FIGURE 7 shows VABs and NABs classification from the video contents using FMSSM where (a) presents the original image of *BasketballDrill* and other than the moving objects, it also contains some visually important areas without motion (e.g. the basket itself is static). So only the motion features are not often sufficient to identify the stationary region which is illustrated in **FIGURE 7** (b). However, the visual saliency feature could successfully recognize the basket like still areas as circled by red and presented in **FIGURE 7** (c). Their combined contributions as shown in **FIGURE 7** (d) reveal the FMSSM classified VABs (reddish) that are encoded with relatively higher-level modes for better quality and the rest of the NABs (bluish) with lower level modes for faster coding.

E. INTER-MODE SELECTION

Like HM, the proposed coding technique uses 64×64 as a CU size and selects the best mode at level-0 using the LCF. It activates the FMSSM criteria from the 32×32 level to select a subset of modes at level-1 to level-3. As mentioned in Section I that the HM requires maximum 22 times ME and MC to encode a block using 8×8 mode. However, for doing this, the proposed technique requires maximum 12 times ME and MC by using the codebook of predefined pattern templates aligned to HEVC recommended block partitioning as analyzed below.

For producing binary matrix, we exploit the 8×8 pixel block from the 32×32 block and for each 32×32, we generate a matrix of 4×4 binary values (i.e. $M(x, y)$ in equation (10)) by applying threshold. The cost function generated 4×4 binary matrix is then compared with the codebook of predefined *binary pattern templates* (BPTs) to select a subset of modes (shown in **FIGURE 8**, and to be shown in TABLE 1 and **FIGURE 9**). Each template in **FIGURE 8** is constructed with a pattern of VABs and NABs (1 and 0 respectively)

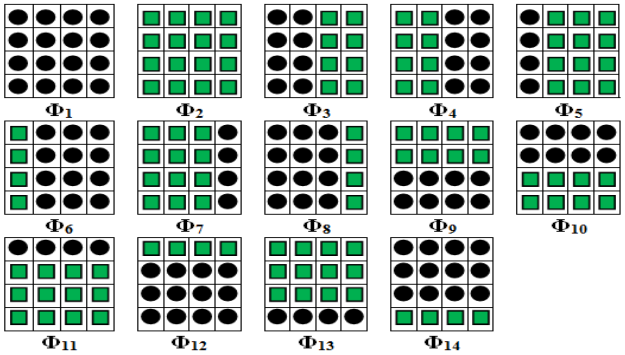


FIGURE 8. Codebook of the proposed predefined binary pattern templates (i.e. Φ1 to Φ14) with VABs (green squares) and NABs (black circles) to perform a subset of inter-mode selection.

focusing on the rectangular and regular object shapes. Both in **FIGURE 8** (for 32×32 level) and **FIGURE 9** (for 16×16 and 8×8 levels), the cells with green square present the VABs and black circle present the NABs. The rationale of keeping such structure similarity between the HEVC block partitioning and proposed template design is due to more appropriate PU mode selection by better motion modeling.

We use a similarity metric using the *Hamming distance* (DH) [37] between the phase correlation generated binary matrix of a 32×32 block and the BPTs presented in **FIGURE 8**. The best-matched BPT that provides the least *sum of the absolute* values of their differences is eventually selected. The DH, denoted by D_h is decided as follows where M is the binary VABs prediction matrix of size 4×4 comprising '1' or '0' combinations to represent a 32×32 block since each '1' or '0' represents VAB or NAB of a 8×8 sub-block within 32×32 and P_n is the n-th BPT:

$$D_h(x,y) = \sum_{x=0}^{4}\sum_{y=0}^{4} |M(x,y) - P_n(x,y)| \tag{10}$$

From all BPTs, the selection of the best-matched j-th BPT is carried out using the following equation:

$$P_j = arg \min_{\forall P_n \in BPT} (D_h) \tag{11}$$

TABLE 1 shows the proposed method's subset of PU mode selection process at 32×32 level where S_K, κ, and $\mathcal{K}$ denote skip, intra and inter modes respectively. Individual template(s) i.e. Φ_1 to Φ_{14} could select either a direct mode (e.g. 24×32 by Φ_{14}) or a subset of modes (e.g. 24×32 or 16×16 by Φ_{13}) where the selection of 16×16 is due to the frequency of more VABs. At 32×32 level, if any 16×16 level mode is selected, then smaller modes including 8×8 at 16×16 level are explored and a subset of modes are selected based on appeared VABs and NABs as shown in **FIGURE 9**. Thus, the previously analyzed example mode- 8×8 could be directly selected according to the pattern of VABs at 16×16 level (see **FIGURE 9**).

From the selected subset of modes at 32×32, 16×16 or 8×8 level, the final mode is determined from the minimum

TABLE 1. Proposed technique adopted mode selection for 32 × 32 coding level using the pattern templates in figure 8.

Predefined Templates of 32×32 Block Level	Selection of Modes at 32×32 Block Level
Φ_1	S_K or $\mathcal{K}$ 32×32
Φ_2	κ 16×16 or $\mathcal{K}$ 16×16
Φ_3 & Φ_4	$\mathcal{K}$ {32×16 or 16×16}
Φ_5	$\mathcal{K}$ {32×8 or 16×16}
Φ_6	$\mathcal{K}$ {32×8}
Φ_7	$\mathcal{K}$ {32×24 or 16×16}
Φ_8	$\mathcal{K}$ {32×24}
Φ_9 & Φ_{10}	$\mathcal{K}$ {16×32 or 16×16}
Φ_{11}	$\mathcal{K}$ {8×32 or 16×16}
Φ_{12}	$\mathcal{K}$ {8×32}
Φ_{13}	$\mathcal{K}$ {24×32 or 16×16}
Φ_{14}	$\mathcal{K}$ {24×32}

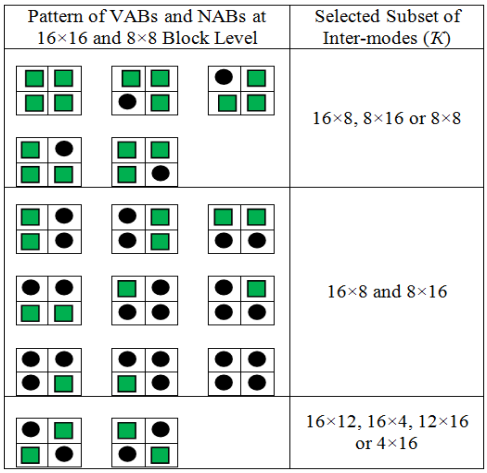


FIGURE 9. Subset of inter-mode selection at 16×16 and 8×8 levels according to the appeared pattern of VABs and NABs.

value of the LCF. The equation for the final mode (Θ) selection is:

$$\Theta_k = arg \min_{\forall m} (j(m)) \tag{12}$$

where $j(m)$ is the LCF for mode selection and Θ_k is the finally selected k th mode. Unlike HM with partially-exhaustive mode selection, the proposed technique checks at most two options for 32×32 level and four for 16×16 and 8×8 levels to select a set of candidate mode(s). Therefore, compared to the partially-exhaustive or full-exhaustive outlook, we entitle this as a direct PU mode selection approach of the proposed scheme.

F. THRESHOLD SELECTION

In the proposed technique, we apply the static threshold and fix it by $\theta_t = 0.25$ although we consider both homogeneous and heterogeneous motion regions in the blocks. We also notice the θ_t value to properly fit with the *joint collaborative team on video coding* (JCT-VC) recommended QP values (i.e. 22, 27, 32, 37) and a wide variety of test sequences

TABLE 2. Test sequences used for this experiment.

Class	Resolution	Sequence	Frames Encoded	Frame Rate (fps)
A	2560×1600	*Traffic*	150	30
		PeopleOnStreet	150	30
B	1920×1080	*Cactus*	500	50
		Kimono	240	24
		ParkScene	240	24
		BQTerrace	600	60
		BasketballDrive	500	50
C	832×480	*BasketballDrill*	500	50
		PartyScene	500	50
		BQMall	600	60
		RaceHorses	300	30
D	416×240	*BQSquare*	600	60
		BasketballPass	500	50
		BlowingBubbles	500	50
		RaceHorses	300	30
E	1080×720	*FourPeople*	600	60
		Johny	600	60
		KristenAndSara	600	60

having different aspects of object motions, camera motions, and resolutions. Whether the value is kept higher or lower than 0.25, detection of VABs for coding also becomes inappropriate and this trend is noticed almost for all sequences. Moreover, in the proposed method, we notice more compact distribution of cost function values that could validate the use of $\theta_t = 0.25$. This value also synchronizes the VABs with higher values of FMSSM across all videos used in this experiment.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

To verify the proposed method's effectiveness, experiments are conducted with the JCT-VC recommended eighteen class sequences including the Class-A, Class-B, Class-C Class-D, and Class-E. The test sequences represent awide range of contents, different aspects of object motions, camera motion, resolutions, and complexity of the contents.The sequence resolutions and the test conditions are presented in TABLE 2. We first assess the performance of proposed method with the HM15.0 and then compare with existing seven recent state-of-the-art methods (to be reported in TABLE 5).

A. EXPERIMENTAL SET-UP

In this work, 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 the HEVC mode selection scheme are developed based on the reference software HM 15.0 [17] and test it under the common test conditions of the HEVC standardization [38]. The motion estimation and motion compensation are carried out only over luma components since there is no additional impact of chroma components on moving region. Thus, the PSNR is calculated over luma only. The RD performance of both schemes are compared considering the maximum CU size 64×64 by enabling both symmetric and asymmetric partitioning block size of 64×64 to 8×8 levels. Performance of both techniques is measured in terms of *Bjontegaard delta peak signal-to-noise ratio* (BD-PSNR), *Bjontegaard delta bit-rate* (BD-BR) [39] and encoding time savings by enabling the *random access* (RA) and *low-delay B* (LD-B) configurations using QP ={22, 27, 32, 37}. We use the search length ±64 for horizontal and vertical directions and run the anchor HM for performance evaluation.

TABLE 3. A theoretical investigation of percentage of time saving by the proposed method against the HM for each sequence type. this is done by checking the average no of inter-modes per coding block by enabling RA test condition.

Sequence Types	HM Checked Average no. of Modes per CU	Proposed Method Checked Average no. of Modes per CU	Average Percentage (%) of time saving
Class- A	21.14	13.02	38.41
Class- B	19.87	12.11	39.06
Class- C	20.96	12.87	38.59
Class- D	17.26	10.82	37.31
Class- E	18.09	10.36	42.73
Average time saving			39.22

B. COMPUTATIONAL TIME ANALYSIS

To justify the computational time saving, let us first calculate the HM and proposed method's average number of modes per coding block for a subset selection. To select a particular mode if any technique checks almost all the modes in one or more coding depth levels in a partially-exhaustive manner, theoretically it requires more computational time compared to the direct mode selection approach. We notice that the HM checks more options for all type of sequences and require more encoding time. The numeric values presented in the second and third column of TABLE 3 indicate average number of modes checked by the HM and proposed technique respectively at RA test condition. The fourth column of the Table presents theoretically obtained average percentage of time saving of the proposed scheme where the Class-E type shows more than 42% time saving as their contents have smoother motions without frequent scene changes. The second highest encoding time saving is obtained for Class-B type (39.06%) and over nineteen sequences of all categories, the proposed method obtained average time saving is 39.22%. However, we experimentally notice the proposed technique to require 6.83% extra time due to carry out phase correlation, saliency and background modeling related preprocessing overheads. Thus, theoretically we anticipate saving 32.39% average encoding time compared to the HM.

FIGURE 10 illustrates the average time saving (ΔT_S) of the proposed method against the HM at RA and LD test conditions. Over eighteen sequences and a wide range of bit-rates, experimentally it obtains on average 33.86% (range: 27.65%∼39%) encoding time saving for RA condition and 29.09% (range: 22.51%∼34.12%) for LD condition

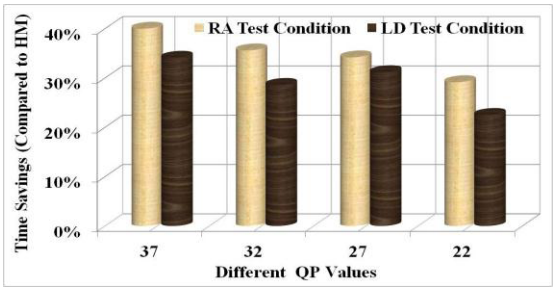


FIGURE 10. Illustration of average time savings (ΔTS) of the proposed method against the HM at RA and LD test conditions.

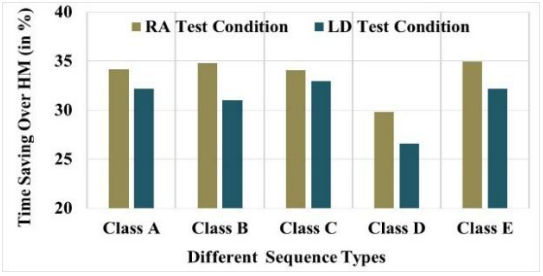


FIGURE 11. Average time savings by the proposed method against the HM based on different video categories.

as shown in **FIGURE 10**. Both for RA and LD configurations, it appears to obtain the highest time savings (39.12% and 34.31% respectively) at QP=37, while at QP=22, time savings are the lowest (29.15% and 22.51% for RA and LD cases respectively). This is due to handle the higher percentage of motion blocks at higher bit-rates and encoding them with appropriate modes towards obtaining the RD performance without sacrificing quality. The execution of time savings (ΔT_s) is carried out by:

$$\Delta T_s = \frac{(T_{HM} - T_{PRO})}{T_{HM}} \times 100\% \tag{13}$$

where T_{HM} and T_{PRO} indicate the total encoding time consumed by the HM and the proposed technique respectively.

For further analysis, we calculate the encoding time of both techniques based on video categories and notice the proposed scheme to achieve 33.18% average encoding time saving at RA test condition and 30.45% at LD condition compared to the HM15.0 as shown in **FIGURE 11**. For Class-E type sequence, the proposed method obtains the highest encoding time saving since it could simply generate a stable background using DBM. In contrast, the *BasketballDrill* sequence of Class-C obtains the lowest encoding time saving both at RA and LD test conditions (23.43% and 20.96% respectively) although it shows more relatively improved RD performance compared to any other Class sequences. Both at bit-rate and video type basis, the RA configuration shows higher time saving than the LD configuration. However, the RD performance at LD condition is superior to the RA condition which is to be discussed in the following Section.

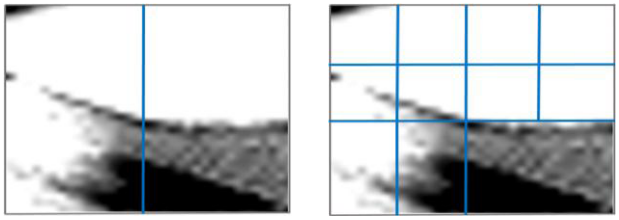


FIGURE 12. Block partitioning modes obtained for the block at (8, 10) position of *BasketballDrill* using the HM and proposed technique. (a) Block partitioning adopted by the HM (whitish indicates high motion). (b) Block partitioning pattern adopted by the proposed technique.

C. RD PERFORMANCE ANALYSIS

FIGURE 12 shows the HM and proposed method selected block partitioning modes for the block at (8, 10) position of *BasketballDrill* sequence as we discussed about this block in **FIGURE 5**. This complex motion block is zoomed in **FIGURE 12** for better visualization in which the whitish indicate high motion areas.

The HM uses 16×32 mode for partitioning it which is not sufficiently appropriate to cover the entire motion areas as the structure of 16×32 mode is not identical with the appeared motion in that block. In contrast, the proposed technique could better identify the object shapes using the DBM and partitions accordingly using a subset of two modes {i.e. 8×16, and 8×8} for partitioning this complex motion block and determines the final mode from 16×16 or 8×8 coding depth level. The rationale of spending few more bits for coding such high motion block is not to sacrifice the image quality. As the mode selection approach of the HM is partially-exhaustive in nature, it could skip some best modes for partitioning a number of blocks having complex video contents in the higher levels.

FIGURE 13 shows the HM and proposed method selected average percentage of four different depth level modes for the *BasketballDrill* sequence using the QPs {22, 27, 32, and 37}. The content of this sequence cover the movements of players with frequent motions and the proposed technique uses higher depth level modes (could be 16×16 and 8×8) for appropriate partitioning of higher motion blocks. Due to encode the extended number of motion blocks, its time savings at RA and LD condition goes the lowest. In contrast, we notice the *Traffic* sequence to have relatively smoother motion areas and the percentage of proposed method selected 16×16 and 8×8 level modes also decrease while increasing the 32×32 and 64×64 level modes. The utilization of higher percentage of skip, 64×64 or 32×32 level modes could determine the smooth background with large block size. An example of mode distributions for the *Traffic* sequence is shown in **FIGURE 14**.

Now let us first concentrate to the frame level PSNR of both techniques in **FIGURE 15** (a) and (b) for *BQSquare* and *Traffic* sequence respectively. Compared to the HM, the proposed technique obtains relatively improved PSNR values almost for all encoded frames of *BQSquare* sequence

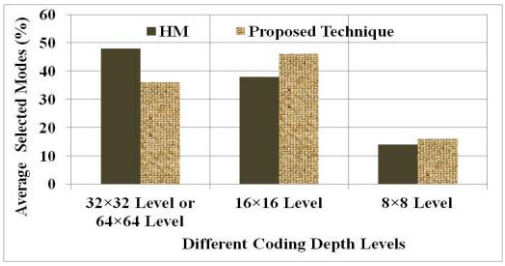


FIGURE 13. The HM and proposed method selected average percentage of depth level-0 to depth level-3 modes for the *BasketballDrill* sequence.

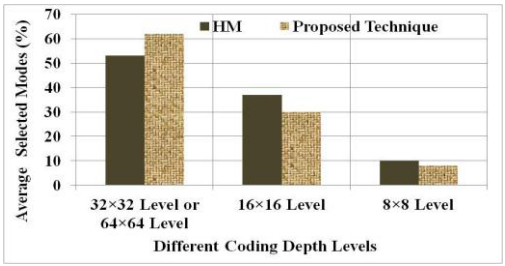


FIGURE 14. The HM and proposed method selected average percentage of depth level-0 to depth level-3 modes for the *Traffic* sequence.

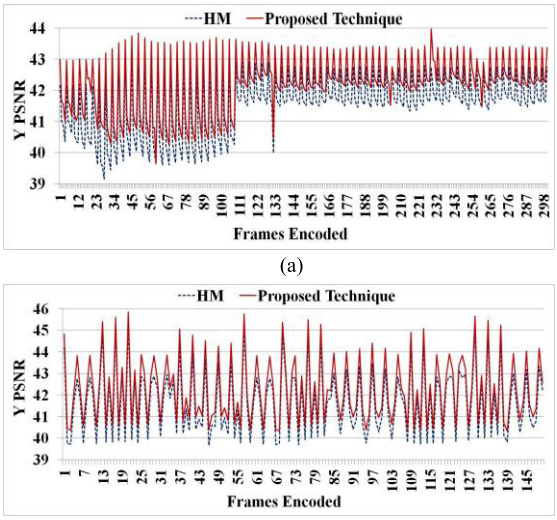


FIGURE 15. Frame by frame level PSNR comparison of HM and proposed technique for the *BQSquare* and *Traffic* sequence. (a) Frame level PSNR for *BQSquare* with LD condition at QP=27. (b) Frame level PSNR for *Traffic* with LD condition at QP=22.

at QP=27. However, for most frames of *Traffic* sequence, it could obtain the similar PSNR values and for few frames it slightly improves PSNR against HM at QP=22. The outcomes of both (a) and (b) in **FIGURE 15** are presented at LD condition as the RD performances of proposed technique are more improved at LD test condition compared to the RA condition. The detailed RD performance results of six sequences with LD test condition and additional results of all sequences with RA and LD conditions are presented in **FIGURE 16** and TABLE 4 respectively.

FIGURE 16 shows RD performance graphs of both techniques for different sequence types at QP = {22, 27, 32, 37}

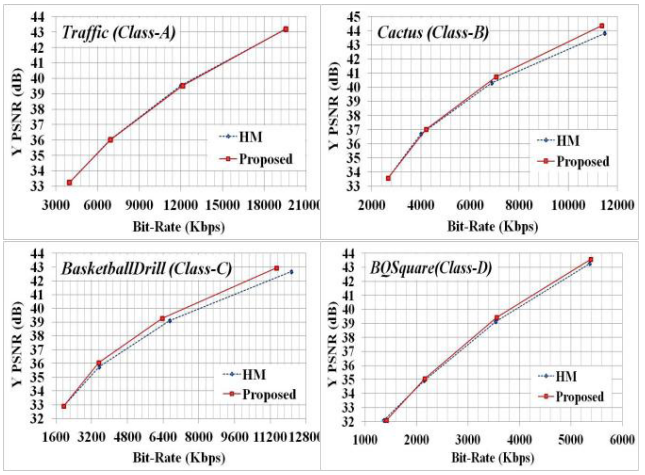


FIGURE 16. RD performance comparison of the proposed technique with the HM15.0.

TABLE 4. Performance comparison of the proposed technique against HM using BD-PSNR, BD-BR, AND ΔTS at RA and LD-B test conditions. The sequences are ordered according to table 2.

Sequence	RA Test Condition			LD-B Test Condition		
	BD-PSNR (dB)	BD-BR (%)	ΔT_S (%)	BD-PSNR (dB)	BD-BR (%)	ΔT_S (%)
Traffic	0.03	-0.11	29.67	0.05	-0.19	27.38
PeopleOnStreet	0.01	-0.08	39.37	0.02	-0.08	37.61
Cactus	0.08	-0.39	27.83	0.13	-0.63	22.89
Kimono	0.04	-0.23	36.94	0.06	-0.29	33.89
ParkScene	-0.03	0.16	33.81	-0.02	0.09	30.92
BQTerrace	-0.02	0.16	38.96	-0.01	0.11	31.24
BasketballDrive	0.00	-0.04	33.81	0.03	-0.28	29.96
BasketballDrill	0.24	-2.21	23.43	0.32	-2.76	20.96
PartyScene	0.03	-0.56	36.69	0.04	-0.59	39.28
BQMall	0.04	-0.32	36.82	0.05	-0.52	31.82
RaceHorses	-0.04	0.41	37.27	-0.03	0.21	34.57
BQSquare	0.08	-0.11	31.52	0.10	-0.16	32.98
BasketballPass	0.06	-0.59	32.07	0.09	-0.93	26.83
BlowingBubbles	0.21	-2.13	28.79	0.29	-2.61	25.44
RaceHorses	-0.01	0.08	24.93	-0.03	0.13	23.77
FourPeople	0.13	-1.28	34.62	0.18	-2.09	30.54
Johny	0.09	-0.96	38.79	0.16	-1.71	35.25
KristenAndSara	0.07	-0.63	35.38	0.11	-0.91	32.95
Average	**0.05**	**-0.49**	**33.37**	**0.08**	**-0.73**	**30.46**

and also reports the proposed method obtained minimum to the maximum PSNR difference (i.e. ΔPSNR) with the HM.

The proposed technique obtains the {min ∼ max} ΔPSNR values {0.00dB ∼ 0.04dB}, {0.00dB ∼ 0.47dB}, {0.01dB ∼ 0.89dB}, and {0.00dB ∼ 0.26dB}, for the *Traffic*, *Cactus*, *BasketballDrill*, and *BQSquare*respectively. Thus, the obtained maximum achievable ΔPSNR is 0.89dB (e.g. at 11200 Kbps for *BasketballDrill* sequence in **FIGURE 16**). For most videos in **FIGURE 16**], the proposed method shows a minor RD performance improvement compared to the HM.

The proposed method produced results (compared to the HM) for eighteen sequences in terms of BD-PSNR, BD-BR, and ΔT_S are reported in TABLE 4. The '+' and '−' sign associated with the BD-PSNR, BD-BR indicate the increment

and decrement respectively, while no sign with ΔT_S indicates the time saving of the proposed method. When we carry out the BD-PSNR and BD-BR calculation according to the individual sequence, we notice the *BasketballDrill* of Class-C to perform the best and *RaceHorses* of Class-C to perform the worst both for RA and LD cases. The inferior RD performance of the *RaceHorses* in Class-C is due to the failure of establishing a stable background for the whole scene.

Once we calculate the average of BD-PSNR and BD-BR for the sequences of each Class type in TABLE 4, we notice the proposed technique to obtain the similar performance with the HM for Class-B type at LD configuration (i.e. improves 0.03dB BD-PSNR and reduces 0.19% BD-BR). However, at RA configuration, it could improve 0.01dB BD-PSNR by reducing 0.06% BD-BR. For the *Cactus* of Class-B, it could improve the RD performance as the contents of this sequence include a large homogeneous region in the background. The sequences in Class-E on the other hand could demonstrate more improved RD performance compared to other Class types. This is because the Class E sequences have relatively large homogeneous background regions with little motion where the DBM could perform its best. Thus, the proposed method can obtain average 0.15dB BD-PSNR improvement and 1.57% BD-BR reduction for LD and 0.09dB BD-PSNR improvement and 0.95% BD-BR reduction for RA test condition. It could also save the highest encoding time saving both for RA (36.26%) and LD (32.91%) conditions using the Class-E type sequences. The still background regions of large portions are decided not to be split which results in large time savings. The overall results of TABLE 4 reveal the proposed method to improve 0.08dB BD-PSNR and decrease 0.73% BD-BR on average at LD test condition and improve 0.05dB BD-PSNR with 0.49% BD-BR reduction at RA condition.

D. OVERALL PERFORMANCE ANALYSIS

To justify the proposed method's effectiveness, its results are compared with seven recent mode selection based fast coding approaches in TABLE 5. The technique presented by Correa *et al.* [14] acquires the highest time saving (65%) at RA condition although they sacrifice 0.06dB BD-PSNR and 1.35% bit-rate increment on average for ten sequences. Pan's method in [10] almost similarly performs like the method presented by Shen *et al.* in [13]. Although the encoding time savings in Pan's method is higher than Shen's method by 15% but the BD-PSNR and BD-BR results reveal Shen's method to perform better in terms of bit-rate savings by 0.04% and 0.17% at RA and LD conditions respectively. The method introduced by Shen *et al.* [13] also performed better than Pan *et al.* [16] in terms of coding gain and encoding time savings at RA and LD test conditions. Over eighteen sequences, the approach presented by Ahn [12] demonstrates the superior performance compared to the one by Xiong [11] in terms of both bit-rate reduction and encoding time savings.

The approach introduced by Lee *et al.* [15] shows virtually no coding loss compared the existing approaches in TABLE 5 in terms of improving the BD-PSNR (i.e. 0.01dB) and reducing the BD-BR (i.e. 0.35%) at LD test configuration. However, at RA condition they could not improve the BD-PSNR but reduce some bit-rates. In both cases, their technique saves $\{30\% \sim 33\%\}$ average encoding time. After all, they improve 0.01dB BD-PSNR by reducing 0.16% BD-BR and saving 32% encoding time on average. The proposed coding technique on the other hand, similarly performs with Lee's method in terms of time savings for both RA and LD test cases. However, it outperforms all the existing state-of-the-art methods presented in TABLE 5 in terms of both reducing bit-rates i.e. 0.79% and 0.51% BD-BR for LD and RA test conditions respectively and improving the BD-PSNR i.e. 0.08dB and 0.05dB for LD and RA conditions respectively.

TABLE 5. Performance comparison of different fast inter-coding methods using BD-PSNR, BD-BR, and ΔTS at RA and LD test conditions.

Algorithms	RA Test Condition			LD Test Condition		
	BD-PSNR (dB)	BD-BR (%)	ΔT_S (%)	BD-PSNR (dB)	BD-BR (%)	ΔT_S (%)
Pan *et al.* [10], 2014	- 0.01	+ 0.40	56	- 0.01	+ 0.40	56
Xiong *et al.* [11], 2014	-	-	-	- 0.07	+ 2.21	40
Ahn *et al.* [12], 2015	-	+ 1.40	49	-	+ 1.00	42
Shen *et al.* [13], 2015	- 0.01	+ 0.36	39	- 0.01	+ 0.23	39
Correa *et al.* [14], 2015	- 0.06	+ 1.35	65	-	-	-
Lee *et al.* [15], 2015	0.00	+ 0.02	33	+ 0.01	- 0.35	30
Pan *et al.* [16], 2016	- 0.03	+ 0.86	12	- 0.02	+ 0.55	15
Proposed Technique	+ 0.05	- 0.51	34	+ 0.08	- 0.79	31

E. SUBJECTIVE QUALITY ASSESSMENT

It is widely recognized that only the higher PSNR values possibly will not always assure better video quality [40]–[41]. Hence we provide the HM and proposed scheme reconstructed images to compare subjective image quality. In this work, we present a subjective quality estimation test using the *Double-Stimulus Continuous Quality Scale*(DSCQS) according to the test conditions of [42]. The sequences were serially organized to conduct this test and the viewing candidates were asked for rating the quality of the HM (called as 'H') and proposed method (called as 'P') generated sequences on a continuous scale ranging between "*Excellent*" and "*Bad*". The assessment results reveal the viewers to recognize the proposed method reconstructed videos having the similar perceptual image quality with the HM almost for all cases. For evaluation purpose, we present an example in **FIGURE 17** where (a) shows the original image of *BasketballDrill* sequence taken for subjective quality test. **FIGURE 17** (b) and (c) illustrate the images reproduced by the HM and the proposed method respectively. For quality comparison, if we concentrate on the entire contents in three

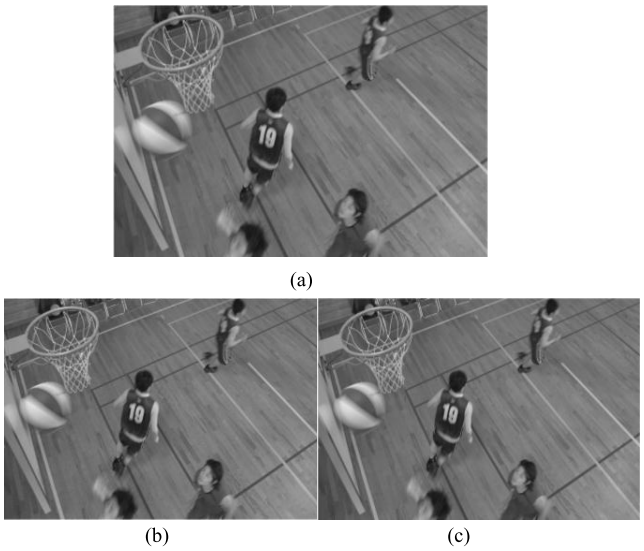


FIGURE 17. **Subjective quality evaluation for the HM and proposed method using the *BasketballDrill* sequence. (a) Original image of *BasketballDrill* sequence. (b) The reproduced image obtained by the HM. (c) The reproduced image obtained by the Proposed method.**

images, the similar quality is appeared and perceived in all cases. It also becomes almost impractical to visually distinguish them from each other. The images in **FIGURE 17** are obtained using the 25th frame of *BasketballDrill* sequence at QP=22 as a random selection. The bits per frame values are 704859 and 704851 and the PSNR values are 43.39 and 43.45 for HM and proposed method respectively.

V. CONCLUSION

In this work, a fast video coding framework has been developed under the existing HEVC recommended coding framework by analyzing various motion and salient features. The existing fast mode selection methods with full dependency on the Lagrangian cost function could not reach improved rate-distortion performance with the HEVC reference test model (HM). To boost-up the HM performance, the proposed technique uses *foreground motion and spatial salient metric* (FMSSM) and its features are captured by dynamic background and visual saliency modeling respectively. Based on the FMSSM value of a coding unit (CU), we select a subset of modes to be explored for encoding the CU. This preprocessing phase is fully independent from the existing Lagrangian cost function (LCF). Since the proposed technique could carry out mode selection with simple criteria, it reduces 32% (raging 21% $\sim$ 40%) average encoding time compared to the HM15.0. Due to exploration of uncovered and static background areas for coding by exploiting the dynamic background modeling, the proposed technique efficiently selects the FMSSM based appropriate block partitioning modes. Consequently, it could also obtain an improvement of 0.08dB BD-PSNR on average (compared to the HM15.0) as a byproduct. The proposed coding framework is expected to facilitate some electronic devices with limited processing and computational resources for faster using the HEVC features without sacrificing image quality.

REFERENCES

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

[2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[3] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, *High Efficiency Video Coding Text Speciation Draft 8*, dovument JTCVC-L1003, 2012.

[4] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[5] Y. Lu, "Real-time CPU based H.265/HEVC encoding solution with Intel platform technology," Intel Corp., Shanghai, China, 2013. Accessed: Dec. 3, 2018. [Online]. Available: https://software.intel.com/sites/default/files/managed/1a/c9/white_paper_real-time_HEVC_encodingSolution_IA_v1.0.pdf

[6] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012.

[7] M. Jiang and N. Ling, "On Lagrange multiplier and quantizer adjustment for H.264 frame-layer video rate control," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 663–669, May 2006.

[8] J. Vanne, M. Viitanen, and T. D. Hamalainen, "Efficient mode decision schemes for HEVC inter prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 9, pp. 1579–1593, Sep. 2014.

[9] L. Shen, Z. Zhang, and Z. Liu, "Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1709–1722, Oct. 2014.

[10] Z. Pan, S. Kwong, M.-T. Sun, and J. Lei, "Early MERGE mode decision based on motion estimation and hierarchical depth correlation for HEVC," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 405–412, Jun. 2014.

[11] J. Xiong, H. Li, Q. Wu, and F. Meng, "A fast HEVC inter CU selection method based on pyramid motion divergence," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 559–564, Feb. 2014.

[12] S. Ahn, B. Lee, and M. Kim, "A novel fast CU encoding scheme based on spatiotemporal encoding parameters for HEVC inter coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 422–435, Mar. 2015.

[13] L. Shen, Z. Zhang, X. Zhang, P. An, and Z. Liu, "Fast TU size decision algorithm for HEVC encoders using Bayesian theorem detection," *Signal Process., Image Commun.*, vol. 32, pp. 121–128, Mar. 2015.

[14] G. Correa, P. A. Assuncao, L. V. Agostini, and L. A. da Silva Cruz, "Fast HEVC encoding decisions using data mining," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 4, pp. 660–673, Apr. 2015.

[15] H. Lee, H. J. Shim, Y. Park, and B. Jeon, "Early skip mode decision for HEVC encoder with emphasis on coding quality," *IEEE Trans. Broadcasting*, vol. 61, no. 3, pp. 388–397, Sep. 2015.

[16] Z. Pan, J. Lei, Y. Zhang, X. Sun, and S. Kwong, "Fast motion estimation based on content property for low-complexity H.265/HEVC encoder," *IEEE Trans. Broadcast.*, vol. 62, no. 3, pp. 675–684, Sep. 2016.

[17] Joint Collaborative Team on Video Coding (JCT-VC). *HM Software Manual, CVS Server*. Accessed: Dec. 12, 2016. [Online]. Available: http://hevc.kw.bbc.co.uk/svn/jctvc-hm/

[18] M. Paul, M. R. Frater, and J. F. Arnold, "An efficient mode selection prior to the actual encoding for H.264/AVC encoder," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 581–588, Jun. 2009.

[19] M. Paul, W. Lin, C. T. Lau, and B.-S. Lee, "Direct intermode selection for H.264 video coding using phase correlation," *IEEE Trans. Image Process.*, vol. 20, no. 2, pp. 461–473, Feb. 2011.

[20] F. Baluch and L. Itti, "Mining videos for features that drive attention," in *Multimedia Data Mining and Analytics*. Cham, Switzerland: Springer, Apr. 2015, pp. 311–326. [Online]. Available: http://ilab.usc.edu/publications/doc/Baluch_Itti15mdma.pdf

[21] L. Shen, Z. Zhang, and P. An, "Fast CU size decision and mode decision algorithm for HEVC intra coding," *IEEE Trans. Consum. Electron.*, vol. 59, no. 1, pp. 207–213, Feb. 2013.

[22] H. Zhang and Z. Ma, "Fast intra mode decision for high efficiency video coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 4, pp. 660–668, Apr. 2014.

[23] L. Shen, Z. Zhang, and Z. Liu, "Effective CU size decision for HEVC intracoding," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4232–4241, Oct. 2014.

[24] K. Lim, J. Lee, S. Kim, and S. Lee, "Fast PU Skip and split termination algorithm for HEVC intra prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 8, pp. 1335–1346, Aug. 2015.
[25] J. Tariq, S. Kwong, and H. Yuan, "HEVC intra mode selection based on rate distortion (RD) cost and sum of absolute difference (SAD)," *J. Vis. Commun. Image Represent.*, vol. 35, pp. 112–119, Feb. 2016, doi: 10.1016/j.jvcir.2015.11.013.
[26] X. HoangVan, J. Park, and B. Jeon, "A probabilistic intra mode decision in distributed video coding," in *Proc. IEEE Int. Conf. Syst., Signals Image Process.*, Apr. 2012, pp. 380–383.
[27] G. Correa, P. Assuncao, L. Agostini, and L. A. da Silva Cruz, "Complexity control of high efficiency video encoders for power-constrained devices," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1866–1874, Nov. 2011.
[28] H. L. Tan, F. Liu, Y. H. Tan, and C. Yeo, "On fast coding tree block and mode decision for high-efficiency video coding (HEVC)," in *Proc. Int. Conf. Acoustic Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 825–828.
[29] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An effective CU size decision method for HEVC encoders," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 465–470, Feb. 2013.
[30] D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.
[31] M. Haque, M. Murshed, and M. Paul, "Improved Gaussian mixtures for robust object detection by adaptive multi-background generation," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.
[32] M. Paul, W. Lin, C.-T. Lau, and B.-S. Lee, "Explore and model better I-frames for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1242–1254, Sep. 2011.
[33] M. Paul, W. Lin, C.-T. Lau, and B. S. Lee, "A long-term reference frame for hierarchical B-picture-based video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1729–1742, Oct. 2014.
[34] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 19, 2007, pp. 545–552.
[35] Z. Li, S. Qin, and L. Itti, "Visual attention guided bit allocation in video compression," *Image Vis. Comput.*, vol. 29, no. 1, pp. 1–14, Jan. 2011.
[36] *Saliency Map Algorithm: MATLAB Source Code*. Accessed: Jan. 2015. [Online]. Available: http://www.vision.caltech.edu/~harel/share/gbvs.php and http://libra.msra.cn/Publication/4113493/graph-based-visual-saliency
[37] H. Yang and Y. Wang, "A LBP-based face recognition method with hamming distance constraint," in *Proc. Int. Conf. Image Graph.*, Aug. 2007, pp. 645–649.
[38] F. Bossen, *Common HM Test Conditions and Software Reference Configurations*, document JCTVC-K1100, Joint Collaborative Team on Video Coding of ISO/IEC and ITU-T, Shanghai, China, Oct. 2012.
[39] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD curves*, document VCEG-M33, ITU-T SC16/Q6, Austin, TX, USA, 2001.
[40] Y. Q. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards*. Boca Raton, FL, USA: CRC Press, 1999.
[41] T. D. Pessemier, L. Martens, and W. Joseph, "Dynamic optimization of the quality of experience during mobile video watching," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast.*, Jun. 2015, pp. 1–6.
[42] *Methodology for the Subjective Assessment of the Quality of Television Pictures*, document Rec. ITU-R BT.500-10, 2000.

PALLAB KANTI PODDER (S'14) received the B.Sc. (Hons.) and M.Sc. degrees from the Department of Information and Communication Engineering in 2008 and 2010, respectively, and the Ph.D. degree from the CM3 Machine Learning Research Unit, Charles Sturt University, Australia, in 2017. After his M.Sc. degree, he joined as a Lecturer with the Computer Science and Engineering Department, Bangladesh University, Dhaka, Bangladesh. Then, he joined as a Lecturer and promoted to an Assistant Professor with the Department of Information and Communication Engineering, Pabna Science and Technology University, Pabna, which is one of the renowned universities in Bangladesh, where he is currently serving as an Associate Professor and a Research Leader with the Human Computer Interaction and Computer Vision-based research laboratory. He has published more than 30 journal articles and conference proceedings in the areas of image processing, video compression, video quality evaluation, human–computer interaction, and computer vision. He is a member of the Bangladesh Computer Society.

MANORANJAN PAUL (M'03–SM'13) received the Ph.D. degree from Monash University in 2005. He was a Post-Doctoral Research Fellow with the University of New South Wales, Monash University, and Nanyang Technological University. He is currently an Associate Professor with the School of Computing and Mathematics, Charles Sturt University. His major research interests are in the fields of image/video coding, EEG signal analysis, and computer vision. To date, he has published more than 155 refereed papers in International journals and conferences. He regularly published journal articles in the IEEE Transactions. He was a keynote speaker in the IEEE DICTA-17, WoWMoM-14 Workshop, DICTA-13, and ICCIT-10. He received $2.5M competitive grant, including two Australian Research Council Discovery Projects. He is a Senior Member of the Australian Computer Society (ACS).

Dr. Paul received the ICT Researcher of the Year 2017 from ACS. He has also served as a guest editor for the *Journal of Multimedia* and the *Journal of Computers* for five special issues, the General Chair for PSIVT 2019, and the Program Chair for PSIVT 2017 and DICTA 2017. He is currently serving as an Associate Editor for the IEEE Transactions on Multimedia, the IEEE Transactions on Circuits and Systems for Video Technology and the *EURASIP Journal on Advances in Signal Processing*.

MANZUR MURSHED (M'96–SM'12) received the master's degree in graduate teaching and the Ph.D. degree in computer science from The Australian National University, Canberra, Australia, in 1997 and 1999, respectively. He is a Federation University Australia's Robert HT Smith Professor and the Personal Chair (formerly Monash University, Gippsland Campus) and was one of the Founding Directors of the Centre for Multimedia Computing, Communications, and Artificial Intelligence Research. He is currently the Emeritus Professor with the School of Science, Engineering and Information Technology, Faculty of Science and Technology, Federation University, VIC, Australia. His major research interests are in the fields of video technology, information theory, wireless communications, distributed computing, and security and privacy. He has so far published more than 190 refereed research papers and received more than $1 million nationally competitive research funding, including three Australian Research Council Discovery Projects grants in 2006, 2010, and 2013, respectively, on video coding and communications, and a large industry grant in 2011 on secured video conferencing. To date, he has successfully supervised 19 Ph.D. students. He received the University Gold Medal from BUET in 1994, the inaugural Early Career Research Excellence Award from the Faculty of Information Technology, Monash University, in 2006, and the Vice-Chancellor's Knowledge Transfer Award (commendation) from the University of Melbourne in 2007. He has served as an Associate Editor for the IEEE Transactions on Circuits and Systems for Video Technology in 2012 and a Guest Editor for special issues of the *Journal of Multimedia* from 2009 to 2012. He is an Editor of the *International Journal of Digital Multimedia Broadcasting*.