Fast Intermode Selection for HEVC Video Coding Using Phase Correlation

ABSTRACT

The recent *High Efficiency Video Coding* (HEVC) Standard demonstrates higher *rate-distortion* (RD) performance compared to its predecessor H.264/AVC using different new tools especially larger and asymmetric inter-mode variable size motion estimation and compensation. This requires more than 4 times computational time compared to H.264/AVC. As a result it has always been a big concern for the researchers to reduce the amount of time while maintaining the standard quality of the video. The reduction of computational time by smart selection of the appropriate modes in HEVC is our motivation. To accomplish this task in this paper, we use phase correlation to approximate the motion information between current and reference blocks by comparing with a number of different binary pattern templates and then select a subset of motion estimation modes without exhaustively exploring all possible modes. The experimental results exhibit that the proposed HEVC-PC (HEVC with Phase Correlation) scheme outperforms the standard HEVC scheme in terms of computational time while preserving-the same quality of the video sequences. More specifically, around 40% encoding time is reduced compared to the exhaustive mode selection in HEVC.

Index Terms— HEVC, Intermode Selection, Motion Estimation, Phase correlation, Phase matched error, Video Coding.

I. INTRODUCTION

The emerging *High Efficiency Video Coding* (HEVC) standard addresses all the existing applications of its predecessor H.264 and specifically focuses on the key issue of high resolution videos [1]. Due to adopting a more adaptive block partitioning, prediction, and transformation methods, HEVC provides excellent performance improvement compared to its predecessor [2]. The picture is partitioned in different *coding units* (CUs) and each of the 64×64 CU can be further partitioned into 2 or 4 smaller units of equal size and the process is continued recursively until square blocks of 8×8 are reached [4]. HEVC also provides asymmetric partitioning such as 64×16, 64×48, 16×64, 48×64, 32×8, 32×24, 8×32, 24×32, 16×4, 16×12, 4×16 and 12×16. In some cases an 8×8 block is also partitioned. Fig. 1 shows block-partitioning option available in HEVC for inter-prediction modes. HEVC requires more than 4 times computational time compared to H.264 mainly due to the extended number of levels (e.g., block size with 64×64, 32×32, 16×16, or 8×8) and different partitioning patterns for *prediction units* (PUs). Thus, a number of electronic devices with limited processing and battery power could not fully exploit HEVC encoding and decoding features.

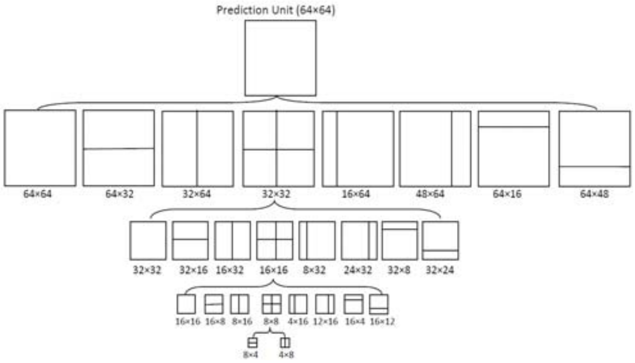


Fig. 1: Block-partitioning in HEVC for motion estimation and motion compensation using inter-prediction modes.

In the mode selection process, if the size of a CU is 2N×2N, HEVC checks the Lagrangian cost function exhaustively using all modes in a level such as 2N×2N, 2N×N, N×2N, N×N, 3N/4×2N, N/4×2N, 2N×3N/4, 2N×N/4 (N=32 in Fig. 1). The Lagrangian cost function is determined for each mode by converting bits into distortion using Lagrangian multiplier (λ). The Lagrangian cost function (J) is defined for a mode selection as

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

where the resultant bits R and the distortion D are determined by a mode for each CU through the motion estimation, motion compensation and encoding process. The best mode of a level is selected based on the minimum value of the cost function. If any mode except N×N is selected then the selected mode is the final mode for the CU. If N×N is selected in the top level (the second row of Fig. 1) based on the cost function, then N×N/2, N/2×N, N/2×N/2, 3N/4×N, N/4×N, N×3N/4, and N×N/4 modes (third row of Fig. 1) are checked against the cost function of N×N mode. If N/2×N/2 mode provides minimum cost function then the next level is also explored, otherwise, the final decision is taken in this level. In this way HEVC selects the best inter-prediction mode for a CU. Moreover, other corresponding intra-modes are also explored to compare with the inter-modes in each level. Thus, HEVC requires minimum 8 and maximum 24 inter-prediction modes to explore for the best mode for a CU.

Without exploring all inter-prediction modes exhaustively, if we can directly select the best mode or a subset of the best candidate modes based on some low computational criteria, we can reduce the computational time significantly. However, to determine a low computational criterion is a challenging task due to the diversified video contents and object/camera motions. Interestingly, HEVC does not use all modes in all levels exhaustively (i.e., total 24 modes at a time of each CU) to reduce the computational time. It uses a simple criterion to select a subset of candidate modes in each level. For example if 32×32 inter-mode is not selected for the top level, it does not explore any smaller block-size modes in other levels (such as 16×16 or 8×8).

To reduce the computational time by selecting a direct mode or a subset of modes, there are a number of existing schemes in the literature using (i) *early CU termination* (ECU) [5], (ii) all zeros in transformed residual [6], (iii) controlling depth level [7] based on the available computational time, (iv) using co-located and neighboring blocks prediction [8], and (v) rate-distortion threshold [9]. In the ECU techniques, based on the analysis of conditional probability of the CU depth selection, the CU depth checking is skipped for all the next sub-CUs when the RD cost of the skip mode is minimum in the current CU. ECU yields approximately 42% time reduction in encoding time with negligible loss on the luminance BD-rate. All zeros in transformed residual [6] is a good indicator to estimate the benefit of using prediction. After the residual block is transformed and quantized with a suitable transformed unit, if all the coefficients in this residual block are zero, the encoder indicates that the prediction is sufficient (no residual coefficients coding). This technique reduces about 41.2% computational complexity with the luminance BD-rate loss 0.85%. These processes do not work properly as the conditional probability might not valid assumption in the ECU for medium to high motion-active videos. Scheme in [6] might not good option for the same kind of videos as it is very hard to get all zeros residual for those videos.

Correa *et al*. [7] use two types of frames namely constrained and unconstrained. They use all possible modes for unconstrained frames but they use limited modes for constrained frame to control the computational time. The experimental results show that their method scarifies 0.8 dB PSNR with 5.7% bit rate increase while reduces 40% to 80% computational time compared to HEVC. Leng *et al*. [8] use co-located and neighboring blocks to predict the current block mode. The experimental results show that 45% computational time can be saved with some loss in bit rate and PSNR. Hou *et al*. [9] use rate-distortion threshold for higher level to terminate the exploring modes in lower level. The experimental results confirm the time saving around 30% with 0.5% quality loss.

Above mentioned methods are all based on the recent HEVC video coding standard. Other fast mode selection methods based on H.264 might be applicable in the HEVC with some modification. For example, Kim *et al.* [10] proposed a fast mode selection technique where they reduced the dominating computing cost incurred by the inter 8×8 and intra 4×4 modes. They analyzed only the cost of already available modes and the experimental results showed that around 53% of total encoding time reduction is achieved at *Quantization Parameter* (QP) at 28 compared to H.264. Zeng *et al*. [11] proposed a fast mode selection algorithm using a couple of thresholds depending on both QP and *motion vector* (MV) in order to classify motion activities. They could be able to save around 62% of encoding time ranging QP=24~40 compared to H.264. But this process suffers from large RD performance degradation if all the modes of HEVC are imposed on it and for QP>40 and QP<24. Paul *et al*. [12] proposed a fast mode selection strategy using the approximated RD cost using the bits of derived MVs and predicted distortion avoiding actual bits for residual errors. The experimental results show that they saved 12% encoding time compared to H.264.

The above mentioned approaches predict motion of the current block from the motion of co-located or surrounding blocks. If the current block is not the part of the same object of the surrounding blocks or the co-located block, the above mentioned approaches do not work well. Even if the current block is a part of the same object of the co-located or surrounding block, the predicted motion of the current block might not be the good approximation due to the different direction or magnitude of motion. Due to these the existing methods have to sacrifice the RD performance if motion estimation is not applied for all modes.

If we approximate motion information using a simple criterion and then we use appropriate partitioning, we can save time but will not loss the quality of coding. The phase correlation can provide us this opportunity. Although the phase correlation is challenging, our goal is to select the inter-prediction mode in a faster manner and thus reducing the number of modes and also time in different video sequences. The phase correlation is able to approximate relative motion of the current block against the reference block. In this paper our motivation is to save the encoding time of HEVC by using phase correlation without sacrificing the quality of the videos. This attempt of reducing huge computational time without degrading the image quality ultimately extends the video coding application in real time and other limited-powered electronics devices.

Paul *et al*. [13] proposed a direct inter-mode selection technique for H.264 using phase correlation where they reduced the number of candidate modes. The method proposed in [13] could not be straightforward applied in the HEVC to select a direct mode or subset of suitable modes due to the 3 times of modes and double/quadruple size of CU compared to H.264. Challenges to apply the strategy [13] are: (i) The CU size of HEVC is 64 or 32 whereas the CU size of H.264 is 16, thus, if we calculate motion indication using 8×8 block using phase correlation, we need to find strategy on how we select larger modes at top level (64×64) or the second top level (32×32); (ii) Need to take decision whether we allow low level modes if a top/second top level mode is selected; (iii) Need to take decision whether we allow more than one modes for a given motion pattern; and (iv) Need to take decision what would be appropriate block size of determining motion pattern for phase correlation, obviously the decisions on (i), (ii) and (iii) also depend on the decision of (iv). In this paper we contribute in the above three challenges (i)–(iii) using two tiers pre-defined binary motion pattern templates to select a subset of inter-modes for motion estimation and compensation.

The rest of the paper is organized as follows: section II presents proposed technique in detail. The experimental results for the proposed scheme are shown in section III while section IV is the conclusion of this paper.

II. PROPOSED FAST MODE SELECTION TECHNIQUE

The phase information corresponds to reliable motion and the phase correlation provides the shifting information between two correlated images by *Fourier Transformation* [14][15]. We calculate the relative displacement between current and co-located blocks of two different frames of a video. By adjusting the displacement we can find the motion-compensated block in the reference frame for the current block. Using the phase of the current block and the magnitude of the motion-compensated reference block, we can generate a matched block. The difference between the current block and the matched block known as phase-matched error block is a good indicator for the existence of motion and the type of motion of the current block compared to the reference block. The whole procedure is shown using a block diagram in Fig. 2.

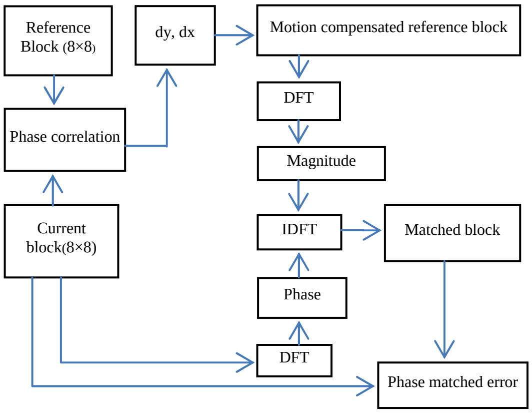


Fig. 2: Block diagram of phase matched error generation

To obtain the exact motion information we consider the entropy ratio of low frequency component and the total entropy of the transformed *phase-matched error* (PME) block. If the ratio is higher than a selected threshold we assume that the current block has motion, otherwise, we assume that it does not have any motion. Normally, the motion indicator is calculated in a block which is smaller than the CU, and we get a binary pattern for a CU by considering '1' as motion and '0' as no motion. We need to take a decision on the inter-prediction mode based on the binary motion pattern of a CU. To make RD performance comparable with HEVC, we may need to select a subset of modes for fully motion estimation and motion compensation. We select a subset of modes and then the final mode is selected based on the Lagrangian cost function. The technique is explained in 3 key steps: (i) binary motion prediction using PME, (ii) mode selection using two tiers binary motion pattern templates at different mode levels, and (iii) dynamic threshold selection for different bit rates.

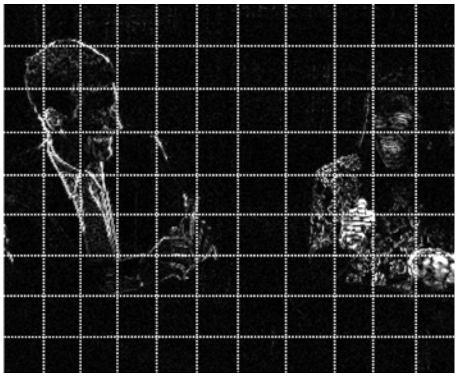
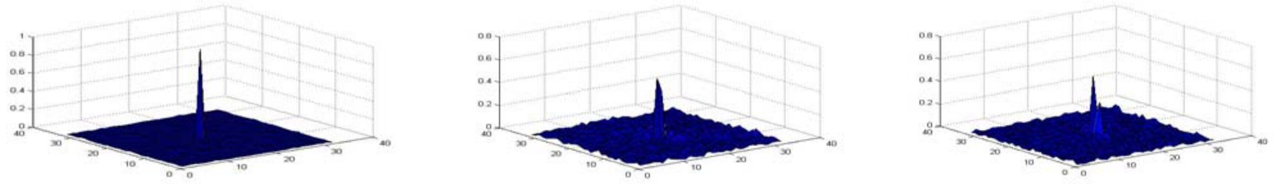


Fig. 3: Motion is represented by showing the difference between 21st and 22nd frame of *Paris* video; the difference is multiplied by 5 for clear visualization.

A. *Motion Prediction*

In our experiment we set the CU blocks as 32×32 pixels and the motion prediction is calculated using block of 8×8 pixels. To predict whether the current block (8×8) has any motion, we apply *Fast Fourier transformation* (FFT) on both the current block and its co-located block in the reference frame. We then calculate the phase difference of the current block from its reference block, perform the inverse FFT on the resulting phase difference and finally calculate the resultant two dimensional (2-D) array [12]. This 2-D array will have a signal peak at the coordinates corresponding to the shift between current block and its co-located block. Fig. 3 shows the difference between the 21st and 22nd frame of *Paris* video where we also find the presence of motion. The residual error, PME is calculated by subtracting the matching block from the original block. Fig. 4 shows phase correlation peaks at different blocks of the 22nd frame of the *Paris* video. Here 32×32 pixel block is considered for clear visualization. We observe that one big magnitude exactly at the middle (a), one smaller peak (b) and two peaks (c) successfully represent different motions in corresponding blocks which are exploited by the proposed method.



(a) No motion

(b) Single motion

(c) Multiple motion

Fig. 4: Phase Correlation Peaks with absolute motion representation at different blocks for the 22nd frame of the Paris video. (a) One big magnitude peak exact at the middle indicating no motion (i.e., CU at (1, 1) position). (b) One small magnitude peak indicates single (simple) motion (i.e., CU at (2, 4) position) and (c) More than one peaks generated using phase correlation is indicating multiple (complex) motions (i.e., CU at (5, 1) position).

To *et al.* [16] showed that the PME provides better motion indication which is calculated by subtracting the current block from its best matched reference block with the phase replaced by the phase of current block. In the phase-matched error generation process the phase component of the matched block is replaced by the phase component of the original block before subtraction. To find the energy compaction of PME, we apply *discrete cosine transformation* (DCT). The detailed procedure can be found in [13][16].

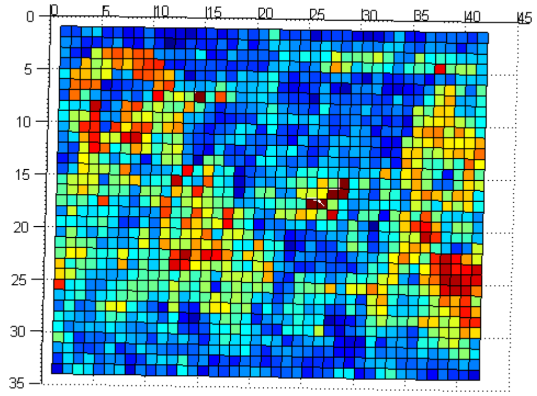


Fig. 5: Ratio of entropies between low-frequency coefficients (taken from top-left triangle) and all coefficients of transformed phase-matched error blocks between 21st and 22nd frame of *Paris* video.

In PME if there is no displacement between the current block and the co-located block then the energy will be concentrated on top-left triangle of the transform PME, otherwise the energy will be scattered throughout the entire area. Thus we can calculate the energy concentration ratio of the top-left triangle (using the third-fourth of each dimension i.e., each horizontal/vertical hand of the triangle is 6 as the block size is 8×8) with respect to the energy of the whole area and then finally predict the presence of motion against a pre-defined threshold.

This phenomenon is presented by calculating the ratio of entropies between the low-frequency coefficients and all coefficients of transformed PME block. Fig. 5 shows an example of entropy ratios (0~1) between frames 21st and 22nd of *Paris* video. Clearly the figure is consistence with the motion scenarios in Fig. 3 by leveling the motion areas with reddish color and non-motion areas with bluish color. If this ratio of energy is greater than the threshold, the motion type is denoted by ‘1’ otherwise motion type is ‘0’. In this way the motion prediction is finally calculated using 8×8 pixel block. Fig. 6 illustrates the motion representation map in which motion and non-motion blocks between 21st and 22nd frames of the *Paris* video are shown. The figure is calculated using Fig. 5 and the threshold 0.37. The motion dominated areas are identified by reddish blocks and the areas with no motion are shown in blue.

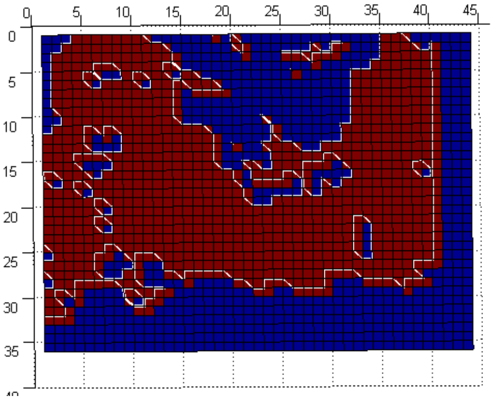


Fig. 6: Motion representation map produced by the presence or absence of motion (1 or 0 respectively) between 21st and 22nd frame of *Paris* video.

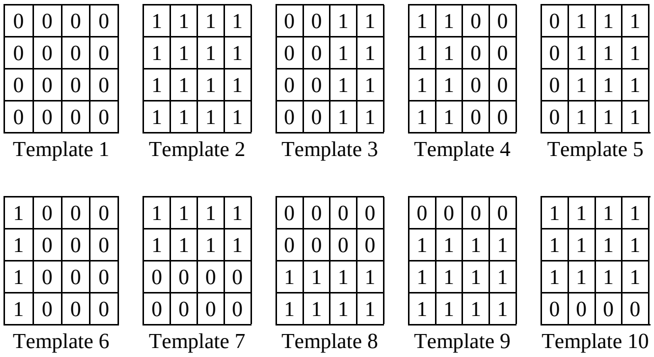
B. *Mode Selection*

In the proposed technique, each CU is set as a 32×32 pixel block and we use 8×8 pixel blocks for binary matrix (‘1’ as motion, ‘0’ as no motion) generation in each of the CUs i.e., a 32×32 CU has 4×4 binary values. Then we calculate the prediction of motion type using ten different predefined binary pattern templates as shown in Fig. 7. In order to select the most appropriate subset of modes, we compare the generated binary matrix for each CU with our proposed templates. A best-matched binary pattern template is selected using a similarity metric for a binary motion block of a CU (generated at Sub-Section *A*) and based on the binary motion pattern template we select a subset of modes for full motion estimation and compensation. The final mode from the selected sub-set modes is determined using Lagrangian cost function. Fig. 7 shows the templates. We design the templates using ‘1’ and ‘0’ combination intuitively focusing on the rectangular and regular object shapes. For example, Template-1 should be selected for the approximation of an object without any motion, Templete-2 with full motion in its entire block area, Template-3 with no-motion in left-half and motion in right-half, Templete-4 is the opposite of Templete-3, and so on. Obviously motion pattern in a CU cannot be a well-structured shape with '1' and '0' combination similar to any binary pattern template. Thus, we need to find the best-matched binary pattern template among ten for a motion prediction binary matrix of a CU through a similar metric. We use a simple similar metric using the *sum of absolute difference* (SAD) between the binary matrix of a CU generated by phase correlation and the *binary pattern templates* (BPTs) in Fig. 7. The best-matched BPT is selected for a CU which provides the minimum SAD. The SAD, D_n is determined as follows where M is the binary motion prediction matrix of a CU comprising 4×4 '1' or '0' combination and P_n is the n-th BPT:

$$D_n(x,y)=\sum_{x=0}^{4}\sum_{y=0}^{4}\left|M(x,y)-P_n(x,y)\right|. \qquad (2)$$

The best-matched j-th BPT is selected from all BPTs as follows

$$P_j=\arg\min_{\forall P_n\in BPT}(D_n). \qquad (3)$$



$\begin{matrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{matrix}$	$\begin{matrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{matrix}$	$\begin{matrix}0&0&1&1\\0&0&1&1\\0&0&1&1\\0&0&1&1\end{matrix}$	$\begin{matrix}1&1&0&0\\1&1&0&0\\1&1&0&0\\1&1&0&0\end{matrix}$	$\begin{matrix}0&1&1&1\\0&1&1&1\\0&1&1&1\\0&1&1&1\end{matrix}$
Template 1	Template 2	Template 3	Template 4	Template 5
$\begin{matrix}1&0&0&0\\1&0&0&0\\1&0&0&0\\1&0&0&0\end{matrix}$	$\begin{matrix}1&1&1&1\\1&1&1&1\\0&0&0&0\\0&0&0&0\end{matrix}$	$\begin{matrix}0&0&0&0\\0&0&0&0\\1&1&1&1\\1&1&1&1\end{matrix}$	$\begin{matrix}0&0&0&0\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{matrix}$	$\begin{matrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\0&0&0&0\end{matrix}$
Template 6	Template 7	Template 8	Template 9	Template 10

Fig. 7: Proposed binary pattern templates for selecting a subset of modes.

TABLE I and TABLE II provide the mode selection process from the BPT at 32×32 and 16×16 levels respectively. The subset of mode selection in both levels is also heuristics and intuitive. The result is verified by the rate-distortion performance against HEVC's performance. It might also be possible that other selection process could provide similar or better rate-distortion performance compared to HEVC. Once a subset of candidate modes is selected by a particular template at 32×32 level we then estimate the final mode by calculating their lowest Lagrangian cost function. If 16×16 mode is selected at 32×32 level, we again explore smaller modes using the motion pattern at 16×16 block (see TABLE II).

TABLE I: PROPOSED SELECTION OF MODES AT 32×32 BLOCK LEVEL USING PRE-DEFINED MOTION PATTERN TEMPLATES.

Motion Pattern at 32×32 block level matches with the Template number	Selection of Modes at 32×32 block level
Template 1	Skip or 32×32
Template 2	Intra 16×16 or Inter 16×16
Template 3 & 4	32×16 or Inter 16×16
Template 5	32×8 or Inter 16×16
Template 6	32×24 or Inter 16×16
Template 7 & 8	16×32 or Inter 16×16
Template 9	8×32 or Inter 16×16
Template 10	24×32 or Inter 16×16

TABLE II: SUBSET OF MODES SELECTED AT INTER 16×16 MODE

Motion Pattern at 16×16 block	Selected Modes
$\begin{matrix}1&1\\1&1\end{matrix}$ $\begin{matrix}1&1\\0&1\end{matrix}$ $\begin{matrix}0&1\\1&1\end{matrix}$ $\begin{matrix}1&0\\1&1\end{matrix}$ $\begin{matrix}1&1\\1&0\end{matrix}$	16×8, 8×16 or 8×8
$\begin{matrix}1&0\\1&0\end{matrix}$ $\begin{matrix}0&1\\0&1\end{matrix}$ $\begin{matrix}1&1\\0&0\end{matrix}$ $\begin{matrix}0&0\\1&1\end{matrix}$ $\begin{matrix}1&0\\0&0\end{matrix}$ $\begin{matrix}0&1\\0&0\end{matrix}$ $\begin{matrix}0&0\\0&1\end{matrix}$ $\begin{matrix}0&0\\1&0\end{matrix}$ $\begin{matrix}0&0\\0&0\end{matrix}$	16×8 and 8×16.
$\begin{matrix}0&1\\1&0\end{matrix}$ $\begin{matrix}1&0\\0&1\end{matrix}$	16×12, 16×4, 12×16, and 4×16.

The proposed algorithm selects comparatively smaller subset of modes than the HEVC. For example if the template 2 is selected, either Inter 16×16 or Intra 16×16 will be the final mode based on the smaller value of the Lagrangian cost function. According to TABLE II, based on the pattern of '1's and '0' combination in the templates we again select a subset of modes generated from Inter 16×16. The mode selection process for a block will be terminated by selecting the final mode from the generated subset of modes having the lowest value using the Lagrangian cost function.

C. Dynamic Threshold Selection

Instead of using any static threshold, we use dynamic thresholds ranging from 0.52 to 0.37 which varies with different *Quantization Parameter* (QP) values as shown in

TABLE III. This is because the number of motion blocks is inversely proportional to the threshold (*Th*). Paul *et al.* [13] proposed a direct inter-mode selection for H.264 using phase correlation where they used a wide variety of thresholds (from 0.27 to 0.91) for different QP values. They derived the average threshold from the best thresholds of 10 videos using different QPs. The dynamic thresholds used in [13] could not perform well with our proposed method as the proposed method is based on the HEVC and thus, it has more modes compared to the H.264. But our observation is that different thresholds compared to the threshold used in [13] work well in our case at different bitrates. Note that the proposed method is less sensitive with thresholds (from 0.37 to 0.52) compared to the method in [13] (where threshold 0.27 to 0.91). For high bitrates normally small blocks are selected while big blocks are selected for low bitrates. Thus in the proposed method for the high bit rates we use lower value of thresholds in order to select the smaller motion dominating blocks in different video sequences.

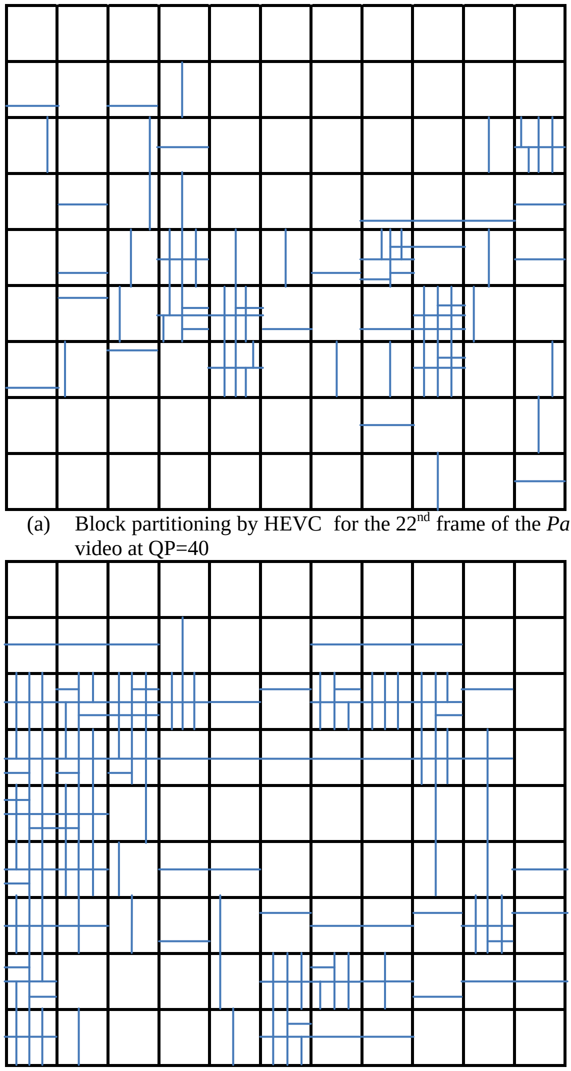
TABLE III : PROPOSED THRESHOLD FOR ALL VIDEOS IN OUR EXPERIMENT AGAINST QPs

QP	Threshold (*Th*)
40	0.52
36	0.49
32	0.46
28	0.43
24	0.40
20	0.37

We also observed that for a relatively high value of *Th* is more appropriate for low motion videos and low value of *Th* is suitable for high motion videos. On the other hand, for the high motion videos if we use high *Th* value then some of the blocks may not be classified as motion-blocks. For this reason, we use a range of thresholds considering the high and low motion areas for all video sequences and thus, better rate-distortion is achieved at different bit rates.

III. EXPERIMENTAL RESULTS

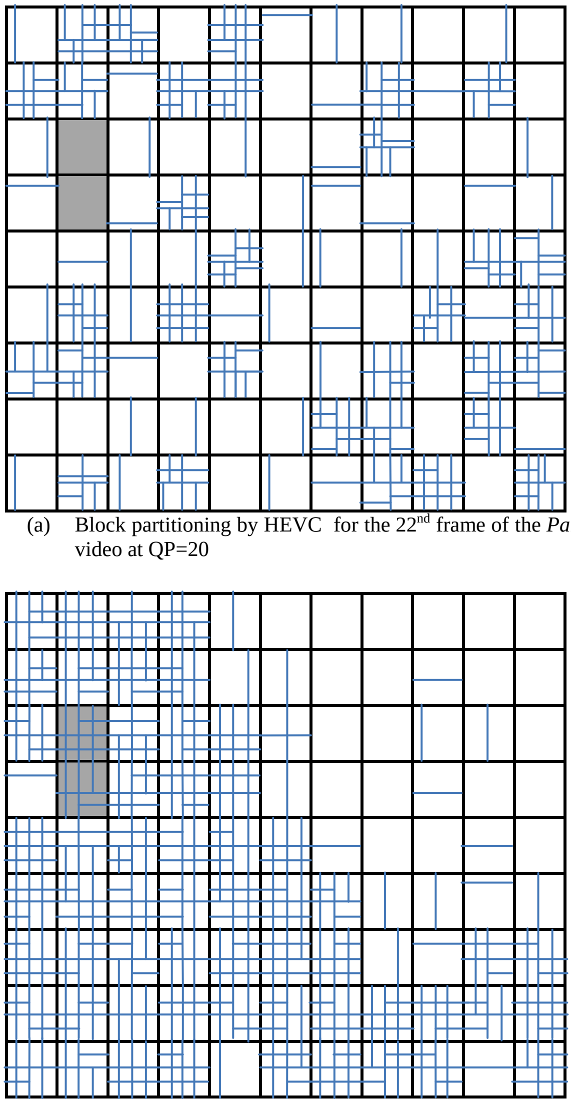
We have conducted our experiments on four *standard definition* (SD) video sequences- *Parris*, *Tempete*, *Silent*, *Tennis* and on two *high definition* (HD) video sequences- *Pedestrian* and *Bluesky*. The SD videos have a frame size of 288×352 (height and width) and HD videos have a frame size of 544×960. For each of the video sequences first 100 frames are encoded with CU size of 32×32 with search length ± 15. With block size of 32×32 for the four SD videos each frame is comprised of 9 rows and 11 columns and for the two HD videos each frame is comprised of 17 rows and 30 columns. We compare our results with HEVC with two reference frame and CU size 32×32 by enabling asymmetric block-partitioning from 32×32 to 8×8 block size. Fig. 8 shows two different block partitioning status for the 22nd frame of the *Paris* video at QP=40 provided by HEVC and the proposed method. For the same QP, the proposed technique selects more small partitions compared to HEVC at low bit rates.


(a) Block partitioning by HEVC for the 22nd frame of the *Paris* video at QP=40

(b) Block partitioning by the Proposed method for the 22nd frame of the *Paris* video at QP=40 and *Th*=0.52

Fig. 8: Block partitioning scenario for the 22nd frame of the *Paris* video at QP=40 with HEVC and the Proposed method.

At high bit rates, compared to HEVC our phase correlation technique selects more motion dominating areas using more small partitioning with the threshold *Th*=0.37. This scenario is illustrated in Fig. 9 for the 22nd frame of the *Paris* video at QP=20. The CUs at (3, 2) and (4, 2) position are marked because HEVC selects those CUs as “no motion” areas whereas our proposed method partitions the block as “high motion” region. As (shown in Fig. 6) the motion dominating areas are identified by reddish blocks and thus comparing with Fig. 6 it is obvious that (3,2) and (4, 2) blocks (block 12×8 and 16×8 in Fig. 6 as it uses 8×8 pixels block) are really a high motion dominating regions (show in Fig. 3). To be more specific, for these blocks, we have observed multiple motions as shown in Fig. 4 (c).



(a) Block partitioning by HEVC for the 22nd frame of the *Paris* video at QP=20

(b) Block partitioning by the Proposed method for the 22nd frame of the *Paris* video at QP=20 and *Th*=0.37

Fig. 9: Block partitioning scenario for 22nd frame of the *Paris* video at QP=20 with HEVC and the proposed method.

As a result, we cannot encode these blocks like HEVC and the proposed phase correlation method split it successfully based on the motion pattern. In this way, compared to HEVC the proposed technique more accurately selects motion dominating blocks in each frame of all the video sequences. For comprehensive analysis, we also provide the average processing time saving graph in Fig. 10 using HEVC and the proposed technique at six different QP values for six standard video sequences. Obviously the proposed technique outperforms the HEVC in terms of computational time for both at a specific QP value or a range of QP values and on average 40% computational time is reduced. The proposed algorithm performs as good as the standard HEVC technique in terms of coding quality (i.e., RD) and achieves a significant time saving for a wide range of bit rates. Moreover from Fig. 10, we also observe that for the best quality video (at QP=20 in our experiment) the proposed algorithm saves up to 55% computing time without image quality degradation.

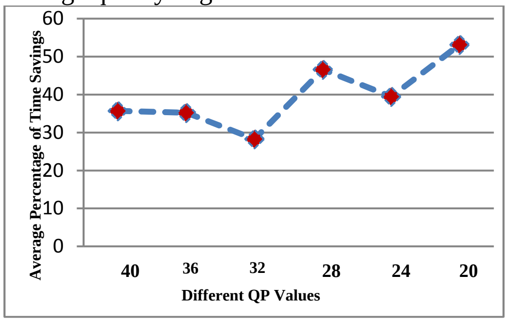


Fig. 10: Average percentage of time savings (compared to HEVC) by our proposed method at different QP values for all videos.

Fig. 11 demonstrates the comparison of RD performances using both the proposed HEVC-PC and the HEVC techniques for six standard video sequences. The experimental results show that almost in all the cases the proposed technique retains image quality similar to the HEVC video coding standard. This is because the proposed method can scan each block of each frame in different videos and perform accurate motion estimation and motion compensation using a simple phase correlation criterion. This means that mode selection process (i.e. how many times a mode will be selected) by using phase correlation works as appropriately as HEVC does for all types of video sequences even using a subset of modes. We also observed that no difference in the RD performance curve especially for high motion videos like *Paris* and *Tennis* as shown in Fig. 11 (a) and Fig. 11 (d) respectively.

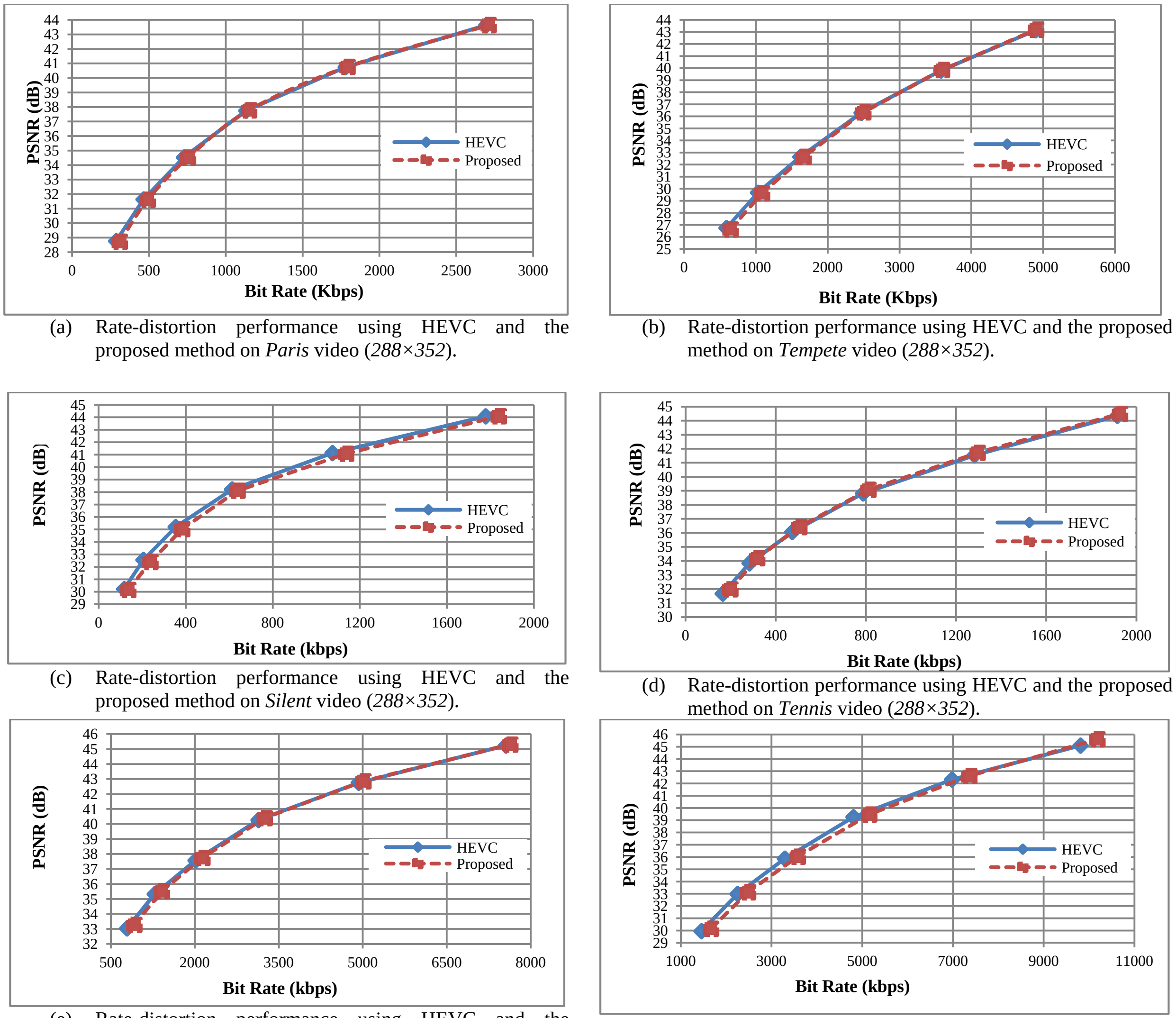
IV. CONCLUSION

In this study, we incorporate the new fast inter-mode selection technique within HEVC coding framework by using phase correlation. Compared to HEVC, instead of exhaustive mode selection search within all possible modes, the proposed scheme successfully exploits mode selection in a faster manner while saving on average 40% of encoding time. We use dynamic threshold for different bit rates and the experimental results exhibit that at a wide range of bit rates our scheme can successfully select the appropriate block-partitions from small to high based on the requirement of the object motion in different video sequences. Our proposed technique outperforms the standalone HEVC by reducing significant amount of computational time without any degradation of the RD performance in most cases thus more suitable for all real time applications and for encoding systems with low processing capacity.

V. REFERENCES

[1] B. Bross, W.-J.Han, G. J. Sullivan, J.-R. Ohm and T. Wiegand "*High efficiency video coding text specification*" Draft 9, document JCTVC-K1003, ITU-T/ISO/IEC, JCT-VC, Oct-2012.

[2] M.B. Cassa, M.Naccari, and F. Pereira, "Fast Rate Distortion Optimization for the emerging HEVC standard", in Picture Coding Symposium 2012, pp. 493-496, May.2012.



(a) Rate-distortion performance using HEVC and the proposed method on *Paris* video (*288×352*).

(b) Rate-distortion performance using HEVC and the proposed method on *Tempete* video (*288×352*).

(c) Rate-distortion performance using HEVC and the proposed method on *Silent* video (*288×352*).

(d) Rate-distortion performance using HEVC and the proposed method on *Tennis* video (*288×352*).

(e) Rate-distortion performance using HEVC and the proposed method on *Pedestrian* video (*544×960*).

(f) Rate-distortion performance using HEVC and the proposed method on *Bluesky* video (*544×960*).

Fig. 11: Comparison of Rate-distortion performance using HEVC and the proposed method on six different video sequences. From (a) to (d) we demonstrate the RD performance on four standard definition videos whereas in (e) & (f) the RD performance of two high definition videos is shown.

[3]

[4] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. “Overview of the H.264/AVC video coding standard.” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, July. 2003.

[5] K. Choi and E. S. Jang, “Coding tree pruning based CU early termination,” JCTVC document, JCTVC-F092, Jul. 2011.

[6] R. H. Gweon and Y. L. Lee, “Early termination of CU encoding to reduce HEVC complexity,” JCTVC-F045, 2011.

[7] G. Correa, P. Assuncao, L. Agostini, and L. A. Da. S. Cruz, “Complexity control of high efficiency video encoders for power-constrained devices,” *IEEE Trans. on Consumer Electronics*, Vol. 57, No. 4, Nov. 2011.

[8] J. Leng, L. Sun, T. Ikenaga, and S. Sakaida, “Content based hierarchical fast coding unit decision algorithm for HEVC,” *IEEE Int. Conference on Multimedia and Signal Proc.*, pp. 56-59, 2011.

[9] X. Hou, Y. Xue “Fast Coding Unit Partitioning Algorithm for HEVC,” *IEEE International. Conference on Consumer Electronics (ICCE)*, pp. 7-10, 2014.

[10] D. Kim and J. Jeong, “A fast mode selection algorithm in H.264 Video coding,” in *Proc. IEEE Int Conf. Multimedia Expo*, 2006, pp. 1709-1712.

[11] H. Zeng, C. Cai, and K.-K Ma, “Fast mode decision for H.264/AVC based on macroblock motion activity,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 4, pp. 1-11, Apr. 2009.

[12] M. Paul, M. Frater, and J. Arnold, "An efficient Mode Selection Prior to the Actual Encoding for H.264/AVC Encoder", *IEEE Transactions on Multimedia*, 11(4), pp. 581-588, 2009.

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

[14] V. Argyriou and T. Vlachos, “A study of sub-pixel motion estimation using phase correlation,” in *Proc. Int. Conf. Brit. Mach. Vis. Assoc.*, 2006, pp. 387–396.

[15] G.A. Thomas, “Television motion measurement for Datv and other applications” *BBC research department, U.K.*, No. 1987/11, 1987.

[16] L. To, M. Pickering, M. Farter, and J. Arnold, “A motion confidence measure from phase information,” in *Proc. IEEE Int. Conf. Image Process.*, 2004, pp. 2583-2586.