Signal Processing: Image Communication

journal homepage: www.elsevier.com

Improved Depth Coding for HEVC Focusing on Depth Edge Approximation

Pallab Podder [a,*], Manoranjan Paul [a], Manzur Murshed [b] and D.M. Motiur Rahaman [a]

[a] *School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW-2795, Australia*
[b] *School of Information Technology, Federation University, VIC-3842, Australia.*

ARTICLE INFO

Article history:
Received
Received in revised form
Accepted
Available online

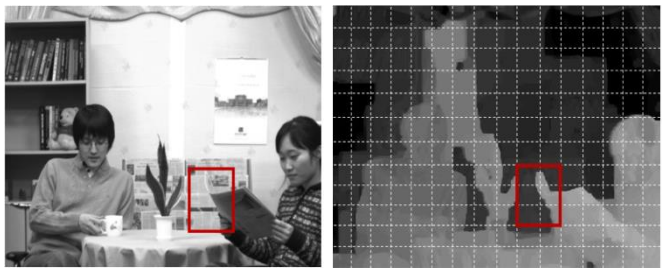
Keywords:
HEVC
Depth Edge
Motion Estimation
Motion Classification
Inter-mode Selection
Pattern Mode

ABSTRACT

The latest *High Efficiency Video Coding* (HEVC) standard has greatly improved the coding efficiency compared to its predecessor H.264. An important share of which is the adoption of hierarchical block partitioning structures and extended number of modes. Although the structure of existing inter-modes are appropriate mainly to handle the rectangular and square aligned motion patterns but they could not be suitable for the block partitioning of depth objects having partial foreground motion with irregular edges and background. In such cases, the *HEVC reference test model* (HM) normally explores finer level block partitioning that require more bits and encoding time to compensate large residuals. Since motion detection is the underlying criteria for mode selection, in this work, we use the energy concentration ratio feature of phase correlation to capture different types of motion in depth object. For better motion modeling focusing at depth edges, the proposed technique also uses an extra Pattern Mode comprising a group of templates with various rectangular and non-rectangular object shapes and edges. As the Pattern Mode could save bits by encoding only the foreground areas and beat all other inter-modes in a block once selected, the proposed technique could improve the rate-distortion performance. It could also reduce encoding time by skipping further branching using the Pattern Mode and selecting a subset of modes using innovative pre-processing criteria. Experimentally it could save 29% average encoding time and improve 0.10dB Bjontegaard Delta peak signal-to-noise ratio compared to the HM.

1. Introduction

The multiview video exploits both texture and depth video information from various angles to create a 3D video [1]-[3] and *free viewpoint video* (FVV) [4] which are gradually becoming more popular for their advanced visual experience with depth perception [5]-[7]. Unlike texture, depth video is determined by a gray scale map indicating distance between camera and 3D points in a scene [8]. If a comparison is drawn between depth and texture image coding, the former one incurs with extensive burden in terms of detecting and encoding its complex edges [9]. Standing on some texture-depth motion correlations, a number of methods in literature encode both texture and depth videos jointly by using texture motion information for the corresponding depth coding. The intention is to reduce the coding complexity by completely/partially avoiding the costly motion estimation process. However, those approaches suffer from the following inescapable limitation: the texture similarity metric is not always equivalent to the corresponding depth similarity metric especially at edge levels. Therefore, those coding techniques could not explicitly detect and encode the complex edge motions of depth objects and reach the similar or improved *rate distortion* (RD) performance of the *HEVC reference test model* (HM) [10].



(a) Gray scale image of *Newspaper* sequence (b) Corresponding depth image of *Newspaper* sequence

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

Compared to the reference texture image in Fig. 1(a), the appeared depth image in Fig. 1(b) demonstrates the lack of appropriate correspondence particularly in the edge like areas due to the existence of irregular motion patterns. For example, the contents of the Red marked block in texture image (Fig. 1 (a)) have been appeared with distinct edges in the corresponding Red marked block of depth image in Fig. 1(b). Due to such misalignment between texture and its corresponding depth edges, additional bits with increased encoding time might be required to compensate large residuals. These constraints deprive a number of electronic devices with limited processing and computational

* Corresponding author
E-mail address: ppodder@csu.edu.au (P. Podder)

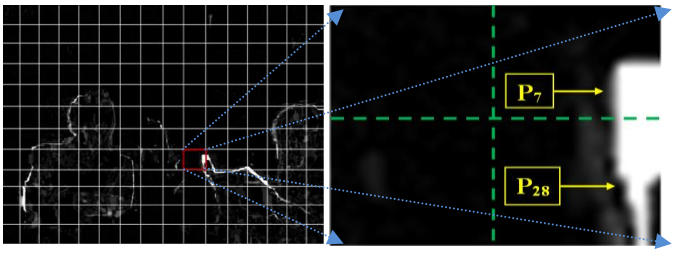
resources to use 3D video and FVV features due to processing a large number of views.

Our motivation is to efficiently encode the depth videos by developing an independent depth coding framework based on the latest HEVC standard [11]-[13]. Compared to the state-of-the-art H.264 [14], the HEVC video coding standard almost doubles the data compression ratio at the same level of video quality, or substantially improved quality at the same bit-rate. This highly improved coding performance gain is due to the *coding unit* (CU) size extension from 16×16 up to 64×64 pixels, variable size *prediction unit* (PU), *transform unit* (TU), symmetric-asymmetric block partitioning patterns and many other advanced features. At 64×64 pixel level, the available inter-prediction modes are 64×64, 64×48, 48×64, 64×32, 32×64, 16×64, 64×16 and 32×32 pixels. The similar partitioning with smaller blocks is revealed down to 8×8 pixels. We denote the block partitioning structure at depth levels 64×64, 32×32, 16×16 and 8×8 by the levels 0, 1, 2, and 3 respectively. For the HM, once a 64×64 mode is selected at level 0, then smaller modes such as 64×48, 48×64, 64×32, 32×64, 16×64, 64×16, and 32×32 are explored. If 32×32 mode is selected from this level, it then further explores smaller modes i.e. 32×16, 16×32, 16×16, 8×32, 24×32, 32×8, and 32×24 at level 1 and so on for the following higher levels. Thus, the HM decides the motion prediction mode of a block by checking all the inter-modes in one or more coding depth levels and minimizing their *Lagrangian cost function* (LCF) [15]. The equation for the LCF ($j(m)$) is defined as:

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

where D means the *sum of squared differences* between the original block and its reconstructed block which is obtained by coding the original block with candidate mode m; λ is the *Lagrangian multiplier* (LM) for the mode selection; $R(m)$ denote the number of bits required for encoding the block with m. To select the best partitioning mode in a coding depth level, the HM explores minimum 8 (i.e. 64×64, 64×48, 48×64, 64×32, 32×64, 16×64, 64×16, and 32×32), and maximum 24 inter-prediction modes (i.e. similar partitioning with smaller blocks from 32×32 to 8×8) with the lowest LCF. This approach of mode selection consumes much higher computational time with respect to the coding depth level increment. Recent literature reveals several times computational complexity increment [16][17] of the HEVC compared to its predecessor H.264.

The design architecture of existing HEVC block partitioning modes may be appropriate mainly to handle the rectangular and square shaped motion patterns in a block. However, they would not be suitable enough for partitioning the depth object having partial foreground motion with irregular edges and background aligned to rectangular and square shapes. Using the frame difference approach, we notice the existence of such irregular motion pattern in the Red marked block at (8, 10) position of the *Newspaper* sequence shown in Fig. 2 (a). The block is highlighted in (b) for better visualization where whitish color indicates motion areas. In such cases, the HM could obtain the best performance by exploring the higher depth level modes (could be 8×8) that require more bits to encode block partitioning header and motion vector information. It also needs higher computational time for the prediction of motion vector in different depth levels. Therefore, to deal with such unusual kinds of motion that have partial foreground of a block, the implementation of more robust motion detection technique could work better to preserve image quality. Paul *et al*. [18][19] introduce a pattern based coding technique in the H.264 standard for texture video where the pattern templates are designed considering a wide variety of regular (i.e. rectangular and square) and non-regular object shapes and edges. As the depth foreground is smoother than texture and residue in depth is less than the texture, this set of templates could better work in depth coding especially for the detection of complex edge motions. Therefore, in the *proposed depth coding method* (PDCM), we incorporate the pattern based coding strategy for explicit motion detection in depth video using the HEVC standard. We also theoretically anticipate the suitability of different patterns i.e. P_7 and P_{28} (see Fig. 6 for more detail) for the detection of foreground motions appeared in Fig. 2 (b) and encode them using the newly incorporated Pattern Mode. Without exhaustive exploration of all modes, if we could determine some pre-processing mode selection criteria, it is then possible not only to save bits by avoiding exhaustive exploration of modes but also reduce encoding time. Moreover, as depth maps are more sensitive to coding errors, approximation of edges by explicit modeling could improve the reconstructed depth quality [20].



(a) Difference between 10th and 11th frame of *Newspaper* sequence. The Red block is taken for analysis

(b) Theoretically anticipated templates for motion detection appeared in the Red marked block in (a)

Fig. 2. Theoretically anticipated templates for depth motion detection for the block at (8, 10) position of the *Newspaper* sequence using the proposed depth coding method (PDCM).

Since motion detection is the underlying criteria for mode selection, the PDCM uses the phase correlation based *energy concentration ratio* (ECR) feature to capture three dissimilar motion information from video contents and performs mode selection. For more accurate motion modeling focusing at depth edges, it also uses an extra Pattern Mode comprising a group of *templates for depth motion detection* (TDMD) with various rectangular and non-rectangular object shapes and edges compared to the existing modes in the HEVC. Using the pre-processing motion criteria, the proposed technique then selects a subset of modes. From the selected subset, the final mode is determined by the minimum value of the LCF.

In general, the subset of existing modes should not improve the RD performance. However, employing the Pattern Mode, the PDCM could improve the RD performance compared to the HM due to the following reasons: (i) since the patterns are designed to encode only the foreground by skipping background areas, it could avoid the necessity of using extra bits; (ii) once the Pattern Mode is selected in a block, it could beat all other inter-modes by detecting complex depth motions and obtaining the lowest cost for that block, and finally; (iii) during selecting the Pattern Mode, the larger blocks are represented by smaller blocks for finer level motion estimation and appropriate mode selection to improve RD performance. On the other hand, this extra Pattern Mode should theoretically require additional computational time. However, the PDCM reduces encoding time by adopting the following strategies: (i) selecting the Pattern Mode in lower depth level, it could ignore the exploration of modes at next higher depth levels, thereby reducing encoding time by avoiding further branching; (ii) it adopts the strategy of encoding only the motion blocks of depth object and finally; (iii) it selects a subset of modes using an innovative preprocessing motion criteria. Moreover, developing an independent depth coding architecture (based on the HEVC) regardless of considering its corresponding texture, the proposed technique could not only improve the interactivity within views by

avoiding texture-depth misalignment issue but also save bits by avoiding the necessity of coding the large residuals. Thus, especially the low processing capacity based electronic devices can get more advantages to use different features of the HEVC.

The major contributions of this paper can be summarized as follows: (i) ECR feature of phase correlation is exploited to capture different types of depth motion which are more accurately classified with mathematically formulated strict criteria; (ii) in addition to the existing block partitioning modes in the HEVC, an extra Pattern Mode is incorporated into the HEVC coding framework to care about the complex depth motions especially at depth edges; (iii) an independent depth coding framework is developed not only to provide better interactivity but also avoid texture-depth misalignment issue.

The rest of the paper is organized as follows. Section-2 reviews the background literature; Section-3 describes the key steps of the proposed technique; Section-4 presents the detail discussion about the experimental results; while Section-5 concludes the paper.

2. Background Review

Many researchers in the literature introduce different forms of block partitioning to approximate the shape of a moving region for better compression efficiency [21][22] in texture videos. With a view to further compression efficiency improvement, Chen *et al*. [23] and Kim *et al*. [24] adopt the strategy with implicit block segmentation instead of explicit encoding of segmentation information. Since these techniques could not skip motion estimation and motion compensation for partitioned background areas, they use additional bits to encode even the almost zero-length motion vector for the background areas. However, Paul *et al*. [18] argue that those approaches may not be suitable for the low to mid bit-rate video coding as the precious extra bits are used for encoding the segment covering almost static background. They also incur with high computational complexity which is a constraint for real-time depth coding applications. Therefore, Paul *et al*. [18] implement a pattern-based coding technique focusing on block partitioning for significantly improving the perceptual image quality of texture video. To speed-up the pattern selection process from the codebook, they also introduce a real-time pattern selection algorithm using different metrics [25]. Compared to the texture, as the bit-rate count in the corresponding depth video is comparatively lower, their approaches would better suit in depth coding for further improving the compression efficiency as they could save bits by encoding only the foreground motion regions. In contrast, the Arithmetic edge coding based arbitrarily shaped motion prediction in macroblock level of depth video is performed by Daribo *et al.* [26] for depth video compression. This process requires extra bits to encode the prediction residuals of the rectangular sub-blocks. Hence the more realistic arbitrary shaped patterns of motion prediction approach in [25] could save more bits by avoiding the necessity of coding large residuals.

By carefully examining the coding mode, motion vector, and structure similarity relationship between the depth and its corresponding texture the authors in [27] successfully implement a depth coding technique to improve coding performance. The performance improvement in depth coding is meant by evolving the fast mode selection algorithms while providing emphasis on coding quality preservation. Aiming this particular goal, a large number of researches have been conducted following two cardinal pipelines that fall into the Inter-coding and Intra-coding.

For Intra-prediction mode decision, Kang *et al.* [28] propose an efficient depth coding method with a view to reduce the loss of boundary information by studying the geometrical and statistical properties of depth video. Experimentally they provide better RD performance compared to the H.264 encoder. Gu *et al*. [29] attempt to reduce computational time for intra depth coding by selecting a subset of available intra modes based on the smoothness of the block. Recently, Park [30] aims at reducing encoding time of intra-prediction for depth coding by selecting a subset of available intra modes based on analyzing the block edge types. His proposed algorithm speeds up the mode decision process by up to 37.65% with negligible loss of coding efficiency. The authors in [31] propose an advanced depth coding technique by introducing intra-picture prediction modes where they utilize geometric primitives along with a residual coding. Their technique substitutes the intra-prediction modes and the residual coding of HEVC for depth intra pictures and intra blocks. Experimentally they obtain about 8% overall bit rate reduction with 3D-HEVC while producing the same quality of synthesized views. On the other side, wedgelet and contour based intra coding could also well-approximate the object edges. However, a series of smaller wedgelets may be required to estimate curved segments of an edge, while contours in an image may be approximated by a wedgelet decomposition which all suffer from pattern matching related overheads. Although the intra-prediction based coding approaches are well-studied for relatively smoother regions, normally they require more bits compared to inter-prediction techniques and its efficiency highly depends on user specified modeling parameters [32].

Good number of citable researches have therefore been introduced for different inter-prediction based fast approaches of multiview and depth video compression to fasten the encoding process. In order to obtain more efficient depth compression, Liu *et al*. [33] analyze the structure similarity between depth and corresponding video and propose two new techniques using the Trilateral Filter and Sparse Dyadic Mode. Regardless of considering the encoding time savings, their approach significantly improve up to about 1.5dB gain on rendering quality compared to the multiview video coding (MVC) technique at the same coding rate. Lin *et al*. [34] propose the *fast mode decision* (FMD) algorithm based on depth information classification. Zhang *et al.* [35] propose the FMD by jointly utilizing the adaptive RD cost threshold, inter-view mode correlation and coded block pattern. For further compression efficiency improvement, recently Pan *et al.* [36] introduce a FMD algorithm based on the texture-depth mode correlation, motion information and coded block pattern to figure out whether to use the same mode as the texture video. Li *et al.* [37] perform pixel-based motion estimation for better prediction in depth coding by exploiting the texture motion. This coding method achieves improved RD performance compared to JM 18.2 by sacrificing over 7.10% average encoding time. For further computational time reduction, Shen *et al.* [38] incorporate an adaptive motion search range determination and a fast mode decision algorithm using the prediction mode and motion vector correlation of color videos and depth maps. Compared to the original H.264 JMVC encoder, they reduce on average 80.2% of average encoding time with *peak signal-to-noise ratio* (PSNR) loss of 0.08dB and the average bitrate increment of 0.60%. To speed-up the mode decision process, Yeh *et al.* [39] attempt to reduce candidate modes by analyzing the RD cost of previously encoded view and determining a threshold for each of the modes in the current view. Compared to the JMVC 4.0, their simulation results reveal a reduction of 76.65% average encoding time by sacrificing 0.07dB PSNR and increasing 0.26% bit-rates. Lei *et al.* [40] propose a FMD method by evaluating inter-view and inter-component coding correlations and activating different early termination strategies for anchor and non-anchor frames. For the non-anchor frame, early termination is decided based on the RD cost of even views, while, for the anchor frames, they apply different mode selection criteria by defining a region of support.

Although this process reduce 78.07% encoding time, results also reveal that it incurs with the quality loss of 0.06dB and bitrate increment of 0.38% on average against the JMVC 8.5.

The above mentioned FMD based depth coding algorithms in the existing literature are developed for different test model versions of the H.264 standard (i.e. the JM). Most of these coding techniques could not reach the similar or improved RD performance with the JM since their algorithmic structures are developed considering motion homogeneity, mode similarity, and complete dependency on existing LCF within the JM framework. Moreover, since the techniques are developed based on different versions of the JM, could not be straight-forward applied to different versions of the HM due to: (i) three times extended number of modes, (ii) CU size extension from 16×16 up to 64×64-pixels, (iii) complex block partitioning structures, (iv) increased motion vector coding length, and (v) other advanced parameter settings in the HEVC. To the best of our knowledge, very few independent depth coding methods in literature could be found for the performance improvement of the HM. Therefore, in the proposed technique, we develop an independent depth coding framework by introducing the Pattern Mode and incorporating it into the HEVC coding framework.

The preliminary idea of the proposed implementation is accepted in a workshop [41], however, a number of significant amendments carried out in this work are summarized as follows: (i) A theoretical anticipation- why the Pattern Mode performs better in terms of capturing foreground-background edge motion using the codebook of predefined pattern templates, (ii) Illustration of more appropriate motion detection for better motion modeling, (iii) Explaining about the HEVC recommended mode distribution (from 64×64 to 8×8 coding depth levels) and the contribution of individually selected pattern mode to improve the RD performance, (iv) Using the HEVC recommended wide range of sequences with various types of motion, resolutions and contents, (v) Detail discussion about the frame-by-frame level PSNR improvement which is a pre-requisite of overall improved RD performance, (vi) Implication of using different weights to the LM and describing with suitable example how ω= 4 performs the best, (vii) Demonstrating the effectiveness of *algorithm with pattern* (AWP) from the *algorithm with no-pattern* (AWNP) and discovering the spatial attribute of the AWP eventually to incorporate it in the proposed algorithm. The ultimate goal is to boost-up the HM coding efficiency by improving the RD performance and reducing the encoding time.

3. Proposed Technique

In the proposed coding technique, we use 64×64 as a CU size. Similar to the HM, the best inter-mode selection at level 0 (i.e. 64×64) is carried out by using the LCF without utilizing any phase correlation-based pre-processing. Once 32×32 size mode is selected (as presented in Fig. 3) from level 0, then we activate the proposed phase correlation based mode selection feature to determine a subset of inter-modes at level 1 to level 3. The phase correlation is a Fourier Transformation based approach to determine the relative translational displacement between current block and the motion compensated block in the reference frame [42]. By exploiting the ECR feature of phase correlation, the PDCM categorizes a block by no-motion, simple/single motion, or complex/multiple motions (i.e. the recognized and classified motion- to be detailed in Section-3.1) using mathematically formulated pre-defined thresholding criteria. The advantages of using the Phase correlation over the *sum of absolute difference* (SAD) or *mean squared error* (MSE) include: (i) it could provide both the absolute direction and amount of motion and (ii) estimates motion more accurately to best figure out whether an image block is associated with simple or complex motion. The classified individual kind of motion is accountable for the selection of individual subset of modes at different coding levels. For example, if the single motion is detected, the proposed technique exploits all inter-modes at 32×32 level and the Pattern Mode. For multiple motions, it exploits the Pattern Mode as well as all the inter-modes of HEVC at 16×16 and 8×8 levels. Note that both at 32×32 and 16×16 level, we ensure the use of Pattern Mode by employing the assorted codebook of different shaped predefined pattern templates that mainly focuses on approximating irregular edges of depth object. While selecting the Pattern Mode for an image block, to ensure the best selected pattern (from the codebook of predefined templates), a real-time pattern selection strategy is appended with the proposed technique. This subset selection phase at different coding depth levels is completely independent from the existing LCF. From the selected subset, the least value of Lagrangian cost function is employed to determine the final mode for a block. The entire process is highlighted as a process diagram in Fig. 3 and the key steps are described in the following Subsections.

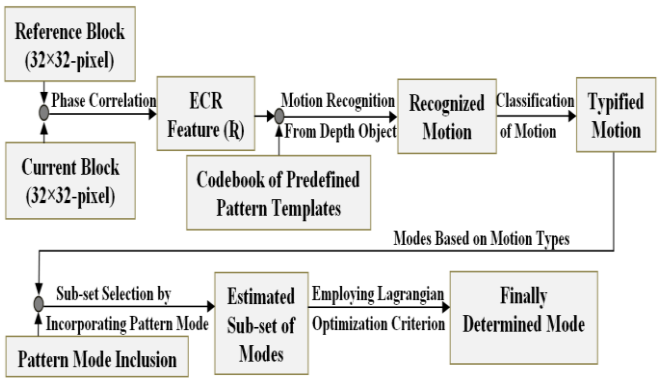


Fig. 3. Process diagram of the proposed mode selection technique for depth coding.

3.1. Motion Recognition and Classification

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

$$ß = Ø\left(\left|þ\left(e^{j(\angle\eta-\angle\delta)}\right)\right|\right) \quad (2)$$

where ß is a phase correlation between the current block *C* and reference block *R* respectively, *Ø* and *þ* means the FFTSHIFT and IFFT respectively, δ and η are the Fast Fourier transformed blocks of the *C* and *R* respectively. The symbol $\angle$ means the phase of the corresponding transformed block. The calculated ß in equation (2) is a two dimensional matrix. The phase correlation peak (Θ) from the position of (dx + Ω/2 + 1, dy + Ω/2 + 1) is calculated by:

$$\Theta = ß(dx + \tfrac{\Omega}{2} + 1, dy + \tfrac{\Omega}{2} + 1) \quad (3)$$

where the blocksize denoted by Ω is 32 since 32×32-pixel block is used by the proposed coding technique to calculate phase correlation and (dx, dy) is the predicted motion vector. Using the phase of the current block and magnitude of the motion-compensated block in the reference frame, we calculate the matched reference block (μ) by:

$$\mu = \left|þ\left(|\eta|\, e^{j(\angle\delta)}\right)\right|. \quad (4)$$

We then subtract the matched reference block from the current block to calculate the residual error (Ɬ). Due to determine the ECR (i.e. Ɍ), we finally apply the *discrete cosine transform* (DCT) to

error ₵ by calculating the ratio from the top-left triangle energy (i. e. ∇_L) with respect to the whole area energy (i. e. ∇_T) by:

$$\text{Ʀ} = (\nabla_L/\nabla_T). \quad (5)$$

In the PDCM, if the calculated value of Ʀ is greater than the predefined *Threshold1* (Ŧ1), motion type is tagged by the "multiple-motions", else if the value of Ʀ is greater than the predefined *Threshold2* (Ŧ2), motion type is tagged by the "single-motion", otherwise motion type is tagged by "no-motion". The implication of using Ŧ1 and Ŧ2 is to be explained in the Section 3.4.

Fig. 4 illustrates the relationship between the ECR and phase correlation peak (i.e. Θ) with respect to the quantitative motion for different blocks of 11th frame in *Newspaper* sequence. Fig. 4 (a) shows the difference between 10th and 11th frame of *Newspaper* where the blocks with Red, Purple, and Blue indicate different categories of motions and the obtained corresponding values of ECR (i.e. Ʀ) for those blocks are also different. If the Ʀ value > Ŧ1 for a block, it encompasses with multiple motions (i.e. Red blocks), else if the value > Ŧ2, it is with single motion (i.e. Purple blocks), else the block does not have any motion (i.e. Blue blocks). From the whole frame, we just illustrate the Red, Purple and Blue blocks at (8, 11), (8, 7), and (4, 8) positions to exemplify the existence of multiple, single and no-motion respectively. Experimentally obtained values of ECR for these blocks are presented in Fig. 4 (b) which shows the highest value for multiple-motions (0.92) and lowest for no-motion (0.28). To display how these motions look like, the phase shifted plots of multiple, single and no-motion are illustrated in (c), (d), and (e) of the same figure with values 0.22, 0.40, and 0.63 respectively. Thus, it is clearly observed from the figure that phase shifted plots have an inverse correlation with motion, while the ECR has a positive correlation with motion.

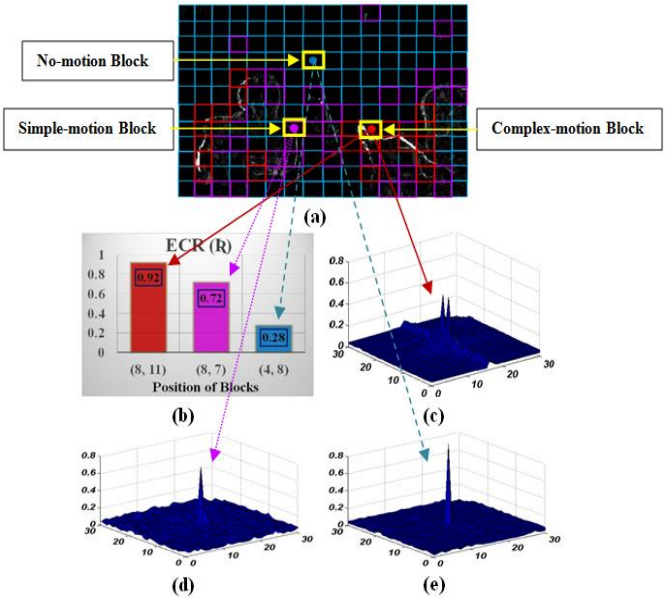


Fig. 4. Illustration of different kinds of motion obtained at different blocks of 11th frame on *Newspaper* video; (a) shows the difference between 10th and 11th frame of *Newspaper* sequence. Based on the appeared complex, simple and no-motion, we mark the blocks with Red, Purple, and Blue respectively. For the blocks with Red and Blue, we obtain the highest and lowest ECR values respectively. In (b), we plot the values of ECR for the blocks at (8, 11), (8, 7), and (4, 8) positions which indicate complex (0.92), simple (0.72) and no-motion (0.28) blocks. Finally, (c), (d), and (e) show the phase shifted plots (i.e. Θ) for the complex, simple and no-motion respectively.

The identified motion type using ECR has been justified with a motion representation map generated between 10th and 11th frame of *Newspaper* sequence using the phase correlation. The map is shown in Fig. 5 where the Reddish blocks indicate the overall motion availability and the Bluish blocks indicate the absence of motion. Fig. 5 has been generated using 32×32 block size for better visualization. If we compare Fig. 4 and Fig. 5 for the blocks at (8, 11), (8, 7), and (4, 8) positions, we also notice the identical motion similarity in terms of presence or absence of motions. These distinguishing characteristics of motion features have been employed for the existing inter-mode (i.e. available in the HEVC standard) selection process. Since the Pattern Mode is distinctly contemplated with the existing HEVC inter-modes for a subset selection, the process of Pattern Mode coding is separately presented in the following Section-3.2.

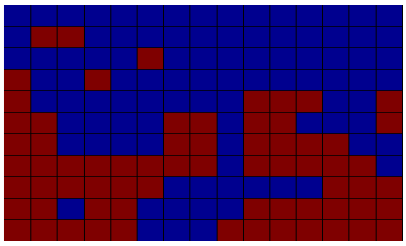


Fig. 5. Motion representation map generated using the Phase correlation indicates entire motion (Reddish) and non-motion (Bluish) blocks in the 11th frame of *Newspaper* sequence.

3.2. Pattern Mode Coding

To detect a wide variety of edge motions with different shapes and edges, the PDCM emphasizes Pattern Mode that comprise with various rectangular and non-rectangular object shapes and edges. Fig. 6 shows 32 different shaped patterns comprising with 64 pixels which are defined in 16×16 pixels block. The patterns are designed placing 1s (white regions) in 64 pixel positions and 0s (black regions) in the remaining 192 pixel positions in 16×16 pixels block. Thus a 32×32 block has four 16×16 sub-blocks, thereby having four Pattern Modes for motion detection having 32 different shapes of motions in a block. The white and black regions inside the templates indicate the presence and absence of motions respectively.

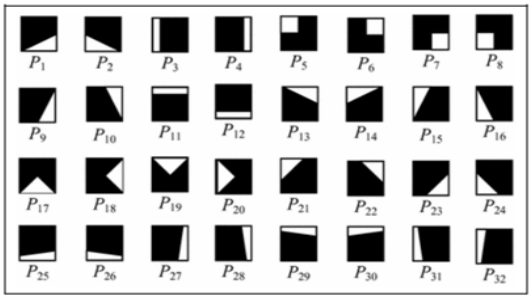


Fig. 6. Templates of 32 different shaped, 64-pixel patterns, defined in 16×16 blocks, in which white regions represent 1 (i.e. presence of motion) and black regions represent 0 (i.e. absence of motion). The term "P" stands for pattern and the subscript numeric values are the numbers of it.

Apparently a real-time pattern selection strategy is appended with the proposed technique to obtain the best selected pattern (from the codebook of predefined templates) for an image block while selecting the Pattern Mode. The selection of the best pattern mainly focuses on using the relevance and similarity metric which jointly provide facility to overcome the trade-off between computational complexity and image quality. The relevance metric focuses on the *gravitational centre* (GC) to represent all moving pixels, while, the similarity metric uses all the pixels to represent that. To describe the Pattern relevance, the GC of a 16×16 binary matrix is first calculated. For the original *codebook of patterns* (CP), the relevance of the *k*th block to a pattern P_t is then measured by $\nabla_{k,t} = D\big(G(M_k), G(P_t)\big)$; where $D(a, b)$ denotes the Manhattan distance between points *a* and *b*. If the

*k*th block is candidate active region block (CRB), then the customized pattern codebook (CPC) is generated based on the following rule: $\forall P_n \in CP : (\nabla_{k,t} \leq T_R) \Rightarrow (P_t \in CPC)$; where T_R is the relevance threshold which is a range of values for dynamic construction of CPCs. Now, to describe the Pattern similarity of the kth block to a pattern $P_t \in$ CPC can be measured using the following distance: $D_{k,t} = \frac{1}{256}\sum_{x=0}^{15}\sum_{y=0}^{15}|M_k(x,y) - P_t(x,y)|$

and the motion region of the *k*th block could be best presented by the pattern P_t such that $D_{k,t} = \min_{\forall P_i \in CPC} (D_{k,i} | D_{k,i} < T_s)$ where Ts

indicate the similarity threshold which is 0.25 since if none of the 64-pixels of a particular pattern cover any part of a motion region, then the pattern similarity metric will be ≥ 64/256=0.25. Further detail of the template selection could be found in [25]. Thus, the relevance metric makes the selection process faster, whereas, the similarity metric targets to the image quality. Through the Pattern Mode selection, only the white regions of the templates are motion compensated for the current block, while and the black regions are treated as skipped. As we encode only one-fourth of a 16×16 block using Pattern Mode, it could effectively reduce bit rates and computational time. We need additional bits to encode the index of the selected pattern template. It is proven in the Section-4 that there are numerous blocks in which the selection of other inter-modes would not be suitable enough for partitioning except the Pattern Mode. The analysis also tells about the RD performance (the objective image quality) improvement especially from the dynamic selection of the Pattern Mode. Note that no extra motion estimation is done in Pattern Mode site. The Pattern Mode is only employed for approximating the motion at depth edges by using a codebook of template matching. Now, making a correlation with Section-3.1 (i.e. for the existing modes in HEVC) and Section-3.2 (i.e. for the newly incorporated Pattern Mode), the Section-3.3 eventually describes the entire process of mode selection.

3.3. Inter-mode Decision

In the PDCM, we use the CU size comprising with 64×64-pixels and similar to the HM, we exhaustively encode all inter-modes at that level (i.e. level 0) using the LCF. Once any 32×32 level mode is selected, then we apply the phase correlation based pre-processing technique due to reduce the computational time from that level to higher levels i.e. level 1 to 3. Since the probability of selecting a 64×64 partition size for the sequences with mid to lower range resolution is below 10%, we do not apply the proposed phase correlation strategy for level 0. In the proposed scheme, there is a high correlation between motion-classification and a subset of inter-mode selection. The motion is classified by analyzing video contents and since the mode selection process is executed from the categorized motion, thus, the probability of selecting the best partitioning mode is also very high. Table 1 depicts the mode selection process of the PDCM at 32×32, 16×16, and 8×8 depth levels based on dissimilar types of motion. It reveals that if there is no existence of motion in a block (i.e. for No-motion block), the proposed algorithm partitions it either by Skip or Inter 32×32 mode. Once the single motion is detected in a block, the subset of eight modes (i.e. intra 16×16, Inter {32×16, 16×32, 32×8, 32×24, 24×32, 8×32} and the Pattern Mode) at 32×32 level is explored. Similarly, a subset of total nine modes is investigated for 16×16 and 8×8 levels. In the figure, $\mathcal{K}$, κ, and Ψ_n stand for Inter, intra, and Pattern Mode respectively and their selection based on classified motion is confirmed by using the 'X' symbol. It is noticed that when more motion dominating blocks are explored, the proposed algorithm selects modes with higher coding depth levels. The rationality of spending few more bits to capture multiple motions is to ensure more appropriate mode selection at higher level so that it could eventually reflect on improving the RD performance.

Table 1. PDCM adopted subset of inter-mode selection process based on dissimilar motion types. In the figure, $\mathcal{K}$, κ, and Ψ_n stand for Inter, intra, and Pattern Mode respectively. The mode selection is symbolized by 'X' marking.

Modes of 32×32, 16×16 and 8×8 Levels	Mode Determination Based on Classified Motion		
	No Motion	Simple/Single Motion	Complex/Multiple Motions
κ {16×16}		X	
S_{kip}	X		
$\mathcal{K}$ {32×32}	X		
$\mathcal{K}$ {32×16}		X	
$\mathcal{K}$ {16×32}		X	
$\mathcal{K}$ {32×8}		X	
$\mathcal{K}$ {32×24}		X	
$\mathcal{K}$ {24×32}		X	
$\mathcal{K}$ {8×32}		X	
Ψ_n		X	X
$\mathcal{K}$ {16×16}			X
$\mathcal{K}$ {16×8}			X
$\mathcal{K}$ {8×16}			X
$\mathcal{K}$ {12×16}			X
$\mathcal{K}$ {4×16}			X
$\mathcal{K}$ {16×12}			X
$\mathcal{K}$ {16×4}			X
$\mathcal{K}$ {8×8}			X

Both at 32×32 and 16×16 level, we ensure the use of Pattern Mode (i.e. Ψ_n, where *n*={1, 2, 3,….,32} and Ψ_n is the *n*th selected pattern for a sub-block) especially focusing on different shapes of moving objects. This is due to cover the whole motion regions usually uncovered by the traditional mode selection process. The ultimate goal is to improve the RD performance. In the proposed coding architecture, the LM is multiplied by a weight (i.e. ω= 4 in this experiment) in the Pattern Mode selection process. The rationality of this multiplication is not only to adjust weight with the HM but also restrict the selection of Pattern Mode for those blocks in which motions are not properly aligned with the pattern templates. This could avoid the necessity of using extra bits for coding with the Pattern Mode. However, an example of the RD performance comparison using ω= 1, 2, 3, 4, 5, 6, and 7 is shown in Fig. 15 and related analysis is provided in Section 4.4 for the *Newspaper* sequence. Other coding configuration may perform better, however, the proposed scheme improves 0.10dB PSNR on average compared to the HM15.0 using ω= 4 (to be reported in Table 3). Moreover, compared to other weights, by selecting ω= 4, it minimizes the time overhead of the Pattern Mode selection.

From the selected subset of motion estimation and motion compensation modes, the final mode is determined by using the lowest value of the Lagrangian optimization function. The equation for the final mode is:

$$¥_t = arg\min_{\forall m}(j(m)) \quad (6)$$

where *j*(*m*) is the Lagrangian cost function for mode selection and $¥_t$ is the finally selected *t*th mode.

3.4. Threshold Specification

For all test sequences, we observe the trend that if we increase Ŧ1 and Ŧ2 values, the number of motion blocks decreases. At a given bit-rate, for relatively high threshold, largely stationary regions in an object are classified as no-motion blocks, however, for relatively lower threshold, those are classified as motion (simple/complex) blocks. On the other hand, if the bit-rate is increased and the thresholds are kept stationary, again, numerous regions in the moving object may not be classified as motion blocks. These trends motivated us to use a range of thresholds in the PDCM. Since the proposed technique is developed based on motion classification strategy, we first derive thresholds against entire range of *quantization parameters* (QPs) used in the HEVC.

This range of QPs and their PDCM generated respective values of Ŧ1 and Ŧ2 are shown in Fig. 7.

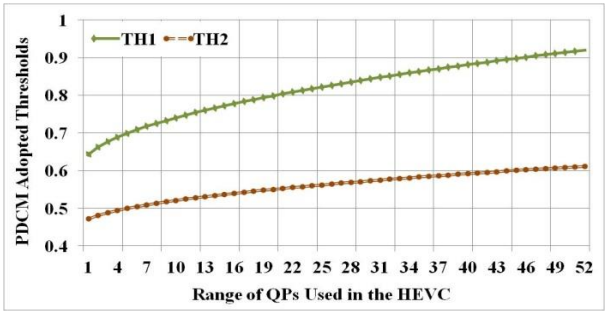


Fig. 7. Proposed range of Ŧ1 and Ŧ2 for the entire QPs used in the HEVC.

We test the proposed technique for the whole range of QPs using the thresholds in Fig. 7 and notice it not to sacrifice the RD performance. From this broad range of QPs, we just select six popularly used sample QPs and their corresponding threshold values that are used in this experiment (presented in Table 2). We approximate the evaluation of Ŧ1 and Ŧ2 using the non-linear functions where QPs are used as independent variables. The approximation of these thresholds are developed in equations (7) and (8) by:

$$Ŧ1 = 0.0445 \times \sqrt{QP} + 0.60 \quad (7)$$

$$Ŧ2 = 0.0225 \times \sqrt{QP} + 0.45 \quad (8)$$

Table 2. Range of thresholds for the QPs used in the proposed experiment for a wide variety of video contents.

QP	Ŧ1	Ŧ2
40	0.88	0.59
36	0.86	0.58
32	0.85	0.57
28	0.83	0.56
24	0.81	0.56
20	0.79	0.55

4. Experimental Results and Discussions

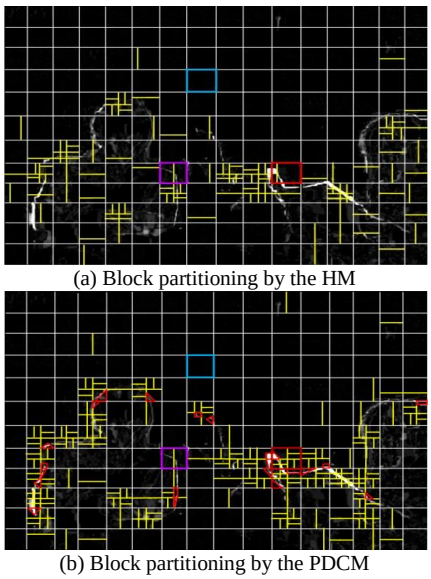
To evaluate the effectiveness of using PDCM, we perform experiments on nine popularly used depth sequences. The sequences with resolutions 1920×1088 (W×H) are *GT_Fly*, *Poznan_Street*, *Poznan_Hall*, *Poznan_CarPark*, and *Undo_Dancer*, while, the sequences with resolutions 1024×768 (W×H) are *Newspaper*, *Lovebird1*, *Kendo*, and *Balloons*. The test sequences selected for the experiment are the representatives in the sense having wide range of contents, different kinds of object motions, camera motion, and the complexity of the contents. For performance evaluation, we first compare the PDCM results with the HM15.0 and then compare the produced results with existing five recent state-of-the-art methods as to be reported in Table 5.

4.1. Experimentation Set-up

The test platform used for the experiment is a 64-bit Microsoft Windows 7 operating system running on a dedicated desktop machine with Intel Core i7 CPU of 3.33 GHz and 32-GB RAM. The proposed depth edge approximation based coding scheme (i.e. the PDCM) and the HEVC with mode selection scheme are developed based on the HM15.0 [10]. Like the HM, we also implement the PDCM by setting the CU as a 64×64 pixel block. Using a wide range of QPs (i.e. 20, 24, 28, 32, 36, and 40), the tested sequences are encoded with 25 frame rate and search range ±64 (horizontal and vertical). We use the *Group of picture* (GOP) size 12 and two reference frames to encode a P-frame for both techniques. The performance evaluation is carried out based on the *Bjontegaard Delta PSNR* (BD-PSNR), *Bjontegaard Delta Bit-rate* (BD-BR) and the average encoding time saving (T_s). The BD-PSNR and BD-BR are calculated according to [43]. Since the proposed technique mainly focuses on improving the depth coding performance, individually the depth views were used during testing. The entire experiment were carried out using the middle view (e.g. view-4 where the View-3 and View-5 are available) of the depth sequences and the PSNR were also computed using that of the middle view (i.e. view-4). However, for comprehensive analysis, we further test the proposed algorithm by generating the synthesized views and using both views from the multiview sequence (e.g. both view-3 and view-5). The calculated PSNR and the synthesized views reproduced by the HM and PDCM are presented in Section 4.6. As the PDCM is fully devoted to the performance improvement of the HEVC only, therefore, in the presented work, all the comparisons have been carried out with the HM in every aspects.

4.2. Block Partitioning Modes Analysis

Fig. 8 shows the distributions of block partitioning modes by the HM and the PDCM for the 11th frame of *Newspaper* sequence at QP=32. We first consider the blocks at (4, 8), (8, 7), and (8, 11) positions which encompass with no-motion, single-motion and multiple-motions respectively as described in Fig. 4. If we observe the blocks at (4, 8) and (8, 7) positions (Blue and Pink blocks respectively) in Fig. 8 (a~b), we observe very similar partitioning patterns in both techniques. However, for the remaining block at (8, 11) position (Red square) in Fig. 8 (a), the HM selects 8×32 mode for that block by considering its partial motion only. This selection is not appropriate enough to obtain the best RD performance due to the lack of proper correspondence between appeared motion and the structure of 8×32 mode. The PDCM on the other hand, partitions that of the block using the Pattern mode (top-left position of the sub-block with the template P_{16}), 8×8 mode (bottom-left sub-block), 16×8 mode (top-right sub-block), and 8×16 mode (bottom-right sub-block). The reason of selecting the P_{16} template is probably due to having its most structure similarity with the appeared motion in that sub-block. This approach of more appropriate mode selection could eventually improve the RD performance.



(a) Block partitioning by the HM

(b) Block partitioning by the PDCM

Fig. 8. Analysis of Block Partitioning Modes for the HM and the Proposed technique with distinct Pattern Mode.

In Fig. 8 (b), we also visualize the selection of other different pattern templates. We try to provide the similar shape of the pattern templates as they were generated in Fig. 6 and mark them

Red according to their selection in the blocks. To complete the analysis, we further take into account the block at (8, 10) position which we first took into discussion in Fig. 2. We theoretically anticipated about the top-right and bottom-right positions of the (8, 10) block to be partitioned using Pattern Mode by selecting P_7 and P_{28} templates respectively. Now in Fig. 8, we experimentally observe these two pattern templates (i.e. P_7 and P_{28}) to be selected by the proposed technique to handle motions in those positions. This is highlighted in Fig. 9 for further analysis.

(a) Sub-block Partitioning pattern adopted by the HM (b) Sub-block partitioning pattern adopted by the PDCM

Fig. 9. Block partitioning modes for the block at (8, 10) position using the HM (a) and the proposed coding technique (b). Compared to (a), the motion detection approach in (b) appears more appropriate.

To partition the top-right sub-block of the block at (8, 10) position in Fig. 9 (a), the HM uses 8×16 mode. However, the partitioning decision for the bottom-right sub-block using 16×8 mode does not reveal appropriate as the appeared motion is not identical to the structure of 16×8 mode in that sub-block. On the other hand, due to properly capture the whole motion area the PDCM uses P_7 and P_{28} templates for the top-right and bottom-right sub-blocks respectively and more appropriately partitions them using the Pattern Mode. This approach of mode selection reveals more appropriate in terms of btaining the improved RD performance.

Fig. 10 draws a comparison on mode selection between the HM and the PDCM for the *Newspaper* sequence. In the Figure, the HM uses higher percentage of lower depth level modes for partitioning blocks. For this sequence, the PDCM distinctly selects 8.85% Pattern Mode by detecting depth motions. The percentage of Pattern Mode selection is the highest for *GT_Fly* sequence which is 11.24. The contents of this sequence reveal rough foreground with more dispersive edges. Therefore, the PDCM extensively exploits different pattern templates to capture those irregular object motions and selects relatively higher percentage of Pattern Modes. For nine sequences used in this test, the proposed technique selects 9.63% Pattern Mode on average.

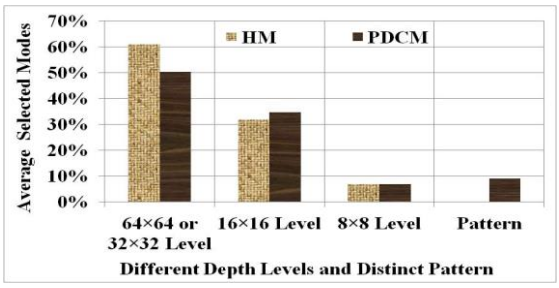


Fig. 10. Average modes selected by the HM and the PDCM for *Newspaper* sequence at different depth levels with distinct Pattern Mode.

4.3. Encoding Time Analysis

To analyze the encoding time of the HM and the PDCM, we first observe the average number of modes attempted at each block in a frame for final mode decision. For all the sequences, this average is noticed higher for the HM compared to the proposed technique and therefore the HM requires more encoding time for final mode execution. However, we experimentally notice that PDCM requires 4.81% extra time for the execution of phase correlation and pattern matching related pre-processing overheads. This extra time is also taken into account for the final encoding time calculation.

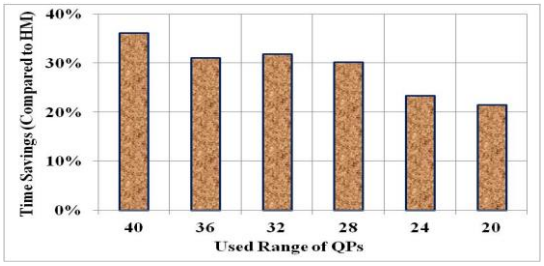


Fig. 11. Demonstration of overall average time saving (for nine sequences) by the PDCM (against the HM) at different QPs.

We demonstrate the computational time analysis of both techniques at the QP and video sequence basis. In both cases, the calculation of encoding time saving (T_s) is executed by:

$$T_s = \frac{(T_{HM} - T_{PRO})}{T_{HM}} \times 100\% \quad (9)$$

where T_{HM} and T_{PRO} mean the encoding time consumed by the HM and the PDCM respectively. The experimental results reveal that for a wide range of QPs, the proposed technique reduces 29.06% encoding time on average although the highest encoding time saving is found at QP=40 (36.12%) as shown in Fig. 11**.** However, at QP=20, it obtains the least amount of time saving (21.47%) due to handle the increased number of motion blocks and classify them in a finer level.

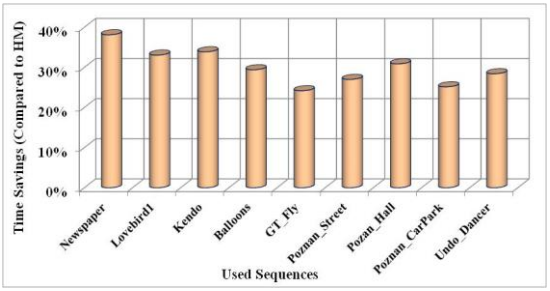


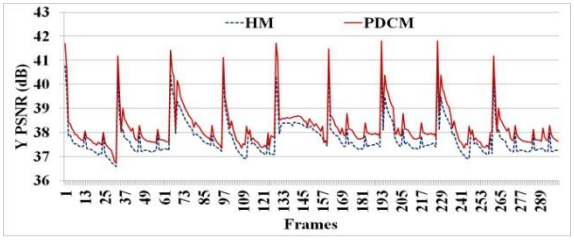
Fig. 12. Average time saving by the PDCM for each individual sequence used in this experiment.

In addition, the results of time saving for each individual sequence in Fig. 12 show that the PDCM could save 30.53% average encoding time with the highest saving for the *Newspaper* sequence (38.65%). In the Figure, for the first four sequences (with resolutions 1024×768), we experimentally obtain 33.61%, while, for the rest of the sequences (with resolution 1920×1088), we obtain 27.12% average encoding time saving. The *GT_Fly* sequence demonstrate the lowest time saving (i.e. 24.17%). This is because the proposed technique exploits the highest percentage of Pattern Mode for this sequence (i.e. 11.24% as discussed in Section 4.2) by extensively exploring pattern templates and requiring more extra time. However, this sequence shows more improved RD performance compared to any other 1920×1088 resolution sequences which is discussed in the following RD performance analysis section.

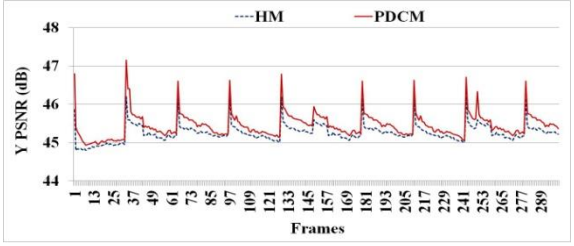
4.4. RD Performance Analysis

Prior to the detail discussion about the RD performance in Fig. 14, let's first analyze the obtained frame level PSNR of the HM and the PDCM at different QPs. To present this results, we consider the *Newspaper* sequence from 1024×768 resolution type and test it at QP=32. Fig. 13(a) shows that almost for all the frames, the PDCM obtains improved PSNR compared to the HM. For this QP, the average PSNR values obtained by the HM and the proposed technique are 42.36dB and 42.51dB respectively. Another similar example for the *Poznan_Street* sequence (from

1920×1088 resolution type) at QP=28 is provided in Fig. 13(b). The outcomes of the this sequence shows average 0.19dB PSNR improvement. At this QP, the average PSNR values for the HM and the PDCM are 45.19dB and 45.38dB respectively.



(a) Frame level PSNR distribution for *Newspaper* sequence at QP=32



(b) Frame level PSNR distribution for *Poznan_Street* sequence at QP=28

Fig. 13. PSNR distribution of the HM and the PDCM for the *Newspaper* and *Poznan_Street* sequence.

The performance is then evaluated for the whole range of QPs used in this experiment. The results of four sequences (two from each resolution types) are demonstrated in Fig. 14 for further analysis. In the Figure, the RD performance comparison curves of both techniques are presented using four QPs (i.e. 20, 24, 28, 32) for better visualization. The Figure also reports minimum to the maximum PSNR difference obtained by the PDCM (with the HM). The PDCM obtained {min ~ max} PSNR difference (i.e. ΔPSNR) values for *Newspaper*, *Lovebird1*, *GT_Fly*, and *Poznan_Street* are {0.03dB ~ 0.19dB}, {0.06dB ~ 0.31dB}, {0.04dB ~ 0.43dB}, and {0.03dB ~ 0.28dB} respectively. Thus, the maximum achievable ΔPSNR of the PDCM is 0.43dB against the HM (e.g. at the bit-rate 790 (Kbps) for the *GT_Fly* sequence in Fig. 14. Thus, the outcomes of the Figure show relatively improved RD performance of the PDCM compared to the mode selection approach in the HM.

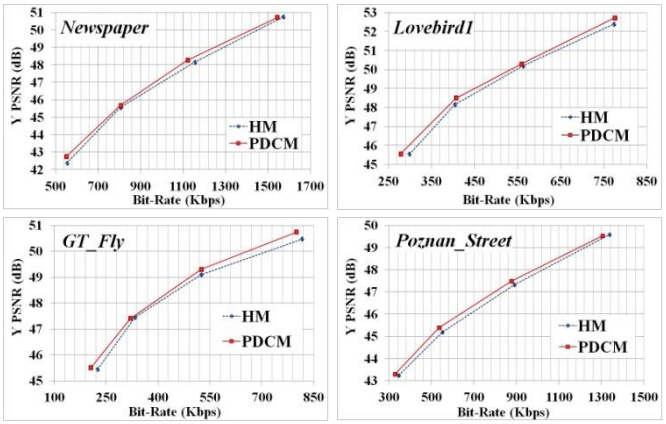


Fig. 14. RD performance comparison curves for four sequences obtained by the HM and the PDCM.

For additional performance analysis, we present the comparison results of the proposed technique against the HM in terms of BD-PSNR and BD-BR in Table 3 where '+' and '-' sign indicate the increment and decrement respectively. The table also presents the resolutions of the sequences. For the 1024×768 resolution sequences, the average values of BD-PSNR and BD-BR are +0.10dB and -0.61% respectively which is higher than the average values obtained for the 1920×1088 resolution sequences. However, the finally calculated results of Table 3 reveal an overall 0.10dB BD-PSNR gain of the PDCM, while, decreasing 0.59% BD-BR on average compared to the mode selection approach in the HM.

Table 3. Performance comparison of the PDCM against the HM15.0 in terms of BD-PSNR and BD-BR; numeric values with + and – signs indicate increment and decrement respectively.

Sequences	Resolutions	BD-PSNR (dB)	BD-BR (%)
Newspaper	1024×768	+ 0.12	- 0.69
Lovebird1	1024×768	+ 0.10	- 0.64
Kendo	1024×768	+ 0.08	- 0.53
Balloons	1024×768	+ 0.09	- 0.59
Average		**+ 0.10**	**- 0.61**
GT_Fly	1920×1088	+ 0.13	- 0.70
Poznan_Street	1920×1088	+ 0.11	- 0.67
Poznan_Hall	1920×1088	+ 0.09	- 0.56
Poznan_Carpark	1920×1088	+ 0.07	- 0.46
Undo_Dancer	1920×1088	+ 0.08	- 0.54
Average		**+ 0.09**	**- 0.58**
Overall-average		**+ 0.10**	**- 0.59**

Fig. 15 illustrates the implication of using ω= 4 in the PDCM. It is experimentally observed that if we gradually decrease the value of ω (e.g. 2), the percentage of Pattern Mode selection increases. Thus, their bit-rate requirement is less, however, the residue still remain large that leads to a large distortion. As a result, the overall calculated value of $j(m)$- in equation (1) loses its suitability to determine the Pattern Mode as the final mode for a block. Eventually, the RD performance becomes inferior. For ω=6, in contrast, the percentage of Pattern Modes selection decreases sharply. Therefore, due to the selection of other existing modes (available in the HEVC standard), the bit-rate requirement becomes much higher, however, since the amount of distortion does not reduce significantly, it leads to the inferior RD performance gain. This trend is noticed almost for all the sequences. In fact, this multiplication results in execution of λ×4×R, which significantly increases the overall calculated value of residue in a block. Thus, even the largest possible residuals could also be handled while selecting the Pattern Mode. Moreover, employing ω=4, we ensure the selection of Pattern mode only for those blocks in which motions are properly aligned with the pattern templates. The Figure also confirms that using both ω={4, 5} the proposed technique could produce more improved RD performance compared to the HM, however, as ω=4 obtains its apex, we eventually employ it in the proposed experiment.

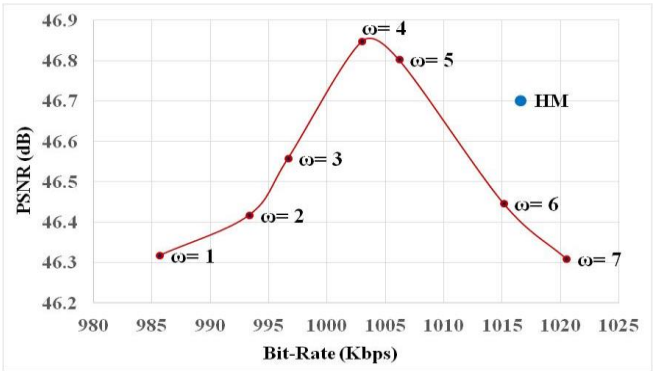


Fig. 15. RD performance outcomes from using different weights to the LM, where ω=4 outperforms all other weight combinations to obtain better RD performance that we use in the PDCM.

Now we provide an analysis regarding the impact of patterns and the selection of Pattern Mode. Fig. 16 illustrates the RD performance comparison curves for two sequences (one from each resolution type) obtained by the HM, the AWP, and the AWNP. It is obvious that the AWNP demonstrates inferior RD performance

in most cases; i.e. sacrifice 0.06dB and 0.09dB BD-PSNR compared to the HM for the *Newspaper* and *GT_Fly* sequence. Conversely, the AWP improves the RD performance with the HM especially caring about the irregular depth motions and selecting more appropriate block partitioning modes (including the average selection of 9.63% distinct Pattern Mode). Thus, it could obtain 0.11dB and 0.13dB BD-PSNR gain while reducing 38.65% and 24.17% encoding time respectively. However, by employing the AWNP based approach, much higher computational time savings could be obtained; i.e. 59.36% and 42.32% which are 20.71% and 18.15% more for the *Newspaper* and *GT_Fly* sequence respectively. The proposed method is developed considering and reviewing all the points of aforesaid discussion and integrating the AWP with it.

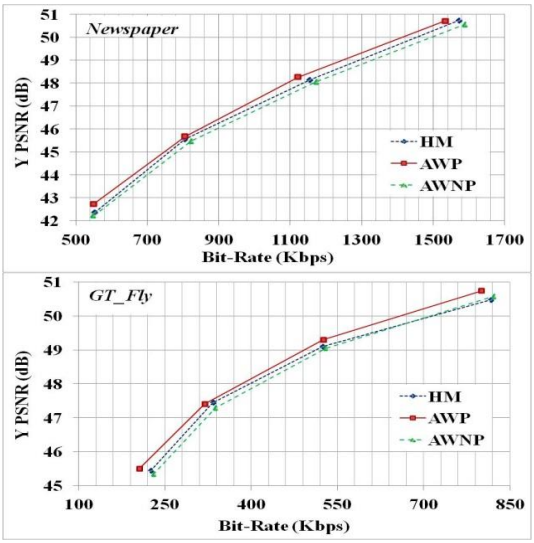


Fig. 16. RD performance comparison curves of two sequences obtained by the HM, the AWP (algorithm with patterns), and the AWNP (algorithm with no pattern).

For more particular performance analysis, we compare different fast mode selection methods (Table 4) with the PDCM using the sequences for the RD curve generation in Fig. 14**.** The technique adopted by Yeh *et al*. [39] sacrifice on average 0.10dB BD-PSNR and increases 2.66% BD-BR. These values go up to 0.23dB and 6.66% respectively for the *Poznan_Street* sequence. However, this method could save 65.11% average encoding time compared to the JMVC 8.5 for those four sequences [40]. On the other hand, Lei's method performs almost similar to JMVC 8.5 where they only sacrifice on average 0.01dB BD-PSNR decrement and 0.34% BD-BR increment, while the average computational time saving is 77.56% [40].

Table 4. Performance comparison of the PDCM with two recent methods of depth video compression for four sequences.

Sequence	Yeh vs. JMVC		Lei vs. JMVC		PDCM vs. HM	
	BD-PSNR	BD-BR	BD-PSNR	BD-BR	BD-PSNR	BD-BR
Newspaper	-0.03	+ 0.69	-0.03	+ 0.63	+ 0.12	-0.69
Lovebird1	-0.10	+ 1.54	+ 0.01	-0.31	+ 0.09	-0.61
GT_Fly	-0.07	+ 1.77	-0.00	+ 0.21	+ 0.11	-0.65
Poznan_Street	-0.23	+ 6.66	-0.02	+ 0.84	+ 0.08	-0.51
Average	**-0.10**	**+ 2.66**	**-0.01**	**+ 0.34**	**+ 0.11**	**-0.67**

Compared to the approaches of Yeh [39] or Lei [40], the proposed technique achieves lower percentage of encoding time savings for those four sequences (31.25%) against the HM15.0 as shown in Fig. 12. However, for the same sequences, the proposed technique notably obtains 0.11dB BD-PSNR gain and 0.67% BD-BR reduction on average compared to the HM15.0.

4.5. Overall Performance Analysis

Table 5 reveals the performance comparison results of the PDCM against five coding techniques recently presented. It can be seen that Yeh's and Lei's methods provide almost the similar performance in terms of BD-PSNR and BD-BR compared to other existing methods. They also save on average 76.65% and 78.07% computational time compared to the JMVC 4.0 and JMVC 8.5 respectively. In contrast, Shen's method performs better compared to Pan's method. Among the existing five schemes, the technique presented by Li *et al*. [37] performs best in terms of BD-PSNR improvement, however it incurs with the encoding time increment of over 7.10% that may limit its uses for a number of faster coding applications. Moreover, the methods presented from [36]-[40] have been developed based on different versions of the JM (i.e. based on H.264) which could not be straight-forward applied to different versions of the HM (i.e. based on HEVC) due to a number of constraints as discussed in Section-2. The proposed strategy-PDCM on the other hand has been developed based on the HEVC standard.

Although the encoding time saving of the PDCM is lower than the state-of-the-art methods presented in [36], [38], [39], and [40], the PDCM outperforms all these methods in terms of both improving the BD-PSNR i.e. +0.10dB and reducing the BD-BR i.e. -0.59% on average. On the other hand, the methods presented in [37] or [33] (discussed in Section-2) outperforms the PDCM in terms of BD-PSNR gain. However, these approaches incur with the limitation of increased encoding time. Since the PDCM targets to the time saving without sacrificing quality, it goes through an independent depth coding process and reduces significant percentage of encoding time compared to [33] or [37]. Since the proposed technique also shows relatively improved coding quality compared to the HM, it could be employed for a number of electronic devices with limited processing and computational resources to use different features of the HEVC standard. However, similar approach of [33] or [37] could also be integrated with the existing PDCM for its further coding gain.

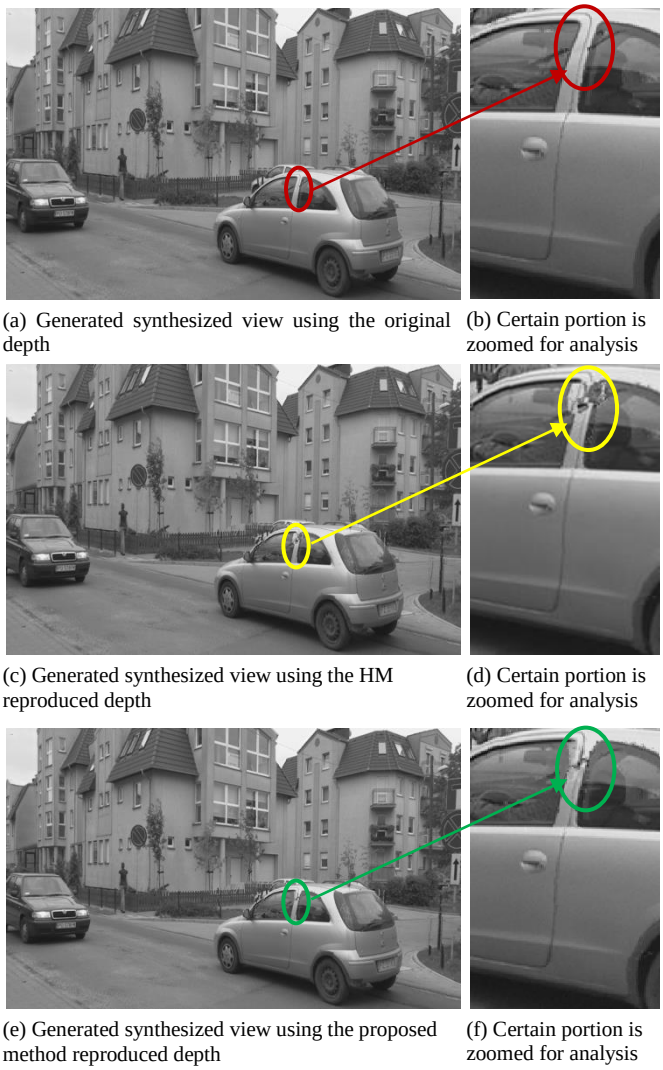
Table 5. Overall performance analysis of different fast depth coding algorithms.

Algorithms	BD-PSNR (dB)	BD-BR (%)	ΔT_S (%)	Videos and Coder Used
Pan et al. [36], 2013	- 0.10	+ 2.14	68.54	06 (JM)
Li et al. [37], 2014	+ 0.83	- 26.28	- 7.14	05 (JM)
Shen et al. [38], 2014	- 0.08	+ 0.60	80.20	06 (JM)
Yeh et al. [39], 2014	- 0.07	+ 0.26	76.65	06 (JM)
Lei et al. [40], 2015	- 0.06	+ 0.38	78.07	06 (JM)
PDCM	+ 0.10	- 0.59	29.06	09 (HM)

4.6. Performance on Synthesized Views

Only independently improved depth map or its corresponding PSNR may not always promise the improved synthesized view. Therefore, we further test the performance of the HM and PDCM on synthesized views. An example of view synthesis is carried out by exploiting the view information from Cam-3 and Cam-5 to generate the view of Cam-4. The outcome is presented in Fig. 17 by employing the *Poznan_Street* sequence (with resolution 1920×1088) and its frame-5 is taken as a random selection. In the Figure, (a), (c) and (e) represent the generated synthesized views by employing the original depth, the HM and PDCM reproduced depth respectively, while placing the original texture in all cases. In most of the areas, they look like very similar at a glance. However, the zooming impact could further help us to distinguish

them from each other. For assessing the image quality, let us first concentrate to the zoomed side view of the car indicated by Red ellipse of Fig. 17 (b) which is almost a defect-less synthesized view achieved by employing the original depth. However, once the same image is presented by the HM reproduced depth (Yellow ellipse in Fig. 17 (d)), it clearly lacks the proper correspondence with (b) in the Figure. This is perhaps the HM generated reproduced depth is not always good enough to generate the synthesized view similar to that one in (b). Although the proposed technique (i.e. Green ellipse in Fig. 17 (f)) could not reveal analogous results with the original in (b), however, it could outperform (d) in terms of quality conservation for the indicated image part. Note that to generate the synthesized views, we first go through the image warping technique and then apply the inverse mapping as a simple post processing filtering. To this end, once the average PSNR of the synthesized views is calculated, the proposed technique could also reveal 0.06dB progress compared to the HM (i.e. 24.63dB and 24.69dB for the HM proposed method respectively).



(a) Generated synthesized view using the original depth

(b) Certain portion is zoomed for analysis

(c) Generated synthesized view using the HM reproduced depth

(d) Certain portion is zoomed for analysis

(e) Generated synthesized view using the proposed method reproduced depth

(f) Certain portion is zoomed for analysis

Fig. 17. Generated synthesized views for the whole frame using the original depth (in (a)), HM reproduced depth (in (c)) and the proposed method reproduced depth (in (e)). For additional analysis with better visualization, their corresponding zoomed view for a specific part is further demonstrated in (b), (d) and (f) respectively.

5. Conclusions

The structure of existing inter-modes in the HEVC standard would not be suitable enough for the block partitioning of depth object having partial foreground motion with irregular edges and background. In such cases, the HEVC reference test model (HM) normally explores finer level block partitioning that require more bits and computational time to compensate large residuals. In this work, the proposed technique uses the energy concentration ratio feature of phase correlation to capture various types of motion in depth object. For more appropriate depth motion modeling, it also exploits an extra Pattern Mode comprising a group of pattern templates with different rectangular and non-rectangular object shapes and edges. Since the Pattern Mode saves bits by encoding only the foreground areas and beat all other inter-modes in a block once selected, the proposed technique could improve the rate distortion performance. Using the Pattern Mode, it could also reduce encoding time by avoiding further branching and selecting a subset of modes using innovative preprocessing motion criteria. Experimental results reveal that the proposed technique could save 29.06% (23.5%~36.12%) encoding time, while improving average 0.10dB (0.07~0.13) BD-PSNR compared to the HM. Work is undergoing towards aiming further amendment of the proposed strategy and then assessing its improved performance with the 3D HEVC.

References

[1] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H. 264/MPEG-4 AVC standard," *Proc. IEEE*, vol. 99, no. 4, pp. 626–642, April 2011.

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

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

[4] D. M. Rahman and M. Paul, "Free View-Point Video Synthesis Using Gaussian Mixture Modeling," accepted in the *IEEE 30th International Conference on Image and Vision Computing New Zealand*, 2015.

[5] H. Liu,M. Yuan, F. Sun, and J. Zhang, "Spatial neighborhood-constrained linear coding for visual object tracking," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 469–480, February 2014.

[6] Y. Fang, J. Wang, M. Narwaria, P. L. Callet, and W. Lin, "Saliency detection for stereoscopic images," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2625–2636, June 2014.

[7] H. Liu, S. Chen, and N. Kubota, "Intelligent video systems and analytics: A survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1222–1233, August 2013.

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

[9] A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems," *IEEE International Conference on Image Processing*, pp. 2448–2451, October 2008.

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

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

[12] G. J. Sullivan, J. R. Ohm, W. J Han, and T. Wiegand "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, 2012.

[13] B. Bross, Han, W. J. Ohm, J. R. Sullivan, and G. J. Wiegand, "High Efficiency Video Coding Text Speciation draft 8" JCTVC- L1003, 2012.

[14] 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, 2003.

[15] M. Jiang and N. Ling "On Lagrange multiplier and quantizer adjustment for H.264 frame-layer video rate control" *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 5, May 2006.

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

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

[18] M. Paul and M. Murshed, "Video coding focusing on block partitioning and occlusions," *IEEE Transactions on Image Processing*, vol. 19, no. 3, pp. 691-701, February 2010.

[19] M. Paul and M. Murshed, "Video coding using arbitrarily shaped block partitions in globally optimal perspective," *Eurasip Journal on Advances in Signal Processing*, 2011, doi:10.1186/1687-6180-2011-16

[20] S. Shahriyar, M. Ali, M. Murshed, and M. Paul, "Efficient coding of depth map by exploiting temporal correlations," *IEEE International conference on Digital Image Computing: Techniques and Applications*, pp. 1-8, November 2014.

[21] O. D. Escoda, P. Yin, and C. Gomila, "Hierarchical B-frame results on geometry-adaptive block partitioning," presented at the VCEG-AH16 Proposal, ITU/SG16/Q6/VCEG, Antalya, Turkey, January 2008.

[22] S. Chen, Q. Sun, X. Wu, and L. Yu, "L-shaped segmentations in motion-compensated prediction of H.264," *IEEE International Conference on Circuits and Systems*, 2008.

[23] J. Chen, S. Lee, K. H. Lee, and W. J. Han, "Object boundary based motion partition for video coding," *Picture Coding Symposium*, 2007.

[24] J. H. Kim, A. Ortega, P. Yin, P. Pandit, and C. Gomila, "Motion compensation based on implicit block segmentation," *IEEE International Conference on Image Processing*, 2008.

[25] M. Paul, M. Murshed, and L. S. Dooley, "A real-time pattern selection algorithm for very low bit-rate video coding using relevance and similarity metrics," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 6, pp. 753-761, June 2005.

[26] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression," *IEEE International Conference on Image Processing*, pp. 1541–1544, October 2012.

[27] J. Lei, S. Li, C. Zhu, M. T. Sun, C. Hou, "Depth Coding Based on Depth-Texture Motion and Structure Similarities," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 2, pp. 275-286, February 2015.

[28] M. K. Kang and Y. S. Ho, "Depth video coding using adaptive geometry based intra prediction for 3-D video systems," *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 121–128, 2012.

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

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

[31] P. Merkle, K. Muller, D. Marpe, and T. Wiegand, "Depth Intra Coding for 3D Video based on Geometric Primitives", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 570-582, March 2016.

[32] X. H. Van, J. Park, and B. Jeon, "A probablistic intra mode decision in distributed video coding," *IEEE International Conference on Systems, Signals and Image Processing*, pp. 380-383, April 2012.

[33] S. Liu, P. Lai, D. Tian, and C. W. Chen, "New Depth Coding Technique With Utilization of Corresponding Video," *IEEE transactions on Broadcasting*, vol.57, no. 2, pp. 551-561, June 2011.

[34] Y. H. Lin and J. Ling, "A Depth Information Based Fast Mode Decision Algorithm for Color Plus Depth-Map 3D Videos," *IEEE transactions on Broadcasting*, vol.57, no. 2, pp. 542-550, June 2011.

[35] Y. Zhang, S. Kwong, L. Xu, and G. Jiang, "DIRECT Mode Early Decision Optimization Based on Rate Distortion Cost Property and Inter-view Correlation," *IEEE transactions on Broadcasting*, vol.59, no. 2, pp. 390-398, June 2013.

[36] Z. Pan, Y. Zhang, and S. Kwong, "Fast Mode Decision Based on Texture-Depth Correlation and Motion Prediction for Multiview Depth Video Coding," *Journal of Real-Time Image Processing*, pp. 1–10, March 2013, doi: 10.1007/s11554-013-0328-3.

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

[38] L. Shen, P. An, Z. Liu, and Z. Zhang, "Low Complexity Depth Coding Assisted by Coding Information From Color Video," *IEEE Transactions on Broadcasting*, vol. 60, no.1, pp. 128-133, March 2014.

[39] C. H. Yeh, M. F. Li, M. J. Chen, M. C. Chi, X. X. Huang, and H. W. Chi, "Fast Mode Decision Algorithm Through Inter-View Rate-Distortion Prediction for Multiview Video Coding System," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 594–603, February 2014.

[40] J. Lei, J. Sun, Z. Pan, S. Kwong, J. Duan, and C. Hou, "Fast Mode Decision Using Inter-View and Inter-Component Correlations for Multiview Depth Video Coding," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 4, pp. 978–986, August 2015.

[41] P. Podder, M. Paul and M. Murshed, "A novel depth edge prioritization based coding technique to boost-up HEVC performance" *IEEE International Conference on Multimedia and Expo Workshops*, July 2016.

[42] M. Paul, W. Lin, C. T. Lau, and B. S. Lee, "Direct Inter-Mode Selection for H.264 Video Coding using Phase Correlation," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 461 – 473, 2011.

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