Defence Science Journal, Vol. 72, No. 1, January 2022, pp. 56-66, DOI : 10.14429/dsj.72.16733

A Motion Estimation based Algorithm for Encoding Time Reduction in HEVC

S. Karthik Sairam* and P. Muralidhar

National Institute of Technology, Warangal - 506 004, India

**E-mail: karthik_sai_ram@yahoo.com*

ABSTRACT

High Efficiency Video Coding (HEVC) is a video compression standard that offers 50% more efficiency at the expense of high encoding time contrasted with the H.264 Advanced Video Coding (AVC) standard. The encoding time must be reduced to satisfy the needs of real-time applications. This paper has proposed the Multi-Level Resolution Vertical Subsampling (MLRVS) algorithm to reduce the encoding time. The vertical subsampling minimises the number of Sum of Absolute Difference (SAD) computations during the motion estimation process. The complexity reduction algorithm is also used for fast coding the coefficients of the quantised block using a flag decision. Two distinct search patterns are suggested: New Cross Diamond Diamond (NCDD) and New Cross Diamond Hexagonal (NCDH) search patterns, which reduce the time needed to locate the motion vectors. In this paper, the MLRVS algorithm with NCDD and MLRVS algorithm with NCDH search patterns are simulated separately and analysed. The results show that the encoding time of the encoder is decreased by 55% with MLRVS algorithm using NCDD search pattern and 56% with MLRVS using NCDH search pattern compared to HM16.5 with Test Zone (TZ) search algorithm. These results are achieved with a slight increase in bit rate and negligible deterioration in output video quality.

Keywords: Test zone search; Vertical subsampling; Encoding time; Search pattern; Complexity; Motion vector

NOMENCLATURE

D_{mp}	Distortion
λ_{pred}	Lagrangian multiplier
R_{mp}	Number of bits needed to transmit 'mp'
BD	Best Distance
MV	Motion Vector
BMV	Best Motion Vector
D_b	Best distance
S_r	Search range
P	Prediction block
HR	Half Resolution frame
QR	Quarter Resolution frame
RD	Rate-Distortion
J	Rate-Distortion cost
VS	Vertical Subsampling
J_B	RD cost of the best coding unit
J_C	RD cost of the current coding unit
J_m	RD cost of the merge mode
J_s	RD cost of the skip mode
J_{md}	Merge mode Rate-Distortion cost at depth d
J_{sd}	Skip mode Rate-Distortion cost at depth d
CBF	Coded Block Flag
NCDD	New Cross Diamond Diamond
NCDH	New Cross Diamond Hexagonal
CFM	Coded Block Flag Fast Method
ECU	Early Coding Unit termination
ESD	Early Skip Detection
GOP	Group of Pictures

Received : 29 January 2021, Revised : 28 November 2021
Accepted : 30 November 2021, Online published : 05 January 2022

1. INTRODUCTION

The High Efficiency Video Coding (HEVC) or H.265[1,2] standard compresses ultra high definition video sequences with approximately 50% less bitrate than the H.264/Advanced Video Coding[3] standard while maintaining the same video quality[4]. The video compression involves splitting the frame into Coding Tree Units (CTUs), intra and inter predictions, finding transform and quantisation for the residual block, and filtering operations using deblocking[5] and Sample Adaptive Offset (SAO)[6] filters. HEVC is widely used in ultra high definition online video streaming and surveillance applications, which are shown in Fig. 1. In HEVC, complexity increases along with an increase in efficiency. In HEVC, most of the time is consumed during the process of finding the Rate-Distortion (RD) cost for Prediction Units (PU), and the motion estimation process. Several authors have developed different algorithms to lessen the encoding time of the video encoder. Hsieh[7], *et al.* developed a power-efficient motion estimation controller to reduce the power dissipation. The dissipation is due to the large coding bandwidth required to access the current or reference pixel values during the motion estimation process. Vayalil[8], *et al*. proposed a hardware implementation system for the improved TZ search algorithm. This method uses the snake scan to get the data of a row or column. In addition, the residue number system is used to improve the speed of the Sum of Absolute Difference (SAD) calculations. This approach helps to decrease the encoding time. Cebrián-Márquez[9], *et al.* uses the pre-analysis stage, which performs block-based motion estimation to estimate Rate-Distortion

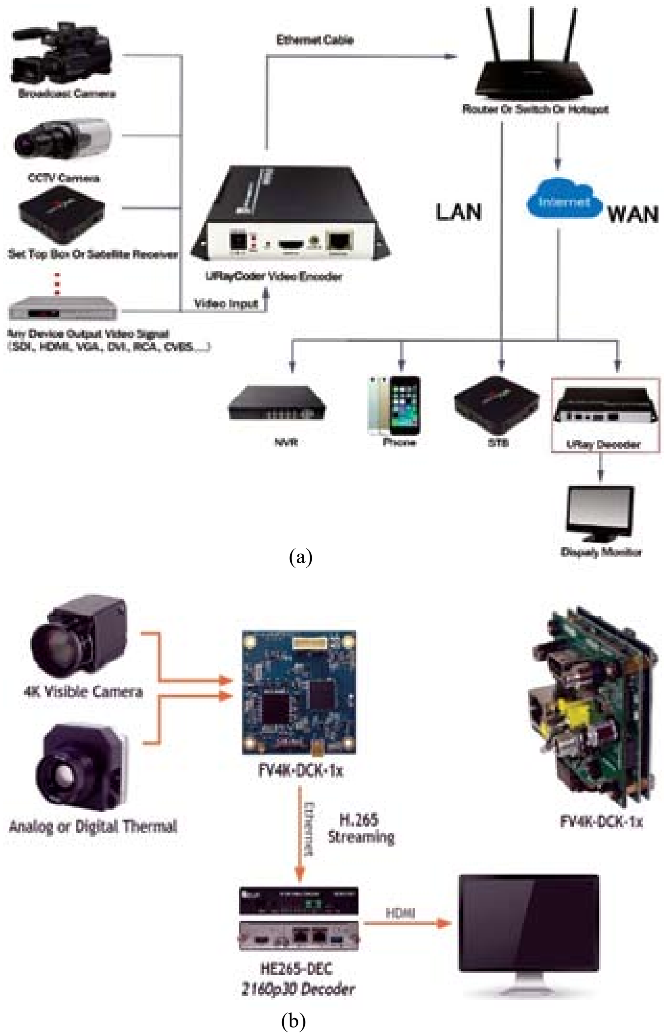


Figure 1. Applications of HEVC in (a) Online video streaming and (b) Surveillance.

cost. The estimated cost is used to build the optimal quad-tree by omitting a large number of unnecessary partitions, which reduces the encoding time. Fan[10], *et al.* take the motion vector of the conventional HEVC merge mode as a center and applies the motion estimation process around it and along the axis in a small search region. This approach improves bitrate saving. However, the encoding time is increased. The Test Zone (TZ) search algorithm in HEVC uses multiple search points at the start, making it difficult for real-time implementation. Pakdaman[11], *et al.* uses a single search point at starting of the TZ search algorithm. The single search point is obtained by using the wavelet transform to analyse the current and reference frames. After analysing, similar points are identified and matched to determine the single search point. Jiang[12], *et al.* proposed the approach to predict the optimised motion vectors by utilizing the motion consistency of the adjoining PUs. Similarly, the spatial correlation of neighboring CUs can be utilised to forecast the depth of the current CU. Gogoi and Peesapati[13], *et al.* proposed a hardware architecture for motion estimation using a hybrid search pattern. The hybrid search pattern consists of hexagonal and square global patterns and two, three, and four-point local search patterns. This method minimises the encoding time by 11%. Bouaafia[14], *et al.* uses the Support Vector Machine (SVM) and Convolutional Neural Network (CNN) approaches to predict the CU partitions during the Rate-Distortion Optimisation (RDO) search process. This approach reduces the encoding time. However, machine learning approaches are less suitable for real-time applications due to their high computational complexity. Erabadda[15], *et al.* use the SVMs to classify the CU. The SVM is trained by using the texture and context features of the coding unit. In addition, the Bayesian probabilistic model is employed to improve the accuracy of the CU split decision. However, the SVM is trained with only five video sequences. Training with minimal data leads to the inaccurate prediction of CTU structure. Kuo[16], *et al.* suggested an approach that lowers the complexity of the RDO search process by predicting the CU depth using the neighboring and co-located CU depth range. After determining the depth range, the context and texture information is used for early termination of the RDO search process and correcting the depth range prediction error. The early termination process decreases the encoding time. Huang[17], *et al.* proposed the RD complexity optimisation scheme to preselect the CU depth and speed up the Transform Unit (TU) tree decision process. Moreover, the early Prediction Unit (PU) and CU termination algorithms are provided to decrease the encoding time. Lu[18], *et al.* minimises the complexity of the encoder by generating the classification trees. The trees are generated by using the intra and inter features obtained after encoding using the conventional HEVC algorithm. The features provide the context and texture properties of PU, CU, and TU. Mallikarachchi[19], *et al.* developed the online trained content-adaptive models to identify the CU size quickly. Moreover, the motion vector reuse scheme is introduced to lessen the encoder's complexity. However, the models use limited data during the training process, providing unreliable results for certain features. Sharma and Arya[20], optimise the parameters of the HEVC using the Non-dominated sorting genetic algorithm II to improve the compressed video quality. This approach concentrates on increasing the quality of video and decreasing the file size. Yan[21], *et al.* reduce the complexity of the intra prediction by using the statistics of the rough mode decision method. In this process, the number of most probable modes is decreased based on the PU size. The decrease in the most probable modes decreases the complexity of the encoder.

Moreno[22], *et al.* have presented an algorithm that minimises the encoder complexity by deciding the CU size based on the early termination condition. Kim[23], *et al.* have proposed a method that bypasses the interpolation process of list 1 when the bi-predicted motion data of list 0 and list 1 are the same. This strategy lessens the intricacy of encoder and decoder. Lee[24], *et al.* have described an early skip mode scheme to reduce the encoder's coding time. Ahn[25], *et al.* use the spatial and temporal parameters to reduce the encoder's coding time. Here, the decision of subdividing the CU is taken based on the motion and texture complexity. Purnachand[26], *et al.* have developed an algorithm that omits the global search step only when the cost difference between the Initial Search Point and the current bock is lower than the threshold. The threshold value in this case is the lowest cost of temporal and spatially co-located blocks. By using this method and rotating hexagonal search pattern, the complexity of the encoder is decreased. Rui Fan[27], *et al.* suggested a technique that utilises the Priority Guided Fast Partial Internal Early Termination algorithm and motion complexity. The PU is categorised here based on motion, i.e., smooth, medium, or complex motion. Pan[28], *et al.* have presented a new algorithm called adaptive Fractional Pixel Motion Estimation skipped algorithm. Here the children type PUs can be encoded based on the best motion vector[29] of root PU using Integer Pixel Motion Estimation. Shen[30], *et al.* suggested an algorithm that reduces complexity by skipping prediction modes that are not prevalently used at higher depths of the CU. The previously discussed algorithms use Three Step Search (TSS)[31], improvements in TSS[32-34], logarithmic search[35], One dimension full search[36], etc., to speed up the motion estimation process. These algorithms may reduce complexity by reducing the number of search points. However, there is a possibility of converging to local minima due to the early termination of the searching process. We have developed a Multi-Level Resolution Vertical Subsampling (MLRVS) algorithm to prevent the early termination of the searching process and improve the motion estimation speed.

In this paper, the MLRVS algorithm is proposed, which uses vertical subsampling and the complexity reduction algorithm to reduce the encoding time of the encoder. In addition, New Cross Diamond Diamond (NCDD) and New Cross Diamond Hexagonal (NCDH) search patterns are proposed to accelerate the motion estimation process.

2. OVERVIEW OF MOTION ESTIMATION PROCESS DURING INTER PREDICTION IN HEVC

In HEVC, each frame is segmented into CTUs. It is possible to subdivide each CTU[37] into coding units or the CTU itself as the CU. The size of CU can be 64, 32, 16, or 8. The CU contains one luma Coding Block (CB) and two associated chroma CBs. Every CU can be additionally partitioned into Prediction Units (PU). The size of PU should be less than or equal to the size of CU. The structure of CTU is shown in Fig. 2. The CTU can be split up to a maximum depth of four.

HEVC supports the PU partition modes like Merge/Skip mode, 2N×2N, 2N×N, N×N, N×2N, nL×2N, nR×2N, 2N×nD, and 2N×nU. The RD cost can be determined for PU partition modes by utilising the Eqn (1).

$$mp^{*} = \arg^{\min}_{mp \in MP} D_{mp} + \lambda_{pred} \times R_{mp} \tag{1}$$

For the reference picture list 'MP,

D_{mp}→ Distortion, R_{mp}→ number of bits needed to transmit *mp*, and λ_{pred}→Lagrangian multiplier.

The distortion or SAD is calculated during the motion estimation process to find the RD cost. Motion estimation in HEVC plays a vital role in decreasing the bit rate for storing or transmitting the video signal. The TZ search algorithm (discussed in section 2.1) is used for motion estimation in HEVC. During the motion estimation process, for every block in the current frame, the appropriate matching block can be found in the previous frame inside the search area. Generally, SAD is the widespread matching criterion to find the distortion, which is used to find the best matching block in the previous frame for the block in the current frame. SAD is obtained by first calculating the absolute difference between each current block pixel and the corresponding pixel in the reference block. Then these differences are summed together to get the final SAD value. The SAD is calculated by using Eqn (2). Then the RD cost is calculated for the partition modes using the SAD value. Among PU modes, the best mode is the mode that has a lower RD cost.

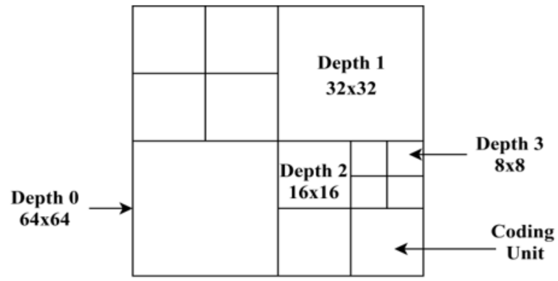


Figure 2. Structure of CTU with coding unit depths ranging from 0 to 3.

$$SAD = \sum_{k,l} \left| S_A(k,l) - S_B(k,l) \right| \tag{2}$$

where, $S_A(k,l)$ → $(k, l)_{\text{th}}$ pixel in current frame-block, $S_B(k,l)$ → $(k, l)_{\text{th}}$ pixel in reference frame-block.

Both current frame-block and reference frame-block are equal in size.

2.1 TZ Search Algorithm

The TZ Search algorithm[38] is explained in the following steps

1. First, calculate the median predictor (discussed in section 2.2).
2. After calculating the median predictor[39], check whether the zero motion vector is the best starting point than the median predictor. Determine the best starting point.
3. Now consider the best starting point as an initial starting point and perform the first search.
4. In the first search, either diamond search[40] or square search patterns can find the best motion vector. Here the search window can have a minimum distance of one to maximum distance of search range. The distance at which the point with minimum distortion occurs is considered as 'Best distance (BD).'
5. Now take the Best distance and check the following three conditions.
   - If the BD is zero, the searching process stops
   - If 1 < BD < iRaster, perform refinement directly.
   - If BD> iRaster, perform a raster scan by taking the value of iRaster as a stride length.

The raster search process can be done on a whole search window if the difference between the starting position and the first phase motion vector is too significant.

6. If the best distance in the previous search is not zero, apply the star or raster refinement. During this refinement stage, the last search's best motion vector is taken as the starting point. Here the distance is in the range of one to search range. Diamond or square search patterns are used in the

refinement process. During star refinement, the distance is multiplied by two in each iteration until it reaches the search range. During raster refinement, the distance can be divided by two in each iteration until it goes one.

2.2 Median calculation in HEVC

In HEVC, the median predictor is obtained

using the Predictors A, B, and C. Predictors A, B, and C are the left, top, and top right predictors for the median predictor, as shown in Fig. 3. The median predictor is calculated by using Eqn (3).

$$Median(A,B,C) = A + B + C - Min(A, Min(B,C)) - Max(A, Max(B,C)) \quad (3)$$

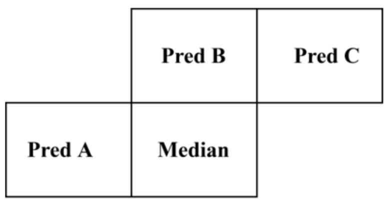


Figure 3. Median predictor prediction using left, top, and top right predictors.

3. PROPOSED WORK

The framework of the proposed research is shown in Fig. 4. The encoding process involves motion estimation, inter prediction, transform, quantisation, and entropy coding[1] to generate the bitstream. This paper proposes the MLRVS algorithm to accelerate the motion estimation process. The algorithm involves vertical subsampling and motion estimation using newly proposed search patterns in the vertical sub-sampled frames. Besides, a complexity reduction algorithm is used during the inter-prediction and quantisation process to reduce the encoding time. The MLRVS algorithm and the complexity reduction algorithm are explained below.

3.1 MLRVS algorithm

The MLRVS algorithm shown in Algorithm 1 is explained in the below steps.

(a) Algorithm 1 MLRVS algorithm

Input: Prediction Block, P; Search region, Sr; Motion Vectors, (MV, M1, M2); zero MV and Neighbours, (MV(0,0) and MVx, MVy, MVz); Frame, Orig; best Distance (Db); Distance, (D1, D2, D3);

Output: Best Motion Vector (BMV)

1. ***Initialisation****: MV=(0,0); TotalCost=∞*
2. *(Start Prediction)*
3. **for** $t_{mp}MV \in (MV, MV_x, MV_y, MV_z)$ **do**
4. $t_{mp}MV = getCost(t_{mp}MV, S_r, P);$
5. **if** $t_{mp}Cost < TotalCost$ **then**
6. $Cost = t_{mp}Cost;\ MV = t_{mp}MV;$
7. **end if**
8. **end for**
9. $D_b = \{1,2,4\};$
10. *HalfResolutionframe(HR)= evenRows(Orig);*
11. *QuarterResolutionframe(QR)=evenRows(HR);*
12. $(D1,M1) = SearchPattern(Db, MV, S_r, P, QR);$
13. $(D2,M2) = SearchPattern(D1, M1, \frac{S_r}{2}, P, HR);$
14. $(D3, BMV) = SearchPattern(D2, M2, \frac{S_r}{4}, P, Orig);$
→(Search pattern can be **NCDD** or **NCDH)**
15. $D_b = D3;$
16. (End Prediction)
17. if $D_b = 0$ **then**
18. Stop the searching process
19. Else
20. Perform refinement operation
21. endif

Step1: Find the median predictor using Eqn (3).

Step2: After median prediction, extract the Half Resolution (HR) and Quarter Resolution (QR) frames using the frame extraction process. To create the HR and QR frames, vertical subsampling is used. The representation of the vertical subsampling frame extraction process can be observed in Fig. 5.

Initially, the original frame (Orig) of (M×N) size is taken and subsampled. M and N represent the number of rows and columns of the frame. The vertical subsampling is used to reduce the resolution of the original frame. The HR frame, which is of size (M/2 × N), is obtained by considering the original frame's even rows, and the QR frame is obtained by considering the even rows of the HR frame.

Step 3: After extracting QR and HR frames, apply the motion estimation process to find the Best Motion Vector (BMV). In this algorithm, the BMV is obtained by first

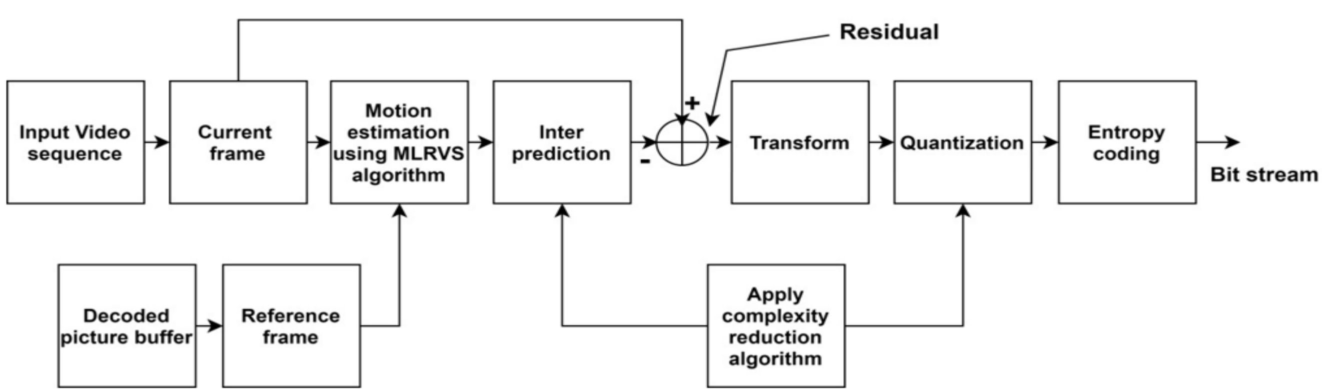


Figure 4. The framework of the proposed research.

Figure 5. Frame extraction process.

calculating the Motion Vector M1 in the QR frame. We use the search patterns like NCDD or NCDH to find the motion vector. After the searching process, take the M1 of the QR frame as an initial search point in the HR frame and find the Motion Vector M2 in the HR frame. Finally, take the M2 of the HR frame as an initial search point in the original frame and find the original frame's Best Motion Vector (BMV) using the search pattern.

The advantage of this process is the possibility of converging towards local minima is significantly less.

The Vertical Subsampling (VS) with SAD computations are obtained by using Eqn (4).

$$VS = \sum_{i=0}^{\frac{M}{2}-1} \sum_{j=0}^{N-1} \left| P(2i, j) - Q(2i, j) \right| \tag{4}$$

where M = the Total number of rows in a block, N = Total no of columns in a block, P = original block, Q = reference block.

(b) Complexity reduction algorithm

This section presents the complexity reduction algorithm to decrease the H.265 encoding time. The flowchart representing the complexity reduction algorithm is shown in Fig. 6. The CUs of size 2N×2N at each depth are taken, where N can be 4, 8, 16, or 32. Then the RD cost (J) is measured by using Eqn (5).

$$J = D + \lambda \times R \tag{5}$$

where, $R \rightarrow$ Number of bits required to transmit, $\lambda \rightarrow$ Lagrangian multiplier, D =Distortion.

Distortion is obtained by calculating SAD between the original frame-block and reference frame-block, shown in Eqn (6).

$$Distortion = \sum_{e,f} \left| S_A(e,f) - S_B(e,f) \right| \tag{6}$$

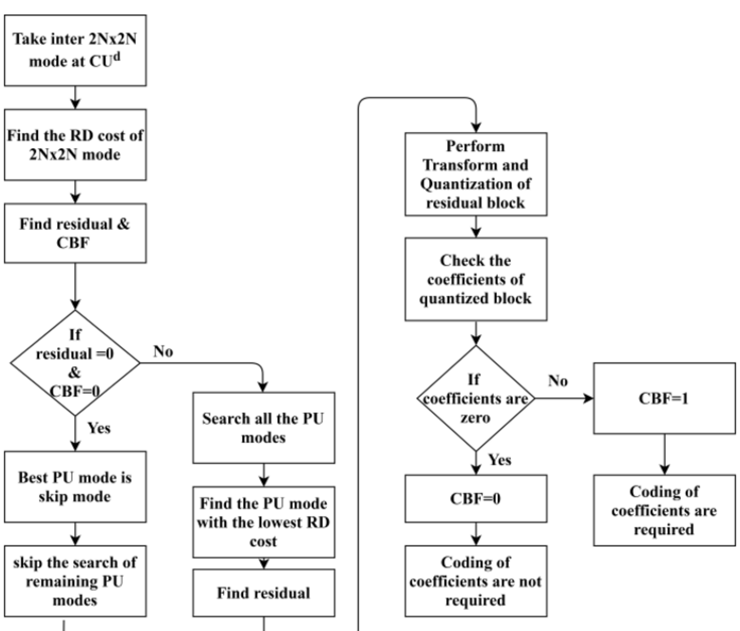


Figure 6. Flowchart of the complexity reduction algorithm.

where, $S_A(e,f) \rightarrow (e,f)_{th}$ pixel in the current frame-block, $S_B(e,f) \rightarrow (e,f)_{th}$ pixel in the reference frame-block.

Let J_C represents the RD cost of Current CU, and J_B represent the RD cost of Best CU. Here, the Best CU is the CU having lower RD cost. The Best CU cost is determined based on the Eqn (7).

$$J_B = \begin{cases} J_C, & if\ J_C < J_B \\ J_B, & otherwise \end{cases} \tag{7}$$

Now update the prediction data and reconstruction data. After finding the RD cost, check whether the RD cost of the skip (J_s) is less than the RD cost of merge mode (J_m) or not. Let N represents the maximum number of merge candidates and skip candidates. The number of merge candidates is signaled in the slice header. Usually, the N value is five. The merge RD cost at each depth can be calculated using Eqn (8).

$$J_{m_d} = \frac{1}{N} \sum_{k=0}^{N-1} J_{m_{u-k}} \tag{8}$$

Similarly, the skip RD cost J_{s_d} can be calculated using Eqn (9).

$$J_{s_d} = \frac{1}{N} \sum_{k=0}^{N-1} J_{s_{v-p}} \tag{9}$$

Here, d is the current CU depth, u, v represents the number of merge modes, skip modes treated as best PU modes for particular CU size, and $J_{m_{u-k}}$, $J_{s_{v-p}}$ represents the rate-distortion cost of k^{th} merge mode and p^{th} skip mode.

During block merging, the Merge flag specifies that block merging is utilised to get the motion data for PU. Merge index is used for determining the candidate present in the merge list. In block merging, the skip mode with the skip flag is incorporated.

If $J_s < J_m$, then skip the computation of RD cost for the remaining modes. Otherwise, perform the RD computation for all other PU modes.

After performing the RD computations, the early skip condition is checked by gathering the Coded Block Flag (CBF) and residual information. The skip condition is shown in Eqn (10).

$$skip\ mode = \begin{cases} True, & if\,(residual = 0\ and\ CBF = 0) \\ False, & otherwise \end{cases} \tag{10}$$

CBF is used to indicate whether the Transform Block (TB) has any significant non-zero coefficients or not. Generally, after calculating the prediction residual, each CU is divided into TBs. Each TB can be 32×32, 16×16, 8×8, or 4×4 in size. The condition of CBF is shown in Eqn (11).

$$CBF = \begin{cases} 0, & if\ all\ coefficients\ in\ TB\ are\ zero \\ 1, & else \end{cases} \tag{11}$$

The time required for encoding is saved by checking the coefficients of the quantised block. The coefficients are checked by using the CBF. If the block has all zeros, then the coding of that block can be skipped, which saves encoding time.

Let 'earlycu' is the variable used for the determination of the CU early. The 'earlycu' condition is checked by using the Eqn (12).

$$earlycu = \begin{cases} True, & if\ skip(0)\ is\ high \\ False, & otherwise \end{cases} \quad (12)$$

Here' skip (0)' checks the skip flag of the luma component. If the skip flag of the luma component is skipped, it returns true, which means the block is skipped. The advantage of this process is for the skipped CUs, the splitting and finding of RD cost can be avoided, which results in a decrease in encoding time.

(c) Search Patterns

This paper proposes two search patterns: New Cross Diamond Diamond (NCDD) and New Cross Diamond Hexagonal (NCDH) search patterns. In these search patterns, center biased searching is used and also allows halfway search stop. The search patterns are explained below.

New Cross Diamond Diamond (NCDD) search and New Cross Diamond Hexagonal (NCDH) search

In the MLRVS algorithm, the NCDH search pattern is used, which is formed by adding the third stage to the cross diamond hexagonal search[41], as shown in Fig. 8. The NCDD and NCDH patterns shown in Figs. 7 and 8 are explained.

Step 1: Perform a small diamond search by considering the median predictor as an origin (0, 0). Here four points around the origin are considered, with distance one for finding the best motion vector. The four search points are indicated by '•.' If the best motion vector is the same as the origin, then the searching process stops; otherwise, move to step2.

Step 2: Again, consider the median predictor and make it an origin. Now take the four search points indicated by 'Δ' with a distance of two around the origin. If the best motion vector after searching is the same as the origin, the search stops; otherwise, move to step 3.

Step 3: Now consider the two nearby search points indicated by '■' close to the best motion vector of step 2. Here the best motion vector can be found among the three search points, including the best motion vector of step 2.

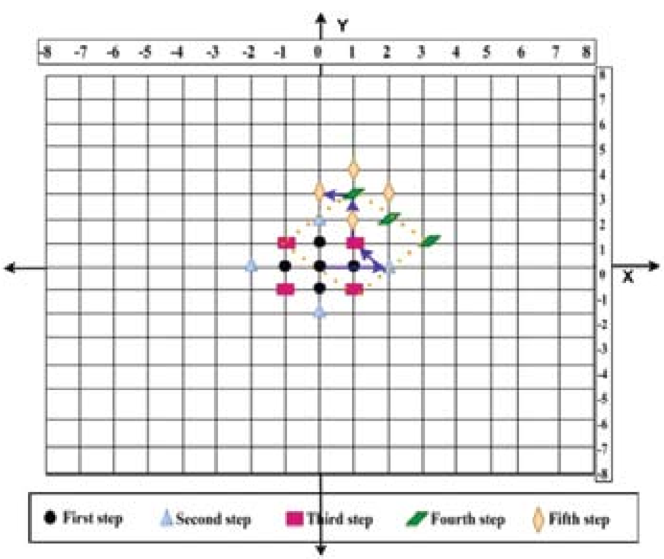


Figure 7. NCDD search pattern.

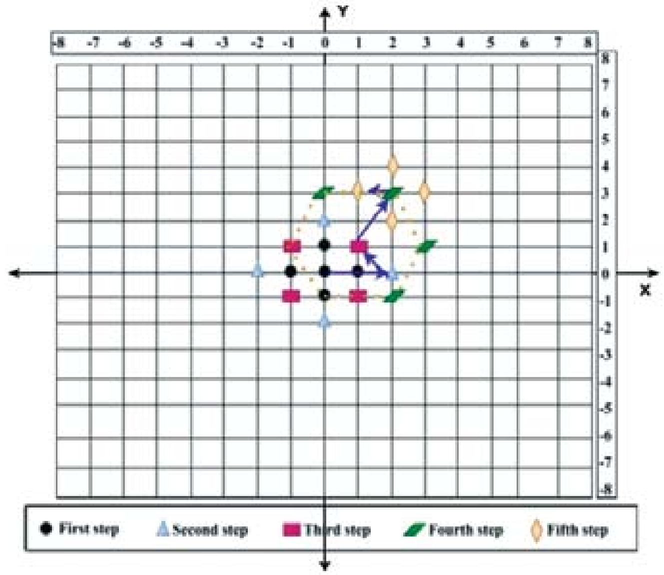


Figure 8. NCDH search pattern.

Step 4: (a) If the search pattern is NCDD, then the eight-point diamond search is applied by considering the best motion vector of step 3 as a center. If the obtained motion vector after searching is the same as the center, stop the searching operation. Otherwise, move to step 5.

(b) If the search pattern is NCDH, then the six-point hexagonal search is applied by considering the best motion vector of step 3 as a center. Stop the searching process if the center point has a minimum distortion value. Otherwise, move to step 5.

Step 5: Perform a small diamond search (like step 1) by considering the obtained motion vector as the center. The point with minimum SAD value is the best matching point.

4. EXPERIMENTAL RESULTS

In this paper, HEVC reference software HM 16.5[42] is used to implement the proposed method. Seventeen different sequences with different resolutions are used to evaluate the output of the proposed method. We also measured the RD performance loss of the proposed algorithm using the Bjontegaard delta bitrate (BD-BR)[43,44] and compared it with the state-of-the-art techniques; Lee[24], *et al.*, Liu[45], *et al.* and Mallikarachchi[19], *et al.* Table 1 show the experimental conditions needed to verify the performance of the MLRVS algorithm. The percentage of Time Saving (TS) can be calculated using Eqn (13).

$$Time\ Saving\ (TS)(\%) = \frac{T_{orig} - T_{prop}}{T_{orig}} \times 100 \quad (13)$$

Table 1. Experimental conditions

Maximum CU and TU size	64×64 and 32×32
Configuration	Encoder_randomaccess_main
QP values	22, 27, 32, and 37
Maximum CU and TU depth	4 and 3
GOP Size	8
Search range	64
Number of frames to be encoded	100

With the fast encoding options, the conventional HM reference software HM 16.5 is used as an anchor method.

As discussed before, the main objective of the proposed method is to minimise the encoding time. The search patterns NCDD or NCDH can be used in place of the proposed method's search pattern. The proposed method using each search pattern is simulated separately and analysed the results. Table 2 compares the proposed method using the NCDD search pattern (Prop +NCDD) with the standard HM 16.5. The findings indicate that the encoding time is reduced by 55% at the cost of a 0.31dB decrease in YPSNR and an 8.06% increase in bit rate. The proposed method using NCDH (Prop+NCDH) search pattern is also simulated and compared with HM 16.5 method. The outcome shows that the approach proposed significantly decreased the encoding time by 56% with minimal video quality degradation, i.e., 0.23dB. The experimental results of the Partyscene video sequence at QP=37 for a proposed method with NCDD and NCDH search patterns are shown in Fig. 9. The proposed method encoded the video sequences with an accuracy of 92%.

Table 3 compares the proposed methods (Prop+NCDD and Prop+NCDH) with Lee[24], *et al.* and Liu[45], *et al.* by making HM 16.5 reference method as an anchor. The authors in Lee[24], *et al.* reduced the encoding time by 32% using the early skip mode decision with slight RD performance loss. The results in Table 3 show the complete domination of the proposed method compared to Lee[24], *et al.* in encoding timesaving. Even though the bit rate is increased, the proposed method's timesaving percentage is almost 40% more than Lee[24], *et al.*

The authors in Liu[45], *et al.* use the machine learning approach to reduce the encoding time for finding the CU size. The proposed method in Liu[45], *et al.* reduces the encoding time by 49% on average with 0.45dB loss in video quality and an 8.83% rise in bit rate. For a few video sequences like RaceHorses, BasketballDrill, and PartyScene, the approach in Liu[45], *et al.* saves more encoding time than our proposed method. The proposed method outperformed the Liu[45], *et al.* method for the remaining video sequences in timesaving, bit rate, and YPSNR. The proposed method can obtain more encoding time saving than the machine learning approach without sacrificing much coding quality. We have also compared the performance of the proposed method with the Mallikarachchi[19], *et al.* approach. The experimental findings show that the state-of-the-art method achieved good RD performance. However, only 47% of encoding time was saved, which is less compared to our proposed method.

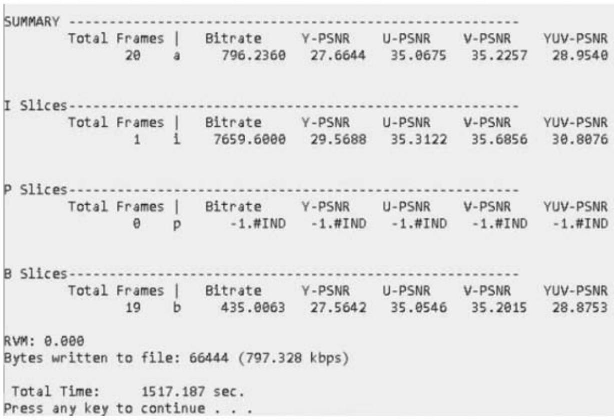
Generally, the Peak Signal to Noise Ratio (PSNR) is calculated using the Eqn (14).

$$PSNR = 10\log_{10}\frac{(2^{bitdepth}-1)^2 \times W \times H}{\sum_i (O_i - D_i)^2} \qquad (14)$$

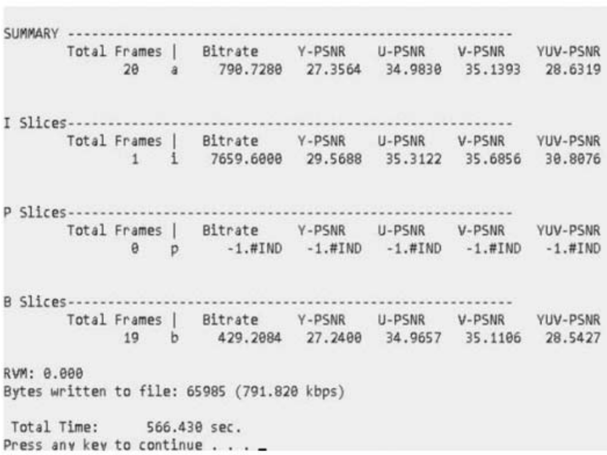
where, bitdepth = each pixel bit depth, H= Number of vertical pixels, W = Number of horizontal pixels, O_i = reference picture pixel value, D_i = Decoded picture pixel value, i = pixel address.

Table 2. Experimental outcomes of the proposed method compared to HM-16.5 standard

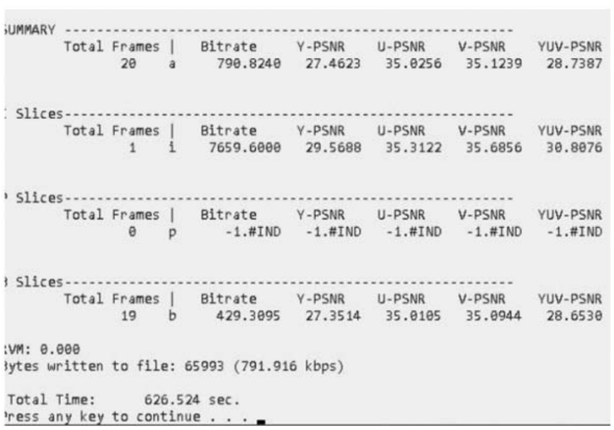
Class	Input	Resolution	Prop+NCDD							Prop+NCDH						
			BD-BR (%)	BD-PSNR (dB)	TS (%)					BD-BR (%)	BD-PSNR (dB)	TS (%)				
					QP=22	QP=27	QP=32	QP=37	Total			QP=22	QP=27	QP=32	QP=37	Total
A	PeopleOnStreet	2560x1600	7.83	-0.51	54.61	58.81	63.92	66.54	61	6.53	-0.26	60.48	62.90	66.59	70.12	65
	Traffic		8.98	-0.37	57.18	63.01	67.94	70.78	65	8.64	-0.28	58.62	61.97	66.52	71.12	65
B	Cactus	1920×1080	11.03	-0.27	34.17	51.09	60.55	68.42	54	9.94	-0.28	34.88	48.06	61.59	68.66	53
	Kimono		9.68	-0.23	39.80	52.50	67.88	76.81	59	7.07	-0.18	41.12	51.54	67.77	76.97	59
	Parkscene		6.92	-0.15	30.47	51.34	69.12	74.58	56	4.18	-0.09	26.05	48.90	65.16	75.22	54
	BasketballDrive		7.79	-0.42	35.40	40.85	45.55	59.51	45	5.26	-0.34	36.54	46.18	53.39	57.99	49
C	BasketballDrill	832×480	10.14	-0.40	27.34	35.38	45.87	54.32	41	12.07	-0.49	32.65	42.63	51.45	61.73	47
	BQMall		13.44	-0.53	32.76	42.66	54.03	58.28	47	10.45	-0.43	26.46	39.74	50.79	55.50	43
	PartyScene		7.05	-0.34	29.13	34.86	48.59	61.07	43	6.78	-0.14	23.92	31.05	41.58	56.28	38
D	BlowingBubbles	416×240	9.26	-0.36	22.10	32.88	42.60	52.15	37	4.50	-0.18	25.23	33.86	43.90	53.80	39
	BQSquare		8.84	-0.42	36.99	47.71	59.22	68.61	53	4.15	-0.10	33.90	53.37	66.87	76.99	58
	BasketballPass		7.36	-0.35	36.56	41.68	54.42	63.49	49	6.87	-0.33	43.01	50.01	59.02	73.33	56
	RaceHorses		10.80	-0.54	29.72	36.96	45.31	57.82	42	9.34	-0.46	29.77	43.20	51.54	63.98	47
E	KristenAndSara	1280×720	3.39	-0.10	56.57	68.08	73.57	77.26	69	2.90	-0.06	56.98	68.01	75.21	76.61	69
	Johnny		5.35	-0.12	61.13	70.85	76.24	78.36	72	2.53	-0.13	50.80	68.29	73.91	76.98	67
	FourPeople		4.17	-0.15	57.98	70.77	75.92	78.35	71	2.45	-0.09	55.71	65.43	67.69	74.78	66
	Stockholm		5.13	-0.14	52.10	60.32	68.41	71.31	63	4.86	-0.17	52.11	62.25	66.74	72.20	63
	Average		**8.06**	**-0.31**	**40.82**	**50.57**	**59.94**	**66.92**	**55**	**6.38**	**-0.23**	**40.48**	**51.60**	**60.56**	**68.36**	**56**



(a)



(b)



(c)

Figure 9. Experimental results for Partyscene video sequence at QP=37 for (a) HM 16.5 (b) Prop+NCDD (c) Prop+NCDH.

As human vision is more sensitive to luminance (Y), the YPSNR is considered instead of PSNR for drawing the RD curve. Figure 10 shows an example of RD curves for BQSquare, BQMall, Cactus, and FourPeople, respectively. The RD curves indicate that the proposed approach can maintain the video's quality the same as that of the regular HM 16.5. The RD performance loss due to the proposed method is slightly larger than standard HM but tolerable and even smaller than the machine learning approach method.

Table 3. Experimental findings of the proposed method and state-of-art techniques using HM-16.5 as an anchor

Class	Input	Resolution	Prop+NCDD			Prop+NCDH			Lee[24], *et al.*			Liu[45], *et al.*			Mallikarachchi[19], *et al.*		
			BD-BR (%)	**BD-PSNR (%)**	**TS (%)**	**BD-BR (%)**	**BD-PSNR (%)**	**TS (%)**	**BD-BR (%)**	**BD-PSNR (%)**	**TS (%)**	**BD-BR (%)**	**BD-PSNR (%)**	**TS (%)**	**BD-BR (%)**	**BD-PSNR (%)**	**TS (%)**
A	PeopleOnStreet	2560x1600	7.83	-0.51	61	6.53	-0.26	**65**	3.71	-0.24	35	7.37	-0.38	56	**1.89**	**-0.21**	53
	Traffic		8.98	-0.37	**65**	8.64	-0.28	60	5.13	-0.26	40	9.07	-0.47	52	**2.01**	**-0.19**	49
	Cactus		11.03	-0.27	**54**	9.94	-0.28	53	6.30	-0.19	32	7.53	-0.24	44	**1.67**	**-0.14**	28
B	Kimono	1920×1080	9.68	-0.23	**59**	7.07	-0.18	**59**	4.20	-0.18	29	6.53	-0.27	51	**1.34**	**-0.16**	41
	Parkscene		6.92	-0.15	**56**	4.18	**-0.09**	54	5.40	-0.22	31	3.63	-0.14	53	**2.12**	**-0.09**	43
	BasketballDrive		7.79	-0.42	45	5.26	-0.34	**49**	6.20	**-0.09**	24	6.34	-0.15	46	**1.30**	-0.12	41
	BasketballDrill		10.14	-0.40	41	12.07	-0.49	47	6.80	-0.21	35	9.81	-0.43	**54**	**1.92**	**-0.07**	34
C	BQMall	832×480	13.44	-0.53	**47**	10.45	-0.43	43	5.60	-0.22	30	9.64	-0.48	42	**1.46**	**-0.10**	32
	PartyScene		7.05	-0.34	43	6.78	**-0.14**	38	3.92	-0.16	32	9.87	-0.76	**59**	**2.32**	-0.17	**35**
	BlowingBubbles		9.26	-0.36	37	4.50	-0.18	**39**	5.42	**-0.09**	26	6.17	-0.37	37	**1.50**	-0.12	30
D	BQSquare	416×240	8.84	-0.42	53	4.15	-0.10	**58**	4.20	-0.12	31	12.34	-0.87	47	**0.98**	**-0.06**	54
	BasketballPass		7.36	-0.35	49	6.87	-0.33	**56**	2.30	-0.18	25	10.05	-0.54	40	**0.54**	**-0.08**	51
	RaceHorses		10.80	-0.54	42	9.34	-0.46	47	5.10	**-0.11**	33	12.89	-0.8	**57**	**1.23**	-0.13	56
	KristenAndSara		3.39	-0.10	**69**	2.90	**-0.06**	**69**	2.50	-0.07	38	13.35	-0.62	58	**2.15**	**-0.06**	54
E	Johnny	1280×720	5.35	-0.12	**72**	**2.53**	-0.13	67	3.50	**-0.10**	28	8.09	-0.32	47	2.70	-0.11	57
	FourPeople		4.17	-0.15	**71**	**2.45**	**-0.09**	66	2.70	-0.13	32	9.07	-0.48	36	2.61	-0.12	62
	Stockholm		5.13	-0.14	63	4.86	-0.17	**63**	2.60	**-0.05**	39	8.44	-0.49	52	**1.94**	-0.07	61
	Average		8.06	-0.31	55	6.38	-0.23	**56**	4.44	-0.15	32	8.83	-0.45	49	**1.73**	**-0.11**	47

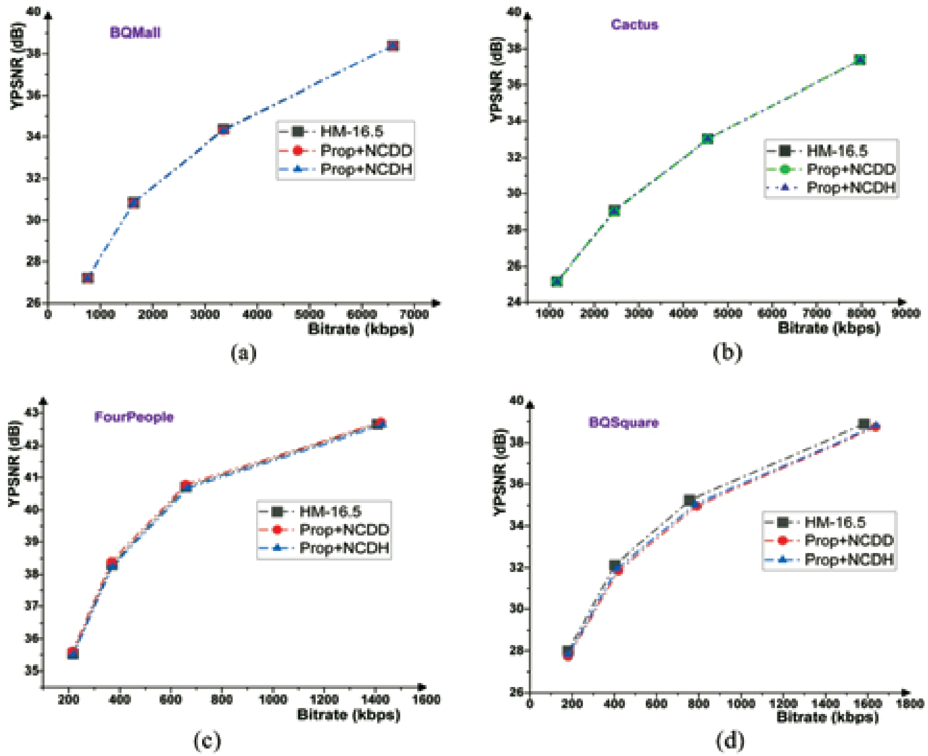


Figure 10. Example RD curves of (a) BQMall (b) Cactus (c) FourPeople (d) BQSquare Video sequences.

In this paper, the encoder_randomaccess_main configuration is used, which uses the hierarchical Bidirectional structures. This configuration provides higher efficiency but with a more significant delay compared to the other configurations.

5. CONCLUSIONS

In this paper, the MLRVS algorithm is used to minimise the encoding time of the HEVC encoder. The algorithm uses vertical subsampling, which decreases the number of computations needed to find the motion vector. Besides, two search patterns are proposed, which helps to quicken the motion estimation process. Moreover, the complexity reduction algorithm is used to lessen the time required for coding the coefficients. The proposed algorithm with two different search patterns is simulated individually. The results exhibit that the proposed algorithm has reduced the encoding time by 56% with NCDH and 55% with NCDD search patterns compared to the HM 16.5 standard. The results exhibit that our proposed method saves more encoding time than the state-of-the-art methods with slight RD performance loss.

In future research work, we will design the Long short-term memory (LSTM) neural network to predict the coding unit size in less encoding time with high efficiency.

REFERENCES

1. Sullivan, G.J.; Ohm, J.R.; Han, W.J. & Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.*, 2012, **22**(12), 1649-1668.
doi: 10.1109/TCSVT.2012.2221191
2. Wiegand, T.; Ohm, J.R.; Sullivan, G.J.; Han, W.J.; Joshi, R.; .; Tan, T.K. & Ugur, K. Special section on the joint call for proposals on High Efficiency Video Coding (HEVC) standardization. *IEEE Trans. Circuits Syst. Video Technol.*, 2010, **20**(12), 1661-1666.
doi: 10.1109/TCSVT.2010.2095692
3. Wiegand, T.; Sullivan, G.J.; Bjontegaard, G. & Luthra, A. Overview of the H.264/AVC Video Coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 2003, **13**(7), 560-576.
doi: 10.1109/TCSVT.2003.815165
4. Ohm, J.R.; Sullivan, G.J.; Schwarz, H.; Tan, T.K. & Wiegand, T. Comparison of the coding efficiency of video coding standards including high efficiency video coding (HEVC). *IEEE Trans. Circuits Syst. Video Technol.*, 2012, **22**(12), 1669-1684.
doi: 10.1109/TCSVT.2012.2221192
5. Norkin, A.; Bjontegaard, G.; Fuldseth, A.; Narroschke, M.; Ikeda, M.; Kenneth, A.; Minhua, Z. & Geert Vander, A. HEVC Deblocking Filter. *IEEE Trans. Circuits Syst. Video Technol.*, 2012, **22**(12), 1746-1754.
doi: 10.1109/TCSVT.2012.22 23053
6. Fu, C.; Alshina, E.; Alshin, A.; Huang, Y.W.; Chen, C.Y.; Tsai, C.Y.; Hsu, C.W.; Lei, S.M.; Park, J.H. & Han, W.J. Sample Adaptive Offset in the HEVC Standard. *IEEE Trans. Circuits Syst. Video Technol.*, 2012, **22**(12), 1755-1764.
doi: 10.1109/TCSVT.2012.22 21529
7. Hsieh, J.; Cai, J.; Wang, Y. & Guo, Z. ML-Assisted DVFS-Aware HEVC Motion Estimation Design Scheme for Mobile APSoC. *IEEE Syst. J.*, 2019, **13**(4), 4464-4473.
doi: 10.1109/JSYST.2018.2885538
8. Vayalil, N. C.; Paul, M. & Kong, Y. A Residue Number System Hardware Design of Fast-Search Variable-Motion-

Estimation Accelerator for HEVC/H.265. *IEEE Trans. Circuits Syst. Video Technol.,* 2019, **29**(2), 572-581. doi: 10.1109/TCSVT.2017.2787194.

9. Cebrián-Márquez, G.; Martínez, J. L. & Cuenca, P. A Motion-Based Partitioning Algorithm for HEVC Using a Pre-Analysis Stage *IEEE Trans. Circuits Syst. Video Technol.*, 2019, **29**(5), 1448-1461. doi: 10.1109/TCSVT.2018.2839026.
10. Fan, K.; Wang, R.; Li, G. & Gao, W. Efficient Prediction Methods With Enhanced Spatial-Temporal Correlation for HEVC. *IEEE Trans. Circuits Syst. Video Technol.*, 2019, **29** (12), 3716-3728. doi :10.1109/TCSVT.2018.2885002.
11. Pakdaman, F.; Hashemi, M.R. & Ghanbari, M. A. low complexity and computationally scalable fast motion estimation algorithm for HEVC. *Multimed. Tools Appl.,* 2020, **79**, 11639–11666. doi:10.1007/s11042-019-08593-y
12. Jiang, X.; Song, T.; Katayama, T. & Leu, J.-S. Spatial Correlation-Based Motion-Vector Prediction for Video-Coding Efficiency Improvement. *Symmetry,* 2019, **11**(2), 129. doi : 10.3390/sym11020129
13. Gogoi, S. & Peesapati, R. A hybrid hardware oriented motion estimation algorithm for HEVC/H.265. *J. Real-Time Image Proc.,* 2021, **18**, 953–966. doi : 10.1007/s11554-020-01056-w
14. Bouaafia, S.; Khemiri, R.; Sayadi, F.E. & Atri, M. Fast CU partition-based machine learning approach for reducing HEVC complexity. *J. Real-Time Image Proc.*, 2020, **17**, 185–196. doi : 10.1007/s11554-019-00936-0
15. Erabadda, B.; Mallikarachchi, T.; Kulupana, G. & Fernando, A. iCUS: Intelligent CU Size Selection for HEVC Inter Prediction. *IEEE Access*, 2020, **8**, 141143-141158. doi: 10.1109/ACCESS.2020.3013804.
16. Kuo, Y.; Chen, P. & Lin, H. A Spatiotemporal Content-Based CU Size Decision Algorithm for HEVC. *IEEE Trans. Broadcasting*, 2020, **66**(1), 100-112. doi: 10.1109/TBC.2019.2960938.
17. Huang, B.; Chen, Z.; Cai, Q.; Zheng, M. & Wu, D. O. Rate-Distortion-Complexity Optimized Coding Mode Decision for HEVC. *IEEE Trans. Circuits Syst. Video Technol.*, 2020, **30**(3), 795-809. doi: 10.1109/TCSVT.2019.2893396.
18. Lu, Y.; Huang, X.; Liu, H.; Zhou, Y.; Yin, H. & Shen, L. Hierarchical Classification for Complexity Reduction in HEVC Inter Coding. *IEEE Access,* 2020, **8**, 41690-41704. doi: 10.1109/ACCESS.2020.2977422.
19. Mallikarachchi, T.; Talagala, D. S.; Arachchi, H. K. & Fernando, A. Content-Adaptive Feature-Based CU Size Prediction for Fast Low-Delay Video Encoding in HEVC. *IEEE Trans. Circuits Syst. Video Technol.*, 2018, **28**(3), 693-705. doi: 10.1109/TCSVT.2016.2619499.
20. Sharma, R. R. & Arya, K. V. Parameter optimization for HEVC/H.265 encoder using multi-objective optimization technique. *In* 2016 11th International Conference on Industrial and Information Systems (ICIIS), 2016, 592-597. doi: 10.1109/ICIINFS.2016.8263008.
21. Yan, S.; Hong, L.; He, W. & Wang, Q. Group-Based Fast Mode Decision Algorithm for Intra Prediction in HEVC. *In* 2012 Eighth International Conference on Signal Image Technology and Internet Based Systems, 2012, pp. 225-229. doi: 10.1109/SITIS.2012.41.
22. Moreno, A.J.; Enriquez, E.M. & Diazde-Maria, F. Complexity Control Basedon a Fast Coding Unit Decision Method in the HEVC Video Coding Standard. *IEEE Trans. Multimedia*, 2016, **18**(4), 563 - 575. doi: 10.1109/TMM.2016.2524995
23. Kim, K.Y.; Kim, H.Y.; Choi, J.S. &Park, G.H. MC Complexity Reduction for Generalized P and B Pictures in HEVC. *IEEE Trans. Circuits Syst. Video Technol.*, 2014, **24**(10), 1723-1728. doi: 10.1109/TCSVT.2014.23 08651
24. Lee, H.; Shim, H.J.; Park, Y. & Jeon, B. Early Skip Mode Decision for HEVC Encoder with Emphasis on Coding Quality. *IEEE Trans. Broadcasting*, 2015, **61**(3), 388-397. doi: 10.1109/TBC.2015.2419172
25. Ahn, S.; Lee, B. & Kim, M. A Novel Fast CU Encoding Scheme Based on Spatiotemporal Encoding Parameters for HEVC Inter Coding. *IEEE Trans.Circuits Syst. Video Technol.*, 2015, **25**(3), 422-435. doi: 10.1109/TCSVT.2014.23 60031
26. Nalluri, P.; Alves, L.N. & Navarro, A. Complexity reduction methods for fast motion estimation in HEVC. *Signal Processing: Image Commun.*, 2015, **39**, 280-292. doi: 10.1016/j.image.2015.09.015
27. Fan, R.; Zhang, Y. & Li, B. Motion Classification-Based Fast Motion Estimation for High-Efficiency Video Coding. *Multimedia IEEE Trans.*, 2017, **19**(5), 893-907. doi: 10.1109/TMM.2016.2642786
28. Pan, Z.; Lei, J.; Zhang, Y. & Wang, F.L. Adaptive Fractional-Pixel MotionEstimation Skipped Algorithm for Efficient HEVC Motion Estimation. *ACM Trans. Multimedia Comput., Commun. Appl.*, 2018, 12. doi: 10.1145/3159170
29. R, V., & Kaimal, M. Motion segmentation algorithm using spectral framework. *Def. Sci. J.* 2010, **60**(1), 39-47. doi: 10.14429/dsj.60.102
30. Shen, L.; Zhang, Z. & An, P. Fast CU size decision and mode decision algorithm for HEVC intra coding. *IEEE Trans. Consumer Electron.*, 2013, **59**(1), 207-213. doi: 10.1109/TCE.2013.6490261
31. Koga, T.; Iinuma, K.; Hirano, A.; Iijima, Y.; Lijima, Y.; Ishiguro-Oonuma, T.; Hirano, M. & Ishiquro, T. Motion compensated interframe coding for video conferencing. *In* Proc. Nat. Telecommunications Conf, pp. G5.3.1-G5.3.5, 1981.
32. Chun, K.W. & Ra, JB An improved block matching algorithm based on successive refinement of motion vector candidates. *Signal Process.: Image Commun.*,

1994, **6**, 115-122.
doi: 10.1016/0923-5965(94)90010-8
33. Lee, L.W.; Wang, J.F.; Lee, J.Y. &Shie, J.D. Dynamic search-window adjustment and interlaced search for block matching algorithm. *IEEE Trans. Circuits Syst. Video Technol.*, 1993, **3**, 85-87.
doi: 10.1109/76.180692
34. Li, R.; Zeng, B. & Liou, M.L. A new three-step search algorithm for block motion estimation. *IEEE Trans. Circuits Syst. Video Technol.*, 1994, **4**, 438-442.
doi: 10.1109/76.313138
35. Jain, J.R. & Jain, A.K. Displacement measurement and its application in interframe image coding. *IEEE Trans. Commun.*, 1981, **29**(12), 1799-1808.
doi: 10.1109/TCOM.1981.1094950
36. Chen, M.J.; Chen, L.G. & Chiueh, T.D. One-dimensional full search motion estimation algorithm for video coding. *IEEE Trans. Circuits Syst. Video Technol.*, 1994, **4**, 504-509.
doi: 10.1109/76.322998
37. Chen, F.; Wen, P.; Peng, Z.; Jiang, G.;Yu, M. & Chen, H. Hierarchical complexity control algorithm for HEVC based on coding unit depth decision. *EURASIP J. Image Video Processing*, 2018, 96.
doi: 10.1186/s13640-018-0341-3
38. Khemiri, R.; Bahri, N.; Belghith, F.;Sayadi, F.; Atri, M. & Masmoudi, N. Fast motion estimation for HEVC videocoding. *In* 2016 International Image Processing, Applications and Systems (IPAS), Hammamet, Tunisia, 2016.
doi: 10.1109/IPAS.2016.7880120
39. Saran, R. Median Predictor based Lossless Video Compression Algorithm for IR Image Sequences. *Def. Sci. J.*, 2009, **59**(2), 183-188.
doi: 10.14429/dsj.59.1507
40. Zhu, S. & Ma, K.K. A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation. *IEEE Trans. Image Processing*, 2000, **9**(2), 287-290.
doi: 10.1109/83.821744
41. Cheung, C.H. & Po, L.M. Novel Cross-Diamond-Hexagonal Search Algorithms for Fast Block Motion Estimation. *IEEE Trans. Multimedia*, 2005, **7**(1), 16-22.
doi: 10.1109/TMM.2004.840 609
42. Software reference test model HM16.5available in https://hevc.hhi.fraunhofer.de/, Nov 2018.
43. Bjontegaard, G. Calculation of Average PSNR Differences between RD Curves. ITU-T SG16/Q6, VCEG-M33, Austin, 2001.
44. Bjontegaard, G. Improvements of the BD-PSNR Model. ITU-T SG16/Q6,VCEG-AI11, Berlin, 2008.
45. Liu, D.; Liu, X. & Li, Y. Fast CU size decisions for HEVC intra frame codingbased on support vector machines. *In* IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing(DASC), 2016.
doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.168

CONTRIBUTORS

Mr S. Karthik Sairam received his BTech in electronics and communication engineering from DVR college of Engineering and Technology in 2012 and his MTech from Vardhaman college of Engineering, Hyderabad in 2015. He is a PhD research scholar at National Institute of Technology Warangal since 2018. His research interests are High Efficiency Video coding and Architectures for video compression algorithms.
In this paper, he devised the idea, developed the method, generated the results, and wrote the manuscript.

Dr P. Muralidhar received BTech (ECE) and MTech in Electronic Instrumentation from National Institute of Technology, Warangal, India, in 1993 and 2004 respectively. Then he has received his PhD from NIT Warangal. He joined NIT Warangal in 1997. Since then he has been working in the ECE Department, NIT Warangal. His research interests include design of embedded systems and VLSI architectures for video processing systems.
In the present study, he has guided the author, supervised the work, and verified the manuscript.