

Research Article

Complexity Analysis of New Future Video Coding (FVC) Standard Technology

Soulef Bouaafia ¹, **Randa Khemiri** ^{1,2}, **Seifeddine Messaoud** ¹
and **Fatma Elzahra Sayadi** ³

¹*Electronics and Microelectronics Laboratory, Faculty of Sciences of Monastir, University of Monastir, Environment Street, 5019 Monastir, Tunisia*

²*Higher Institute of Computer Science and Multimedia of Gabes, University of Gabes, Gabes, Tunisia*

³*National Engineering School of Sousse, University of Sousse, Sousse, Tunisia*

Correspondence should be addressed to Soulef Bouaafia; soulefbouaafia@gmail.com

Received 28 December 2020; Revised 17 May 2021; Accepted 14 July 2021; Published 2 August 2021

Academic Editor: Floriano De Rango

Copyright © 2021 Soulef Bouaafia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Future Video Coding (FVC) is a modern standard in the field of video coding that offers much higher compression efficiency than the HEVC standard. FVC was developed by the Joint Video Exploration Team (JVET), formed through collaboration between the ISO/IEC MPEG and ITU-T VCEG. New tools emerging with the FVC bring in super resolution implementation schemes that are being recommended for Ultra-High-Definition (UHD) video coding in both SDR and HDR images. However, a new flexible block structure is adopted in the FVC standard, which is named quadtree plus binary tree (QTBT) in order to enhance compression efficiency. In this paper, we provide a fast FVC algorithm to achieve better performance and to reduce encoding complexity. First, we evaluate the FVC profiles under All Intra, Low-Delay P, and Random Access to determine which coding components consume the most time. Second, a fast FVC mode decision is proposed to reduce encoding computational complexity. Then, a comparison between three configurations, namely, Random Access, Low-Delay B, and Low-Delay P, is proposed, in terms of Bitrate, PSNR, and encoding time. Compared to previous works, the experimental results prove that the time saving reaches 13% with a decrease in the Bitrate of about 0.6% and in the PSNR of 0.01 to 0.2 dB.

1. Introduction

High-Efficiency Video Coding (HEVC) is the leading video coding standard [1], standardized in 2013 by the Joint Collaborative Team on Video Coding (JCT-VC) forming the Motion Picture Experts' Group (MPEG) and the Video Coding Expert Group (VCEG). HEVC achieves an increase of about 50% in coding efficiency while maintaining the same visual quality than previous standards, such as H.264/Advanced Video Coding (AVC) [2]. With the development of video technologies, better qualities and higher resolutions are demanding. For this reason, the new video codec is very interesting to improve the compression efficiency and the quality of the predecessor standards. In October 2015, the new group, Joint Video Exploration Team (JVET), has been working on a new video coding standard,

called post-HEVC or Future Video Coding (FVC), as the successor of HEVC [3]. The Versatile Video Coding (VVC) standard is a new video coding technology, which can be standardized from 2020. At the same video quality, especially for UHD video, the FVC standard currently provides between 25 and 30% Bitrate saving compared to HEVC [4].

These new FVC technologies are being evaluated in order to improve the compression efficiency using an experimental platform, namely, the Joint Exploration Model (JEM) software, which was developed from the reference software HM [3]. FVC is developed to essentially meet all existing HEVC and H.264/AVC applications, such as broadcast, surveillance, and smart home, and focus on two goals: higher video resolution and parallel architectures [5]. On the other hand, video coding has a high potential for being deployed in wireless

networks due to its unique features like independent frame coding and low-complexity encoding operations. In fact, the growing complexity has hampered the adoption of video encoding in real-time streaming over mobile wireless networks, such as 4G networks and upcoming 5G networks. However, the video encoding with high computational complexity and the great contribution to a node's power consumption and video transmission over an erroneous wireless channel are the main reasons for these challenges [6]. As in [7], the authors proposed a unified Quality of Experience (QoE) prediction framework for HEVC-encoded video streaming over the wireless network. In addition, the work in [8] proposes a novel frame-level rate control algorithm for videos with a complex scene in HEVC over the wireless network. Many other works in this nascent field of video coding in the wireless network can be found in [9, 10].

This paper focuses on the optimization of the new video coding standard (FVC) through fast methods in terms of encoding time. Therefore, to reduce the computational complexity, we propose in this paper a fast FVC scheme-based fast-mode decision. The rest of this paper is structured as follows. An overview of the FVC standard is defined in Section 2. Section 3 presents some existing algorithms developed on fast-mode decisions in order to reduce the HEVC and FVC complexity in terms of encoding time. Section 4 presents the JEM configuration. Section 5 gives experimental results. At the end, the conclusion of this paper is presented in Section 6.

2. FVC Overview

As in most previous standards, FVC has a hybrid block-based encoding architecture, containing intra and inter prediction and transform coding with entropy coding [1]. The FVC is developed by the JVET based on the HM test model (HM 16.6) [11]. The picture partitioning structure divides the input video into blocks called Coding Tree Units (CTUs). A CTU is split using a quadtree with a nested binary tree structure into Coding Units (CUs), with a leaf CU defining a region sharing the same prediction mode (e.g., intra or inter). Figure 1 shows a general block diagram of the FVC standard. The new coding features in the FVC standard are listed as follows [12].

2.1. Block Partitioning. A Coding Tree Unit (CTU) becomes the principal block partition in the HEVC standard, which replaced the macroblock for the H.264/AVC encoder. The CTU has a size of 64×64 up to 8×8 . We talk about quadtree in HEVC, but recently, the quadtree plus binary tree (QTBT) block structure was introduced in the new video coding FVC [13, 14]. The concepts of multiple partition types have been removed in the FVC standard; it means that the sizes of CU, PU, and TU are similar in the QTBT structure [15]. There are two types of binary trees: a value of 1 determines a symmetric vertical split for a CU, while a 0 value specifies a symmetric horizontal split. These improvements notably increase compression efficiency. The leaf nodes of the binary tree are called coding units (CU), and this segmentation is used for transformation and prediction

processing without further splitting; a CU can be square or rectangular in shape [16]. Figure 2 illustrates the QTBT block structure in JEM software.

The Coding Tree Unit (CTU) with P/B slice coding is presented in Figure 3. In the QTBT structure, CTU is firstly divided into a quadtree partition, and then it can be further partitioned into a binary tree partition. With a QTDepth from 0 to 4 levels, the quadtree nodes have a block size from 256×256 (CTU) to 16×16 (MinQTSIZE). The maximum allowed size of the root node of the binary tree is 128×128 , corresponding to a BTDepth from 0 to 3 [17].

2.2. Intra Prediction. The intra prediction modes have been also enhanced; JEM software has 67 intra modes: 65 angular, Direct Current (DC) and planar prediction modes, instead of 35 modes in HEVC [3, 14]. Figure 4 illustrates intra prediction modes. The black line represents the existing directional mode in HEVC, and the red line means the newly added directional mode in FVC. The planar and DC modes remain the same.

2.3. Inter Prediction. Compared to HEVC, inter prediction has many improvements in JEM software. There are two Motion Vector Predictions (MVPs) including Alternative Temporal Motion Vector Prediction (ATMVP) and Spatial Temporal Motion Vector Prediction (STMVP). The ATMVP is enhanced by allowing each CU to report multiple sets of motion information from multiple blocks smaller than the current CU in the collocated reference frame, as illustrated in Figure 5.

In the STMVP process, based on the neighboring spatio-temporal motion vector predictor, the motion vectors of the sub-CUs are derived recursively, as shown in Figure 6.

2.4. Transforms. The prediction residual is encoded using a transform block. There are two types of transforms: DST and DCT. FVC introduced multiple transforms such as DST (I and VII) and DCT (II, V, and VIII). The size of transform block is increased from 4×4 to 64×64 in the new video coding compared to the HEVC standard.

2.5. Filter Improvements. There are three filtering methods introduced for the FVC standard: a deblocking filter, a Sample Adaptive Offset (SAO), and an Adaptive Loop Filter (ALF). The deblocking filter is designed to minimize the visibility of artifacts and is only used for samples found at block boundaries. The SAO filter is aimed at enhancing the quality of reconstruction of the amplitudes of the original signal and is adaptively applied to all samples. ALF minimizes the mean-absolute-error between the decoded frame and the original image.

2.6. Entropy Coding. "Context Adaptive Binary Arithmetic Coding (CABAC)" is the entropy coder in HEVC [18]. In the FVC standard, the improved version of CABAC is adopted with a changed context model selection for the evaluation of multihypothesis possibilities, transformation coefficients with context-dependent update rate, and adaptive model initialization.

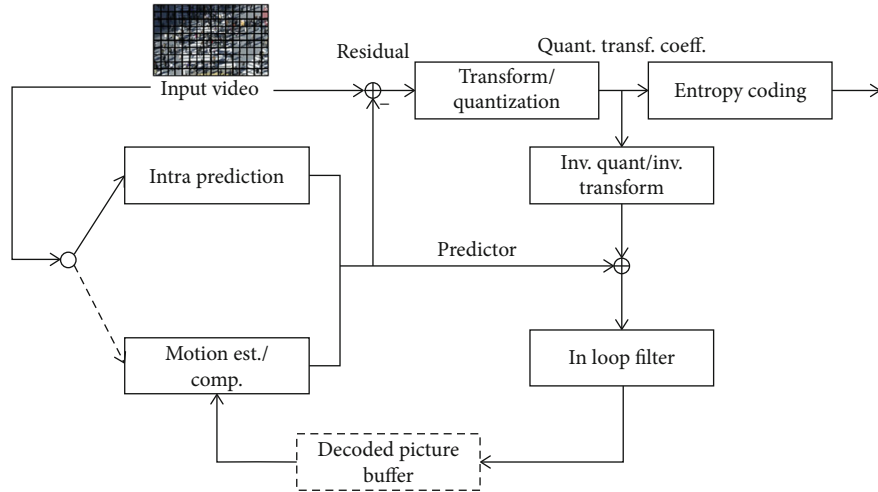


FIGURE 1: FVC block diagram.

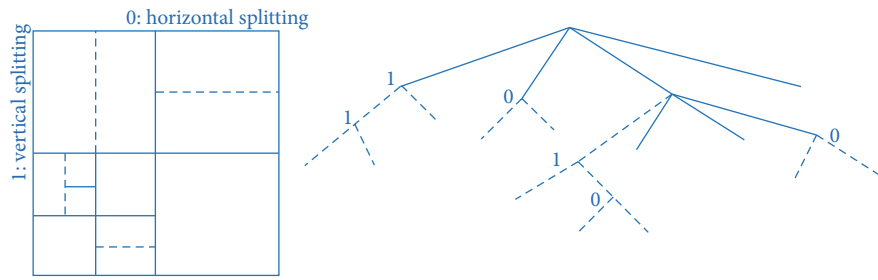


FIGURE 2: QTBT block partitioning structure.

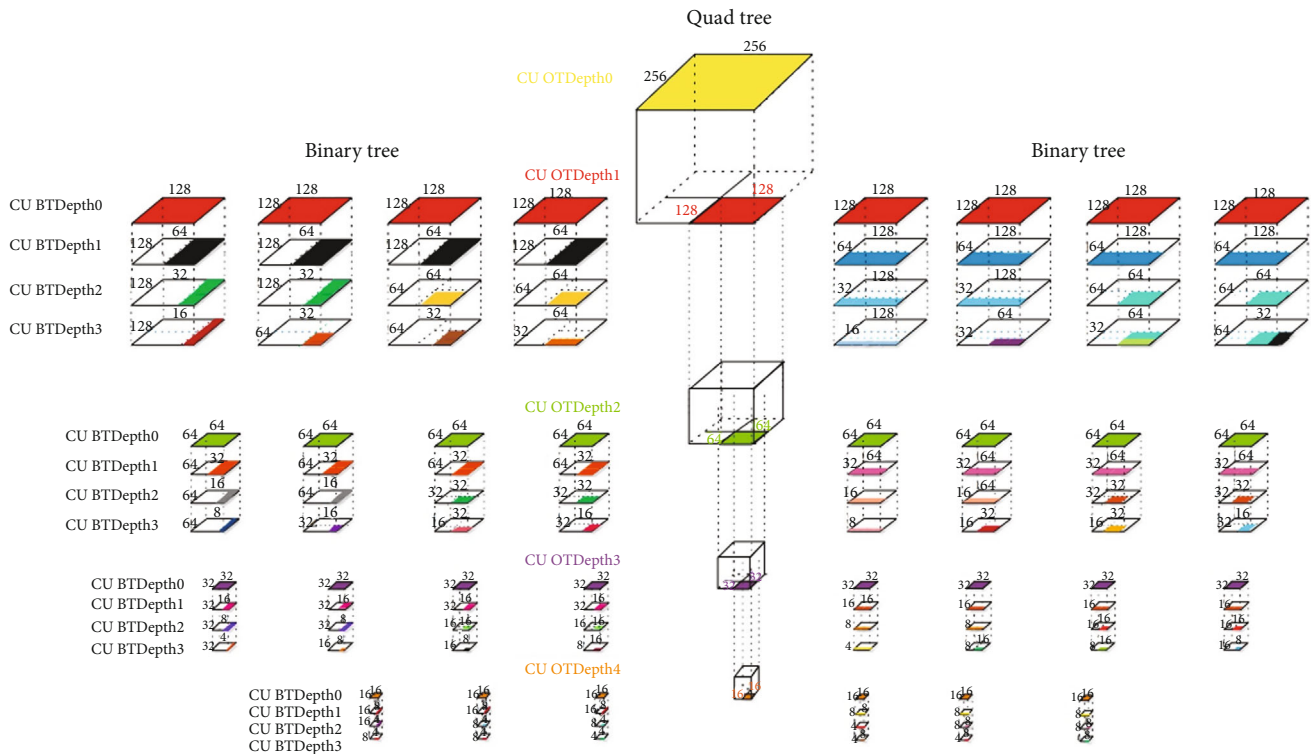


FIGURE 3: Quadtree plus binary tree structure.

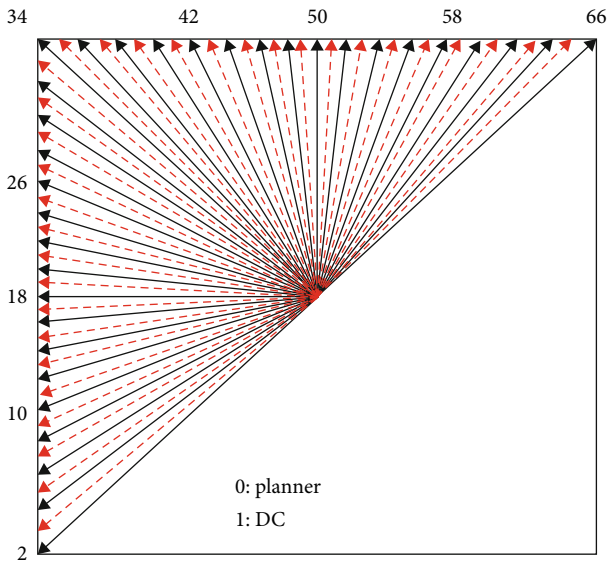


FIGURE 4: Intra prediction modes.

In fact, many researchers aim to reduce the complexity for each standard module in terms of encoding time through software and hardware methods [19]. An overview of previous works, which introduced fast algorithms using RA and LD configurations, is presented in the next section.

3. Related Works

In fact, computational complexity remains a serious issue in video compression, especially when real-time application is desired. Consequently, several optimizations are required to reduce the computational complexity. The main goal of FVC is to solve the critical issues in the HEVC [20]. Being in continuous development, the works detailing these techniques and methods are very rare. This is why some efforts are being made using HEVC to develop FVC. Besides, to minimize the encoding time for JEM software, especially, the ME time, several fast approaches have been proposed [21].

García-Lucas et al. [21] proposed a fast scheme in order to accelerate the ME in JEM called “pre-analysis algorithm.” This algorithm allows reducing the size of the search range and the number of the reference frame. Numerical results prove that the proposed algorithm achieves more than 62% of the execution time with a negligible BD-rate of 0.11%. Moreover, the work in [22] proposes a Naïve-Bayes model-based fast CU mode prediction in HEVC-JEM transcoding to improve coding efficiency. This technique reduces the computational time by 12.71%.

In [23], Khemiri et al. suggested an algorithm using the parallel-difference-reduction process to optimize the ME module of HEVC. The proposed scheme achieves on average 56.17% and 30.4% reduction in coding time with a PSNR loss of 0.095 dB and a reduction in Bitrate of 0.64%. Three algorithms are presented in [24] to improve the TZ search (test zonal search) algorithm. The computational complexity reaches 75% with 0.12 dB PSNR and a decrease by 0.5% in the Bitrate in RA configuration. Another fast algorithm

named “early skip mode decision” for the HEVC is presented in [25]. The obtained results reveal that the fast scheme saves on average 58.5% and 54.8% of execution for several video sequences under RA and LDB configurations.

Two other fast-mode decisions are presented in [26]: the Coded Block Flag and the Early CU Termination. This work reduced the encoder complexity by 58.7%, while maintaining the same level of coding efficiency. Kim et al. [27] proposed two fast-mode decisions in order to accelerate the inter prediction time: ESD and CBF. This proposed algorithm achieves 34.55% of reduction in execution time using RA configuration and 36.48% using the LD configuration. Lee et al. [28] suggest an algorithm called “Adaptive Search Range” (ASR) to reduce the ME complexity by replacing the fixed ME search range with an adaptive one. That is, ASR can also be adopted for several search models in the software execution to minimize the number of search points. The results obtained show that the proposed algorithm can reduce the execution time by up to 53% for different sequences in fast ME schemes. Another interesting work proposed by Park et al. [29] is aimed at reducing the complexity of encoding JVET JEM with the QTBT partition technique. The proposed “Reference Frame Search” method allows the encoder to skip over important reference frame searches by using the strong correlation between parent and child nodes in the QTBT partition. The experiment was carried on with a quad-core Intel i7 4.00 GHz CPU with 16 GB RAM. Results revealed that this technique reduced the ME time by 34% compared with JEM 3.1, while maintaining less than a 0.3% BD-rate.

In order to optimize the TZSearch motion estimation, Purnachand et al. [30] replaced the “diamond search pattern” with the “Hexagonal search pattern.” In addition, the proposed algorithm is improved by changing the search threshold in the search area for each grid. All simulations prove that the computational complexity for ME is decreased by almost 50% compared to the TZSearch algorithm with a nonsignificant change in PSNR and Bitrate. In addition, Ahn et al. [31] introduced a fast inter-HEVC encoding scheme. The achieved results prove that the proposed scheme makes it possible to obtain an average time saving of 49.6% and 42.7% with an average Bitrate of 1.4% and 1% under the RA and LDB configurations for various sequences of test.

On the other hand, in [32], the authors proposed an effective quadtree plus binary tree (QTBT) partition method to reach a good compromise between compression performances. Experimental results provide an average time reduction of 64% with only 1.26% increase in Bitrate. A fast algorithm combining both CU and PU early termination decisions to solve the problem of high computational complexity of HEVC is proposed by Chen et al. in [33]. Results show that the proposed method achieves 57% time saving with an increase of 0.43% in the BD-rate. Wang et al. [34] proposed an algorithm named “Confidence Interval-Based Early Termination” for QTBT partition, to classify the redundant partition methods in terms of RD cost technique. The results obtained prove that the proposed scheme can speed up the QTBT partition process by reducing the execution time by 54.7% with an increase of only 1.12% in Bitrate.

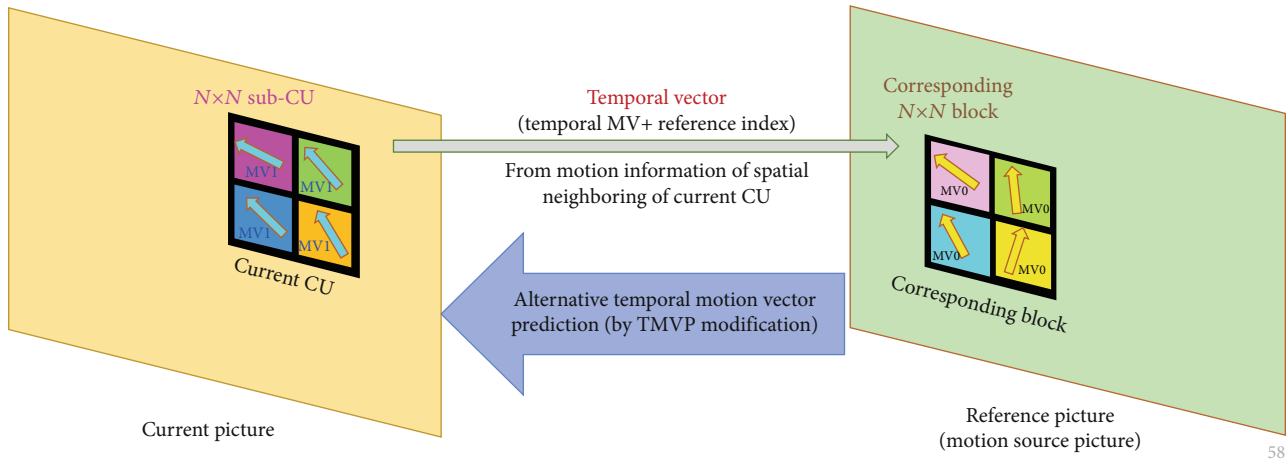


FIGURE 5: ATMVP motion vector prediction.

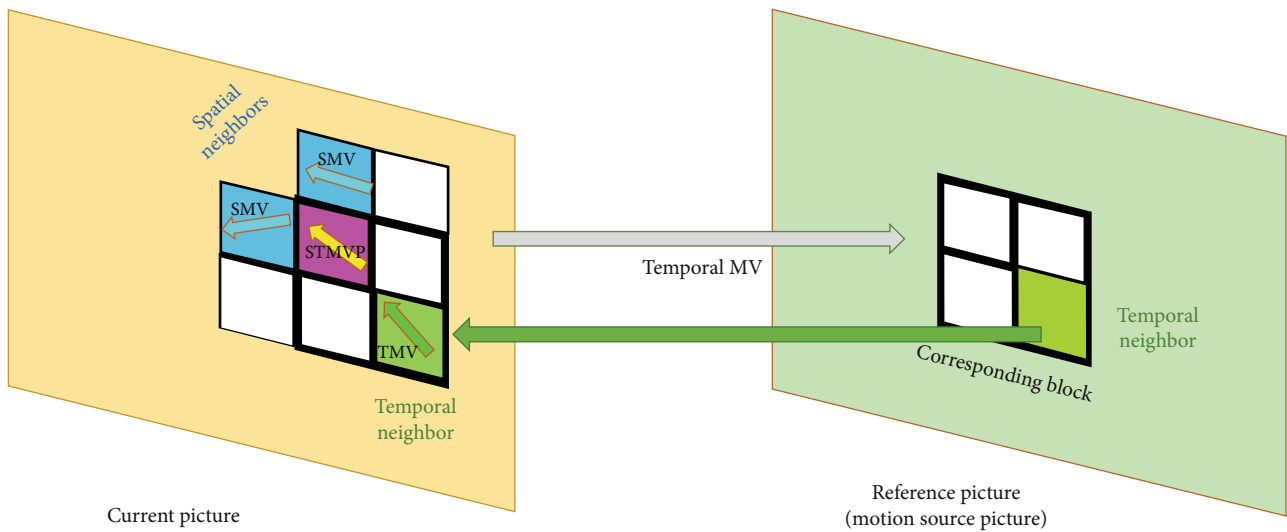


FIGURE 6: STMVP process.

In short, to reduce the FVC complexity, several schemes have been introduced. Some of them are aimed at reducing the number of searches in order to improve ME. Others adopted fast-mode decisions to improve the TZSearch motion estimation using different configurations.

4. JEM Configuration Overview

As with HM for HEVC, the reference software JEM supports four types of coding configurations, as indicated in the Common Test Conditions [35]. The four modes provided are as follows: All Intra, Low-Delay B, Random Access, and Low-Delay P slices only.

4.1. All Intra (AI). All pictures are encoded using I-slices. The Quantization Parameter (QP) is constant for all images. For the AI configuration, a temporal subsampling of the sequences is performed in JEM. The subsampling can be enabled in the JEM software using the parameter “Temporal-

SubsampleRatio.” This parameter of the AI “encoder_intra_jvet10.cfg” configuration file is 8, indicating that one frame is encoded every 8 frames [35]. The number related with each image signifies the encoding and display order. The QPI represents QP for the IDR (Instantaneous Decoder Refresh) picture which is the same for all pictures. Figure 7 gives a graphical presentation of AI configuration.

4.2. Low-Delay (LD). For this configuration, there are two subtypes, which are “Low-Delay P” and “Low-Delay B.” In the LD configuration, only the first frame is encoded in the Intra mode. So, in the LDP mode, all pictures are encoded as a P-slice only while all frames are taken as P and B slices for LDB. The coding order is represented by the associated number for each frame. The QP of each intercoded frame must be calculated by adding an offset to the QP of the intracoded frame as a function of the temporal layer. Figure 8 represents a Low-Delay configuration graphical presentation.

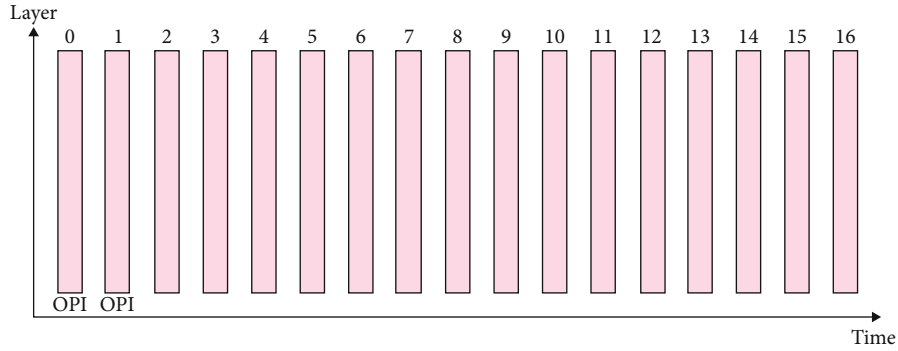


FIGURE 7: All Intra configuration.

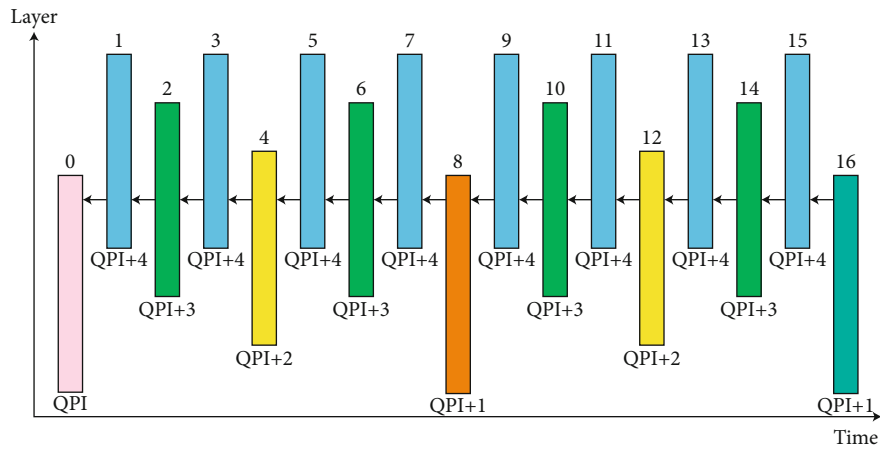


FIGURE 8: Low-Delay configuration.

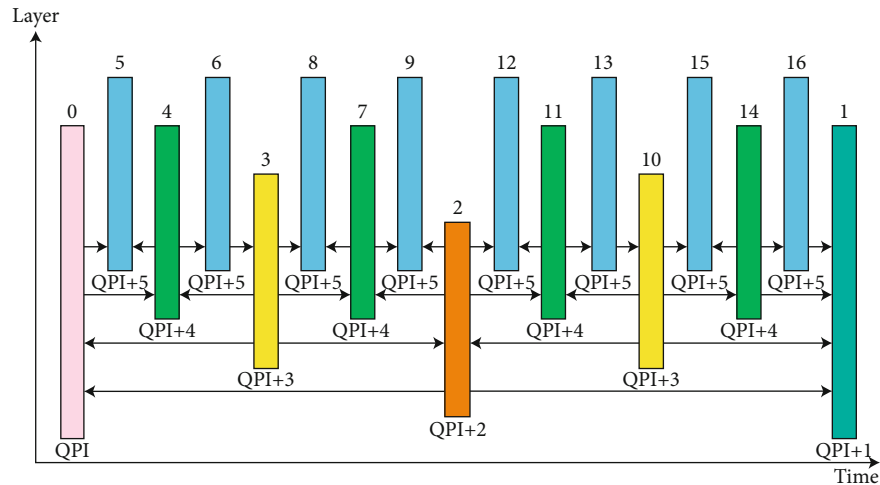


FIGURE 9: Random access configuration.

4.3. *Random Access (RA)*. For the JEM reference software encoder, the hierarchical B-picture coding structure is used in the RA configuration. The “encoder_randomaccess_jvet10.cfg” is selected. The size of the Group of Picture (GOP) is fixed to 16 frames [36]. Figure 9 shows a random access configuration graphical presentation. In RA mode, only the first frame in the video sequence is encoded as intra-

frame. Other successive pictures will be encoded as generalized P and B pictures.

5. Experimental Results

5.1. *Experimental Condition*. In this section, we have evaluated the performance of the FVC standard and compared

TABLE 1: Evaluated sequences.

Classes	Resolution	Videos
Class A1	3840 × 2160	Drums100, and CampfireParty
Class A2	3840 × 2160	CatRobot, and DaylightRoad
Class B	1920 × 1080	Kimono, ParkScene, Cactus, BQTerrace, BasketballDrive
Class C	832 × 480	RaceHorsesC, BQMall, PartyScene, BasketballDrill
Class D	416 × 240	RaceHorses, BQSquare, BlowingBubbles, BasketballPass
Class E	1280 × 720	Vidyo 1, Vidyo 3 and Vidyo 4

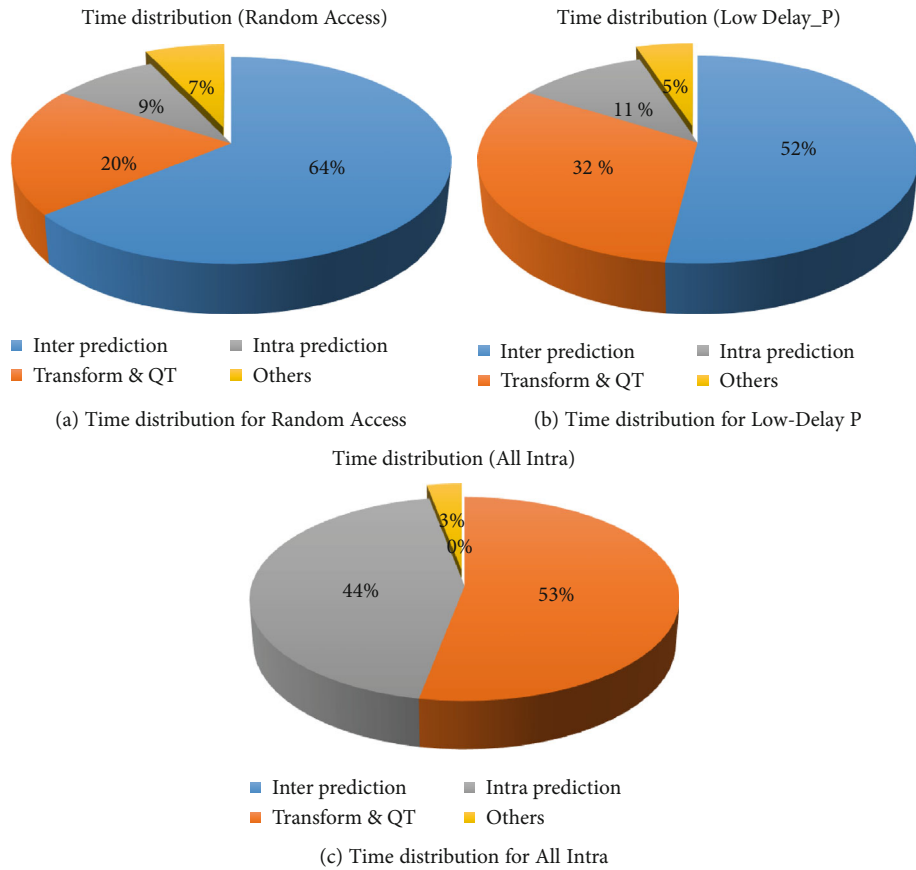


FIGURE 10: Encoding time distribution for (a) Random Access, (b) Low-Delay P, and (c) All Intra.

the three configurations (RA, LDB and LDP) of JEM in terms of Bitrate (BR), PSNR, and Encoding Time (T); a set of simulations have been launched. The proposed scheme has been implemented using reference software JEM-7.1 [37]. In each sequence, the number of frames is limited to 100. All experiments were performed on an Intel® core™ i7 3770 @ 3.4 GHz CPU and 16 GB RAM. All resolutions have been tested in QP 22 to 37 and presented as classes. Giving to the JVET Common Test Conditions (CTC), the test sequences include new Ultra-High-Definition (UHD or 4K/8K) sequences (Class A1-A2, 10-bit) and HEVC test sequences (Class B-E, 8-bit) [34]. Each class consists of different videos with different scenarios and features, as shown in Table 1.

5.2. *Evaluation Criteria.* The coding performance is evaluated through: PSNR, BR, and T, which are defined as follows:

- (i) PSNR (dB): PSNR loss $\longrightarrow \Delta\text{PSNR} = \text{PSNR}_p - \text{PSNR}_o$ (dB)
- (ii) BR (Kbps): Bitrate increase $\longrightarrow \Delta\text{BR} = (\text{BR}_p - \text{BR}_o / \text{BR}_o) \times 100(\%)$
- (iii) T(s): Encoding time speedup $\longrightarrow \Delta T = (T_p - T_o / T_o) \times 100(\%)$

Here, PSNR_o , BR_o , and T_o define the PSNR, Bitrate, and execution time of the original scheme. BR_p , PSNR_p and T_p

defines the PSNR, Bitrate, and execution time of the proposed scheme.

5.3. FVC Time Profile. In this section, we evaluate the profiling results obtained by JEM-7.1, in order to define the encoding components that consume the most time. The time distribution of the JEM encoder for three profiles, namely, All Intra, Random Access, and Low-Delay P is illustrated in Figure 10. These profiling results were obtained with Valgrind tools when processing the “Drums100” sequence encoded with QP = 37 in RA and All Intra configuration, and the “BasketballDrive” sequence encoded in LDP configuration with QP = 37.

For All Intra, the most critical block in execution time is the Transform and Quantization module. The encoding time consumed in intra prediction exceeds 30%, while in Low-Delay P, more than 60% is devoted to the inter prediction. Likewise, the inter prediction consumes 60% of the execution time in an RA configuration.

The complexity of the inter prediction is explained by the huge number of redundant operations that the standard must perform on the same frame and with different block partitions.

5.4. FVC Fast Mode Decision. To reduce the encoding time of the new standard FVC, many fast-mode decision algorithms for splitting were adopted in JEM software, such as Early Skip Detection (ESD), Coded Block Flag (CBF), and Early CU termination (ECU) algorithms, which are clarified as follows.

5.4.1. Early CU Termination (ECU). The early CU termination algorithm is used in the passage from depth p to depth $p + 1$. The best mode is determined by computing the RD cost. After selecting the skip mode having the minimum of RD cost, there is no need to continue the partitioning [24].

5.4.2. Early Skip Detection (ESD). Some works show that the most modes chosen were the SKIP mode. The skip mode is a very efficient coding tool. It can represent a coded block without residual information. After searching the best inter $2N \times 2N$, the Early Skip Detection (ESD) algorithm represents a simple checking of the differential motion vector (DMV) and the Coded Block Flag (CBF) which are the two conditions called as “early Skip conditions.” After selecting the best mode having the minimum of RD cost, the proposed method checks its DMV and CBF. If the DMV and CBF of the best inter $2N \times 2N$ mode are, respectively, equal to (0, 0) and zero, the best mode is determined as the SKIP mode. In other words, the remaining CU modes are not searched for inter mode decision [26].

5.4.3. Coded Block Flag Algorithm (CBF). The detection of the optimal predicted mode will be provided by the coded block flag fast method (CFM) algorithm. For each mode of the CU, RD cost will be calculated. If CBF is zero (all transform coefficients are zeros: CBF_Y, CBF_U, and CBF_V), the other remaining modes will not be tested anymore [25, 26].

Figure 11 shows the flowchart of the mode decision process. The Early_Skip condition checks if the motion vector difference of inter partition mode $2N \times 2N$ is equal to

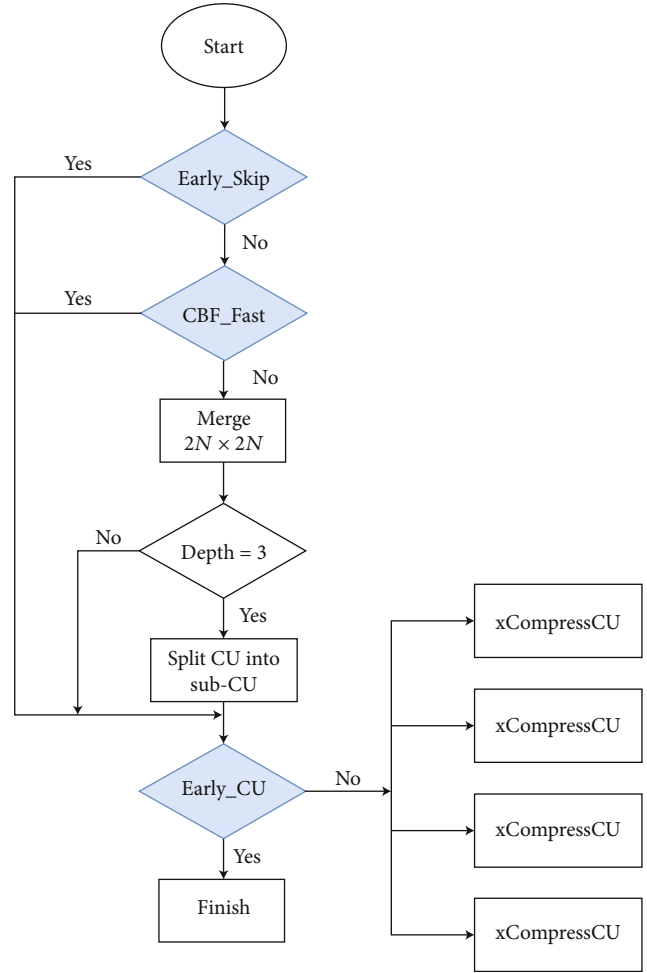


FIGURE 11: Flowchart of the mode decision process.

(0, 0). The CBF_Fast condition checks if inter partition mode $2N \times 2N$ does not contain nonzero transform coefficients. The algorithm evaluates the Early_CU condition directly when the condition is true. This Early_CU condition checks if the best inter coding mode is Skip. If the condition is true, the algorithm stops. Otherwise, it evaluates the next CU level of recursive mode decision if the current CU depth is not the maximum. The aforementioned process is repeated recursively for every coding depth 0, 1, 2, and 3, being the corresponding CU sizes. For every prediction mode, it is necessary to calculate the RD cost with its associated high computational cost. The combined fast algorithms (ECU, ESD, CBF) were proposed in order to reduce the FVC computational complexity and improve the RD performance.

5.5. Results. The comparative performance of the proposed scheme to the original algorithm in terms of Bitrate, encoding time, and PSNR is listed in Table 2. The test configurations are (randomaccess_jvet10, lowdelay_jvet10 (P and B)) based on the JEM CTC [35].

As shown in Table 2 on average, runtime was reduced by 13% for the RA configuration and slightly less for the LDP and LDB configurations. Regarding the performance of the

TABLE 2: Results achieved from the proposed scheme versus the original FVC scheme.

Class	Sequences	Random Access			Low-Delay P			Low-Delay B		
		ΔBR (%)	$\Delta PSNR$ (dB)	ΔT (%)	ΔBR (%)	$\Delta PSNR$ (dB)	ΔT (%)	ΔBR (%)	$\Delta PSNR$ (dB)	ΔT (%)
A1	Drums	-0.45	-0.03	-15.75	-0.32	-0.27	-12.5	-0.82	-0.025	-15.16
	CampfireParty	-0.07	-0.01	-14.65	-0.22	-0.021	-11.72	-0.33	-0.015	-13.22
A2	CatRobot	-0.03	-0.012	-16.68	-0.52	-0.15	-15.75	-0.76	-0.02	-18.75
	DaylightRoad	-0.42	-0.022	-20.89	-0.47	-0.035	-15.95	-0.39	-0.017	-19.7
B	Kimono	-0.206	-0.017	-16.33	-0.011	-0.005	-13.85	-0.06	-0.002	-12.16
	ParkScene	-1.46	-0.13	-15.94	-0.826	-0.072	-9.12	-0.677	-0.06	-12.75
	Cactus	-0.77	-0.04	-13.92	-0.515	-0.027	-10.57	-0.535	-0.02	-11.46
	BQTerrace	-2.36	-0.13	-13.9	-3.205	-0.129	-9.456	-2.034	-0.086	-12.28
	BasketballDrive	-0.274	-0.022	-13.45	-0.189	-0.015	-9.84	-0.021	-0.009	-11.64
C	BasketballDrill	-0.409	-0.056	-10.32	-0.191	-0.033	-5.93	-0.207	-0.023	-6.48
	BQMall	-0.8	-0.18	-9.61	-0.272	-0.077	-5.55	-0.236	-0.075	-5.93
	PartyScene	-1.27	-0.26	-7.92	-1.717	-0.166	-5.71	-0.958	-0.11	-4.482
	RaceHorses	-0.48	-0.07	-7.88	-0.153	-0.013	-5.42	-0.054	-0.014	-3.67
D	BasketballPass	-0.68	-0.27	-7.92	-0.281	-0.164	-4.253	-0.339	-0.164	-4.57
	BQSquare	-2.15	-0.21	-13.97	-4.922	-0.419	-9.996	-3.124	-0.336	-7.393
	BlowingBubbles	-0.55	-0.13	-8.72	-0.684	-0.079	-5.38	-0.548	-0.066	-5.664
	RaceHorses	-0.489	-0.06	-7.85	-0.129	-0.007	-4.56	-0.087	-0.008	-2.826
E	Vidyo 1	-0.71	-0.05	-16.78	-1.284	-0.071	-14.508	-0.77	-0.05	-15.459
	Vidyo 3	-0.64	-0.06	-15.95	-0.401	-0.09	-14.814	-0.357	-0.069	-16.869
	Vidyo 4	-0.51	-0.04	-13.64	-0.769	-0.042	-15.415	-0.169	-0.025	-17.17
Average		-0.749	-0.089	-13.103	-0.841	-0.094	-10.014	-0.613	-0.059	-10.881

fast JEM algorithm, the Bitrate decreases by 0.6% with a loss of 0.05 dB in PSNR for the LDB configuration compared to the RA and LDP conditions.

Furthermore, the results obtained show that the proposed approach performs well with high-resolution video sequences, since it can achieve up to 20% reduction in time. For low-resolution class C and D sequences, where a block of 128×128 pixels depicts a huge part of an image, the splitting chances of this block into a quadtree are therefore higher. However, the time reduction is less compared to other classes, but with an insignificant impact of the BR. In summary, the fast FVC algorithm provides a good trade-off between encoding time and coding efficiency.

Figure 12 evaluates the RD curves for video sequences. The four points shown on these graphs represent the QP parameters 22, 27, 22, and 37. The Bitrate (kbps) is shown on the horizontal axis, while PSNR (dB) is shown on the vertical axis, in each chart. The achieved results demonstrate that the fast JEM algorithm offers almost similar performances to the original JEM software, with negligible loss of quality and Bitrate. According to Figure 12, the degradation of quality is important for lower values of QPs.

Figure 13 shows the time saving for sequences of class A1 and B coded in RA configuration while varying the QP from 22 to 37. The reduction in time decreases in proportion to the increase in the QP value. This proposed algorithm achieves 14.65% reduction in execution time for the Camp-

fireParty video and 13.9% for the BQTerrace video for lower QP values.

5.6. Comparative Performance with Other Approaches. For a more in-depth evaluation of the proposed scheme's encoding performance, a comparison with other approaches proposed in [22, 38, 39] is given below. Comparing the two execution times, our proposed scheme saves 13.10%, where only 12.5% is saved by [22], with an insignificant degradation of Bitrate, around 0.7%. Therefore, we confirm that our proposed scheme outperforms the method proposed in [22], and this is due to its ability to quickly split the QTBT partition, which ensures a low FVC complexity.

In the work cited in [38], authors proposed an enhanced fast algorithm of the QTBT structure. This proposed approach skips some partition processes in QTBT to enhance the encoding efficiency. The obtained results show that the proposed method in [38] achieves 10% encoding time saving with less than 0.2% BD-rate loss under the RA profile. Therefore, our proposed scheme outperforms the work cited in [38] in terms of encoding time by 13.10% with 0.7% of Bitrate.

In the work cited in [39], Huang et al. proposed an algorithm to reduce the encoding complexity by reusing the encoder decisions of the same CU explored in previous partition choices. The simulation results report that the proposed fast algorithms can achieve 9% encoding time

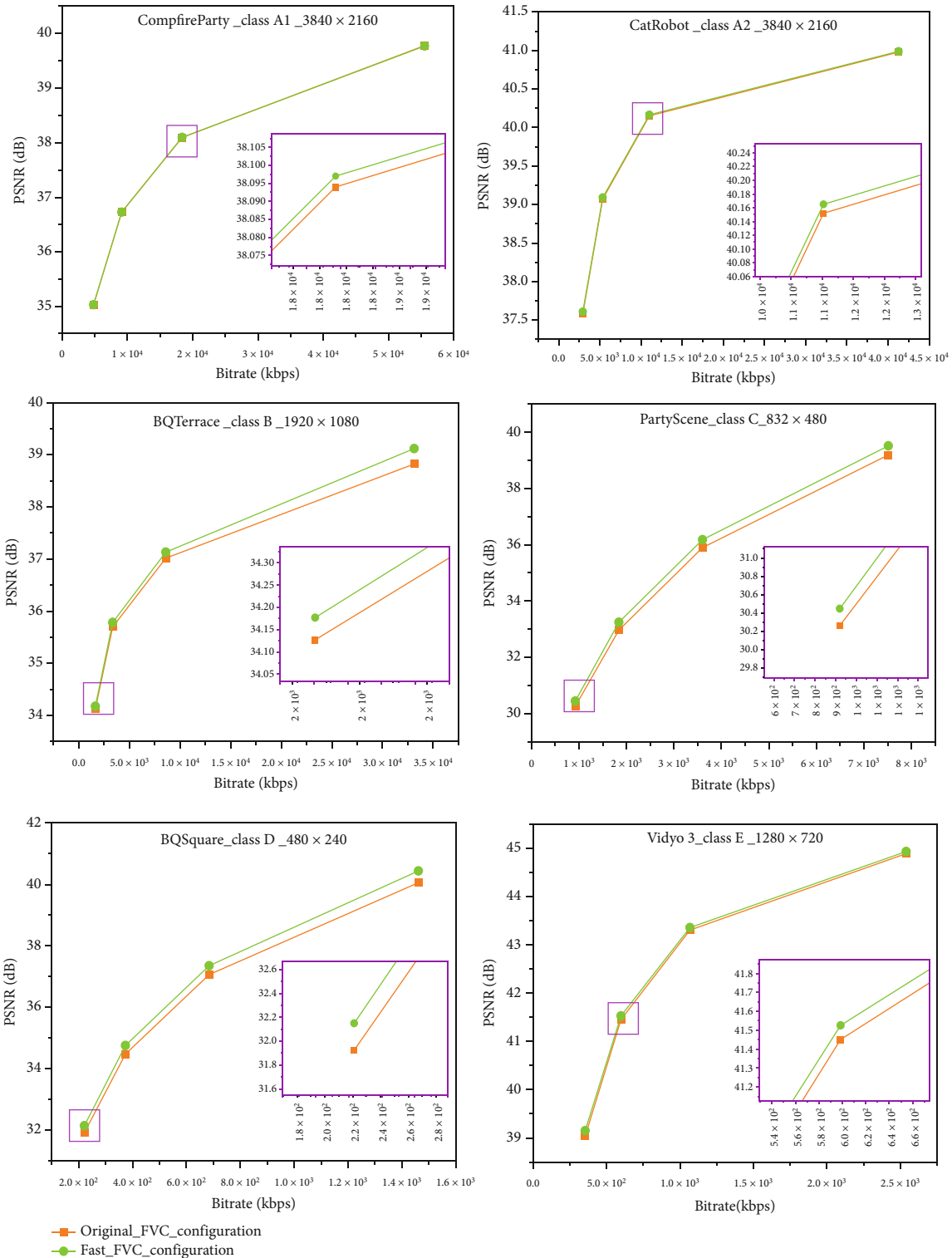


FIGURE 12: RD curves for various test sequences.

reduction with a 0.1% BD-rate in RA configuration, while our proposed approach saves encoding time of 13.10% with an insignificant degradation of Bitrate, around 0.7%. When comparing our work to the state-of-the-art approaches in [22, 38, 39], we can conclude that our proposed scheme performs better in terms of the encoding efficiency.

6. Open Issue and Future Works in FVC Based on Artificial Intelligence Tools

6.1. *Lightweight Machine Learning Approaches for FVC.* Block partition structure is a critical module in the video coding scheme to achieve a significant gap in compression

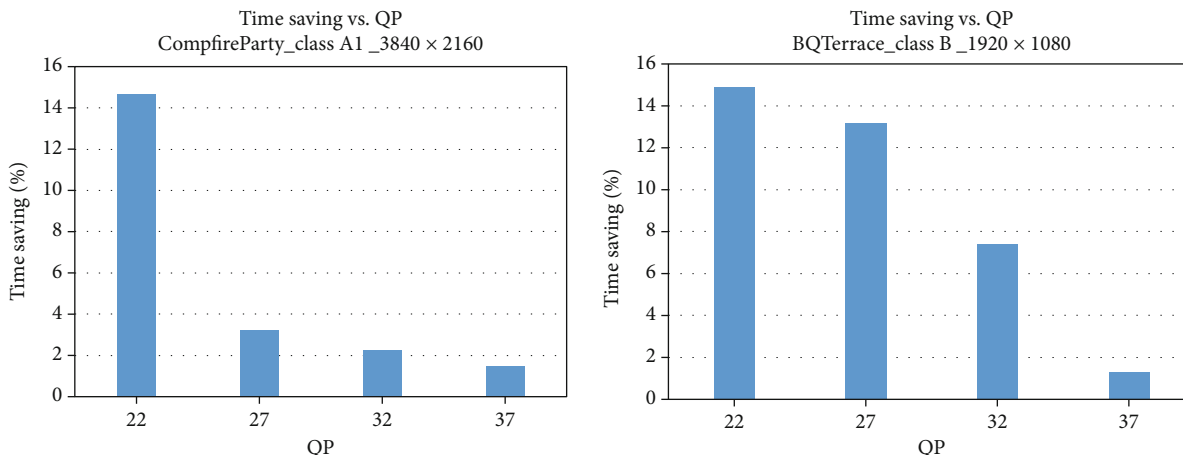


FIGURE 13: Time saving curves for sequences encoded with QP37, 32, 27, and 22 in RA.

performance. Under the exploration of the FVC standard, a new quadtree binary tree (QTBT) block partition structure has been introduced. In addition to the QT block partitioning defined in High-Efficiency Video Coding (HEVC) standard, new horizontal and vertical BT partitions are enabled, which drastically increases the encoding time compared to HEVC. In this regard, a lightweight and tunable QTBT partitioning scheme based on a machine learning approach could resolve this issue.

6.2. End-to-End Deep Learning Approaches for FVC. Conventional video compression approaches use the predictive coding architecture and encode the corresponding motion information and residual information. In this regard, taking advantage of both classical architectures in the conventional video compression method and the powerful nonlinear representation ability of deep neural networks, we have a deep end-to-end video compression model that jointly optimizes all the components for video compression [40, 41]. In this context, all the modules will jointly learn through a single loss function, in which they will collaborate with each other by considering the trade-off between reducing the number of compression bits and improving the quality of the decoded video. Thus, the deep end-to-end video compression model can be advantageous to enhance FVC performance.

6.3. Deep Learning Approaches for FVC. Multimedia video streaming requirement has increased exponentially and the video currently consumes 75% of the internet traffic. Due to which video streaming and storage are a huge challenge for service providers. Image and video compression algorithms rely on FVC codecs which are encoders and decoders that lack adaptability. Due to the advent and advances in deep learning, these issues can be solved by replacing the coding tools for FVC with a new deep learning model. Yet, an intelligent fast algorithm based on deep learning models [42] will be proposed to achieve higher encoding efficiency, lower computational complexity, and better visual quality of the next generation video coding developed on 2020, named VVC [43, 44].

7. Conclusion

In this paper, an overview of the FVC standard versus HEVC has been presented. We propose to compare the three JEM configurations in terms of evaluation metrics (encoding time, Bitrate, and PSNR). The most important feature of FVC is QTBT partition, which simplifies coding units and improves compression efficiency. We adopt a fast decision algorithm for reducing FVC encoding complexity. Experimental results reveal that the proposed method can consistently achieve promising performance for various video sequences.

Abbreviations

FVC:	Future Video Coding
JVET:	Joint Video Exploration Ream
HEVC:	High-Efficiency Video Coding
JCT-VC:	Joint Collaborative Team on Video Coding
MPEG:	Motion Picture Experts' Group
VCEG:	Video Coding Expert Group
AVC:	Advanced Video Coding
JEM:	Joint Exploration Model
VVC:	Versatile Video Coding
UHD:	Ultra-high definition
QTBT:	Quadtree plus binary tree
CTU:	Coding Tree Unit
CU:	Coding Unit
DC:	Direct Current
ATMVP:	Alternative Temporal Motion Vector Prediction
STMVP:	Spatial Temporal Motion Vector Prediction
SAO:	Sample Adaptive Offset
ALF:	Adaptive Loop Filter
CABAC:	Context Adaptive Binary Arithmetic Coding
RA:	Random Access
LD:	Low Delay
ME:	Motion estimation
ASR:	Adaptive Search Range
TZSearch:	Test zonal search
QP:	Quantization Parameter
AI:	All Intra
IDR:	Instantaneous Decoder Refresh

GOP:	Group of Picture
BR:	Bitrate
PSNR:	Peak signal-to-noise ratio
T:	Encoding time
CTC:	Common Test Conditions
ESD:	Early Skip Detection
CBF:	Coded Block Flag
ECU:	Early CU Termination
RD:	Rate distortion
DMV:	Differential motion vector
QoE:	Quality of Experience.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

- [1] 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, pp. 1649–1668, 2012.
- [2] N. Bahri and R. Khemiri, "Optimised HEVC encoder intra-only configuration," *IET Computers & Digital Techniques*, vol. 14, no. 6, pp. 256–262, 2020.
- [3] J. Chen, E. Alshina, G. J. Sullivan, J. R. Ohm, and J. Boyce, "Algorithm description of joint exploration test model 1," *Joint Video Exploration Team (JVET) of ITU-T SG*, vol. 16, pp. 20–26, 2016.
- [4] H. Schwarz, C. Rudat, M. Siekmann, B. Bross, D. Marpe, and T. Wiegand, "Coding efficiency/complexity analysis of JEM 1.0 coding tools for the random access configuration," in *Document JVET-B0044 3rd 2nd JVET Meeting*, San Diego, CA, USA, 2016.
- [5] "Requirements for Future Video Coding (H.FVC)," Report SG16-R1, Annex I, 2017.
- [6] F. De Rango, M. Tropea, and P. Fazio, "Multimedia traffic and video distribution over broadband wireless networks," *In-tech*, pp. 001–030, 2010.
- [7] Z. Cheng, L. Ding, W. Huang, F. Yang, and L. Qian, "A unified QoE prediction framework for HEVC encoded video streaming over wireless networks," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Cagliari, 2017.
- [8] Q. Tu, X. Guo, A. Men, and J. Q. Jun Xu, "A frame-level HEVC rate control algorithm for videos with complex scene over wireless network," in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, Seoul, Korea (South), 2014.
- [9] L. Chen, M. Yang, L. Hao, and D. Rawat, "Framework and challenges: H. 265/HEVC rate control in realtime transmission over 5G mobile networks," in *Proceedings of the 10th EAI International Conference on Mobile Multimedia Communications*, pp. 192–198, China, December 2017.
- [10] M. Nikzad, A. Bohlooli, and K. Jamshidi, "Video quality analysis of distributed video coding in wireless multimedia sensor networks," *International Journal of Information Technology & Computer Science*, vol. 7, no. 1, pp. 12–20, 2014.
- [11] ISO/IEC, ITU-T, "HEVC test model (HM) reference software," 2016, January 2017. <https://hevc.hhi.fraunhofer.de/>.
- [12] J.-R. Ohm and M. Wien, "Status and perspectives of video coding standardization beyond HEVC. IBC2006 paper proposal," in *IBC2006 Paper Proposal*, RWTH Aachen University, Institut für Nachrichtentechnik, Germany, 2017.
- [13] Y.-C. Lin, J.-C. Lai, and H.-C. Cheng, "Coding unit partition prediction technique for fast video encoding in HEVC," *Multimedia Tools and Applications*, vol. 75, no. 16, pp. 9861–9884, 2016.
- [14] J. An, H. Huang, K. Zhang, Y.-W. Huang, and S. Lei, "Quadtree plus binary tree structure integration with JEM tools," *Technical Report JVET-B0023*, 2016.
- [15] J. Chen, E. Alshina, G. J. Sullivan, J. R. Ohm, and J. Boyce, "Algorithm description of joint exploration test model 7 (JEM 7)," *Joint Video Exploration Team (JVET) of ITU-T SG*, vol. 16, pp. 13–21, 2017.
- [16] Z. Wang, S. Wang, J. Zhang, and S. Ma, "Local-constrained quadtree plus binary tree block partition structure for enhanced video coding," in *IEEE Visual Communications and Image Process*, pp. 1–4, Chengdu, China, 2016.
- [17] Y. Yamamoto, "AHG5: fast QTBT encoding configuration," *Technical Report JVET-D0095*, 2016.
- [18] J. Li, C. Wang, X. Chen, Z. Tang, G. Hui, and C.-C. Chang, "A selective encryption scheme of CABAC based on video context in high efficiency video coding," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12837–12851, 2018.
- [19] R. Khemiri, H. Kibeya, H. Loukil, F.-E. Sayadi, M. Atri, and N. Masmoudi, "Real-time motion estimation diamond search algorithm for the new high efficiency video coding on FPGA," *Analog Integrated Circuits and Signal Processing*, vol. 94, no. 2, pp. 259–276, 2018.
- [20] J. Kim, D. Jun, S. Jeong et al., "An SAD-based selective bi-prediction method for fast motion estimation in high efficiency video coding," *ETRI Journal*, vol. 34, no. 5, pp. 753–758, 2012.
- [21] D. García-Lucas, G. Cebrián-Márquez, A.-J. Díaz-Honrubia, and P. Cuenca, "Acceleration of the integer motion estimation in JEM through pre-analysis techniques," *The Journal of Supercomputing*, vol. 75, no. 3, pp. 1203–1214, 2019.
- [22] D. García-Lucas, G. Cebrián-Márquez, A. J. Díaz-Honrubia, and P. Cuenca, "Accelerating the CU partitioning decision in an HEVC-JEM transcoder," *Multimedia Tools and Applications*, vol. 79, no. 3-4, pp. 2047–2067, 2020.
- [23] R. Khemiri, H. Kibeya, F.-E. Sayadi, N. Bahri, M. Atri, and N. Masmoudi, "Optimisation of HEVC motion estimation exploiting SAD and SSD GPU-based implementation," *IET Image Processing*, vol. 12, no. 2, pp. 243–253, 2018.
- [24] R. Khemiri, H. Kibeya, F. E. Sayadi, N. Bahri, M. Atri, and N. Masmoudi, "Optimization of HEVC motion estimation exploiting SAD and SSD GPU-based implementation," *IET Image Processing*, vol. 12, pp. 243–253, 2017.
- [25] Y. Li, G. Yang, Y. Zhu, X. Ding, and X. Sun, "Unimodal stopping model-based early SKIP mode decision for high-efficiency video coding," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1431–1441, 2017.
- [26] R.-H. Gweon and Y.-L. Lee, "Early termination of CU encoding to reduce HEVC complexity," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 95, pp. 1215–1218, 2012.

- [27] J. Kim, J. Yang, K. Won, and B. Jeon, "Early determination of mode decision for HEVC," in *IEEE in Picture Coding Symposium*, pp. 449–452, Krakow, Poland, 2012.
- [28] T.-K. Lee, Y.-L. Chan, and W.-C. Siu, "Adaptive search range for HEVC motion estimation based on depth information," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 10, pp. 2216–2230, 2017.
- [29] S.-H. Park, T. Dong, and E.-S. Jang, "Low complexity reference frame selection in QTBT structure for JVET future video coding," in *IEEE in Advanced Image Technology, International Workshop*, pp. 1–4, Chiang Mai, Thailand, 2018.
- [30] N. Purnachand, L.-N. Alves, and A. Navarro, "Improvements to TZ search motion estimation algorithm for multi-view video coding," in *IEEE in Systems, Signals and Image Processing, 19th International Conference*, pp. 388–391, Vienna, Austria, 2012.
- [31] S. Ahn, B. Lee, and M. Kim, "A novel fast CU encoding scheme based on spatiotemporal encoding parameters for HEVC inter coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 422–435, 2015.
- [32] Z. Wang, S. Wang, J. Zhang, S. Wang, and S. Ma, "Effective quadtree plus binary tree block partition decision for future video coding," in *IEEE in Data Compression Conference*, pp. 23–32, Snowbird, UT, USA, 2017.
- [33] M.-J. Chen, Y. D. Wu, C.-H. Yeh, K.-M. Lin, and S.-D. Lin, "Efficient CU and PU decision based on motion information for inter-prediction of HEVC," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4735–4745, 2018.
- [34] Z. Wang, S. Wang, J. Zhang, S. Wang, and S. Ma, "Probabilistic decision based block partitioning for future video coding," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1475–1486, 2018.
- [35] K. Suehring and X. Li, "JVET common test conditions and software reference configurations," *Technical Report JVET-H1010*, 2017.
- [36] D. Grois, T. Nguyen, and D. Marpe, "Performance comparison of AV1, JEM, VP9, and HEVC encoders," in *Applications of Digital Image Processing XL*, California, USA, 2017.
- [37] ISO/IEC, ITU-T, "Joint exploration test model (JEM) reference software (JEM-7.1)," 2017, October 2017, <https://jvet.hhi.fraunhofer.de/>.
- [38] P. H. Lin, C. L. Lin, and Y. Jen, "AHG5: Enhanced Fast Algorithm of JVETe0078," 2017, Document JVET-F0063.
- [39] H. Huang, S. Liu, and Y.-W. Huang, "AHG5: Speed-Up for JEM-3.1," 2016, Document JVET-D0077.
- [40] S. Bouaafia, R. Khemiri, A. Maraoui, and F. E. Sayadi, "CNN-LSTM learning approach-based complexity reduction for high-efficiency video coding standard," *Scientific Programming*, vol. 2021, 10 pages, 2021.
- [41] S. Bouaafia, S. Messaoud, R. Khemiri, and F. E. Sayadi, "VVC in-loop filtering based on deep convolutional neural network," *Computational Intelligence and Neuroscience*, vol. 2021, 9 pages, 2021.
- [42] S. Bouaafia, R. Khemiri, F. E. Sayadi, and M. Atri, "Fast CU partition-based machine learning approach for reducing HEVC complexity," *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 185–196, 2020.
- [43] M. Z. Wang, S. Wan, H. Gong, and M. Y. Ma, "Attention-based dual-scale CNN in-loop filter for versatile video coding," *IEEE Access*, vol. 7, pp. 145214–145226, 2019.
- [44] S. Bouaafia, R. Khemiri, and F. E. Sayadi, "Rate-distortion performance comparison: VVC vs. HEVC," in *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 440–444, Monastir, Tunisia, March 2021.