Received: May 10, 2024. Revised: May 1, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Fast Depth Decision Algorithm with Spatial Homogeneity and Threshold Value Modified on Versatile Video Coding (VVC) Partitioning

Alexander V. Bukit[1] **Suwadi[1]*** **Wirawan[1]** **Titiek Suryani[1]** **Endroyono[1]**

[1]*Department of Electrical Engineering, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*
* Corresponding author's suwadi@ee.its.ac.id

Abstract: The VVC standard has achieved great performance improvements and has outperformed its predecessor HEVC (High Efficiency Video Coding) with more than 30% compression rate by VVC with no significant loss in quality. The VVC standard has a very broad use, which can be used for camera capture results up to immersive applications with HDR (High dynamic range) and SDR (Standard Dynamic Range) quality. However, with an increase in VVC's excellent compression capability, it comes with an increase in the encoding process complexity, especially in the partitioning process because VVC implements a block partition method that is far more flexible than its predecessor HEVC, namely the Quadtree method with multi-type tree structures (QT+MTT) so it requires a lot of work. In this research, we propose a fast depth decision algorithm with spatial homogeneity and threshold value modified method to save the amount of time and overall process complexity using uniform areas and decision-making of CU (coding unit) split type. The VVC reference program VTM 14.2 implements the suggested fast-depth decision algorithm and tests it under common test conditions (CTC) to determine how effective it is. Fourteen sequences and the All-Intra primary configuration are used in the simulation. The average encoding time savings (T_{saving}) are evaluated and obtained 36.70% with insignificant quality degradation Bjontegaard delta bit rate (BDBR) of 1.37%.

Keywords: VVC, Partitioning, Fast decision.

1. Introduction

The design of VVC follows the hybrid principle in the video coding method. Frames of video are divided into blocks. Versatile Video Coding (VVC) is a video coding technique that is currently being developed and has been designed to be the standard for many video applications and offers significant reliability compared to previous generations, such as the H.265/HEVC encoding technique [1]. Even though VVC has excellent compression capabilities by more than 30% over HEVC, besides that it requires a heavy processor load and considerable time for encoding [2]. Based on this, we need a video coding technology that can reduce video size without significant loss of quality, so that it can be transmitted and saved more efficiently and economically.

Media video generates quite large file sizes as it handles sound and moving images. Substantially larger video size is produced by the trend of using higher video resolutions, which attempts to improve the appearance of the video. However, the VVC complexity is 18 times higher than HEVC [3]. An effective encoding is required due to the growing popularity of videos with low resolution up to high-definition quality, which increases the requirement for storage, data transfer, and display.

Like HEVC and other advance video codecs, VVC is intended to be a block-based hybrid video codec. The new standard offers a wide range of additional high-level extensions and block-level coding tools that enable better prediction and residual coding to provide the promised bit-rate reduction and flexibility. Because VVC has high coding complexity, it needs to be optimized to decrease the coding time. The biggest time and processor load used is during the partitioning stage. In the process of video coding, the VVC standard provides maximum depth for partitioning up to binary and ternary tree (BTT) structures so that the coding complexity increases exponentially. The partition space, quadtree with nested multi-type tree (QTMT) is centralized as well as 95% of the time [3].

In this paper, we propose an optimization technique with spatial homogeneity and threshold value modified method for VVC, which minimizes coding time. The optimal CU partitioning pattern will be obtained by minimizing the Rate Distortion Cost (RD_{cost}) from all possible partitioning patterns by deciding the maximum depth in the partitioning process. In this study, optimization of the VVC standard will be carried out using the Fast Depth Decision method. Therefore, we formulated fast depth decision algorithm method to optimize VVC operating conditions to save the amount of time. The results of the experiments show that our proposed method can decrease the typical coding time (t%) average to 36.70% with negligible quality loss.

The remainder of this paper is structured as follows. In Section II, related works. Section III is a brief review of VVC and the scheme of partitioning. In Section IV Proposed Methodology. Results and Discussion. Finally, Section VI are the conclusions.

2. Related works

The optimization of VVC has been extensively discussed in the literature. As speed and encoding time become one of the most important factors in video encoding, there is still work to be done. Low Complexity VVC standard for low bit rate applications to achieve better bit rates in VVC standard to determine the effectiveness of some of the new coding methods included in VVC at low bit rate and low resolution [4]. To ascertain the effectiveness of some of the new coding methods included in VVC at low resolution and low bitrate, they suggested an optimization framework. By turning off certain of the VVC coding methods, the model test results showed a reduction in encoding complexity of up to 56% at 384 × 216 resolution. A similar study by Abdoli et al. [5], short-range internal display content prediction in Versatile Video Coding (VVC), The approach is called in-loop residual coding with scalar quantization and uses in-block pixels as reference instead of the usual out-block ones. In order to partially recreate the block at the pixel level during the prediction, they suggested adding another in-loop residual signal. The suggested approach is primarily intended for high detail textures, which call for a deep block partitioning structure. As a result, it is only designed to function on 4 × 4 blocks; additional block splitting is prohibited, and the usual algorithm is still unable to accurately anticipate the texture. A similar study by Jing et al. [6] proposed a Fast VVC Intra Prediction Based on Gradient Analysis and Multi-Feature Fusion CNN. This paper suggests a method that leverages a multi-feature fusion CNN and gradient analysis to streamline VVC intra prediction. The Sobel operator is utilized to calculate the gradient of CUs, and the outcomes of the calculation are utilized to make predecisions. Tang et al. proposed a fast block partition algorithm for both intra-coding and inter-coding, based of Canny edge detector [7]. They suggested a quick block partition method for both intra- and inter-coding in order to better balance encoder complexity with coding performance. In intra coding, edge features are extracted using the block-level based Canny edge detector in order to skip vertical or horizontal partition modes and perform early termination. A similar study by Cui et al. [8], this research suggested a fast CU partition technique for VVC intra coding that is based on perception. To imitate the human visual system, the just noticeable difference model (JND) is used. JND variance is used to reflect the perceived texture consistency and is utilized for early partition termination and division mode selection. According to experimental findings, the suggested approach typically saves 31.13% time complexity reduction, with 1.32% Bjontegaard delta bit rate (BDBR) increase. Below a particular threshold, which may be accurately approximated using the JND model, the human visual system (HVS) is unable to notice distortion due to the poor visual resolution. This work primarily uses the enhanced pixel-domain JND model, taking into account the effects of contrast masking, luminance adaptation, and content regularity. In [9] they suggested the method to estimate the ideal partition map from the pixels using a convolutional neural network (CNN). They provide a partition map prediction technique for quick block partitioning in VVC intra-frame encoding. CNN structure for partition map prediction called Down-Up-CNN, which mimics the recursive nature of the partition search procedure, and create a post-processing algorithm to modify the network output partition map in order to produce a block partitioning structure that complies with standards.

A variety of studies are being conducted to use online and offline learning techniques including machine learning (ML) and deep learning (DL) to lessen the computational strain of VVC as data science advances [10]. While utilizing learning techniques may result in some time savings, these methods mainly rely on user-defined traits and orderly training datasets. The majority of training datasets for low-complexity video coding are private and only a tiny subset is available for academic usage. Online learning methodologies also require some additional costs for the training procedure. Therefore, there is no concern about using the pricey learning technique with a private database effectively in low-complexity video coding.

While there is some time that may be saved using learning techniques, these techniques heavily rely on user-defined characteristics and tidy datasets of training. Only a small number of training datasets are accessible for academic use for low-complexity video coding, and the majority are private. Online learning methodologies also require some additional costs for the training procedure. Therefore, there is no concern about using the pricey learning technique with a private database effectively in low-complexity video coding.

This article focuses on the applications of VVC standard encoding tools. The biggest time and processor load used is during the partitioning stage. In the process of video coding, the VVC standard provides maximum depth for partitioning up to binary and ternary tree (BTT) structures so that the coding complexity increases exponentially. We have developed optimizations for video under VVC conditions to reduce the complexity of the VVC standard with a fast depth decision algorithm with negligible effect on decreasing BDBR (Bjontegaard delta bit rate) level.

This paper contributes two different fast block partitioning methods. We first propose an optimization with the threshold value of texture detail scenarios. More specifically, the threshold (Th) value is used to assess the textural complexity (TC) of CU. As a result, when a CU texture complexity value is less than Th, it will be determined to forego the ISP coding mode during video coding. Second, we experimentally show that CUs with simple texture video encoding at the partitioning stage or decision-making of CU split type will tend to follow a repeating pattern, namely the tendency to follow the same pattern as before (previous pattern). An effective termination and bypass method is used to lower the computing needs and accelerate the intra-coding process.

To the best of our knowledge, up to now, this is the first work in literature that proposed fast depth decision algorithm with spatial homogeneity and threshold value modified method on Versatile Video Coding (VVC) Partitioning.

3. Versatile video coding (VVC)

The latest video encoding standard, VVC is developed by JVET (Joint Video Experts Team), which aims at improving compression efficiency in comparison with the existing High Efficiency Video Compression Standard and saving by more than 30% over HEVC. Compared with the predecessor, the quality of video compression has improved. For instance, standard video coding could lower the video transmission cost and media storage while improving or maintaining video quality. By maintaining the same channel capacity, this upgrade can lower the cost of video data transmission and storage equipment and regularly increase the video quality of TV (television) broadcasts. The increased VVC compression efficiency comes at a cost of complexity. A lot of things are complicated, such as the computing complexity of the compute and the local and global memory requirements, as well as memory bandwidth [11]. VVC utilizes a hybrid video encoding method, similar to HEVC and other advanced video codecs.

Inter/intra prediction, conversion procedure, and block release. The input image is initially divided into square coding tree units (CTUs) in this process with fixed size (up to 128 x 128 exposure samples) for encoding. Advanced extensions allow further division of the drawing based on these basic subdivisions and the definition of logical sub-drawing areas such as subplots, tiles, and slices. Using a flexible partitioning system each CTU is divided into a group of rectangular coding units (CUs).

The operation of the video compression calculation requires more decisions which increase CPU load when encoding a video. It is possible in VVC to use QuadTree using binary and ternary structures with nested multi-type trees (MTT) foreach CTU quad-tree plus binary tree (QTBT). In this scenario, a binary or inverted partition structure can be used to divide the quadtree leaf nodes either horizontally or vertically. Then, to predict and code the residues directly, a square or rectangular CU is used.

Assuming (u,v) is the block's center and (d) is the corresponding difference vector, as shown in Fig. 1, VVC processes potential displacement values for block Y(u,v) in the current view block (u_R, v_R) = (u+d, v). The reference block YR(u_R, v_R) points to the prediction unit whose location is (u_R, v_R) in the

Received: May 10, 2024. Revised: May 1, 2024.

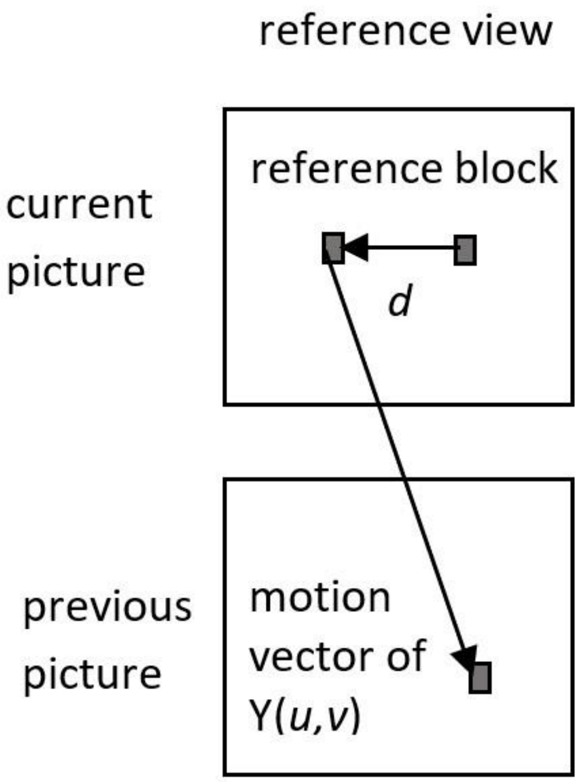

Figure. 1 Motion vector correspondences employing the disparity vector d between a block in the current image and the previously encoded reference view

reference view from the sample position of this block.

VVC Test Model (VTM) 2.0 to VTM 20.2 are some of the open-source VVC software versions that are available on the GitHub website [12]. The VVC Test Model (VTM) 14.2 was employed in this study; it comprises 65 angle prediction modes, including DC mode and planar mode.

The prediction for each block created and the transformation that identity can be mapping is applied to the residues. A block-based video codec also used in earlier H.265/HEVC and H.264/AVC standards. This is followed by entropy coding of predictions and transformation state data. Intra-prediction forms the basis for the application VVC standard. This can be seen in the following two scenarios. First, to ensure occasional use (not including incremental decoding update scenario), I-frames are added to each video sequence. Then the local temporal scene changes. Even if in the usual series of videos, between predictions the most frequently used blocks are coded for internal prediction.

The biggest time and processor load used is during the partitioning stage. In the process of video coding, the VVC standard provides maximum depth for partitioning up to binary and ternary tree (BTT) structures so that the coding complexity increases exponentially. The previous standards, namely HEVC and H.264/AVC, adopted a hybrid video coding framework, which was further developed to the H.266/VVC standard by adopting a block-based hybrid video coding structure in addition to what was previously explained, namely the QTMT coding block structure.

VVC needs to be optimized to balance coding efficiency and computing cost. The best splitting strategy for a CU is found by reducing the RD_{cost} across all feasible splitting strategies. The fundamental distinction between the quad-tree partitioning structure in HEVC and QTMT in VVC is that QTMT permits ternary or binary splitting at any quad-tree leaf node. As a result, reducing computing complexity while maintaining a high-quality visual experience is a critical research goal that most academics are attempting to solve. The possible number of intra modes for VVC intra-coding is 65, which is extremely significant and consumes the encoding time. Due to this, the majority of research works aimed to reduce the candidate modes, i.e., to quickly decide inside the mode by statistically evaluating the original HM (HEVC Test Mode) nature and accumulating data to avoid unnecessary intra-candidate modes before RD optimization (RDO) verification. The Coding Tree Unit (CTU) approach used in HEVC is the same. The CTU is divided, first into a quad-tree, and then the leaf nodes of the quad-tree are successively partitioned using the multi-type tree structure. The multi-type tree structure has four splitting types: vertical binary splitting, horizontal binary splitting, vertical ternary splitting, and horizontal ternary splitting as shown in Fig. 2.

To restrict the sizes of the quad-tree and multi-type tree, certain encoding settings are established. The Maximum Ternary Size (MaxTtSize), Maximum Binary Tree Size (MinBTSize), and Maximum Quad-tree Size (MaxQTSize) parameters, for example, limit the maximum root node size for quaternary, binary, and ternary trees, respectively. [13] The minimal root node size for quaternary, binary, and ternary trees is limited by the Minimum Ternary Size (MinTtSize), Minimum Binary Tree Size (MinBTSize), and minimal Quad-tree Size (MinQTSize).

The methods to reduce the computational overhead effectively divide intra-mode determination into three major steps: most probable modes (MPMs), rough mode decision (RMD), and RDO. RMD creates a candidate set of N modes with the lowest HAD costs by computing the

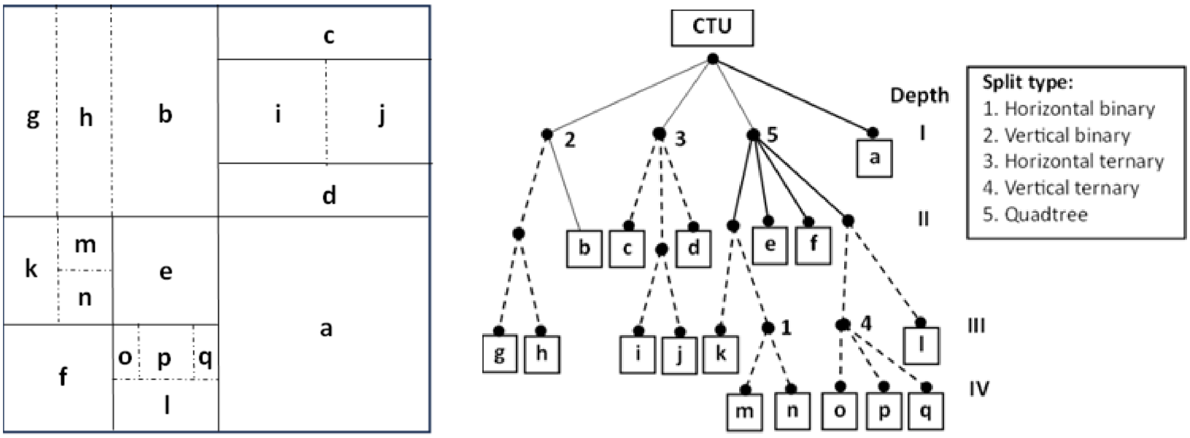

Figure. 2 The multi-type tree structure of CU partitioning

approximate RD_{cost} of each viable mode based on the Hadamard transform (HAD), also known as the sum of absolute transformed differences (SATD) cost. Taking advantage of the high spatial rectification of nearby prediction units (PUs), the MPMs set from the adjacent PUs are retrieved and united with the set of candidates of the present PU. To select the best intra-prediction mode at the lowest RDO_{cost}, the final candidate list will be employed in the RDO process [14].

The size of the CU can range from 128×128 to 8x4 or 4x8 depending on these parameters and splitting types. The coding units (CUs), are known as multi-type tree leaf nodes which are also used for transformation and prediction. As a result, the notions of PU and TU are no longer present in the new QTMT framework. Furthermore, unlike HEVC, the partitioning procedure in I-slices is distinct for the chroma and luma components. To more reliably anticipate the present CU and 3 extra MPMs, VVC increases intra-prediction modes from 35 to 67, which results in higher coding efficiency for intra-coding when compared to the cost of H.265/HEVC computational. Extended angular modes indicate that VVC adjusts certain modifications in three phases in HEVC.

In VVC, a new mode called intra sub-partitions coding (ISP) has been proposed. It splits the luminance intra prediction blocks into two or four equal-sized sub-partitions, each of which has at least sixteen samples, either vertically or horizontally. The block size that can utilize ISP coding mode is 64 × 64 and the minimum sub-partition size is 4 × 8 (or 8 × 4). The matching block is partitioned into four sub-partitions if the size of the block exceeds 4 × 8 (or 8 × 4). When a block’s dimensions equal 4 × 8 (or 8 × 4), the matching block is split into two smaller halves. Each sub-partition’s processing in VVC is comparable to the intra prediction block’s processing. Based on the prediction signal, the encoder generates the residual signal, which is further modified, quantized, and entropy encoded before being delivered to the decoder. After a sequence of operations at the decoder side, such as inverse quantization, entropy decoding, and inverse transformation, the residual signal is recovered. The reconstructed samples are then generated by combining the residual signal with the prediction signal. Following processing of a sub-partition, the prediction signal of the subsequent sub-partition can be determined using the reconstructed samples. This process is repeated until all sub-partitions have been encoded.

The fast CU splitting algorithms for HEVC are insufficient or unsuitable for the evolving standard, even though both standards are based on a block-based hybrid video coding architecture. The total number of potential modes in the candidate list is decreased from 35 to 3 due to the three-step intra-mode selection process. Although there are still reducible intra-modes to ease the computing strain of the exhausted intra-mode selection for all conceivable coding unit (CU) sizes, RMD must still compute the HAD cost for 35 modes. By first determining the gradient direction for each pixel in the current PU (prediction unit) and then sorting all available modes according to their total gradient amplitude, the gradient-based intra-mode choice technique. According to the size of the PU, an assortment of candidate prediction modes with the highest values are selected to provide a mode list of smaller candidate for RDO and RMD than the initial HM. The list of candidates includes DC and planar modes fairly because they are the most likely intra-modes. The five closest PUs’ most relevant modes are added to the candidate list in order to make the most of the spatial correlation among them.

Wang et al. [15] in addition to the prior strategies that use additional processing such as gradient computing and edge detection, statistically evaluate the reference samples of the nearby CUs to adaptively pick the set of best candidate for intra-prediction from three predetermined candidate sets. Determines the top candidate group for RMD out of eight groups by calculating the mean of horizontal and vertical gradients and determining four empirical thresholds based on the strong correlation between the textural feature and the ratios of the horizontal and vertical gradients [16]. All of the aforementioned methods can successfully shorten the encoding process by a certain degree. To obtain relevant information for removing pointless intra-modes from the candidate list, an additional step, such as edge recognition and gradient computation, must be carried out.

To more reliably anticipate the present CU and 3 extra MPMs, VVC increases the modes of intra-prediction from 35 to 67, which results in higher coding efficiency for intra-coding when compared to the H.265/HEVC computational cost. Extended angular modes suggest that three phases of adjustment are used by VVC for some adjustments. The SATD process has two stages, the first of which is for original angular modes and the second of which is for extended angular modes, according to [17]. In the first stage, 35 modes (DC, planar, with 32 even angular prediction modes) are estimated to have initial SATD costs. The first round then selects the candidate list of the optimal N prediction modes with the lowest SATD costs; N depends on the width and height of the present CU. Only the SATD costs of two close modes are calculated in the second round for any parent mode that is greater than 3 and lower than 66. As a result, the total number of modes required to compute the SATD cost can be decreased from 67 to no more than (35 + 2N). The RMD method then chooses the RMD modes with the lowest SATD costs from the candidate list of N intra modes. After the RMD process, MPMs and RMD modes are combined to form the final candidate list for the RDO operation. Additionally, because HEVC employs DCT (discrete cosine transform) type 2 (DCT2) for transform coding, VTM introduces two additional transforms known as discrete sine transform type 8 (DST8) and discrete cosine transform type 7 (DCT7) for the transform coding. The RDO technique must ascertain the RDcost for each intra mode in the final candidate list since there are five different transform combinations for the horizontal and vertical directions, including (DCT2, DCT2), (DST7, DST7), (DCT8, DST7), and (DCT8, DCT8). The expanded angular modes with MTS (Multiple Transform Selection) can provide outstanding coding efficiency even with very complex computational requirements. Fig. 2 shows the ternary tree structure of the CU partition and the quadtree plus binary debut of VCC. The sophisticated CU splitting structure and expanded angular modes indicate that VVC experiences an enormous computing burden. Therefore, complexity reduction video coding has become increasingly popular for video applications on power-constrained devices.

After all, MPMs in the current CU have had their Hadamard costs searched, the best MPM with the lowest Hadamard cost is selected as the first search mode. Then, a gradient descent search based on a bidirectional technique with adjustable step size is used to find the candidate mode list for the RMD process. By their recommendations, Zhang et al. [17] categorize all angular modes into four groups (0-degree, 45-degree, 90-degree, and 135-degree angles) and compute the gradient value for each pixel of the current CU using the Canny operator. The energy for each group is then computed and placed in descending order by projecting the gradient vector. Then, as opposed to computations, the SATD costs are only assessed for the angular modes associated with the primary and/or secondary groups out of the 35 angular modes in the initial stage of the RMD process. This technique can save 54.91% coding time.

4. Proposed methodology

The intra-prediction effectiveness of VVC can be increased by using Fast Depth Decision Algorithm method. However, based on our observations and study, the coding effectiveness of Fast Depth Decision Algorithm coding mode would deteriorate when the partition process of a CU. This section suggests a Fast Depth Decision Algorithm coding mode optimization method based on CU texture complexity that may assess whether a CU has to employ Fast Depth Decision coding mode depending on CU texture complexity analysis in advance, to reduce code time with minimal RD_{cost}.

In VVC, only the nearby samples were used for intra-prediction. Due to the fact that intra-predicted blocks of a picture are not always parallelizable at a decoding, they present an additional difficulty for real-time decoding, in contrast to motion compensated prediction. The spatial redundancy could not be fully removed in this way. The optimal CU size is determined by recursively traversing all CU levels in the VVC intra-prediction coding process, during which the RD_{cost} of each CU at each coding depth level is assessed. VTM supports the

Received: May 10, 2024. Revised: May 1, 2024.

Table 1. Notation lists that are utilized in the proposed method

Notation	Meaning
$Energy_{AC}$	AC coefficients energy of a numbers of row x column
f(x,y)	Intensity of the pixels in the sample at point (x, y)
fmin	Intensity alternately match the permitted minimum
fmax	Intensity alternately match the permitted maximum
D_{CU}	Detail of CU
E_R	Error ratio
M_D	The number of blocks encoded using coding depth 0
N_D	The number of blocks encoded using a coding depth bigger than 0
Th_U	Upper threshold
Th_L	Lower threshold
Th_C	Candidate value threshold
RD_{Cost}	Rate-distortion cost of each division type's
R	The encoded bits number

recently introduced QTMT structure, unlike HM. A CU may examine the horizontal binary splitting, vertical binary splitting, horizontal ternary splitting, vertical ternary splitting, and then the quad-tree splitting in QTMT. The produced encoding information can be used to skip ahead to the remaining partitions after the horizontal splitting. The parent CU's and its two child CUs' intra prediction modes have previously been established, after the horizontal binary partition of a CU is finished, and the ultimate splitting patterns of these children's CUs are also determined and reachable. The notation lists that are utilized in the proposed method listed in Table 1.

4.1 Spatial texture homogeneity

It is not essential to compute the RD_{cost} for CUs of all potential partitioning sizes since some implausible CU sizes can be ruled out beforehand. An effective termination and bypass method is desired to lower the computing needs and accelerate the intra-coding process. The RD-cost of CU in ISP coding mode and the No-ISP (without using ISP) coding mode are computed in [18], respectively. This research examines the CU rate-distortion cost with various texture details in ISP coding mode and finds that, despite additional calculation cost, the coding efficiency of simple texture CU is not much

Figure. 3 Frame of KristenAndSara sequence

increased in ISP coding mode. This method proposed a quick decision algorithm for ISP coding mode based on CU texture complexity. Its goal is to calculate CU texture complexity and determine if CUs must utilize ISP coding mode beforehand to clearly reduce ISP's computation complexity while maintaining almost no degradation in coding efficiency. CUs will be divided into two groups based on the value of TC (texture complexity): complex texture and simple texture. Fig. 3 shows the frame of KristenAndSara sequence. Blocks A and B are found to have rich textures, whereas blocks C and D have smooth textures.

The current CU is classified as plain texture when TC is smaller than the specified threshold (Th). The CU is categorized as a complex texture when TC exceeds Th. The threshold of Th is used to assess the textural complexity of CUs. As a result, Th is set to 20 in this method. When a CU's TC value is less than 20, it will be determined to forego the ISP coding mode during coding. In this method, the threshold is fixed. As for our method, we would propose the adaptive thresholds. The sequences used in this study are Campfire, Tango2 (Class A), BasketballDrive, Kimono, Cactus (Class B), BasketballDrill, RaceHorses, PartyScene (Class C), RaceHorses, BQSquare, BlowingBubbles (Class D), Johnny, KristenAndSara, FourPeople (Class E).

4.2 Threshold value of texture detail

The energy distribution of a CU in the frequency domain can be used to describe its spatial homogeneity. Thus, to assess the textural complexity of an LCU (largest coding unit), we use the AC (alternating current) coefficients of energy. Most of the frequency domain energy in an image is typically found in the low-frequency components in a homogenous area, while the DCT energy is more evenly spread throughout different AC coefficients in a region with rich detail. The energy conservation principle states that the AC coefficients energy

($Energy_{AC}$) of a numbers of row x column (r × c) CU can be stated numerically as:

$$Energy_{AC} = \sum_{x=0}^{r-1}\sum_{y=0}^{c-1} f^2(x,y) - \frac{1}{rc}\left(\sum_{x=0}^{r-1}\sum_{y=0}^{c-1} f(x,y)\right)^2 \quad (1)$$

Where the intensity of the pixels in the sample at point (x, y) is denoted as f(x, y). Remarkably, the AC energy of a CU may be calculated in advance to its highest value, or $Energy_{max}$. $Energy_{max}$ is derived from the CU that has a checkerboard pattern with neighboring pixel samples whose intensities alternately match the permitted minimum (f_{min}) and maximum (f_{max}) values. Consequently, the formula for figuring out a r × c (row x column) CU's $Energy_{max}$ is

$$Energy_{max} = \frac{rc}{2}\left[(f_{max}^2 + f_{min}^2) - \frac{1}{2}(f_{max} + f_{min})^2\right] \quad (2)$$

As a result, we establish D_{CU} (CU Detail) as the standard by which a CU's textural complexity is evaluated.

$$D_{CU} = \frac{\ln(Energy_{AC})}{\ln(Energy_{max})} \quad (3)$$

One way to quantify an LCU's textural complexity is to use its AC energy. Lower CU sizes at higher coding depth levels are better suited for areas with high detail, which is generally indicated by a big D_{CU} number. Conversely, uniform areas typically provide smaller D_{CU} values, thus it is better to use lower coding depth levels.

Rich spatial detail is provided in the LCU and the evaluation of big CU sizes might be skipped when the value of D_{CU} surpasses an upper threshold Th_U. Higher coding depth levels can be pre-eliminated when D_{CU} is below a lower threshold Th_L, which typically results in low texture detail in the LCU. The remaining D_{CU} values are then sorted in ascending order [D_{min},..., D_{max}]. D_{med} is the median value of D_{CU}, and the set of [D_{med},..., D_{max}] is used to select the upper threshold Th_U. Th_C is regarded as a candidate value for each D_{CU} value in the set of [D_{med},..., D_{max}]. The upper threshold Th_U is chosen based on a criterion known as the error ratio (E_R). The following formula is used to determine the value of E_R for each Th_C for the LCUs whose D_{CU} values exceed Th_C :

$$E_R = M_D / (M_D + N_D) \quad (4)$$

M_D denotes the number of blocks encoded using coding depth 0 in all the LCUs whose D_{CU} values are greater than Th_C, and N_D is the number of blocks encoded using a coding depth bigger than 0. The current Th_C value is selected as the upper threshold Th_U, and this process ends when E_R approaches 0. This Th_U value is used in each of the subsequent fast encoding frames. Similar to Th_U, the lower threshold Th_L is established, however Th_L is selected from [D_{min},..., D_{med}). If the present LCU is in an area with a high level of spatial detail then the D_{CU} value is larger than the Th_U. As a result, depth level 0 is skipped and the coding depth level range to be analyzed is set to [I, III] as seen in Fig. 2. With a depth range of [0, II], the LCUs whose D_{CU} value is smaller than Th_L.

Ting Fu et al. [13] discover that the ultimate optimal partition for a parent CU tends to be horizontal splitting as well as the parent CU and both of its offspring CUs are split horizontally. Furthermore, the parent CU is highly unlikely to select vertical split if it and its two offspring CUs select the horizontal intra mode. For these two cases, the recursive vertical splitting can therefore be skipped early. Besides that, for CUs with simple texture video encoding at the partitioning stage or decision-making of CU split type will tend to follow a repeating pattern, namely the tendency to follow the same pattern as before (previous pattern). The CU split type will be decided based on the results of the calculation of the minimum RD cost, so that with the proposed method, when at the decision-making stage the CU Split type to be used starts with the CU Split type pattern that has been taken in the previous pattern. The sequences of video with varying texture and resolutions complexities are evaluated using VTM14.2 in order to examine this correlation. Fourteen frames are examined for each sequence in the tests, those employ the JVET common test conditions (CTC). Table 2 provides test conditions summary, which serve as the default circumstances for the remainder of this work unless otherwise noted. Fig. 4 provides a summary of the test outcomes by modifying UnitTools.cpp file of VTM, the percentage of CU split type pattern where previous pattern greater than different pattern, which indicates that Fast CUs splitting type searching are suitable.

Table 2. Common Test Condition (CTC)

Quantization Parameter (QP)	22, 27, 32, 37
Configuration File	Encoder_intra_vtm.cfg
Number of encoded frames	30

Received: May 10, 2024. Revised: May 1, 2024.

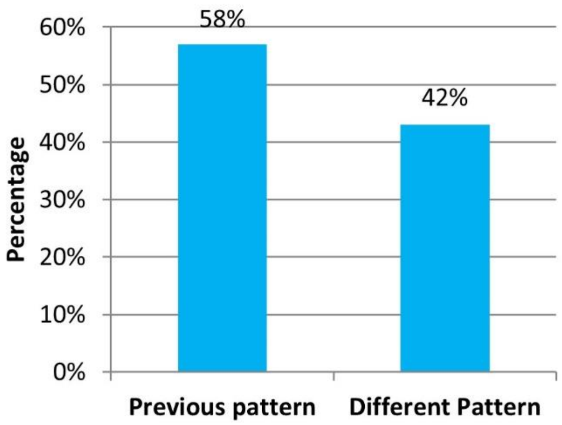

Figure. 4 CU splitting patterns distribution

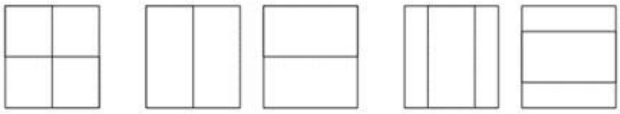

Figure. 5 From left to right: CU_QUAD_SPLIT, CU_HORZ_SPLIT, CU_VERT_SPLIT, CU_TRIH_SPLIT and CU_ TRIV_SPLIT

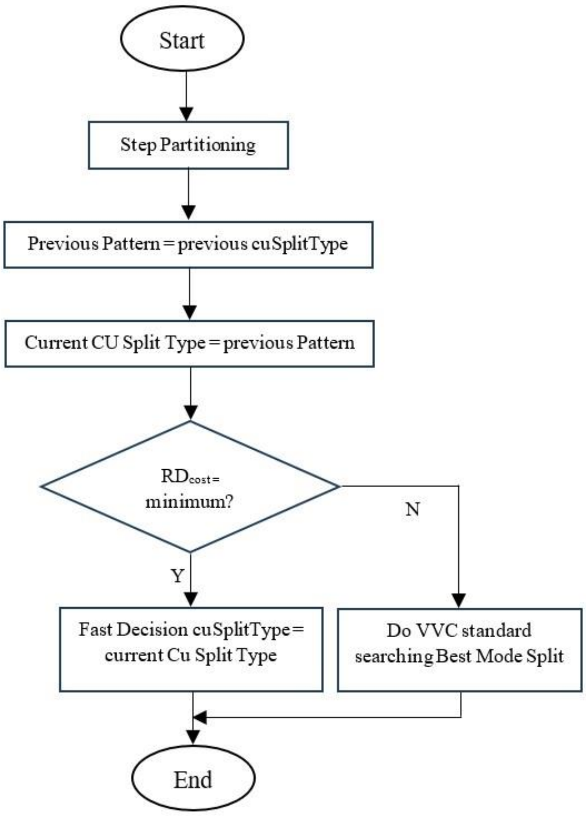

Figure. 6 Proposed method algorithm flow chart of the overall Results of the fast depth decision in VTM

In VTM 14.2, Split Type is divided as CU_QUAD_SPLIT, CU_HORZ_SPLIT, CU_ VERT_SPLIT, CU_TRIH_SPLIT and CU_TRIV_ SPLIT that will be given the values as 1, 2, 3, 4 and 5 respectively as in Fig. 5.

The localized texture consistency in the image is reflected in the aforementioned analysis. When two sub-CUs select the vertical split type, the parent CU may display vertical texture features, and vice versa, because sub-CUs share some of the material of their parent CU. The ideal intra mode for the parent CU and its sub-CU is another characteristic that can represent texture distribution. As a result, we looked at the relationship between the parent CU's split type and the intra mode of the sub-CU. In Algorithm 1 explains the fast depth decision in VVC partitioning method.

Algorithm 1 Pseudocode of fast depth decision in VVC partitioning method

---

Input: Previous Pattern : Previous CU Split Type Pattern; CurSplitType : Decision of Current Split Type; CurSplitType_C : Candidate of Current Split Type; SplitType : CU_QUAD_SPLIT, CU_HORZ_SPLIT, CU_ VERT_SPLIT, CU_TRIH_SPLIT. CU_TRIV_ SPLIT

Output: CUSplitType;

1. CurSplitType_C ← Previous Pattern;
2. Test RD_{cost} for all SplitType
3. **if** RD_{cost} of CurSplitType_C == $BestRD_{cost}$
4. CurSplitType ← CurSplitType_C;
5. **else**
6. do VTM split search procedure
7. return CUSplitType;

Based on the above analysis, the overall algorithm is summarized in Fig. 6. Firstly, a Candidate of Current Split Type is composed of the previous split mode CU to be used in the next step. The pattern of CU split mode is used as the input features in the next process. If the RD_{cost} of CurSplitType_C = min (RD_{cost} of all SplitType) then the CurSplitType is utilized to CUSplitType as Fast Depth Decision splitting.

RDO (Rate distortion optimization) is used in VVC to determine the CTU partition type. The particular process is as the encoder iterates through every conceivable division type, determines each division type's rate-distortion (RD) cost, and determines which division type is the optimal division technique for the CU based on which division type has the lowest RD_{cost}. The following is the formula used to determine RD_{cost}.

Received: May 10, 2024. Revised: May 1, 2024.

$$RD_{cost} = D + \lambda \, . \, R \qquad (5)$$

$$D = SSL + W_C \, . \, SSC \qquad (6)$$

Where R represents the encoded bits number, λ represents Lagrange multipliers, D denotes distortion measure, WC represents chroma distortion weights, SSC and SSL indicates the square sum of chrominance and luminance respectively.

In video frames, the approach seeks to either skip the CU partitioning of complex texture regions or terminate the CU partitioning of homogeneous parts beforehand. Instead of evaluating every depth to find the ideal CU size, the split/non-split decision result is based on a homogeneous classification algorithm. Finding the ideal coding parameters is essential to enhancing the encoder's performance. Traditionally, the optimal coding parameter search is carried out in the rate-distortion (RD) sense, where the size of bits needed to encode a block of images and any distortion produced by utilizing these bits can be balanced. In the intra-prediction rate-distortion optimization process, RDcost is a great indicator to gauge the accuracy of predictions.

The CU split type will be decided based on the results of the calculation of the minimum RDcost, so that with the proposed method, when at the decision-making stage the CU Split type to be used starts with the CU Split type pattern that has been taken in the previous pattern. However, if after being tested on RDcost and not a minimum value, then it will follow the rules in VVC video coding or in this study using VTM. However, if we have obtained the minimum RDcost value, a fast decision CU Split type will be carried out. This step is expected to reduce the time needed during the partitioning stage or CU split type decision making which is the stage that requires the largest time of the total video encoding process.

In this paper using PC running Linux Ubuntu 20.04.4 LTS and an Intel® CoreTM i7 - 67000T Core @ 2.80 GHz processor was used to test VTM 14.2. Thirty frames per sequence are assessed in the tests, which are conducted under the JVET common test conditions (CTC).

4.3 Fast intra-coding algorithm

A fast intra-coding strategy is created by integrating the two previously stated algorithms. Algorithm 2 explains the two-step fast intra-coding strategy in the VVC partitioning method:

Algorithm 2 The hybrid fast intra-coding strategy in VVC partitioning

---

Input: frame to be evaluated, ThU : Threshold upper; THL : Threshold Lower; DCU : CU Detail level; ER : error ratio Previous Pattern : Previous CU Split Type Pattern; CurSplitType : Decision of Current Split Type; CurSplitType_C : Candidate of Current Split Type; SplitType : CU_QUAD_SPLIT, CU_HORZ_SPLIT, CU_ VERT_SPLIT, CU_TRIH_SPLIT. CU_TRIV_ SPLIT

Output: [d0,d1], /* coding depth range
CUSplitType;

---

1. Calculate DCU for each CU using Eq. (3)
2. Sort DCU in ascending order [Dmin,..., Dmax]
3. Divide DCU list into two part, [Dmin… Dmid] and [Dmid… Dmax]
4. Select Th_L from [D_{min}… D_{mid}] where E_R = 0 using Eq. (4)
5. Select Th_U from [D_{mid}… D_{max}] where E_R = 0 using Eq. (4)
6. **if** $D_{CU} < Th_L$
   [d_0,d_1] ← [0,I];
7. **else if** $D_{CU} > Th_U$
   [d_0,d_1] ← [I,III];
8. CurSplitType_C ← Previous Pattern;
9. Test RD_{cost} for all SplitType
10. **if** RD_{cost} of CurSplitType_C == $BestRD_{cost}$
    CurSplitType □ CurSplitType_C;
11. **else**
12. **do** VTM split search procedure
13. return CUSplitType

The D_{CU} values are arranged [D_{min},..., D_{max}] in increasing order. The upper threshold Th_U is chosen from the set of [D_{med},..., D_{max}], where D_{med} is the median value of D_{CU}. With reference to each D_{CU} value in the set of [D_{med},..., D_{max}], Th_C is considered a potential value. An error ratio (E_R) criterion is used to determine the upper threshold Th_U. This process terminates when E_R approaches zero, and the current Th_C value is determined as the upper threshold Th_U using Eq. (4). Every one of the ensuing fast encoding frames uses this Th_U value. The lower threshold Th_L is set, same as Th_U, but Th_L is chosen from [D_{min},..., D_{med}]. When a region with a high degree of spatial information surrounds the current LCU, the Th_U is smaller than the D_{CU} value. Hence, as can be seen in Fig. 2, depth level 0 is bypassed and the level coding depth range is set to [I, III]. These LCUs have a depth range of [0, II] if their D_{CU} value is less than Th_L.The following action is based on algorithm 1.

The VVC/H.266 reference program VTM 14.2 implements the suggested fast-depth decision algorithm and tests it under common test conditions (CTC) to determine how effective it is. The test sequences and the All-Intra primary configuration are used in the simulation. The encoding time savings (T_{saving}) are evaluated in comparison to the original VTM software ($Time_{VTM}$) and proposed method ($Time_{proposed}$) ,and the coding performance is assessed using BDBR. The average encoding time savings, which measures complexity reduction, is provided by:

$$T_{saving} = \frac{Time_{VTM} - Time_{proposed}}{Time_{VTM}} \times 100\% \quad (7)$$

5. Result and discussion

This subsection presents a comparison of the original VTM's performance with our suggested significant mode selection. We used the Bjontegaard Delta Rate (BDBR) [19] and Tsaving to evaluate the results and the coding time reduction rate compared to the anchor test results.

Table 3. The Result of Proposed Method

Class	Sequence	Proposed Method	
		BDBR (%)	**T_{saving} (%)**
A (3840 x 2160)	Campfire	1.72	39.51
	Tango2	1.54	38.34
B (1920 x 1080)	BasketballDrive	1.64	34.61
	Kimono	1.03	36.78
	Cactus	1.32	37.3
C (832 x 480)	BasketballDrill	0.97	34.56
	RaceHorses	1.25	35.82
	PartyScene	0.85	36.85
D (416 x 240)	RaceHorses	1.75	32.27
	BQSquare	1.27	37.64
	BlowingBubbles	1.62	33.58
E (1280 x 720)	Johnny	0.94	37.98
	KristenAndSara	1.51	39.87
	FourPeople	1.74	38.73
Average		1.37	36.70

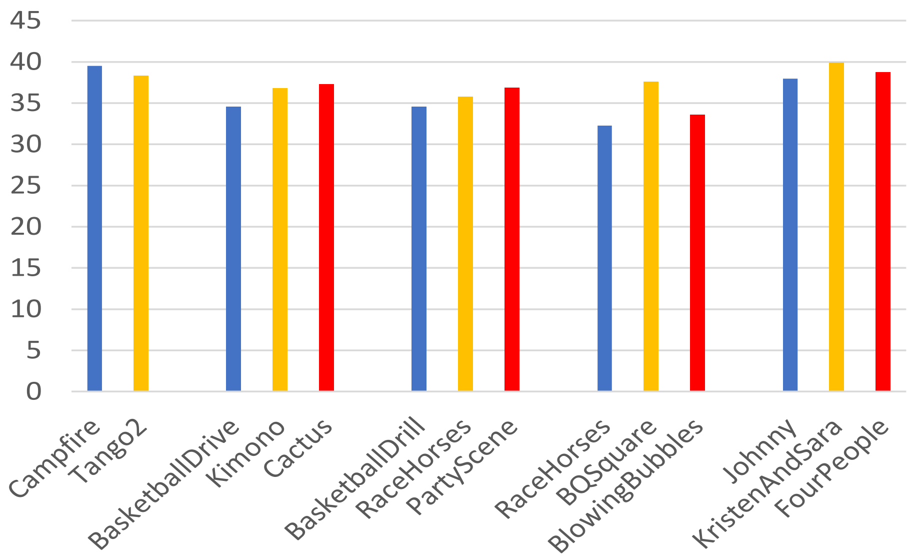

Figure. 7 The test results with the proposed method

Received: May 10, 2024. Revised: May 1, 2024.

It could benefit from a significant decrease in Tsaving% compared to a decrease in BDBR that was small enough to be applied to online and real time video communication, because it needs as little video coding time as possible to minimize delay time. In the test results with the proposed method, the average BDBR% was 1.37% and the decrease in Tsaving% average was 36.70% as shown in Table 3 and Fig. 7. The sequences for statistics are Campfire, Tango2 (Class A), BasketballDrive, Kimono, Cactus (Class B), BasketballDrill, RaceHorses, PartyScene (Class C), RaceHorses, BQSquare, BlowingBubbles (Class D), Johnny, KristenAndSara, FourPeople (Class E).

Table 3 shows that the approach can, on average, reduce the encoding time at VTM 14.2 by 36.70% while causing only a negligible loss in quality for the sequences test. In conclusion, all QPs have consistent processes, and there is a notable average time savings compared to VTM 14.2. The suggested approach allows the system to save a respectable amount of time during the entire VTM 14.2 encoding process because the time savings and BDBR process are closely correlated with the time savings across the entire encoding process. The algorithm suggested in this article performed the best among them for various categories, with an average reduction in coding times of 39.51% and 38.34% in Class A sequences. This demonstrates that when aimed at higher-resolution sequences, the proposed approach can perform better. The suggested approach reduced the encoding time for a single sequence by a minimum of 32.27% for the RaceHorse sequence and better by 38.73% for the FourPeople sequence. The suggested technique works well for videos with a lot of smooth areas and slow motion, like "FourPeople," and it gives less Tsaving for the video sequence "RaceHorse," which has fast motion. This proves that the suggested technique can work well for all sequences and is universal and efficient in lowering coding complexity. The approach suggested in this study can significantly reduce the encoding time under the assumption of preserving the encoding quality since it can reduce the time coding and the partition prediction result is essentially equal to that of the original algorithm.

In order for an objective comparison, we match the proposed algorithm to the several state-of-the-art works, such as the Zhiyong Jing et al. (ZJ) algorithm in [6], Na Tang et al. (NT) algorithm in [7], Xin-Yi Cui et al. (XY) algorithm in [8] and Aolin Feng (AF) algorithm in [9]. Every algorithm is tested using the VTM video coding standard and the sequences (Class A, B, C, D and E). The metric is used to assess the algorithm's performance: average time savings (Tsaving %) average. The ratio of reduction in computation complexity is shown by the value of Tsaving as shown in Fig. 8.

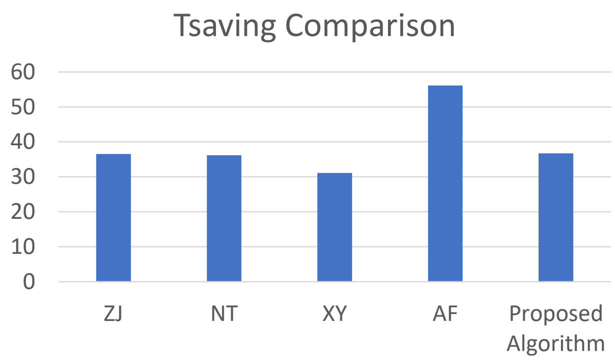

Figure. 8 The comparison of the proposed algorithm experimental average results with the previous method

The proposed algorithm shows slight advantages over the existing algorithms. First, when compared to [6], our algorithm performed better in terms of reducing the coding complexity average, the encoding time was reduced by 0.14% compared to the algorithm in [6], and the BDBR increased by 0.31%. Next, when compared to [7], our method reduced the encoding time by 0.71%, and the BDBR increased by 0.66%. Finally, if compared to [8], our method reduced the encoding time by 5.57%, and the BDBR increased by 0.05%. Finally, when compared to [9], the proposed method increased the encoding time by 19.41% on average, and the BDBR decreased by 0.28%. Although learning approaches can save time, they mostly rely on training datasets and user-defined characteristics. The majority of training datasets for low-complexity video coding are proprietary, and only a tiny percentage are available for academic usage. Online learning approaches can come with certain extra expenses related to the training process.

Based on our knowledge, this is the first work of literature to propose the method of fast depth decision algorithm on VVC partitioning for Real-Time Video Communication. Due to the consistency of each distinct sequence, the complexity reduction will vary. This approach will work better with sequences that are consistent and less well with sequences that change quickly. This approach is anticipated to apply these findings uniformly to all VTM-compliant video encodings.

6. Conclusions

Fast Depth Decision Algorithm with spatial homogeneity and threshold value modified method on Versatile Video Coding (VVC) Partitioning has been reported and experimentally validated. Real-time video communication is validated using the VTM 14.2 (VVC test mode) standard. By using the preceding CU split type's pattern, the optimization can lower the coding time (t%) average by 36.70%, which is rather considerable in comparison to the declining BDBR average level, which is small that it can be negligible (1.37%).

By cutting down on the time needed to finish the code necessary for real-time video transmission, the test findings demonstrate that the results are rather good. It is envisaged that the same outcomes would hold for other sequences as well, allowing for more widespread application.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The contributions of authors are as follows: Conceptualization, Alexander V. Bukit; methodology, Alexander V. Bukit, software, Alexander V. Bukit; validation, Alexander V. Bukit and Wirawan; formal analysis, Alexander V. Bukit, Suwadi, and Wirawan; investigation, Alexander V. Bukit and Titiek Suryani; resources, Alexander V. Bukit; data curation, Alexander V. Bukit and Endroyono; writing—original draft preparation, Alexander V. Bukit; writing—review and editing, Alexander V. Bukit, Titiek Suryani and Endroyono; supervision, Alexander V. Bukit, Suwadi, and Wirawan; project administration, Alexander V. Bukit.

Acknowledgments

Authors would like to acknowledge supports provided by Electrical Engineering Department, Faculty of Intelligent Electrical and Informatics Technology, ITS on research facilities and financial support during completion of this research.

References

[1] ITU-T and ISO/IEC JTC 1, Versatile video coding. Rec. ITU-T H.266 and ISO/IEC 23090-3 (VVC), 2020.

[2] A. Mercat, A. Mäkinen, J. Sainio, A. Lemmetti, M. Viitanen, and J. Vanne, "Comparative Rate-Distortion-Complexity Analysis of VVC and HEVC Video Codecs", *IEEE Access*, Vol. 9, pp. 67813-67828, 2021.

[3] G. Wu, Y. Huang, C. Zhu, L. Song and W. Zhang, "SVM Based Fast CU Partitioning Algorithm for VVC Intra Coding", *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5, 2021.

[4] M. Aklouf, M. Leny, F. Dufaux and M. Kieffer, "Low Complexity Versatile Video Coding (VVC) for Low Bitrate Applications", *8th European Workshop on Visual Information Processing (EUVIP)*, pp. 22-27, 2019.

[5] M. Abdoli, F. Henry, P. Brault, P. Duhamel and F. Dufaux, "Short-Distance Intra Prediction of Screen Content in Versatile Video Coding (VVC)", *IEEE Signal Processing Letters*, Vol. 25, No. 11, pp. 1690-1694, 2018.

[6] Z. Jing, W. Zhu and Q. Zhang, "A Fast VVC Intra Prediction Based on Gradient Analysis and Multi-Feature Fusion CNN", *Electronics 2023*, pp. 12, 2023.

[7] N. Tang, J. Cao, F. Liang, J. Wang, H. Liu, X. Wang and X. Du, "Fast CTU Partition Decision Algorithm for VVC Intra and Inter Coding", In: *Proc. of IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp 361-364, 2019.

[8] X. -Y. Cui and F. Liang, "Perceptual Based Fast CU Partition Algorithm for VVC Intra Coding", In: *Proc. of TENCON 2023 - 2023 IEEE Region 10 Conference (TENCON)*, Chiang Mai, Thailand, 2023, pp. 807-810, 2023

[9] A. Feng, K. Liu, D. Liu, L. Li and F. Wu, "Partition Map Prediction for Fast Block Partitioning in VVC Intra-Frame Coding", *IEEE Transactions on Image Processing*, Vol. 32, pp. 2237-2251, 2023

[10] L. Shen, Z. Liu, X. Zhang, W. Zhao and Z. Zhang, "An Effective CU Size Decision Method for HEVC Encoders", *IEEE Transactions on Multimedia*, Vol. 15, No. 2, pp. 465-470, 2013.

[11] A. Tissier, W. Hamidouche, J. Vanne, F. Galpin and D. Menard, "CNN Oriented Complexity Reduction of VVC Intra Encoder," In: *Proc. of IEEE International Conference on Image Processing (ICIP)*, pp. 3139-31432020, 2020.

[12] A. Wieckowski, B. Bross and D. Marpe, "Fast partitioning strategies for VVC and their implementation in an Open Optimized Encoder", *Picture Coding Symposium (PCS)*, pp. 1-5, 2021.

[13] "VTM 2.0 - VTM 20.2" [Online]. Available: https://github.com/fraunhoferhhi/vvenc

[14] T. Fu, H. Zhang, F. Mu and H. Chen, “Fast CU Partitioning Algorithm for H.266/VVC Intra-Frame Coding”, In: *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, pp. 55-60, 2019.
[15] E. Tun, S. Aramvith and T. Onoye, “Low Complexity Mode Selection for H.266/VVC Intra Coding”, *ICT Express*, Vol 8, Issue 1, pp. 83-90, 2022.
[16] L. Wang and W. Siu, “Novel Adaptive Algorithm for Intra Prediction With Compromised Modes Skipping and Signaling Processes in HEVC”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 23, No. 10, pp. 1686-1694, 2013.
[17] T. Zhang, M. Sun, D. Zhao and W. Gao, “Fast Intra-Mode and CU Size Decision for HEVC”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 27, No. 8, pp. 1714-1726, 2017.
[18] Q. Zhang, Y. Wang, L. Huang and B. Jiang, “Fast CU Partition and Intra Mode Decision Method for H.266/VVC”, *IEEE Access*, Vol. 8, pp. 117539-117550, 2020.
[19] G. Bjøntegaard, Calculation of average PSNR differences between R-D curves, document VCEG-M33, ITU-T VCEG, Austin, TX, USA, 2001.