IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 20, 2019, accepted November 22, 2019, date of publication November 27, 2019, date of current version December 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956196

Context-Based Ternary Tree Decision Method in Versatile Video Coding for Fast Intra Coding

SANG-HYO PARK, (Member, IEEE), AND JE-WON KANG, (Member, IEEE)
Department of Electronics and Electrical Engineering, Ewha Woman's University, Seoul 03760, South Korea

Corresponding author: Je-Won Kang (sagittak@gmail.com)

This work was supported in part by the National Research Foundation of Korea (NRF) through the Korea Government (MSIT) under Grant NRF-2019R1C1C1010249, and in part by the Institute of Information and communications Technology Planning and Evaluation (IITP) through the Korea Government (MSIT) (Development of Compression and Transmission Technologies for Ultra High Quality Immersive Videos Supporting 6DoF) under Grant 2018-0-00765.

ABSTRACT This paper presents a fast encoding method for versatile video coding (VVC) using an early determination scheme that skips redundant multi-type tree (MTT) pruning. MTT consists of binary and ternary trees (TTs) with traditional quadtrees and has recently attracted considerable VVC research interest due to coding efficiency beyond HEVC. However, the additional trees, particularly TTs, significantly increased VVC complexity, which has rarely been studied previously. Therefore, we identified TT characteristics in the VVC encoding context, and hence propose a simple early decision technique that can effectively reduce TT complexity. Experiments over various test sequences show that the proposed method substantially decreased the total encoding time for a VVC test model 4.0 (VTM 4.0) with only slight coding loss under the all-intra configuration. In particular, the proposed method reduced TT encoding time by up to 66% compared with VTM 4.0.

INDEX TERMS Versatile video coding, high efficiency video coding, multi-type tree, encoder complexity, intra-prediction.

I. INTRODUCTION

Versatile video coding (VVC) is a new standardization project established by the Joint Video Experts Team (JVET) of ISO/IEC JTC 1/SC 29/WG 11 and ITU-T SG16 WP3 to improve compression efficiency compared with HEVC. VVC was developed to meet various requirements associated with video coding application [1], including emerging video contents such as an ultra-high definition video and 360-degree video. The recent VVC test model (i.e., VTM 4.0) has already achieved 31% average bitrate saving for the same visual quality compared with the latest HEVC test model (HM) under the random-access configuration [2]. Therefore, VVC is expected to be widely used for broadcasting systems, streaming services, and storage systems in the near future.

Multi-type tree (MTT) structure is a pivot that unifies traditional separated concepts of coding, prediction, and transform units; and can support more block shapes than HEVC [3]. MTT also employs binary (BT) and ternary tree (TT), compared with just quadtree (QT) in HEVC, which are recursively applied to QT. Let N denote the width or height of a block. Then, given block of size 4N × 4N, BT allows the block to be split to the half, i.e., 4N × 2N or 2N × 4N, as in the HEVC prediction unit (PU); and TT can split a block into three (4N × N, 4N × 2N, and 4N × N (horizontal direction); or N × 4N, 2N × 4N, and N × 4N (vertical direction)), providing more flexible area partitioning.

However, VVC encoder complexity can be increased further due to complicated TT partitioning and further splits in recursion as Fig. 1 shows that the TT time portion is about 33% among the entire encoding time. Fig. 1 presents encoding time for BT, TT, and others including QT in quantization parameter (QP) 37 and the all-intra configuration that encodes all frames as intra-frames [18]. Herein, the encoding time for BT and TT includes the corresponding prediction, transform and coding times for the CU encoding test of each partition. BT requires most of total encoding time (more than 50%). However, reducing BT complexity could significantly degrade compression performance; not only due to reduced BT efficiency but also some BT conditions, e.g., splitting rules on picture boundary as described in [3]. TT also consumes about one-third of total encoding time. In addition,

The associate editor coordinating the review of this manuscript and approving it for publication was Byeungwoo Jeon.

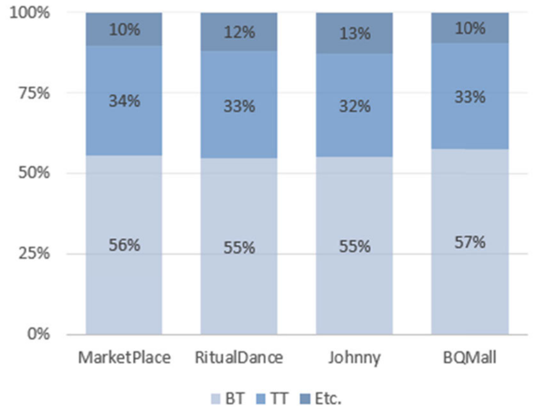


FIGURE 1. **Time distributions for MTT partitioning under all-intra configurations with quantization parameter 37. 100% represents the total encoding time of VTM 4.0.**

TABLE 1. **Running time of VTM 4.0.**

Sequence	Encoding time (s)	TT time (s)	TT time per frame (s)
MarketPlace	6465.15	2190.28	4.56
RitualDance	4708.49	1568.96	3.27
Johnny	1876.14	604.80	1.26
BQMall	2130.30	701.49	1.46

as shown in Table 1, the actual running time for TT per frame takes more than one second, which is far beyond real-time encoding. Few previous studies have considered reducing TT complexity, although it has some redundancy with other block partitions. Thus, TT redundancy should be addressed for low complexity video applications, such as real-time broadcasting, low-delay streaming, and battery constrained encoding devices.

Early tree decision approaches for coding tree units (CTUs) or coding units (CUs) have been widely studied previously for fast video encoding. One pioneer work used early termination based on QT pruning in HEVC [4]. Whether to early terminate or not the sub-QT partitioning was judged by checking the best CU prediction mode. Several studies have attempted to reduce QT plus BT (QTBT) encoding complexity by extending pruning methods [5], [6]. However, TT-specific fast encoding methods have not yet been thoroughly investigated. Although JVET experts have considered mechanisms to relieve VTM 4.0 encoding complexity, they have focused on TT use under several predefined conditions [7].

This paper proposes a context-based TT decision (C-TTD) method to significantly reduce TT computational complexity in VVC intra-coding. The proposed method aims to find an early decision criterion based on a Bayesian approach that exploits rate-distortion (RD) costs related to TT splits to determine if the current TT split can be skipped. The contribution of the paper includes:

1) A new feature is identified, which has strong correlation of BT costs with the choice of TT. To the best of our knowledge, this feature is the first attempt to reduce the TT complexity in the VVC literature.
2) From a practical point of view, the proposed method has a lightweight termination scheme that effectively reduces TT complexity. In contrast, the state-of-the-art [16] requires complexity-heavy texture extraction process which may hinder implementing practical low-complexity encoders.
3) The proposed TT pruning scheme can reduce the encoding complexity of both TT and BT. Since a node in TT can have child nodes of either one of BT and TT or both, it is critical for an encoder to find the redundancy of tree split tests. Thus, by skipping redundant TT split tests, the proposed method can improve the entire VVC encoding complexity.

Experimental results show that the proposed method can reduce the total VTM 4.0 encoding time by 67% on average with only very slight coding loss under an all-intra configuration.

II. RELATED WORK

A. OVERVIEW OF MULTI-TYPE TREE IN VVC

In addition to splitting blocks using QT structures in HEVC, VVC can use BT or TT structures for each leaf node of the QT [8]. If the parent node is QT, the current node can be split with QT, BT, or TT; whereas if the parent node is BT or TT, the current and child nodes cannot use QT, hence only BT or TT is used for subsequent partitioning. Fig. 2 shows several examples. There is also a non-split block, i.e., a leaf node after testing all possible tree structures. The block is encoded using intra-prediction and residual coding. Importantly, BT and TT have directions in the splits, i.e., vertical and horizontal directions, as shown in Fig. 2.

MTT performs RD cost competition for all possible partitioning trees in a CU to determine the best structure. The CU is split if the best RD cost is achieved by QT, BT, or TT; and otherwise remains a non-split block. Other MTT aspects should also be considered, such as split restrictions by depth and picture boundary. For depth constraints by depth, if BT or TT depth reaches the maximum depth in a tile, then the tree is not further split. CUs can also be forced to split at a picture boundary. More detailed CTU partitioning processes are described in [8], with associated syntax and semantic elements described in [3].

B. EXISTING TREE DECISION METHODS FOR FAST VIDEO ENCODING

Various methods have been proposed to reduce complexity to make a recursive QT structure for a square block in HEVC. A pruning-based early CU decision was proposed to judge whether to split a block according to the best prediction mode [4]. Context-based early CU pruning approaches have been considered Subsequently, exploiting correlations among CUs [9] or RD costs of CUs within QT [10].

Learning-based approaches have been studied recently for fast HEVC intra coding by pruning CUs in a QT structure.

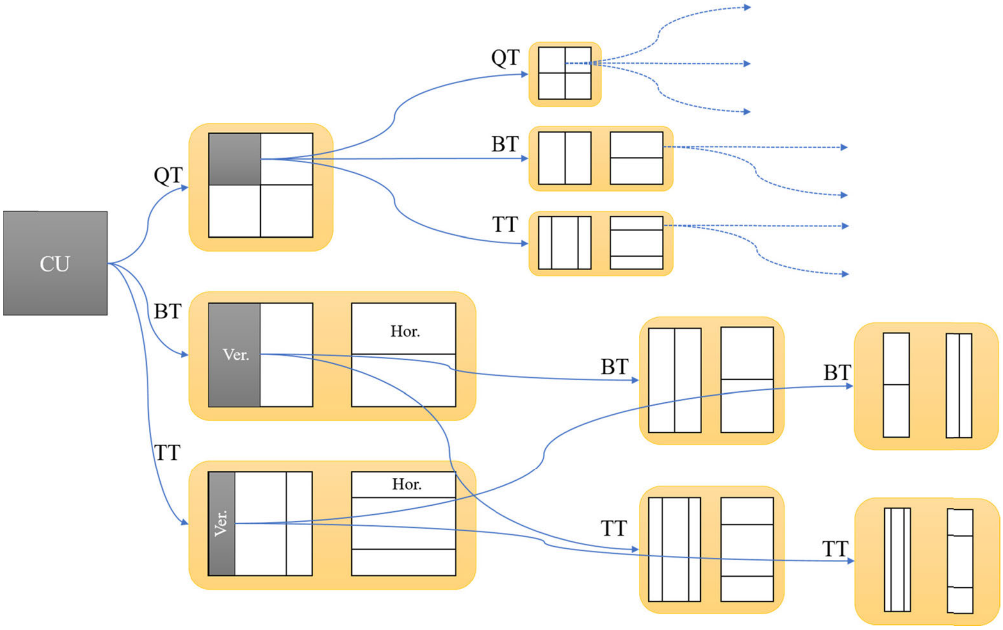


FIGURE 2. An example of possible split cases of MTT for a CU in VVC. *Ver* denotes vertical direction split and *Hor* denotes horizontal split. Each gray node can split further among QT, BT and TT depending on the current tree structure and some restriction rules.

A fast encoding method based on decision tree was proposed, targeting partitioning and prediction mode complexity [21]. The random forest technique within a CU, using an ensemble-based randomized tree model, was proposed to tackle encoding complexity for HVEC intra-prediction modes [19]. Support vector machines or neural networks have been employed to classify redundant CUs for fast CU depth decision [11], [12], [20]. Convolutional neural networks (CNNs) are widely used for classification tasks and have also been employed to predict the best CU partition [14]. Since learning-based approaches are fitted to specific coding structures or features, they would be difficult to extend to BT and TT tree structures.

Recent studies have addresses BT and TT complexity, along with other tree types, utilizing similar approaches to previous QT related studies [5], [6], [13]. A probabilistic approach exploiting RD context was proposed for QTBT partitions [6]. Learning-based approaches have been developed to facilitate BT classification, using various classification methods, such as a decision tree [13] and CNN [5]. However, TT complexity has been only rarely tackled, even though TT demands high VVC encoding complexity. JVET conducted several experiments to consider TT complexity by restricting split conditions [7], [15]. A fast encoding method was proposed that targets QT, BT, and TT [16], exploiting various features, including gradient and texture variance. Early determination of MTT partitioning was established in [16] using the decision tree classifier for each split mode. In [22], another machine learning based approach is investigated, which utilizes random forest technique and extracts various features including gradient-based texture information to reduce the MTT complexity. In random-access configuration, this approach showed 0.61% coding loss with 30% encoding time saving. In the random-access configuration, this approach showed 0.61% coding loss with 30% total encoding time saving. However, these learning-based approach still required complex and intensive preprocessing to extract texture information for every CU, greatly increasing total encoder complexity.

Decision tree-based classifiers can be easily overfit to the training set, and thus, studies should carefully establish training and testing sets. Although the current methods can contribute to reducing MTT complexity, considerable room remains to reduce TT complexity.

III. PROPOSED METHOD

Encoding complexity for TT is high, which is a bottleneck to develop low-complexity VVC encoders. Although TT might not be the best choice during RD optimization, current VTM 4.0 design needs to check almost all split and non-split cases, which requires considerable computational complexity by redundancy. For the problem considered in this paper, redundancy can be quantified by the number of TT RD evaluations that might be unnecessary from initialization to final decision. Thus, we try to find a critical factor correlated to the redundancy and exploit it to make an early decision. Specifically, motivated by the Bayesian probability approach, we propose a simple C-TTD scheme for a VVC encoder with MTT structure. The detailed C-TTD description is presented in this section on top of VTM 4.0.

A. MOTIVATION AND ANALYSIS

Table 2 shows probability $p(\tau)$, measuring the number of events where the split mode τ is redundant in VTM 4.0

TABLE 2. **Prior probability when encoding using VTM 4.0.**

Sequence	$p(\tau = \text{TT\_H})$ (%)		$p(\tau = \text{TT\_V})$ (%)	
	QP 22	QP 37	QP 22	QP 37
Cactus	0.880	0.907	0.921	0.900
BasketballDrive	0.866	0.886	0.970	0.953
BQTerrace	0.899	0.886	0.915	0.889
BasketballDrill	0.838	0.890	0.856	0.907
BQMall	0.863	0.879	0.838	0.858
RaceHorsesC	0.873	0.892	0.893	0.860
FourPeople	0.892	0.891	0.886	0.881
KristenAndSara	0.919	0.892	0.888	0.879
Average	0.879	0.890	0.896	0.891

Notes: hor = horizontal direction. ver = vertical direction for TT split; $p(\tau = \text{TT\_H})$ and $p(\tau = \text{TT\_V})$ = prior probability for horizontal and vertical direction splits, respectively.

TABLE 3. **Test material information for statistics.**

Class	Sequence	Cropped resolution
B	Cactus BasketballDrive BQTerrace	1920x1024
C	BasketballDrill BQMall RaceHorsesC	832x448
E	FourPeople KristenAndSara	1280x704

Notes: Only the first frame is used for all sequences and encoded with QP 22 and QP37 under the all intra configuration.

software. The term 'redundant' means that the RD cost of τ is larger than that of the best mode in a CU. Table 2 shows that $p(\tau)$ is high when evaluating TT RD costs. For example, the average prior probability a horizontal TT split mode is redundant, $p(\tau = \text{TT\_H}) = 0.879$ and 0.89 in QP 22 and 37, respectively. Similarly, average vertical TT split mode redundancy $p(\tau = \text{TT\_V}) = 0.896$ and 0.891 in QP 22 and 37, respectively. Table 3 shows the detailed test conditions and material details. We used eight test videos and two QPs but also observed similar numbers in different scenarios.

Since the best mode is determined after comparing RD costs for up to six block partitioning tests, including no CU split and five different split modes, i.e., QT, horizontal BT (BT_H), vertical BT (BT_V), horizontal TT (TT_H), and vertical TT (TT_V), the level of redundancy can incur unnecessary computations. However, according to Bayes' theorem, the degree of belief after observing the evidence greatly increases with initial degree of belief in τ. Thus, if we can find an appropriate evidence ε and exploit the posterior $p(\tau | \varepsilon)$ after the observation, an early decision can be made to efficiently sort out TT redundancy.

We investigated BT split RD costs for evidence ε. We assumed that a block with a given texture direction should be encoded by the most suitable partition shape in view of RD cost. Thus, blocks with vertical texture direction should be encoded by BT_V or TT_V rather than horizontal partitioning modes. In this case, BT_V or TT_V RD costs tend to be smaller than BT_H or TT_H. Thus, cost comparisons for BT modes (i.e., BT directions) could be useful to predict the best TT direction. Furthermore, BT direction can be suitable evidence because certain TT direction pixels significantly overlap with the same BT direction. Fig. 3 shows the level of sub-CUs of TT_V overlap with corresponding sub-CUs of BTs. For example, 100% pixels in "A" of TT_V are included in "A" of BT_V, whereas only 50% of pixels are included in "A" of BT_H. Therefore, TT_V tends to provide similar encoding results to BT_V because it has more overlapped pixels than BT_H. Therefore, we compare BT_H and BT_V costs as an evidence and use them to enhance the belief for TT redundancy.

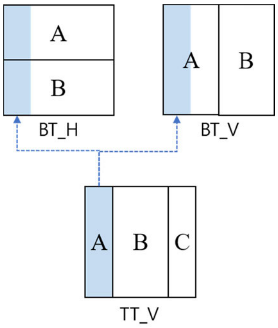


FIGURE 3. **Example mapping TT_V area to BT split cases in a CU encoding.**

Thus, we define a suitable evidence, ε, where RD cost of BT in a certain direction is better than the other cases, is defined as below:

$$\varepsilon(\delta) = \begin{cases} 1, & \text{if } J_{\text{BT}}(\delta) < J_{\text{BT}}(\sim \delta) \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where δ and $\sim \delta$ denote horizontal or vertical direction BT split, respectively; $\varepsilon(\delta)$ denotes evidence ε; and J_{BT} denotes the BT RD cost for a given δ. Let $p(\varepsilon)$ denote the probability that $\varepsilon = 1$. Table 4 shows the posterior $p(\tau|\varepsilon)$, gathered under the same test conditions of Table 1. τ can be either TT_H or TT_V, and δ can be either horizontal (hor) or vertical (ver) direction. Posterior increases with given ε. On average, posterior reaches more than 0.955 and 0.945 in horizontal and vertical directions, respectively. The probability is substantially high.

B. CONTEXT-BASED TT DECISION (C-TTD)

We propose a C-TTD scheme that skips TT early to decrease encoding complexity. The proposed method utilizes $\varepsilon(\delta)$ to select redundant TTs more accurately. Since TT implies two possible split cases, the proposed method has two phases (horizontal (TT_H), and vertical (TT_V) oriented TT split.

The phases determine whether or not to conduct the corresponding split, based on ε, e.g. ε(ver) when $\tau = \text{TT\_H}$. Since the proposed method needs the prior RD costs in (1), all BT split tests should be preceded. The proposed method does not

TABLE 4. Posterior probability when encoding using VTM 4.0.

Sequence	$p(\tau = \text{TT\_H} \mid \varepsilon(\text{hor}) = 0)$ (%)		$p(\tau = \text{TT\_V} \mid \varepsilon(\text{ver}) = 0)$ (%)	
	QP 22	QP 37	QP 22	QP 37
Cactus	0.948	0.954	0.968	0.953
BasketballDrive	0.956	0.949	0.991	0.980
BQTerrace	0.957	0.942	0.962	0.942
BasketballDrill	0.979	0.946	0.910	0.949
BQMall	0.933	0.934	0.913	0.921
RaceHorsesC	0.947	0.944	0.953	0.925
FourPeople	0.953	0.949	0.955	0.949
KristenAndSara	0.965	0.943	0.950	0.943
Average	0.955	0.945	0.950	0.945

Notes: hor means horizontal direction and ver means vertical direction for BT or TT split; $p(\tau = \text{TT\_H} \mid \varepsilon(\text{hor}) = 0)$ and $p(\tau = \text{TT\_V} \mid \varepsilon(\text{hor}) = 0)$ = posterior probability

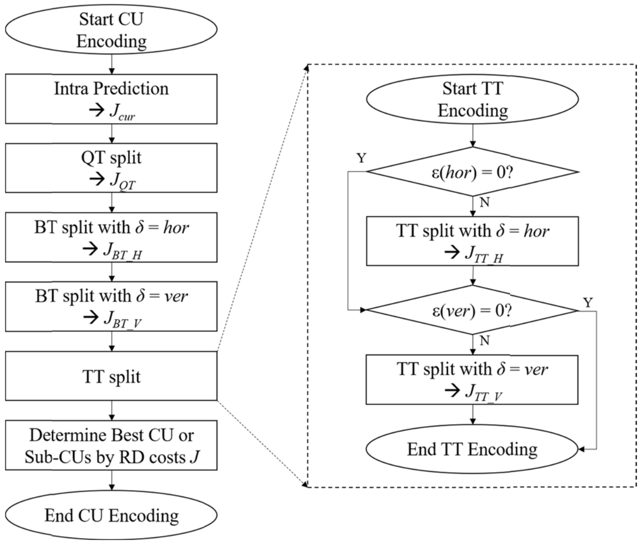


FIGURE 4. Proposed method structure in CU encoding on top of VTM. hor = horizontal and ver = vertical direction BT or TT split.

skip TT splits for the case that one BT split tests is restricted or does not preceded.

Fig. 4 shows the proposed procedure on top of VTM 4.0. Specifically, the maximum number of test cases for a CU encoding. In general, the CU is first tested by the current block encoding mode (i.e., no child nodes exist for the CU), with J_{cur} being the RD cost for the current block mode. If the current CU can have child nodes by QT structure, then QT split mode is tested and measured as J_{QT}. Unless QT split is possible, J_{QT} is set to the maximum RD cost value. $J_{\text{BT\_H}}$ and $J_{\text{BT\_V}}$ are measured similarly. The proposed method uses these prior RD costs to decides whether a TT split is required. Specifically, the proposed method skips TT_H when $\varepsilon(\text{hor}) = 0$, and TT_V when an evidence $\varepsilon(\text{ver}) = 0$. Pruning the TT split not only reduces TT split encoding complexity, but also reduces all available BT and TT split encoding complexities (see Fig. 2). Thus, the proposed method can substantially decrease both BT and TT encoding complexity by pruning TT split processes.

IV. EXPERIMENTAL RESULT

This section evaluates the proposed method performance compared with VVC reference software (i.e., VTM version 4.0) [17]. We chose frames that did not overlap the training samples, skipping the first 2 seconds frames in each sequence, e.g. 100 frames were skipped for the *Cactus* sequence. Other conditions are the same as CTC [18]. We evaluated both RD performance using Bjontegaard Delta bitrate (BDBR) and encoding complexity using measurement time [18] compared with the reference method. The computer environment was a PC with Windows 10 64-bit, Intel CPU at 3.6 GHz with 8 cores without Turbo Boost or hyperthreading technology. We calculate BDBR for Y, U and V components, denoting BDBR-YUV, as

$$BDBR-YUV = \frac{(6BDBR-Y+BDBR-U+BDBR-V)}{8}, \tag{2}$$

where BDBR-Y, BDBR-U, and BDBR-V, are BDBR of Y, U, and V components, respectively. The total encoding time reduction, *ETR*, was measured as

$$ETR = \frac{T_{\text{org}} - T_{\text{method}}}{T_{\text{org}}}, \tag{3}$$

where T_{org} and T_{method} denote the original (reference) and proposed method total encoding time, respectively.

We also tested the state-of-the-art (SOTA) MTT decision method [16] for VVC. To ensure fair comparison, the SOTA method was trained with the same sequence used in this paper. The SOTA decision tree was implemented using the python *scikit-learn* machine learning library, with maximum depth = 5. Since our main focus was on the TT decision, two SOTA classifiers were implemented: for horizontal and vertical TT splits. The SOTA encoding framework was implemented on top of VTM 4.0. In comparison with VTM 2.0 that SOTA employed, VTM 4.0 has additional tools that SOTA cannot directly access. Thus, SOTA classifiers were not applied to VTM 4.0 tools, such as separate chroma trees.

Table 5 compares SOTA and proposed method outcomes. SOTA case reduced average total encoding time by 6%, with slight coding loss, exhibiting 0.75% increased BDBR-Y component and 0.62% increased BDBR-YUV. On the other hand, the proposed method reduced average total encoding time by 33% with 0.93% and 1.02% increased BDBR-Y and BDBR-YUV, respectively. Thus, the proposed method provided approximately 7 times faster encoding than SOTA with slight coding loss.

Inspecting sequence-by-sequence results, SOTA was sometimes slower than the reference, particularly for 4K sequences *Tango2*, *FoodMarket4*, *Campfire*, *CatRobot1*, and *ParkRunning3*. This performance degradation could be due to not only false classifier prediction, but also inefficient encoding framework. In particular, the changed encoding order in SOTA may interrupt fast encoding options activated in the recent VTMs (e.g., VTM 4.0), increasing total encoding complexity. Moreover, Sobel operators and variance computation per CU that SOTA employed for classification may

TABLE 5. SOTA [16] and proposed method performance.

Class	Sequence	FPS	SOTA [16]					Proposed				
			BDBR-Y	BDBR-U	BDBR-V	BDBR-YUV	ETR	BDBR-Y	BDBR-U	BDBR-V	BDBR-YUV	ETR
Class A1 (3840×1260)	Tango2	60	0.34%	-1.48%	-1.38%	0.18%	-23%	0.58%	0.46%	0.51%	0.58%	29%
	FoodMarket4	60	0.39%	-0.16%	-0.28%	0.32%	-9%	0.63%	0.53%	0.50%	0.64%	33%
	Campfire	30	0.44%	0.07%	-0.28%	0.34%	-5%	0.72%	1.14%	1.07%	0.80%	33%
Class A1 (3840×1260)	CatRobot1	60	0.67%	-0.24%	-0.44%	0.53%	-9%	1.19%	2.03%	1.61%	1.28%	32%
	DaylightRoad2	60	0.89%	0.27%	-0.18%	0.84%	3%	1.16%	2.34%	1.69%	1.26%	31%
	ParkRunning3	50	0.30%	0.13%	0.15%	0.23%	-6%	0.57%	0.81%	0.80%	0.69%	35%
Class B (1920×1080)	MarketPlace	60	0.39%	-0.20%	0.06%	0.32%	4%	0.54%	1.04%	1.02%	0.59%	34%
	RitualDance	60	0.64%	-0.64%	-0.81%	0.49%	0%	0.92%	1.43%	1.50%	0.99%	34%
	Cactus	50	0.74%	0.17%	-0.25%	0.65%	8%	1.00%	1.43%	1.89%	1.08%	34%
	BasketballDrive	50	0.88%	-0.54%	-0.70%	0.74%	5%	1.16%	1.93%	1.72%	1.24%	33%
	BQTerrace	60	0.87%	0.32%	0.59%	0.89%	12%	0.89%	2.05%	2.97%	1.00%	31%
Class C (832×480)	BasketballDrill	30	1.37%	-0.54%	-0.62%	1.10%	15%	1.55%	2.19%	2.71%	1.67%	34%
	BQMall	60	1.11%	-0.61%	-0.74%	0.91%	15%	1.29%	2.33%	2.53%	1.42%	35%
	PartyScene	50	0.70%	0.01%	-0.12%	0.63%	25%	0.66%	1.50%	1.90%	0.76%	36%
	RaceHorsesC	50	0.64%	-0.08%	-0.50%	0.52%	12%	0.76%	1.11%	1.62%	0.82%	36%
Class D (416×240)	BasketballPass	30	1.05%	-0.97%	-0.63%	0.82%	14%	1.05%	2.18%	2.23%	1.23%	35%
	BQSquare	60	0.76%	-0.21%	-0.75%	0.68%	22%	0.54%	1.84%	2.00%	0.61%	34%
	BlowingBubbles	50	0.73%	-0.38%	-0.80%	0.59%	22%	0.65%	1.34%	1.73%	0.74%	35%
	RaceHorses	30	0.73%	-0.72%	-0.97%	0.52%	12%	0.80%	1.67%	2.27%	0.95%	36%
Class E (1280×720)	FourPeople	60	1.00%	-0.03%	-0.31%	0.89%	9%	1.34%	1.52%	1.86%	1.38%	35%
	Johnny	60	1.00%	-0.62%	-0.60%	0.84%	-3%	1.37%	1.87%	2.18%	1.44%	33%
	KristenAndSara	60	0.89%	-0.67%	-0.96%	0.69%	0%	1.14%	1.39%	1.75%	1.19%	33%
Average	Class A1		0.39%	-0.52%	-0.65%	0.28%	-10%	0.64%	0.71%	0.69%	0.67%	32%
	Class A2		0.62%	0.05%	-0.16%	0.53%	-3%	0.97%	1.73%	1.37%	1.07%	33%
	Class B		0.70%	-0.18%	-0.22%	0.62%	6%	0.90%	1.58%	1.82%	0.98%	33%
	Class C		0.95%	-0.31%	-0.50%	0.79%	17%	1.06%	1.78%	2.19%	1.17%	35%
	Class D		0.82%	-0.57%	-0.79%	0.65%	18%	0.76%	1.76%	2.06%	0.88%	35%
	Class E		0.96%	-0.44%	-0.62%	0.81%	2%	1.29%	1.59%	1.93%	1.34%	34%
	Overall		**0.75%**	**-0.33%**	**-0.48%**	**0.62%**	**6%**	**0.93%**	**1.55%**	**1.73%**	**1.02%**	**34%**

Notes: FPS denotes frame per second.

burden the encoder. In contrast, the proposed method reduced complexity reduction for all sequences by 29–36%, hence the proposed method would be extremely suitable for video sequences requiring fast encoding. In terms of compression performance, maximum BDBR-Y loss occurred for the *BasketballDrill* sequence for both SOTA and proposed methods, returning similar BDBR-Y loss for both methods (1.37% and 1.55% increase, respectively.

We compared the proposed method performance for different ETR and coding loss operating points by applying the method only when MTT depth (D_{MTT}) for the current CU was larger than the threshold, where $D_{MTT} = 0$ to 3 represents the BT and/or TT depth. Table 6 compares the different thresholds, providing details to allow trade-off between computational complexity and RD performance.

The results show the proposed method outperforms the SOTA for both coding efficiency and ETR (0.46% and 20% respectively for "Proposed-$\mathrm{D_{MTT}} > 0$")., which implies the proposed method provides better trade-offs. Coding loss was dramatically reduced for "Proposed-$\mathrm{D_{MTT}} > 1$" (0.19% BDBR-Y loss and 8% ETR). The proposed method could be easily configured to achieve different ETR and coding loss by restricting the range.

We evaluated proposed method performance for other fast coding methods applied to QT and BT compared with SOTA for the same conditions, to evaluate whether the proposed method still works appropriately under the condition. We set a new anchor: QTDR + $\mathrm{SOTA_{BT}}$, combining quad-tree depth restriction (QTDR), QT depth > 1, and a fast BT encoding SOTA technique ($\mathrm{SOTA_{BT}}$). By including both fast algorithms for QT and BT, the new anchor should have much lower complexity while substantially degrading coding efficiency compared with the original anchor. Accordingly, we made two implementations on top of the new anchor (QTDR + $\mathrm{SOTA_{BT}}$+ $\mathrm{SOTA_{TT}}$ and QTDR + $\mathrm{SOTA_{BT}}$+ Proposed method), applying different fast TT decision methods: $\mathrm{SOTA_{TT}}$ and the proposed method.

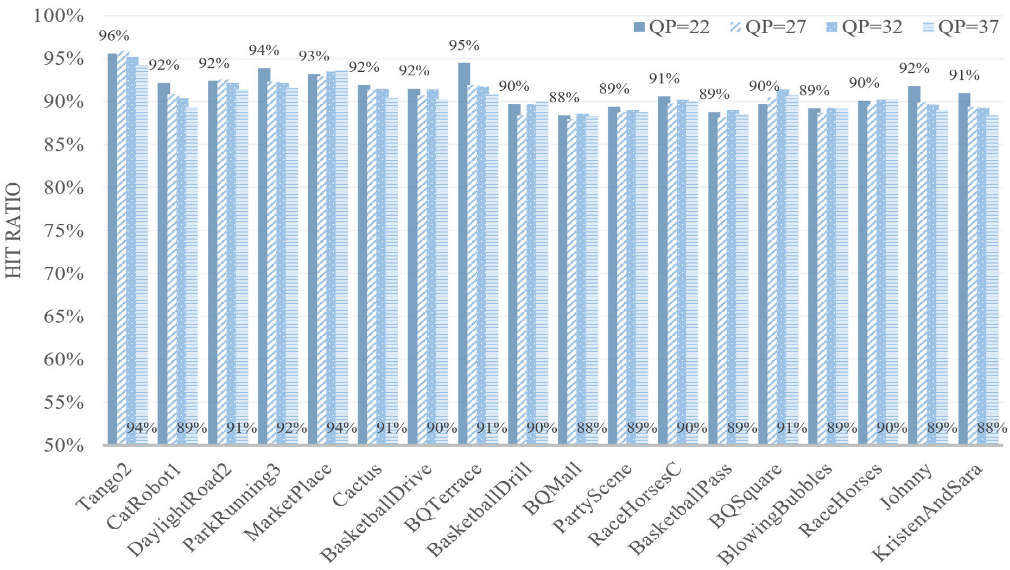


FIGURE 5. Hit ratio for the proposed method on the anchor. Ratio for QP=22 is drawn above bar and for QP=37 is in bar.

TABLE 6. SOTA [16] and proposed method performance with depth restriction.

Sequence	SOTA [16]		Proposed-No-Limit		Proposed-$D_{MTT} > 0$		Proposed-$D_{MTT} > 1$	
	BDBR-Y	ETR	BDBR-Y	ETR	BDBR-Y	ETR	BDBR-Y	ETR
Tango2	0.34%	-23%	0.58%	29%	0.30%	18%	0.11%	8%
FoodMarket4	0.39%	-9%	0.63%	33%	0.37%	21%	0.15%	11%
Campfire	0.44%	-5%	0.72%	33%	0.37%	20%	0.15%	9%
CatRobot1	0.67%	-9%	1.19%	32%	0.63%	20%	0.30%	9%
DaylightRoad2	0.89%	3%	1.16%	31%	0.59%	19%	0.23%	8%
ParkRunning3	0.30%	-6%	0.57%	35%	0.34%	23%	0.14%	11%
MarketPlace	0.39%	4%	0.54%	34%	0.33%	21%	0.14%	9%
RitualDance	0.64%	0%	0.92%	34%	0.48%	20%	0.20%	10%
Cactus	0.74%	8%	1.00%	34%	0.50%	20%	0.21%	8%
BasketballDrive	0.88%	5%	1.16%	33%	0.58%	19%	0.19%	8%
BQTerrace	0.87%	12%	0.89%	31%	0.45%	19%	0.19%	7%
BasketballDrill	1.37%	15%	1.55%	34%	0.80%	21%	0.36%	9%
BQMall	1.11%	15%	1.29%	35%	0.56%	20%	0.20%	8%
PartyScene	0.70%	25%	0.66%	36%	0.28%	20%	0.10%	8%
RaceHorses	0.64%	12%	0.76%	36%	0.38%	21%	0.18%	9%
BasketballPass	1.05%	14%	1.05%	35%	0.46%	20%	0.20%	8%
BQSquare	0.76%	22%	0.54%	34%	0.22%	18%	0.06%	6%
BlowingBubbles	0.73%	22%	0.65%	35%	0.28%	21%	0.08%	7%
RaceHorses	0.73%	12%	0.80%	36%	0.34%	20%	0.12%	8%
FourPeople	1.00%	9%	1.34%	35%	0.66%	19%	0.29%	8%
Johnny	1.00%	-3%	1.37%	33%	0.69%	19%	0.28%	8%
KristenAndSara	0.89%	0%	1.14%	33%	0.57%	19%	0.24%	8%
Overall	**0.75%**	**6%**	**0.93%**	**34%**	**0.46%**	**20%**	**0.19%**	**8%**

Notes: D_{MTT} = MTT depth (0 to 3); Proposed no-limit = proposed method was applied without any restriction; Proposed-$D_{MTT} > 0$ = proposed method was only applied when MTT depth > 0; Proposed-$D_{MTT} > 0$ = proposed method was only applied when MTT depth > 1.

Table 7 shows that SOTA and the proposed methods on top of QTDR + SOTA_{BT} provided significant ETR (37%) compared with the new anchor. However, coding efficiency for the two methods differed substantially, providing 2.77% and 1.84% BDBR-Y loss, respectively, i.e., approximately 1% BDBR difference. Thus, the proposed method significantly

TABLE 7. SOTA [16] and proposed method performance for fast encoding condition.

Sequence	QTDR + SOTA$_{\mathrm{BT}}$ + SOTA$_{\mathrm{TT}}$		QTDR + SOTA$_{\mathrm{BT}}$ + Proposed	
	BDBR-Y	ETR	BDBR-Y	ETR
Tango2	2.75%	47%	1.55%	34%
FoodMarket4	2.41%	42%	1.39%	37%
Campfire	2.00%	33%	1.25%	36%
CatRobot1	3.42%	37%	2.20%	37%
DaylightRoad2	2.58%	36%	1.54%	37%
ParkRunning3	1.42%	26%	1.22%	38%
MarketPlace	1.46%	38%	0.96%	38%
RitualDance	2.72%	42%	1.67%	36%
Cactus	2.83%	35%	1.85%	38%
BasketballDrive	2.94%	37%	1.86%	36%
BQTerrace	2.61%	38%	1.36%	41%
BasketballDrill	3.95%	35%	2.88%	40%
BQMall	3.64%	38%	2.29%	38%
PartyScene	2.18%	37%	1.48%	40%
RaceHorses	2.49%	36%	1.81%	39%
BasketballPass	2.07%	29%	2.03%	36%
BQSquare	2.18%	38%	1.33%	38%
BlowingBubbles	2.15%	32%	1.67%	40%
RaceHorses	2.46%	33%	2.08%	40%
FourPeople	3.98%	39%	2.35%	36%
Johnny	4.46%	41%	3.19%	35%
KristenAndSara	4.32%	38%	2.39%	34%
Overall	**2.77%**	**37%**	**1.84%**	**37%**

Notes: QTDR denotes QT depth restriction; SOTA$_{\mathrm{BT}}$ and SOTA$_{\mathrm{TT}}$ denote fast BT and TT decision methods for SOTA, respectively; anchor is QTDR + SOTA$_{\mathrm{BT}}$ on top of VTM 4.0.

outperformed the SOTA method, providing significant computational complexity reduction in cooperation with other fast encoding schemes in QT and BT.

Fig. 5 compares proposed method accuracy in view of hit ratio, i.e., the ratio of how many times the proposed method hits the redundancy of TT splits in terms of RD cost on top of the anchor. We evaluated the best and the worst coding efficiency cases for each resolution in Table 5. Hit ratio ranged between 96% to 89%, hence the proposed method hit TT split redundancy with high accuracy.

An additional experiment was conducted to investigate how much the proposed method influenced MTT structure encoding complexity. As discussed in Section III, final paragraph, pruning TT split can also reduce BT encoding complexity, because a CU in TT structure can be further split by TT structure as well as BT structure according to the original encoding order in VTM 4.0. Therefore, we measured running time consumed by BT and TT during encoding for the anchor and proposed method, as shown in Table 8. Since the proposed method focused on reducing TT encoding complexity, TR_{TT} provided substantial average reduction (62%). TR_{TT} maximum reduction occurred for *RitualDance* (66%). TR_{BT}, *PartyScene*, and *BlowingBubbles* sequences exhibited largest time reduction for the proposed method (25%). The proposed method also effectively decreasing BT encoding complexity.

TABLE 8. BT and TT Encoding time reduction of the proposed method compared to the anchor.

Sequence		TR$_{\mathrm{BT}}$	TR$_{\mathrm{TT}}$
Class B	MarketPlace	18%	64%
	RitualDance	18%	66%
	Cactus	20%	63%
	BasketballDrive	18%	63%
	BQTerrace	20%	60%
Class C	BasketballDrill	23%	62%
	BQMall	22%	62%
	PartyScene	25%	61%
	RaceHorsesC	23%	63%
Class D	BasketballPass	23%	60%
	BQSquare	24%	57%
	BlowingBubbles	25%	60%
	RaceHorses	24%	60%
Class E	FourPeople	20%	64%
	Johnny	19%	62%
	KristenAndSara	19%	63%
Average	Class B	19%	63%
	Class C	23%	62%
	Class D	24%	59%
	Class E	20%	63%
	Overall	**21%**	**62%**

Notes: TR$_{\mathrm{BT}}$ and TR$_{\mathrm{TT}}$ = time reduction for BT and TT compared with the anchor, respectively.

V. CONCLUSION

This paper considered MTT structure complexity in VVC, particularly TT structure, which has higher complexity than BT. We also proposed a method to efficiently skip TT, exploiting RD costs from the previously encoded CU data based on probabilistic approach. Experimental results confirmed that the proposed method significantly reduced VTM 4.0 encoding time while sustaining reasonable coding efficiency. The proposed TT pruning method also decreased BT encoding complexity. Our method shows better or comparable performance in comparison with the SOTA while ours utilizes a lightweight feature. Moreover, concerning HM encoding time, we believe that a 34% VTM encoding time reduction is significant because VTM 4.0 has much higher complexity than HM according to the report [2].

A CU encoding process would be a bottleneck in parallel processing due to the nested tree structure as discussed above. Algorithmic optimization, such as the proposed approach, could be exploited to solve the bottleneck. Thus, the current study provides a useful base to develop low complexity VVC encoders.

REFERENCES

[1] Segall, V. Baroncini, J. Boyce, J. Chen, and T. Suzuki, *Joint Call for Proposals on Video Compression With Capability Beyond HEVC*, document JVET-H1002, Joint Video Experts Team (JVET) of ITU-T and ISO/IEC, Oct. 2017.

[2] F. Bossen, X. Li, K. Suehring, and A. Norkin, *AHG Report: Test Model Software Development (AHG3)*, document JVET-N0003, Joint Video Experts Team (JVET) of ITU-T and ISO/IEC, Mar. 2019.

[3] J. Chen, Y. Ye, and S. H. Kim, *Algorithm Description for Versatile Video Coding and Test Model 4*, document JVET-M1002-v2, Joint Video Experts Team (JVET) of ITU-T and ISO/IEC, Jan. 2019.

[4] K. Choi, S.-H. Park, and E. S. Jang, *Coding Tree Pruning Based CU Early Termination*, document JCTVC-F092, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T and ISO/IEC, Jul. 2019.

[5] Z. Jin, P. An, C. Yang, and L. Shen, "Fast QTBT partition algorithm for intra frame coding through convolutional neural network," *IEEE Access*, vol. 6, pp. 54660–54673, 2018.

[6] Z. Wang, S. Wang, J. Zhang, S. Wang, and S. Ma, "Probabilistic decision based block partitioning for future video coding," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1475–1486, Mar. 2018.

[7] J. Ma, F. Le Léannec, and M. W. Park, *Description of Core Experiment: Partitioning*, document JVET-J1021, Joint Video Experts Team (JVET) of ITU-T and ISO/IEC, Apr. 2018.

[8] J. Chen, Y. Ye, and S. H. Kim, *Algorithm Description for Versatile Video Coding and Test Model 4 (VTM 4)*, document JVET-M1002-v2, Joint Video Experts Team (JVET) of ITU-T and ISO/IEC, Jan. 2019.

[9] L. Shen, Z. Zhang, and P. An, "Fast CU size decision and mode decision algorithm for HEVC intra coding," *IEEE Trans. Consum. Electron.*, vol. 59, no. 1, pp. 207–213, Feb. 2013.

[10] J. Gu, M. Tang, J. Wen, and Y. Han, "Adaptive intra candidate selection with early depth decision for fast intra prediction in HEVC," *IEEE Signal Process. Lett.*, vol. 25, no. 2, pp. 159–163, Feb. 2018.

[11] Y. Zhang, Z. Pan, N. Li, X. Wang, G. Jiang, and S. Kwong, "Effective data driven coding unit size decision approaches for HEVC INTRA coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 11, pp. 3208–3222, Nov. 2018.

[12] K. Kim and W. W. Ro, "Fast CU depth decision for HEVC using neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 5, pp. 1462–1473, May 2019.

[13] Z. Wang, S. Wang, J. Zhang, S. Wang, and S. Ma, "Effective quadtree plus binary tree block partition decision for future video coding," in *Proc. Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Apr. 2017, pp. 23–32.

[14] M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, and Z. Guan, "Reducing complexity of HEVC: A deep learning approach," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5044–5059, Oct. 2018.

[15] T.-D. Chuang, C.-Y. Chen, Y.-W. Huang, and C.-M. Lei, *C1-Related: Separate Tree Partitioning at 64×64-Luma/32×32-Chroma Unit Level*, document JVET-K0230, Joint Video Experts Team (JVET) of ITU-T and ISO/IEC, Jul. 2018.

[16] H. Yang, L. Shen, X. Dong, Q. Ding, P. An, and G. Jiang, "Low complexity CTU partition structure decision and fast intra mode decision for versatile video coding," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2019.2904198.

[17] *VVC Test Model (VTM) Software*. Accessed: Jul. 24, 2019. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/

[18] F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, *JVET Common Test Conditions and Software Reference Configurations for SDR Video*, document JVET-M1010, Joint Video Experts Team (JVET) of ITU-T and ISO/IEC, Jan. 2019.

[19] S. Ryu and J. Kang, "Machine learning-based fast angular prediction mode decision technique in video coding," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5525–5538, Nov. 2018.

[20] C. Huang, Z. Peng, F. Chen, Q. Jiang, G. Jiang, and Q. Hu, "Efficient CU and PU decision based on neural network and gray level co-occurrence matrix for intra prediction of screen content coding," *IEEE Access*, vol. 6, pp. 46643–46655, 2018.

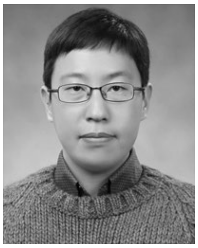
[21] F. Duanmu, Z. Ma, and Y. Wang, "Fast mode and partition decision using machine learning for intra-frame coding in HEVC screen content coding extension," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 6, no. 4, pp. 517–531, Dec. 2016.

[22] T. Amestoy, A. Mercat, W. Hamidouche, D. Menard, and C. Bergeron, "Tunable VVC frame partitioning based on lightweight machine learning," *IEEE Trans. Image Process.*, vol. 29, pp. 1313–1328, 2020, doi: 10.1109/TIP.2019.2938670.

SANG-HYO PARK received the B.S. and Ph.D. degrees in computer engineering and computer science from Hanyang University, Seoul, South Korea, in 2011 and 2017, respectively.

From 2017 to 2018, he held a Postdoctoral position with the Intelligent Image Processing Center, Korea Electronics Technology Institute, and a Research Fellow with the Barun ICT Research Center, Yonsei University, in 2018. Since 2019, he holds a Postdoctoral position at the Department of Electronic and Electrical Engineering, Ewha Woman's University, Seoul. He is the author/coauthor of several scientific/technical articles in international conferences/journals, several pending or approved patents, and more than 50 MPEG/JCT-VC/JVET contribution documents. His research interests include HEVC, VVC, encoding complexity, omnidirectional video, and deep learning.

Dr. Park has been actively participating in the standardization work of the MPEG, JCT–VC, since 2011, and JVET, since 2015. He has served as the Co-Editor of the Internet Video Coding (IVC, ISO/IEC 14496-33) for standardization for six years.



JE-WON KANG received the B.S. and M.S. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2006 and 2008, respectively, and Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2012.

He was a Senior Engineer with the Multimedia RnD and Standard Team at Qualcomm Technologies, Inc., San Diego, CA, USA, from 2012 to 2014. He was a Visiting Researcher with the Nokia Research Center, Tampere University, Tampere, Finland, in 2011, and the Mitsubishi Electric Research Laboratory, in Boston, USA, in 2010. He has been an Active Contributor to the recent international video coding standards in JCT–VC, including High-Efficiency Video Coding (HEVC) standard and the extensions to multiview videos, 3D videos, and screen content videos. He is currently an Associate Professor with Ewha Woman's University, Seoul, South Korea, and the Head of the Information Coding and Processing Laboratory, Department of Electronics and Electrical Engineering in the University. His current research topics include image and video processing and compression, computer vision, and machine learning.

• • •