PAPER

Lookahead Search-Based Low-Complexity Multi-Type Tree Pruning Method for Versatile Video Coding (VVC) Intra Coding

Qi TENG[†,††], **Guowei TENG**[†,††a)], **Xiang LI**[†,††], **Ran MA**[†,††], **Ping AN**[†,††], ***Nonmembers***, ***and*** **Zhenglong YANG**[†††], ***Member***

SUMMARY The latest versatile video coding (VVC) introduces some novel techniques such as quadtree with nested multi-type tree (QTMT), multiple transform selection (MTS) and multiple reference line (MRL). These tools improve compression efficiency compared with the previous standard H.265/HEVC, but they suffer from very high computational complexity. One of the most time-consuming parts of VVC intra coding is the coding tree unit (CTU) structure decision. In this paper, we propose a low-complexity multi-type tree (MT) pruning method for VVC intra coding. This method consists of lookahead search and MT pruning. The lookahead search process is performed to derive the approximate rate-distortion (RD) cost of each MT node at depth 2 or 3. Subsequently, the improbable MT nodes are pruned by different strategies under different cost errors. These strategies are designed according to the priority of the node. Experimental results show that the overall proposed algorithm can achieve 47.15% time saving with only 0.93% Bjøntegaard delta bit rate (BDBR) increase over natural scene sequences, and 45.39% time saving with 1.55% BDBR increase over screen content sequences, compared with the VVC reference software VTM 10.0. Such results demonstrate that our method achieves a good trade-off between computational complexity and compression quality compared to recent methods.

key words: *versatile video coding, multi-type tree, intra coding, fast pruning method, lookahead search*

1. Introduction

Versatile video coding (VVC) version 1 [1], [2] has been formally finalized at the 19th Joint Video Exploration Team (JVET) meeting on July 1, 2020. As the latest video coding standard, VVC achieves more than 40% bitrate savings while maintaining the same objective quality as compared with its predecessor, High Efficiency Video Coding (HEVC) [3], [4]. The bitrate saving can be more than 50% if subjective quality is used as the metric instead of peak signal-to-noise ratio (PSNR). That means the VVC standard can provide great user experience at half the cost of bandwidth.

The excellent coding performance of VVC is due to its many novel techniques and tools. The maximum coding tree unit (CTU) size of VVC is increased from 64 × 64, as used in HEVC, to 128 × 128, and the partition structure of CTU is also developed from quadtree (QT) to quadtree with nested multi-type tree (QTMT). The multi-type tree (MT) consists of binary trees (BT) and ternary trees (TT). BT/TT has two partition types called horizontal splitting (BH/TH) and vertical splitting (BV/TV), as shown in Fig. 1. Moreover, a coding unit (CU) can be further divided into 2 or 4 partitions owing to the newly introduced intra sub-partition (ISP). These more flexible partition structures and novel tools make it possible to model video content more precisely.

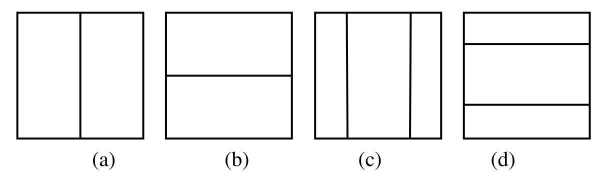

Fig. 1 Multi-type tree partition modes: (a) BV, (b) BH, (c) TV and (d) TH.

The intra prediction modes of VVC are extended from 35 to 67, including Planar mode, DC mode and 65 directional modes. For better precision of intra prediction, VVC deploys multiple reference line (MRL) which uses more reference lines rather than the nearest reference line. The newly introduced matrix weighted intra prediction (MIP), which utilizes a trained matrix to perform matrix-vector multiplication and interpolation, can also reduce the intra prediction errors. In addition to square intra prediction (45° to −135°), wide-angle intra prediction (WAIP) is used to re-map the square directional modes for the directional prediction of non-square blocks. In order to improve the residual coding efficiency, two-pass multiple transform selection (MTS) and low-frequency non-separable transform (LFNST) are also adopted. These techniques and tools improve the performance of intra prediction.

Benefiting from these new techniques, the coding performance of VVC is greatly improved. However, the coding complexity has increased dramatically as well. Under all-intra, low-delay and random-access configurations, the average coding complexity of VVC is 31 times, 5 times and 7 times that of HEVC, respectively [5]. The very high coding complexity becomes a bottleneck in the development of VVC encoder, and makes it a tough task to realize real-time video coding. Several schemes, such as [6], [7], attempt to tackle the problem by specialized hardware and they achieve

Manuscript received February 28, 2022.
Manuscript revised July 9, 2022.
Manuscript publicized August 24, 2022.

[†] The authors are with the Shanghai Institute for Advanced Communication and Data Science, Shanghai University, 333 Nanchen Road, Shanghai, China, 200444.
[††] The authors are with the School of Communication and Information Engineering, Shanghai University, 333 Nanchen Road, Shanghai, China, 200444.
[†††] The author is with the Shanghai University of Engineering Science, 333 Longteng Road, Shanghai, China, 201620.
a) E-mail: tenggw@shu.edu.cn

DOI: 10.1587/transfun.2022EAP1022

good results. However, using a specialized hardware-based encoder is both costly and inconvenient for individual users. Therefore, software optimizations are a highly cost-effective choice.

After the release of the first version of the VVC test model (VTM 1.0), many researchers have been devoting their efforts to reducing the complexity of VVC intra coding. Yang et al. [8] presented a novel QTMT decision framework based on texture features (such as gradient and local difference) and context features, but the features of classifiers are very effective for QT and BT but less so for TT. Dong et al. [9] analyzed the characteristics of RMD candidates, and then utilized four maximum likelihood estimation-based estimators to prune improbable candidates. These estimators can dynamically estimate the probability that the candidate is the best one. Tang et al. [10] utilized a block-based Canny edge detector to extract edge features for skipping the vertical partition or horizontal partition. Saldanha et al. [11] proposed two fast decision-making strategies to determine whether to skip the partition of BT or TT. One strategy uses the variance of CU and the optimal intra prediction mode, while the other strategy uses the intra sub-partition mode (ISP). Cui et al. [12] proposed an early termination strategy for CU partition based on four directional gradients. In [13], Fu et al. proposed a Bayesian rule-based early skip scheme that fully explores the information contained in horizontal binary splitting. Amestoy et al. [14] utilized random forest to determine the most probable structure of CTU. Chen et al. [15] proposed a fast intra partition algorithm based on variance and gradient. Wieckowski et al. [16] described 13 fast block partitioning selection approaches for VVC. Lei et al. [17] exploited the rate-distortion cost information of sub-CUs for pruning redundant MT partitions. LY et al. [18] exploited the prediction error to establish a tunable decision model for the early skipping of BT and TT partition. Zhang et al. [19] proposed a prediction tool based on DenseNet. This tool can predict the partition boundaries of various blocks. Tissier et al. [20] utilized the convolutional neural network (CNN) and decision tree (DT) to predict the most likely split in each block. Li et al. [21] proposed a deep learning approach to predict the QTMT-based CU partition. The aforementioned methods significantly reduce the complexity of VVC intra coding. However, most of them consider that QT, BT, and TT have the same priority (or equal weight), which leads to a worse trade-off between encoding efficiency and complexity. Additionally, BT and TT are actually hard to classify accurately by existing texture features. One reason for this is that the number of TT samples is insignificant in comparison to the QT and BT samples.

In this paper, we propose a fast MT pruning method using a lookahead search for VVC intra coding. To reduce unnecessary CTU partition structure search processes, this method employs low-complexity lookahead search, which is a breath-first search (BFS) method [22]. The lookahead search is performed before the current CU partition, because the original search in VTM is a depth-first search (DFS) method [23]. The improbable BT and TT nodes are pruned according to the result of the lookahead search. Different pruning strategies are used when the cost error is different. Experimental results show that the proposed algorithm significantly reduces complexity for VVC intra coding with negligible coding performance loss and achieves a high objective quality for a variety of video sequences. The main contributions of this paper are summarized as follows:

1) A lookahead search-based fast MT pruning method is proposed. The lookahead search is a BFS method and obtains the approximate rate-distortion (RD) cost of MT nodes. According to the result of the lookahead search, the unpromising MT partition modes are removed by different strategies under different cost errors. It's worth noting that the lookahead search does not interfere with the original search process in VTM, which means the lookahead search does not cause the encoder to produce unknowable coding results.
2) We qualitatively analyze the priorities of QT, BT, and TT by comparing their coding efficiency. In order to achieve less coding performance loss, the partition modes with high priority are more likely to be selected, while the partition modes with low priority are more likely to be removed. In view of this idea, two strategies are proposed and utilized in MT pruning.

The reminder of this paper is organized as follows. VTM intra coding is introduced in Sect. 2. Analyses of the Partition Structures are provided in Sect. 3. Section 4 presents the proposed algorithm. The experimental results and conclusions are given in Sect. 5 and Sect. 6, respectively.

2. VTM Intra Coding

Figure 2 illustrates the VTM intra prediction process. The rough mode decision (RMD) process, which consists of two steps: RMD-1 and RMD-2, uses the sum of absolute transformed difference (SATD) costs to select N candidates from 67 intra modes. These N modes form the RMD-list and are sorted in ascending order according to their SATD costs. And then, the multiple reference line (MRL), matrix weighted intra prediction (MIP), the most probable mode (MPM) and ISP are initialized and merged with the RMD-

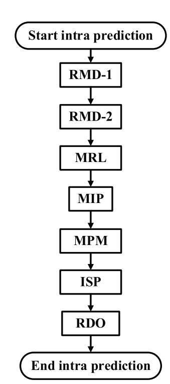

Fig. 2 VTM intra prediction.

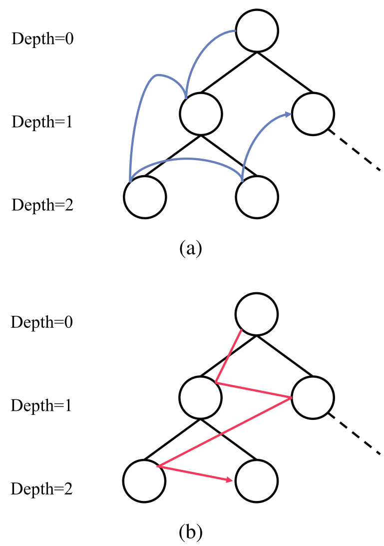

Fig. 3 Two traversal algorithms for a tree: (a) DFS and (b) BFS.

list to form a RD-list. The final step of intra prediction is the time-consuming rate-distortion optimization (RDO) process, which uses RD cost to select the optimal mode from the RD-list.

As shown in Fig. 3, two traversal algorithms can be employed for a tree. The DFS explores as far as possible along each branch before backtracking, while the BFS explores all nodes at the current depth level before moving on to the nodes at the next depth level. In VTM, the DFS algorithm is employed, and the tree consists of QT, BT, and TT. Each tree node can be regarded as a CU, a prediction unit (PU), and a transform unit (TU). The intra prediction processes are performed to derive the RD cost at each leaf node. When the leaf node splits, the RD cost of this node is the sum of the RD costs of its child nodes. This node can be regarded as a set of nodes, and the RD cost of this set is the RD cost of it. Finally, the non-conflicting node set with the minimum RD cost is the partition structure of CTU.

Theoretically, there are five partition modes for each depth, with a total of 14 possible sub-CUs (4 sub-CUs for QT, 2 sub-CUs for each BT, and 3 sub-CUs for each TT). Thus, $\sum_{Depth=0}^{7} 14^{Depth} = 113,522,235$ intra predictions should be performed in a CTU. However, not all partition modes are permitted at all depth levels in VVC. To keep things simple, n_i is defined as the number of nodes at a depth of i, and the n_0 is always 1. According to the VVC specification, the nodes at depths of 0 and 1 can only be split by QT. Considering that MT nodes cannot be split by QT, the $n_1 = n_0 \times 4$, $n_2 = n_1 \times 4$, while $n_3 = n_2 \times 14$, $n_4 = n_2 \times (4 \times 14 + 10 \times 10)$, $n_5 = n_2 \times (4 \times (4 \times 14 + 10 \times 10) + 10 \times 10 \times 10)$, and so on. Therefore, VTM only performs intra prediction $\sum_{i=0}^{7} n_i = 2,948,405$ times in a CTU. In fact, this number is lower. One reason for this is that duplicate sub-CUs are restricted due to the syntax elements in VTM. However, many redundant intra predictions are still being checked. Effective algorithms with reduced complexity and low RD performance loss are highly desired.

Table 1 Performance of different partition structures.

Partition Structure	BDBR(%)	TS(%)	TS/BDBR
QT	20.13	70.55	3.50
BTTT	1.51	22.06	14.61
QTBT	0.25	16.59	66.36
QTTT	0.88	25.16	28.59

3. Analyses of the Partition Structures

In order to evaluate the performance of five partition modes (QT, BH, BV, TH, and TV) of VVC, we perform the experiment at CU depth 2. Th experiment is conducted on VTM 10.0 [2] under common test conditions (CTC) [24]. Six video sequences with different contents and resolutions are used for statistical analyses. *Campfire* contains a dark background, *Cactus* contains rotational motions, *BasketballPass* has a simple background, *BQMall* contains global and local motion, *CatRobot1* has various motion and complex texture, and *FourPeople* contains local motion. Each sequence is coded with all-intra (AI) configuration and four Quantization Parameter (QP), 22, 27, 32, 37.

As shown in Table 1, the Bjøntegaard Delta Bit Rate (BDBR) and Bjøntegaard Delta PSNR (BDPSNR) are used to quantify the difference in RD performance between the partition structure in question and the QTMT structure in VTM. Higher BDBR means worse rate-distortion performance. The time saving (TS) reflects the change in coding complexity. To have an intuitive evaluation of the performances of different partition structures, the TS/BDBR [25] is introduced. Higher TS/BDBR value means better trade-off between time reduction and encoding efficiency.

The QT structure gets an average complexity reduction of 70.55%, while BDBR increases by 20.13%, with only 3.50 in TS/BDBR. It demonstrates that MT plays an important role in improving the RD performance, but also introduces complexity overhead as a result of numerous unnecessary and redundant partition mode decisions.

With this in mind, we can further discuss the performance of the MT structure. BTTT gets an average complexity reduction of 22.06%, while BDBR increases by 1.51%, with 14.61 in TS/BDBR. The QTBT gets an average complexity reduction of 16.59%, while BDBR increases by 0.25%, with 66.36 in TS/BDBR. The QTTT achieves an average complexity reduction of 25.16% with 0.88% BDBR increases and 28.59 in TS/BDBR. By comparing the results of QTBT, QTTT, and BTTT, we find that BTTT's trade-off is about 2 times less than that of QTTT, 5 times less than that of QTBT. Thus, QTBT has a higher priority than QTTT, while QTTT has a higher priority than BTTT. That is, the fast intra algorithm can ignore the checking processes of TT partition mode under certain conditions.

4. The Proposed Algorithm

In this section, the proposed method is introduced in detail. To determine which MT partition modes should be adopted,

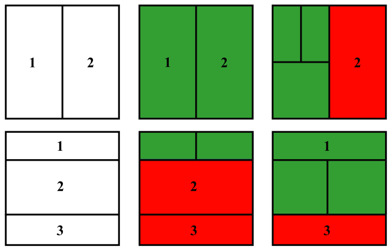

Fig. 4 Examples leading to errors between lookahead search and original search: green parts and red parts represent the error-free zone and the error zone, respectively.

a lookahead search is performed to obtain the approximate RD cost of each MT node. And then, these nodes are sorted by the RD cost, and the fast MT pruning process utilizes the sorted results to remove unpromising MT partition modes at the current CU. Considering the complexity of the lookahead search and the performance of the overall method, the proposed method only performs at CU depths 2 and 3.

4.1 Lookahead Search

The VTM obtains the RD cost of each node by the DFS method, which means the original search in the VTM cannot simultaneously obtain the RD cost of nodes at the same depth. Such a situation is harmful since pruning is difficult. Thus, we design a BFS-based lookahead search method and analyze its performance in this subsection. The "cost error" is defined as the absolute difference between the RD cost of the lookahead search and the original search.

The results of the lookahead search are the approximation results of original search due to the dependence between intra prediction processes. As shown in Fig. 4, the buffer data of part 2/3 need to be updated by the buffer data of part 1/2. In view of the different partition structure of part 1/2 in the original search and the lookahead search, the buffer data in part 2/3 is different, which makes the results obtained by the two search methods have cost errors.

We analyse the cost error magnitude through an experiment. First, high depth values are probably chosen for CUs whose neighboring CUs are at high CU depths [26]. This means that the error is greater for CUs whose neighboring CUs have high CU depths, that is, the error magnitude can be represented by the depths of neighboring CUs. Therefore, we introduce a depth distance d as the distance between the depths of the current CU and the neighboring CUs, and d can be denoted by

$$d = (D_{left} + D_{up})/2 - D_{cur} \tag{1}$$

where D_{left}, D_{up} and D_{cur} are the depths of the left CU, above CU, and current CU, respectively. As shown in Fig. 5, the normalized mean error is the normalized average of the absolute difference between the RD costs of the lookahead search and the original search. It dramatically increases with the d when the d is greater than 0.0.

The lookahead search's complexity overhead is also taken into account. Two options, baseline lookahead search (LS1) and simplified lookahead search (LS2), are evaluated to find a lookahead search method that could achieve a better balance between complexity and accuracy. The intra prediction process in LS1 is complete, while the intra prediction process in LS2 turns off the MTS, ISP, MIP, and MRL. As shown in Table 2, p_i is the probability that the ith candidate selected as the best partition mode. LS2 can achieve 60.23% time savings compared with LS1, while maintaining a similar probability distribution.

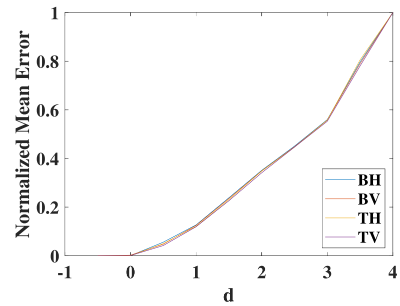

Fig. 5 An illustration of the relationship between d and the lookahead search cost errors.

Table 2 The probability distribution of the best partition mode in different lookahead search schemes (%).

Scheme	p_1	p_2	p_3	p_4	TS(%)
LS1	59.45	22.99	11.16	6.39	baseline
LS2[1]	52.22	27.86	13.04	6.88	60.23

[1] turns off the MTS, ISP, MIP and MRL

4.2 Fast MT Pruning

In the fast MT pruning method, two strategies are utilized. These strategies give preference to BT due to the higher priority of it compared with TT. In order to reduce the influence of the lookahead search cost error on the overall method, the strategy to be adopted depends on the value of d: if $d < Th$, Strategy A is adopted; otherwise, Strategy B is adopted.

Strategy A: This strategy only retains the QT candidate and two MT candidates for partition, so it is suitable for the case of a minor error. If at least one TT candidate is in front of all BT candidates in the sorted list derived from the lookahead search, the first TT is selected. And then, the first BT is selected. Otherwise, the first BT and the first candidate behind it are selected. Unselected candidates are removed, as shown in Fig. 6.

Strategy B: This strategy retains the QT candidate and two or three MT candidates for partition, so it is suitable for the case of a major error. If at least one TT candidate is in front of all BT candidates in the sorted list derived from the lookahead search, the first TT is selected. And then, the first BT and the first candidate behind it are selected. Otherwise, the first BT and the first candidate behind it are selected. Unselected candidates are removed, as shown in Fig. 6.

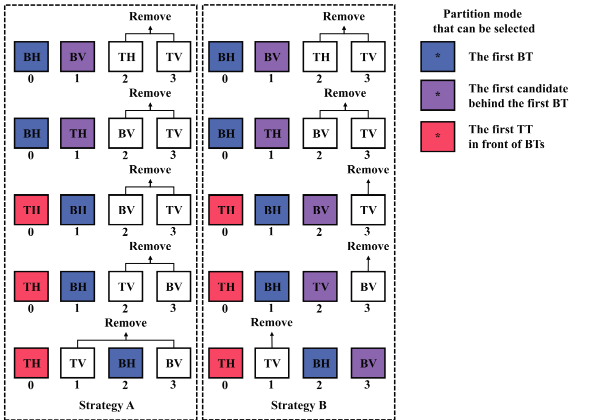

Fig. 6 Examples of two strategies in fast MT pruning: the numeric label indicates the position of the candidate in the sorted list.

4.3 Trigger Condition Selection

Although the elapsed time of the LS2 is much less than that of the LS1, the LS2 still suffers from a runtime overhead. Therefore, the overall method cannot be employed at each CU depth level. Moreover, it is not necessary to be performed at depths of 0 and 1, because VVC only allows QT partition at these depths.

From the structure of the tree, we can see that the lower the depth of pruning the tree, the fewer nodes may be traversed, which means that fewer intra predictions are performed. We can list the examples at depths of 2 and 3 to intuitively analyze this idea. In these examples, the computational burden of the lookahead search is not considered.

Example 1: At depths of 2 and 3, only a QT and the other two candidates are retained. When only QT and BT are retained, $n_1 = n_0 \times 4$, $n_2 = n_1 \times 4$, while $n_3 = n_2 \times 8$, $n_4 = n_2 \times (4\times8+4\times4)$, $n_5 = n_2 \times (4\times(4\times14+4\times10)+4\times4\times10)$, and so on. Therefore, the minimum number of execution times of intra prediction is $\sum_{i=0}^{7} n_i = 1{,}028{,}501$. When only QT and TT are retained, $n_1 = n_0 \times 4$, $n_2 = n_1 \times 4$, while $n_3 = n_2 \times 10$, $n_4 = n_2 \times (4 \times 10 + 6 \times 6)$, $n_5 = n_2 \times (4 \times (4 \times 14 + 6 \times 10) + 6 \times 6 \times 10)$, and so on. Therefore, the maximum number of execution times of intra prediction is $\sum_{i=0}^{7} n_i = 1{,}526{,}261$. In this example, the complexity of intra prediction can be reduced by 48.23% to 65.12%.

Example 2: At depths of 2 and 3, only a QT and the other three candidates are retained. The minimum number of execution times is 1,828,469, while the maximum number of execution times is 2,308,309. Therefore, the complexity of intra prediction can be reduced by 21.71% to 37.98% in this example. The calculation process is similar to that of Example 1.

In view of this, the overall method can be executed at depths of 2 and 3.

4.4 Framework of the Proposed Algorithm

As described before, the proposed method focuses on the reduction of MT complexity, as shown in Fig. 7. It consists of two stages, the lookahead search and fast MT pruning. The proposed algorithm is only employed at depths of 2 and 3 due to the complexity of the lookahead search. This trigger condition is determined by calculation.

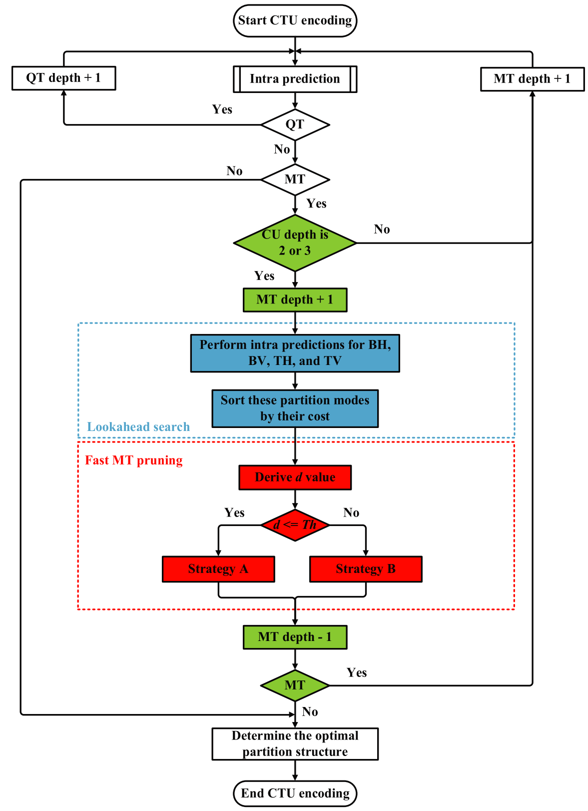

Fig. 7 The flowchart of the overall method: blue parts, red parts, and green parts represent the proposed algorithm, and the other parts are the original modules of VTM intra coding.

In the lookahead search, the approximate RD costs of BH, BV, TH, and TV are obtained by intra predictions. The error between the results of the lookahead search and the original search in VTM is inevitable, because the former is carried out in a BFS way and the latter adopts the DFS. To reduce the complexity, some intra tools are turned off in the lookahead search. Finally, four MT candidates are sorted by their costs.

In the fast MT pruning, a depth distance d is derived from the Eq. (1). Higher d values usually mean higher cost errors in the lookahead search. In order to avoid the influence of error on the performance of the fast MT pruning as much as possible, we use different pruning strategies when the value of d is different. Strategy A only retains the QT candidate and two MT candidates, while Strategy B retains the QT candidate and two or three MT candidates. If the d is less than or equal to the Th, Strategy A is adopted; otherwise, Strategy B is adopted.

5. Experimental Results

In this section, the experiments are conducted to evaluate

Table 3 The results of the proposed algorithm over Class A1, Class A2, and Class B–E.

Class	Sequence	LS2(Th = 0.0)*		LS2(Th = 2.0)		LS1(Th = 0.0)		LS1(First Two)	
		BDBR(%)	TS(%)	BDBR(%)	TS(%)	BDBR(%)	TS(%)	BDBR(%)	TS(%)
A1	Tango2	0.92	44.69	1.44	48.91	0.59	26.00	0.91	32.48
	FoodMarket4	0.95	44.01	1.47	49.47	0.68	27.29	0.97	35.54
	Campfire	0.87	46.07	1.38	51.98	0.68	34.65	1.19	45.40
A2	CatRobot1	1.13	47.85	1.72	53.62	0.81	35.12	1.34	44.28
	DaylightRoad2	1.07	51.42	1.51	58.32	0.88	36.84	1.30	49.04
	ParkRunning3	0.47	45.28	0.64	48.66	0.32	35.43	0.45	40.84
B	BasketballDrive	0.85	48.43	1.55	55.80	0.60	33.90	1.33	46.10
	BQTerrace	0.84	48.69	1.22	53.04	0.67	36.62	1.12	48.00
	Cactus	0.98	50.23	1.53	56.81	0.70	39.02	1.35	49.89
	MarketPlace	0.95	50.91	1.49	57.58	0.61	36.96	1.03	47.67
	RitualDance	1.51	40.90	2.32	47.75	1.08	26.75	1.96	37.74
C	BasketballDrill	1.71	47.83	2.64	51.92	1.46	36.94	2.26	48.22
	BQMall	0.87	45.98	1.35	51.77	0.63	35.37	1.46	49.05
	PartyScene	0.54	47.46	0.76	50.03	0.49	37.28	0.93	49.96
	RaceHorsesC	0.78	48.09	1.08	53.33	0.57	37.47	1.10	49.11
D	BasketballPass	0.64	45.59	1.24	51.07	0.66	36.42	1.67	49.10
	BlowingBubbles	0.53	46.18	0.82	49.53	0.59	36.45	1.13	48.95
	BQSquare	0.66	46.41	0.88	48.69	0.55	34.50	1.05	47.78
	RaceHorses	0.47	46.06	0.82	49.35	0.50	36.24	0.80	46.58
E	FourPeople	1.31	48.34	2.12	55.32	1.18	37.19	1.99	50.21
	Johnny	1.43	49.93	2.32	55.53	1.15	37.35	1.81	48.34
	KristenAndSara	1.06	46.99	1.73	53.48	0.83	34.26	1.55	46.48
	Average	**0.93**	**47.15**	**1.45**	**52.36**	**0.74**	**34.91**	**1.30**	**45.94**
	TS/BDBR		**50.51**		**35.99**		**47.34**		**35.24**

* indicates the main proposal in this paper

the performance of the proposed algorithm. To validate its effectiveness and robustness, we implement the proposed algorithm in VVC reference software VTM 10.0 [2] and compare it to several related works. Section 5.1 presents the setup of the experiments. Then, the performance of the overall framework achieved on VTM 10.0 is revealed in Sect. 5.2. Finally, the performance comparison with other recent works is shown in Sect. 5.3.

5.1 Experimental Setup

The sequences of Class A1, Class A2 and Class B-F in CTC [24] are utilized. Class A1 and A2 are the ultra high-definition (UHD) sequence sets with 10-bit depth. Class B and C-F are the high-definition (HD) and low-definition sequence sets with 8-bit depth, respectively. All the sequences have different and various contents and scenes. Class A1-E includes material mostly captured with video cameras, and Class F contains some computer-generated material as well as typical screen content. AI configuration is adopted. Four QPs, 22, 27, 32 and 37, are used in our experiments.

The BDBR and BDPSNR [27] are used to evaluate the RD performance of the algorithm. Higher BDBR or lower BDPSNR means worse RD performance. Time saving (TS) is employed to represent the coding time change in percentage as shown in

$$TS = \frac{T_{VTM10.0} - T_{Proposed}}{T_{VTM10.0}} \times 100\% \tag{2}$$

where $T_{VTM10.0}$ and $T_{Proposed}$ represent the encoding time of the proposed algorithm and original VTM 10.0 algorithm, respectively. The experimental results are averaged over four QPs.

To have an intuitive evaluation of the performances of different methods, the TS/BDBR [25] is introduced. Higher TS/BDBR value means better trade-off between time reduction and encoding efficiency.

5.2 Performance Evaluation of Overall

Table 3 provides the proposed algorithm results for Classes A1, A2 and B-E. “LS2 (Th = 0.0)” and “LS2 (Th = 2.0)” denote the LS2 option with $Th = 0.0$ and the LS2 option with $Th = 2.0$, respectively. “LS1 (Th = 0.0)” represents the LS1 option with $Th = 0.0$, while “LS1 (First Two)” indicates that the LS1 option retains the QT candidate and directly selects the first two candidates in the sorted list without any strategy. It can be observed that LS2 (Th = 0.0), LS2 (Th = 2.0), LS1 (Th = 0.0) and LS1(First Two) can greatly reduce coding time for all natural scene sequences, and get an average complexity reduction of 47.15%, 52.36%, 34.91% and 45.94%, with 0.93%, 1.45%, 0.74% and 1.30% BDBR increase, respectively. It shows that LS1 (Th = 0.0) has the better TS/BDBR compared with LS1 (First Two), which means that our strategies are effective. Additionally, the TS/BDBR of LS2 (Th = 0.0) is similar to that of LS1 (Th = 0.0), while the time reduction of the former is greater than that of the latter and the RD performance loss of the latter is less than that of the former. Therefore, LS2 (Th = 0.0) can be applied in scenes requiring low complexity, while LS1 (Th = 0.0) can be applied in scenes requiring high visual quality. It also shows LS2 (Th = 2.0) has the lower

Table 4 The results of the proposed algorithm over Class F.

Sequence	LS2 (Th = 0.0)*		LS2 (Th = 2.0)		LS1 (Th = 0.0)		LS1 (First Two)	
	BDBR(%)	TS(%)	BDBR(%)	TS(%)	BDBR(%)	TS(%)	BDBR(%)	TS(%)
BasketballDrillText	1.79	44.58	2.62	49.62	1.34	33.77	2.51	45.05
SlideEditing	1.31	45.36	2.05	52.64	1.09	36.76	1.70	47.46
SlideShow	1.54	46.23	2.01	52.18	1.11	32.80	1.84	43.70
Average	**1.55**	**45.39**	**2.23**	**51.48**	**1.18**	**34.44**	**2.01**	**45.40**
TS/BDBR		**29.28**		**23.09**		**29.19**		**22.59**

* indicates the main proposal in this paper

Table 5 Summary results of LS2 (Th = 0.0) over different classes.

Bitdepth	Class	BDBR(%)	BDPSNR(dB)	TS(%)
10	A1	0.91	-0.03	44.93
10	A2	0.89	-0.03	48.18
8	B	1.03	-0.04	47.83
8	C	0.98	-0.05	47.34
8	D	0.57	-0.04	46.06
8	E	1.27	-0.05	48.42
8	F	1.55	-0.11	45.39

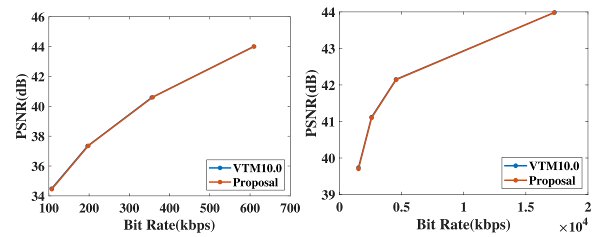

Fig. 8 Performances of the overall algorithms: (a) RD curves of *BasketballPass*(Class D) and (b) RD curves of *Tango2*(Class A1).

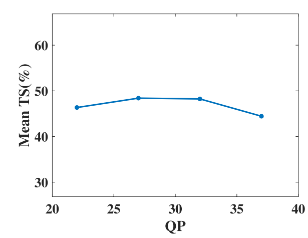

Fig. 9 Mean TS curve under different QPs.

TS/BDBR compared with LS2 (Th = 0.0). The reason for this is that Strategy A is used more often in LS2 (Th = 2.0).

Table 4 provides the proposed algorithm results over Class F. Simulation results show that the complexity of the proposed algorithm is reduced by 45.39%, 51.48%, 34.44% and 45.40% on average when the methods are LS2 (Th = 0.0), LS2 (Th = 2.0), LS1 (Th = 0.0) and LS1 (First Two), respectively. Meanwhile, there are only 1.55%, 2.23%, 1.18% and 2.01% BDBR increases, accordingly. This demonstrates that the proposed algorithm performs similar complexity reduction over screen content sequences, but performs better RD performance over natural scene sequences.

Specifically, in LS2 (Th = 0.0), the highest time saving is 51.42% for *DaylightRoad2*, and the lowest time saving is 44.01% for *FoodMarket4*. *FoodMarket4* contains a large number of complex textures, while the texture of *DaylightRoad2* is relatively regular and simple. It shows that the performance of our algorithm is related to the texture complexity. Additionally, a consistent gain is achieved with a minimum of 44.93% in Class A1 and a maximum of 48.42% in Class E, and this method provides a 48.46% time reduction with a very low BDBR increase of 0.57% in Class D, as shown in Table 5. The results also show that 10-bit sequences achieve 46.56% complexity reduction with 0.90% BDBR increase or 0.03 dB BDPNSR decrease on average, and 8-bit sequences obtain 47.01% complexity reduction with 1.08% BDBR increase or 0.06 dB BDPSNR decrease on average. The speed gain of 8-bit sequences is similar to that of 10-bit sequences, while the RD performance of the latter is better.

Furthermore, RD curves of our approach and the VTM 10.0 are provided to verify our algorithm's performance under different bit rates, including the low-definition sequence, *BasketballPass* in Class D, and the high-definition sequence, *Tango2* in Class A1. As shown in Fig. 8, our approach exhibits similar RD performance compared with VTM 10.0 for all bit rate points. This demonstrates that the RD loss of our method is negligible over different bandwidth.

Figure 9 shows the mean TS curve of our method under different QPs. The results are averaged over all sequences. It can be observed that our approach achieves a consistent time saving over different QPs.

5.3 Performance Comparison with Other Works

The results of four recent works are presented in Table 6 for objective comparisons, including Li [21], LY [18], ZQ [19], and Tissier [20]. Li, ZQ, and Tissier are CNN-based methods, while LY uses handcrafted features. They all focus on low-complexity QTMT structure decisions. Among them, Li and LY are implemented on the VTM 7, while ZQ and Tissier are implemented on the VTM 10. Because VTM 7 and VTM 10 use the same CU splitting scheme, our comparison with Li and LY is reasonable. Since the test results given by Li are not complete according to CTC, we only compare the existing data in it with ours. The data of other algorithms is calculated from the data given in their papers.

As shown in Table 6, ZQ reduces the complexity from 20.31% in Class D to 74.97% in Class A1, 51.19% on average, and the BDBR increment is from 0.62 in Class D to 2.81 in Class E, 1.84% on average. The results also show that the complexity reduction of ZQ in the Class F is low.

Table 6 Performance of other works and the proposed algorithm.

Class	Li [21], VTM 7.0		LY [18], VTM 7.1		ZQ [19], VTM 10.0		Tissier [20], VTM 10.2		Proposed, VTM10.0	
	BDBR(%)	TS(%)	BDBR(%)	TS(%)	BDBR(%)	TS(%)	BDBR(%)	TS(%)	BDBR(%)	TS(%)
A1	1.60	43.9	1.90	46.6	2.24	74.97	0.60	47.5	0.91	44.93
A2	1.49	45.5	1.02	42.9	1.88	74.25	0.80	46.4	0.89	48.18
B	1.15	47.7	1.11	42.1	1.86	66.10	0.85	51.0	1.03	47.83
C	1.09	45.2	1.57	44.4	1.41	34.49	0.96	45.7	0.98	47.34
D	1.07	43.5	1.22	43.4	0.62	20.31	0.71	43.3	0.57	46.06
E	1.81	49.5	1.57	40.4	2.81	53.77	1.23	43.9	1.27	48.42
Average[1]	**1.37**	**45.9**	**1.40**	**43.3**	**1.80**	**53.98**	**0.86**	**46.3**	**0.94**	**47.13**
F	-	-	2.20	48.1	2.03	34.42	1.43	25.3	1.55	45.39
Average[2]	**1.37**	**45.9**	**1.51**	**44.0**	**1.84**	**51.19**	**0.94**	**43.3**	**1.03**	**46.88**
TS/BDBR		**33.5**		**29.1**		**27.88**		**46.1**		**45.58**

[1] indicates the average over Class A1, Class A2 and Class B-E
[2] indicates the average over all classes

This demonstrates that ZQ can not achieve consistent performance over different classes, and ZQ only performs better on high-resolution natural scene sequences. One of the reasons is that the training database of ZQ only consists of 2K sequences and does not contain screen content sequences. If the training database contains more different sequences, the performance can be improved.

The Tissier gets an average complexity reduction of 43.3% with 0.94% BDBR increases. Similar to ZQ, the screen content sequences are not considered in the learning process of Tissier. As a result, the performance of Tissier in the Class F is not good.

In addition to ZQ and Tissier, Li also uses a CNN-based method. Li gets an average complexity reduction of 45.9% with 1.37% BDBR increases. The average complexity reduction of Li is similar to that of our method, while the RD performance of Li is lower.

As discussed above, we can see that the time saving of CNN-based methods is lower than that of our method in most instances. It demonstrates that the computational burden of CNN affects the performance to a certain extent. Therefore, it is necessary to optimize the structure and algorithm of CNN. Moreover, once the video signal is disturbed, the performance of them will decline because CNNs are generally prone to noise interruptions, i.e., small image noise can cause drastic changes in the output [28].

LY obtains 44.0% complexity reduction with 1.51% BDBR increases. The BDBR increment of each class is greater than 1.00%, especially in Class F. Our algorithm can achieve better speed gain with a better BDBR increment compared with LY.

By using the TS/BDBR metric, the value 45.58 in our method is higher than 33.5 in Li, 29.1 in LY and 27.88 in ZQ, showing that our method is competitive with these algorithms. Although the TS/BDBR value 46.1 in Tissier is similar to that of our method, our method achieves a better performance in Class F.

In view of the above experiments and analysis, our method achieves a good trade-off between coding efficiency and encoding time saving compared with other recent approaches. Furthermore, our scheme achieves a consistent time saving over different classes. This promising result is due to our stable lookahead search processes and effective pruning strategies.

6. Conclusion

In this paper, we propose a fast MT pruning algorithm for VVC intra coding. This method utilizes lookahead search and MT pruning strategies to determine the proper partition modes. Additionally, the intra prediction in lookahead search can turn off some intra tools according to the application scenarios. Experimental results show that the proposed algorithm can averagely reduce the encoding time by 47.15% on the natural scene sequences compared with the original test model, VTM10.0, while maintaining similar coding efficiency. Furthermore, the proposed algorithm verifies the performance of the screen content sequences and obtains 45.39% complexity reduction with 1.55% BDBR increase. Therefore, our scheme achieves a good trade-off between coding efficiency and coding speed with strong robustness and stability.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62071287, in part by the Chen Guang Project supported by Shanghai Municipal Education Commission and Shanghai Education Development Foundation under Grant No. 17CG41, and in part by the Open Fund of Key Laboratory of Advanced Display and System Applications of Ministry of Education (Shanghai University).

References

[1] B. Bross, J. Chen, S. Liu, and Y.K. Wang, “Versatile video coding (draft 10),” document JVET-S2001, Joint Video Exploration Team (JVET), July 2020.
[2] J. Chen, Y. Ye, and S.H. Kim, “Algorithm description for versatile video coding and test model 10 (vtm 10),” document JVET-S2002, Joint Video Exploration Team (JVET), July 2020.
[3] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol.22, no.12, pp.1649–1668, Dec. 2012.

[4] C. Hsu, C. Chen, T. Chuang, H. Huang, S. Hsiang, C. Chen, M. Chiang, C. Lai, C. Tsai, Y. Su, Z. Lin, Y. Hsiao, J. Klopp, I. Wang, Y. Huang, and S. Lei, "Description of SDR video coding technology proposal by mediatek," document JVET-J0018, Joint Video Exploration Team (JVET), April 2018.
[5] P. Farhad, A.M. Ali, G. Moncef, and H.M. Reza, "Complexity analysis of next-generation VVC encoding and decoding," Proc. IEEE Int. Conf. on Image Process. (ICIP), Virtual Conference, pp.3134–3138, Oct. 2020.
[6] A. Kammoun, W. Hamidouche, P. Philippe, O. Déforges, F. Belghith, N. Masmoudi, and J.F. Nezan, "Forward-inverse 2D hardware implementation of approximate transform core for the VVC standard," IEEE Trans. Circuits Syst. Video Technol., vol.30, no.11, pp.4340–4354, Nov. 2020.
[7] M.J. Garrido, F. Pescador, M. Chavarrías, P.J. Lobo, and C. Sanz, "A 2-D multiple transform processor for the versatile video coding standard," IEEE Trans. Consumer Electron., vol.65, no.3, pp.274–283, April 2019.
[8] 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., vol.30, no.6, pp.1668–1682, April 2020.
[9] X. Dong, L. Shen, M. Yu, and H. Yang, "Fast intra mode decision algorithm for versatile video coding," IEEE Trans. Multimedia, vol.24, pp.400–414, 2022.
[10] 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," Proc. IEEE Asia Pac. Conf. Circuits Syst. (APCCAS), Bangkok, Thailand, pp.361–364, Nov. 2019.
[11] M. Saldanha, G. Sanchez, C. Marcon, and L. Agostini, "Fast partitioning decision scheme for versatile video coding intra-frame prediction," Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Seville, Spain, pp.1–5, Oct. 2020.
[12] J. Cui, T. Zhang, C. Gu, X. Zhang, and S. Ma, "Gradient-based early termination of CU partition in VVC intra coding," Proc. Data Compression Conf. (DCC), Snowbird, UT, USA, pp.103–112, March 2020.
[13] T. Fu, H. Zhang, F. Mu, and H. Chen, "Fast CU partitioning algorithm for H.266/VVC intra-frame coding," Proc. Int. Conf. Multimedia Expo (ICME), Shanghai, China, pp.55–60, July 2019.
[14] 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, Sept. 2019.
[15] J. Chen, H. Sun, J. Katto, X. Zeng, and Y. Fan, "Fast QTMT partition decision algorithm in VVC intra coding based on variance and gradient," Proc. IEEE Int. Conf. Vis. Commun. Image Process. (VCIP), Sydney, NSW, Australia, pp.1–4, Dec. 2019.
[16] A. Wieckowski, J. Ma, H. Schwarz, D. Marpe, and T. Wiegand, "Fast partitioning decision strategies for the upcoming versatile video coding (VVC) standard," Proc. IEEE Int. Conf. on Image Process. (ICIP), Taipei, Taiwan, pp.4130–4134, Sept. 2019.
[17] M. Lei, F. Luo, X. Zhang, S. Wang, and S. Ma, "Look-ahead prediction based coding unit size pruning for VVC intra coding," Proc. IEEE Int. Conf. Image Process. (ICIP), Taipei, Taiwan, pp.4120–4124, Sept. 2019.
[18] Y. Li, G. Yang, Y. Song, H. Zhang, X. Ding, and D. Zhang, "Early intra CU size decision for versatile video coding based on a tunable decision model," IEEE Trans. Broadcast., vol.67, no.3, pp.710–720, 2021.
[19] Q. Zhang, R. Guo, B. Jiang, and R. Su, "Fast CU decision-making algorithm based on DenseNet network for VVC," IEEE Access, vol.9, pp.119289–119297, 2021.
[20] A. Tissier, W. Hamidouche, S. Belhadj, D. Mdalsi, J. Vanne, F. Galpin, and D. Menard, "Machine learning based efficient QT-MTT partitioning scheme for VVC intra encoders," arXiv:2103.05319v2 [eess.IV], March 2021.
[21] T. Li, M. Xu, R. Tang, Y. Chen, and Q. Xing, "Deepqtmt: A deep learning approach for fast QTMT-based CU partition of intra-mode VVC," arXiv:2006.13125 [cs, eess], July 2020.
[22] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithms, 3rd ed., ch. 22.2, MIT Press, 2009.
[23] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithms, 3rd ed., ch. 22.3, MIT Press, 2009.
[24] 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 Exploration Team (JVET), Jan. 2019.
[25] G. Tang, M. Jing, X. Zeng, and Y. Fan, "Adaptive CU split decision swith pooling-variable CNN for VVC intra encoding," Proc. IEEE Int. Conf. Vis. Commun. Image Process. (VCIP), Sydney, NSW, Australia, pp.1–4, Dec. 2019.
[26] L. Shen, Z. Zhang, and P. An, "Fast CU size decision and mode decision algorithm for HEVC," IEEE Trans. Consumer Electron., vol.59, no.1, pp.207–213, Fab. 2013.
[27] G. Bjøntegaard, "Calculation of average psnr differences between r-d curves," document VCEG-M33, Joint Video Exploration Team (JVET), April 2001.
[28] Q. Li, L. Shen, S. Guo, and Z. Lai, "Wavelet integrated cnns for noise-robust image classification," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp.7243–7252, 2020.

Qi Teng received his B.E. degree in electronic information engineering from the Changchun University of Science and Technology in 2018. He is currently pursuing a M.S. degree at the School of Communication and Information Engineering, Shanghai University, China. His current research interests include video coding and computer vision.

Guowei Teng received his Ph.D. degree in Communication and Information Systems from Shanghai University in 2005 and his M.S. degree in electronics from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. He is currently a senior engineer at the School of Communication and Information Engineering, Shanghai University, China. His research interests include video compression, image and video processing and computer vision.

Xiang Li received the B.S. degree from Tianjin University of Technology and Education, Tianjin, China, in 2001, and the M.S. degree from Jiangsu University, Zhenjiang, China, in 2008. He is currently pursuing a Ph.D. degree in Signal and Information Processing at Shanghai University, China. His current research interests include artificial intelligence and deep learning.

Ran Ma received her B.S. degree from Yangzhou University, Yangzhou, China, and her M.S. and Ph.D. degrees from Shanghai University, Shanghai, in 1997, 2000, and 2018, respectively. She is currently an associate professor at the School of Communication and Information Engineering, Shanghai University, China. Her research interests include stereoscopic image and video processing, coding, and application.

Ping An received the B.E. and M.S. degrees from the Hefei University of Technology, Hefei, China, in 1990 and 1993, respectively, and the Ph.D. degree from Shanghai University, Shanghai, China, in 2002. She is currently a professor at the School of Communication and Information Engineering, Shanghai University, China. Her research interests include image and video processing, with a focus on immersive video processing.

Zhenglong Yang received the Ph.D. degree from Shanghai University, Shanghai, China, in 2018. He is currently a lecturer at the Shanghai University of Engineering Science and a member of the China AVS Working Group. His research interests include video coding, artificial intelligence and video processing.