



Communication Visual Perception Based Intra Coding Algorithm for H.266/VVC

Yu-Hsiang Tsai¹, Chen-Rung Lu¹, Mei-Juan Chen¹, Meng-Chun Hsieh¹, Chieh-Ming Yang¹ and Chia-Hung Yeh^{2,3,*}

- ¹ Department of Electrical Engineering, National Dong Hwa University, Hualien 974301, Taiwan
- ² Department of Electrical Engineering, National Taiwan Normal University, Taipei 106308, Taiwan
- ³ Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung 804201, Taiwan
- * Correspondence: chyeh@ntnu.edu.tw

Abstract: The latest international video coding standard, H.266/Versatile Video Coding (VVC), supports high-definition videos, with resolutions from 4 K to 8 K or even larger. It offers a higher compression ratio than its predecessor, H.265/High Efficiency Video Coding (HEVC). In addition to the quadtree partition structure of H.265/HEVC, the nested multi-type tree (MTT) structure of H.266/VVC provides more diverse splits through binary and ternary trees. It also includes many new coding tools, which tremendously increases the encoding complexity. This paper proposes a fast intra coding algorithm for H.266/VVC based on visual perception analysis. The algorithm applies the factor of average background luminance for just-noticeable-distortion to identify the visually distinguishable (VD) pixels within a coding unit (CU). We propose calculating the variances of the numbers of VD pixels in various MTT splits of a CU. Intra sub-partitions and matrix weighted intra prediction are turned off conditionally based on the variance of the four variances for MTT splits and a thresholding criterion. The fast horizontal/vertical splitting decisions for binary and ternary trees are proposed by utilizing random forest classifiers of machine learning techniques, which use the information of VD pixels and the quantization parameter. Experimental results show that the proposed algorithm achieves around 47.26% encoding time reduction with a Bjøntegaard Delta Bitrate (BDBR) of 1.535% on average under the All Intra configuration. Overall, this algorithm can significantly speed up H.266/VVC intra coding and outperform previous studies.

Keywords: H.266; versatile video coding; multi-type tree; coding tool; visual perception; machine learning; intra coding

1. Introduction

The applications of multimedia and network communication technologies have grown rapidly. High-definition videos have increased the amount of data transmission required for video communication. Therefore, video compression to reduce the amount of video transmission and data storage is essential. H.266/Versatile Video Coding (VVC) [1–3] was developed to improve the performance of video compression and can support ultra-high-definition videos, with resolutions from 4 K to 8 K or even larger. H.266/VVC, the latest generation of international video coding standards, was completed in July 2020 [4,5].

In H.265/High Efficiency Video Coding (HEVC), the previous generation video coding standard, the maximum allowed size of a coding tree unit (CTU) is 64×64 . A picture is divided into non-overlapping CTUs. Then, a recursive quadtree (QT) partition can be applied to each CTU to divide it into four equal-sized coding units (CUs), and each CU can be divided recursively until the CU size is 8×8 . Besides the existing QT partition structure in HEVC, the new multi-type tree (MTT) in H.266/VVC can be separated into two partition types: binary tree (BT) and ternary tree (TT). The partition direction of the binary and ternary trees can be further classified into horizontal and vertical directions, as displayed in Figure 1. In H.266/VVC, the maximum allowed size of a CTU is 128×128 [2]. The QT partition also occurs during the CTU encoding process. In the QT partition process,



Citation: Tsai, Y.-H.; Lu, C.-R.; Chen, M.-J.; Hsieh, M.-C.; Yang, C.-M.; Yeh, C.-H. Visual Perception Based Intra Coding Algorithm for H.266/VVC. *Electronics* 2023, *12*, 2079. https:// doi.org/10.3390/electronics12092079

Academic Editor: Stefanos Kollias

Received: 28 February 2023 Revised: 22 April 2023 Accepted: 25 April 2023 Published: 1 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). 128 × 128 corresponds to QtDepth 0, while 64 × 64 corresponds to QtDepth 1, and so on. Then, the nested MTT partition starts if the maximum size of the current prediction configuration that can perform the MTT partition is reached in the partition process. The MttDepth begins at 0 and increases by 1 for each BT or TT split. An H.266/VVC partition example of a CU is illustrated in Figure 2. In addition, H.266/VVC also includes many new coding tools and can be adapted to different videos. Some of these coding tools are used in intra coding, such as intra sub-partitions (ISP) [6], matrix weighted intra prediction (MIP) [7], multiple reference line intra prediction (MRL) [8], cross-component linear model (CCLM) and position dependent intra prediction combination (PDPC) [9], etc. ISP enables the luma CUs equal to 4×8 (or 8×4) and larger than 4×8 (or 8×4) to be divided into two and four sub-partitions horizontally or vertically, respectively. All sub-partitions have the same intra mode. In the MIP mode, smaller boundaries are obtained by averaging neighboring boundary samples. The averaged samples are the input of a matrix vector multiplication which is followed by the addition of an offset. Then, the predicted pixels are generated through linear interpolation [2].



Figure 1. H.266/VVC QT-MTT partition diagram.



Figure 2. Example of H.266/VVC QT-MTT partition.

However, more flexible partition modes and advanced coding tools result in large computations. This paper presents a fast intra coding algorithm based on visual perception, to efficiently decide the execution of the coding tools and MTT partition in H.266/VVC and save the encoding time. The proposed algorithm detects the visually distinguishable (VD) pixels that may affect visual perception. We observe that the variances computed by the VD pixels among MTT partitions within the coding unit are related to the usage of the coding tools and their corresponding MTT splitting modes. Therefore, the variances representing the perceptual information of human vision are used to be the cues for the decision of turning off coding tools and the input features of machine learning. Since the proposed method applies the factor of average background luminance for just-noticeable-distortion (JND) by using spatial average filtering to identify the VD pixels, we focus on All Intra configuration for the study.

2. Related Work

The fast algorithms of H.266/VVC intra coding have been explored to solve the problem of the high computational requirement of H.266/VVC. The methods can be roughly categorized into probability-based [10–12], learning-based [13–20], probability-and learning-based [21,22], texture-based [23–26], gradient-based [27,28], and texture- and

gradient-based [29–31] techniques. The related work on the fast intra coding of H.266/VVC is discussed below.

Fu et al. [10] decided to terminate the vertical splits and TT horizontal split early based on the information of the two sub-CUs after the BT horizontal split. Park and Kang [11] determined the TT split direction of the current CU based on the Bayesian approach by comparing the rate-distortion costs (RD costs) of the horizontal and vertical splits of the BT split. If the RD cost of the horizontal direction is greater than or equal to the RD cost in the vertical direction, the horizontal direction split will be skipped during the TT split; otherwise, if the RD cost in the vertical direction is greater than or equal to the RD cost in the horizontal direction, then the TT vertical split will be skipped. Zhang et al. [12] employed the Bayesian principle to make the CU splitting decision, and utilized the improved de-blocking filter to determine the CU splitting mode.

Yang et al. [13] used the decision tree to propose a cascade decision structure for QT and MTT partitions. A fast intra mode decision by gradient descent search quickly found a suitable intra frame prediction mode. Chen et al. [14] utilized visual perception to extract the VD pixels. The distributions acquired by the horizontal and vertical projections of VD pixels and the quantization parameter (QP) were the input features of random forest classifiers to quickly select the MTT partition. Li et al. [15] proposed a multi-stage exit convolutional neural network model with an early-exit mechanism to decide the CU partition. Wu et al. [16] trained two support vector machine classifiers to predict the split or non-split, and horizontal or vertical split for CUs of different sizes. Wang et al. [17] devised a multi-stage early termination convolutional neural network model that can predict all the partition information of a 32×32 CU and its sub-CUs. Zouidi et al. [18] proposed an intra mode decision to skip unlikely intra prediction modes by using a multi-task learning convolutional neural network. Taabane et al. [19] used five binary light gradient boosting machine classifiers, which were offline-trained to predict the QT/MTT partitions. Park and Kang [20] developed two types of features which can be derived during encoding. The two lightweight neural network models use the features as input vectors to determine the TT structures subsequent to a quadtree partition.

Zhao et al. [21] proposed a complexity reduction algorithm based on statistical theory and size-adaptive convolutional neural network to decide whether to divide CUs of different sizes. Zhao et al. [22] extracted the spatial-temporal neighboring coding features by the deep convolutional network and fused all reference features acquired by different convolutional kernels to decide an optimal intra coding depth. The probability estimation and the spatial-temporal coherence are employed to further select the candidate partition modes within the optimal coding depth.

Peng et al. [23] adaptively set texture thresholds to classify CUs into three categories: simple, common, and complex. For simple CUs, the partition modes were skipped early, and for complex CUs, the non-partition mode was skipped. By computing the directional features, unnecessary horizontal or vertical partitions are omitted for common and complex CUs. Zhang et al. [24] proposed a fast intra block partition algorithm by utilizing gray level co-occurrence matrix to calculate the texture direction information of the CU and terminate the horizontal or vertical split of the BT and TT. Shu et al. [25] designed a complexity control algorithm for VVC intra coding by using texture entropy at the CU level. Zhang et al. [26] decided whether to divide the CU into sub-CUs according to the texture complexity. The CU splitting mode is determined by the texture direction.

Cui et al. [27] computed the gradients to omit unnecessary CU sizes and judge the BT split or TT split in the horizontal or vertical direction. Gou et al. [28] employed the histogram of the oriented gradient and the intra modes of the upper and left CUs for a fast intra mode decision.

Yoon et al. [29] utilized the gradient calculation used in the adaptive loop filter to propose the activity-based block partitioning decision method. The activity indicates the block texture complexity. Jing et al. [30] proposed a fast QT partition decision based on gradient structure similarity. The texture direction was determined by calculating the

standard deviations of the vertical and horizontal directions of the current CU to decide the MTT partition direction. Fan et al. [31] utilized the variances of the pixels in a CU and the gradient features extracted by the Sobel operator to terminate the further splitting of a CU, decide the QT partition and selecting one partition from the five QT/MTT partitions.

3. Proposed Method

Except for the objective pixel distortion, the perceptible interval for gray level variation also affects the decoded video quality. Finding the luminance variation on the regions truly sensitive to human eyes is more feasible than the gradient or texture measurement [14]. People may just distinguish the change once the difference on the gray level between the current pixel and background exceeds the visible threshold. A just-noticeable-distortion (JND) can be utilized to quantify perceptual redundancy [32]. This paper applies the factor of average background luminance for JND [32] to detect the VD pixels within a CU. The average background luminance A(i, j) is computed by 3×3 average filtering for efficient calculation in this paper, as shown in (1). The I(i, j) represents the gray level of the current pixel with the coordinate (i, j). The visible thresholds [32] due to background luminance JND(i, j) are listed in (2), in which T_0 and γ are 17 and 3/128, respectively. VD(i, j) indicates whether or not the current pixel, with the coordinate (i, j), is the VD pixel according to (3), in which VD(i, j) is gerater than or equal to JND(i, j).

$$A(i,j) = \begin{bmatrix} I(i-1,j-1) & I(i,j-1) & I(i+1,j-1) \\ I(i-1,j) & I(i,j) & I(i+1,j) \\ I(i-1,j+1) & I(i,j+1) & I(i+1,j+1) \end{bmatrix} \times \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$$
(1)

$$JND(i,j) = \begin{cases} T_0 \times \left(1 - \left(\frac{A(i,j)}{127}\right)^{\frac{1}{2}}\right) + 3, \text{ for } A(i,j) \le 127\\ \gamma \times (A(i,j) - 127) + 3, \text{ for } A(i,j) > 127 \end{cases}$$
(2)

$$VD(i,j) = \begin{cases} 1, if |I(i,j) - A(i,j)| \ge JND(i,j) \\ 0, otherwise \end{cases}$$
(3)

BTH, BTV, TTH, and TTV are the four kinds of MTT partitions in the QT-MTT structure. The MTT partition is allowed for the CUs no larger than 32×32 , which corresponds to QtDepth 2 under the All Intra configuration. As a foundation for the fast algorithm, we propose calculating the variances of the numbers of VD pixels in various MTT splits of a CU. We compute *varBTH*, *varBTV*, *varTTH*, and *varTTV* for the four kinds of MTT splits in the square CUs (MttDepth 0), for the 32×32 CU (QtDepth 2) and four child 16×16 CUs, as shown in (4)–(7). $VD_{sub_{CU_k}}$ represents the number of VD pixels in each sub-CU for the four kinds of MTT splits. In particular, the $VD_{sub_{CU_k}}$ of the middle sub-CU of the TT split is further divided by 2 for normalization. The μ_{BTH} , μ_{BTV} , μ_{TTH} , and μ_{TTV} represent the mean values of all $VD_{sub_{CU_k}}$ for each kind of MTT splits. The individual variances, *varBTH*, *varBTV*, *varTTH*, and *varTTV* of the four MTT splits, are calculated, and μ_{var} is the average of the above four variances, as shown in (8). The *varMTT*, as in (9), denotes the variance of *varBTH*, *varBTV*, *varTTH*, and *varTTV*.

$$varBTH = \frac{1}{2} \sum_{k=1}^{2} \left(VD_{sub_{CU_k}} - \mu_{BTH} \right)^2$$
(4)

$$varBTV = \frac{1}{2} \sum_{k=1}^{2} \left(VD_{sub_{CU_k}} - \mu_{BTV} \right)^2$$
(5)

$$varTTH = \frac{1}{3} \sum_{k=1}^{3} \left(VD_{sub_{CU_k}} - \mu_{TTH} \right)^2$$
(6)

$$varTTV = \frac{1}{3} \sum_{k=1}^{3} \left(VD_{sub_{CU_k}} - \mu_{TTV} \right)^2$$
(7)

$$\mu_{var} = \frac{varBTH + varBTV + varTTH + varTTV}{4} \tag{8}$$

$$varMTT = \frac{(varBTH - \mu_{var})^2 + (varBTV - \mu_{var})^2 + (varTTH - \mu_{var})^2 + (varTTV - \mu_{var})^2}{4}$$
(9)

The sets of variances for the parent 32×32 CU (QtDepth 2) and four child 16×16 CUs are indicated in (10) and (11), respectively.

$$varSet_{32} = \{varBTH_{32}, varBTV_{32}, varTTH_{32}, varTTV_{32}\}$$
(10)

$$varSet_{16} = \{varBTH_{16,s}, varBTV_{16,s}, varTTH_{16,s}, varTTV_{16,s} | s = 1, 2, 3, 4\}$$
(11)

3.1. Fast Decision for Intra Coding Tools

The intra coding tools in H.266/VVC can ameliorate the compression efficiency, but increase the coding time. Six video sequences, including Class A1 Tango2 (3840×2160), Class A2 ParkRunning3 (3840×2160), Class B BQTerrace (1920×1080), Class C PartyScene (832×480), Class D RaceHorses (416×240), and Class E KristenAndSara (1280×720), are used to observe the contribution of the intra coding tools of ISP and MIP. The statistics of the probabilities, and the performance of Bjøntegaard Delta Bitrate (BDBR) [33] and Time Saving (TS) using ISP and MIP for each MttDepth in VTM10.0 [2,3] and All Intra configuration are shown in Tables 1 and 2, respectively. The definition of TS is shown in (12). The probability distributions show that the usage of the two coding tools decreases as the MttDepth increases. The BDBR is insignificant when the MttDepth is 3. From the observations, we assume that the ISP and MIP can be turned off when the MttDepth is larger than 2.

Time Saving(TS) =
$$\frac{1}{4} \sum_{QP \in \{22, 27, 32, 37\}} \frac{T_{VTM10.0}(QP) - T_{proposed}(QP)}{T_{VTM10.0}(QP)} \times 100\%$$
 (12)

MttDepth	ISP	MIP	Average
0	67.18%	77.16%	72.17%
1	38.32%	42.78%	40.55%
2	27.33%	30.13%	28.73%
3	16.89%	10.88%	13.89%

Table 1. Probability distributions of ISP and MIP in each MttDepth.

Table 2. Performance of turning off ISP and MIP in each MttDepth.

MttDepth	BDBR (%)	TS (%)
0	1.21	12.70
1	0.77	8.48
2	0.54	5.14
3	0.12	4.32

When the QtDepth is 2 or 3, and the MttDepth of CU is larger than 2, the ISP and MIP are turned off conditionally by our method. The decision to turn off the ISP and MIP tools is determined by the criterion exhibited in (13). The threshold, Th_{Tools} , related to the quantization parameter is defined as in (14). The choice of α in (14) is particularly important, and selecting a larger α will lead to an increase in BDBR. Figure 3 shows the relationship between α , and the encoding time ratio and BDBR by the average of the experiments, using the six sequences mentioned above. It can be found that the overall encoding time and BDBR have relatively moderate changes when α is in the range of 0.7 to 1.5. When α exceeds 1.5, the BDBR begins to increase dramatically. Therefore, the final parameter α is chosen as 1.5.

$$varMTT \le Th_{Tools} \tag{13}$$



Figure 3. Selection of α in *Th*_{*Tools*}.

3.2. Fast Decision for Multi-Type Tree Partition

The proposed algorithm adopts the random forest (RF) model [34] in machine learning for the fast decision of the MTT partition. In the RF model, multiple decision trees are constructed using a random sampling method, and the prediction is made by voting. Compared to a single decision tree model, the RF model has stronger generalization ability and a lower overfitting risk. Two binary classifiers using RF are trained individually for the BT split and TT split to quickly determine the split direction in the 32 × 32 CUs corresponding to QtDepth 2. The input features consist of *varSet*₃₂, *varSet*₁₆, and QP. A total of 21 features are the input to the two RF classifiers, as shown in Figure 4. The horizontal or vertical split direction is the output of the binary classifier.

The video frames in the training dataset consist of two parts and are different from the test video sequences in our experiments. The training dataset for the first part contains still images of four different resolutions (4928×3264 , 2880×1920 , 1536×1024 , 768×512) from the CPIH dataset [35]. For the second part of the training dataset, four video sequences with a 4 K (3840×2160) resolution are selected from the Ultra Video Group (UVG) dataset [36]. In the UVG dataset, Jockey, FlowerFocus, YachtRide, and RiverBank video sequences are used. The RF classifiers are trained by a total of 177,894 CUs for the horizontal/vertical classifier of BT and a total of 141,654 CUs for the horizontal/vertical classifier of TT. We use OpenCV 4.5.4 [37] to train the RF models. The overall training dataset is divided into a 75% training set and a 25% validation set. In addition, we adjust the hyperparameters of the random forest model and select 35 decision trees in the BT classifier and 25 decision trees in the TT classifier.



Figure 4. Illustration of input features for the horizontal/vertical random forest classifier.

3.3. Overall Algorithm

The overall algorithm in this paper mainly consists of two strategies based on visual perception information. The first strategy is the fast algorithm for turning off the ISP and MIP intra picture prediction tools, based on the condition decided by the variance when the QtDepth is 2 or 3 and the MttDepth of CU is larger than 2. The second strategy is to use the information of visual perception and QP to propose horizontal/vertical split models trained by the random forest classifiers of machine learning, to select the splitting direction for BT and TT individually when the QtDepth is 2. In addition, if both *varTTH* and *varTTV*, calculated by (6) and (7), are 0 for the case of 32×32 CU (QtDepth 2) or 16×16 CU (QtDepth 3), the subsequent TT splits are terminated in the corresponding QtDepth. The flowchart of the overall algorithm is shown in Figure 5.



Figure 5. The flowchart of the overall fast algorithm.

4. Results

We implement the H.266/VVC reference software version VTM 10.0 [2,3] in our experiments to show the performance under the All Intra configuration, with the QPs of 22, 27, 32, and 37. Class A1, Class A2, Class B, Class C, Class D, Class E, and Class F are included in the test video sequences. The experimental setup follows the common test conditions (CTC) [38]. The information of the test sequences is shown in Table 3. The experiments are executed on Windows 10 64 bit with an Intel i7-7700 K processor @4.20 GHz and a 64 GB RAM. Bjøntegaard Delta Bitrate (BDBR), Bjøntegaard Delta Peak Signal-to-Noise Ratio (BDPSNR), Time Saving (TS), and TS/BDBR are used to evaluate the performance and compare with [16]. The tradeoff between coding efficiency and computation time is evaluated by TS/BDBR, which is computed by the average TS divided by the average BDBR. The greater the ratio, the better the performance.

Class	Resolution	Sequence	Frame Rate (fps)	Bit Depth	Frame Count
A1	3840×2160	Campfire	30	10	300
	3840×2160	FoodMarket4	60	10	300
	3840 × 2160	Tango2	60	10	294
	3840×2160	CatRobot	60	10	300
A2	3840×2160	DaylightRoad2	60	10	300
	3840×2160	ParkRunning3	50	10	300
	1920 imes 1080	BasketballDrive	50	8	500
	1920 imes 1080	BQTerrace	60	8	600
В	1920×1080	Cactus	50	8	500
	1920×1080	MarketPlace	60	10	600
	1920 imes 1080	RitualDance	60	10	600
	832×480	BasketballDrill	50	8	500
	832×480	BQMall	60	8	600
C	832×480	PartyScene	50	8	500
	832×480	RaceHorses	30	8	300
	416×240	BasketballPass	50	8	500
D	416×240	BlowingBubbles	50	8	500
D	416×240	BQSquare	60	8	600
	416×240	RaceHorses	30	8	300
E	1280×720	FourPeople	60	8	600
	1280×720	Johnny	60	8	600
	1280×720	KristenAndSara	60	8	600
	1920 imes 1080	ArenaOfValor	60	8	600
_	832 × 480	BasketballDrillTe	xt 50	8	500
F	1280×720	SlideEditing	30	8	300
	1280×720	SlideShow	20	8	500

Table 3. Information of the test sequences.

Table 4 lists the experimental results of the proposed algorithm for all test sequences. The average BDBR, BDPSNR, and TS are 1.535%, -0.0814 dB, and 47.26%, respectively. For comparison with [16], the proposed algorithm results in only 1.454% BDBR on average, lower than the 2.71% in [16]. The average BDBR of each Class sequence for the proposed

algorithm is lower than that in [16]. The average BDBRs in Class C and Class D are less than half of those in [16], and the average TSs are also well maintained. Furthermore, our method achieves a TS/BDBR of 32.68, which is better than the 23.31 of [16].

		[16]	Proposed		
Class	Sequence	BDBR (%)	TS (%)	BDPSNR (dB)	BDBR (%)	TS (%)
	Campfire	2.65	64.74	-0.0361	1.165	52.48
Λ 1	FoodMarket4	1.47	46.93	-0.0593	1.806	50.08
AI	Tango2	2.42	64.45	-0.0266	1.713	50.73
	Average	2.18	58.71	-0.0407	1.561	51.10
	CatRobot	3.27	63.81	-0.0492	1.696	52.68
4.2	DaylightRoad2	2.02	70.39	-0.0528	1.249	53.11
AZ	ParkRunning3	1.46	55.14	-0.0559	0.955	59.75
	Average	2.25	63.11	-0.0526	1.300	55.18
	BasketballDrive	2.38	67.81	-0.0605	2.209	48.57
	BQTerrace	2.43	64.25	-0.0526	1.235	40.90
р	Cactus	2.78	66.61	-0.0540	1.540	47.86
D	MarketPlace	2.58	71.93	-0.0641	1.586	56.74
	RitualDance	4.21	64.06	-0.0938	1.868	52.15
	Average	2.88	66.93	-0.0650	1.688	49.24
	BasketballDrill	5.39	65.29	-0.0885	1.842	44.15
	BQMall	2.92	62.93	-0.0887	1.656	44.80
С	PartyScene	1.40	58.77	-0.0394	0.557	39.94
	RaceHorses	2.00	62.10	-0.0610	1.073	46.97
	Average	2.93	62.27	-0.0694	1.282	43.97
	BasketballPass	2.34	61.15	-0.0984	1.613	45.00
	BlowingBubbles	2.24	59.94	-0.0317	0.476	38.18
D	BQSquare	1.68	59.98	-0.0320	0.409	35.90
	RaceHorses	1.69	58.98	-0.0723	1.126	46.26
	Average	1.99	60.01	-0.0586	0.906	41.34
	FourPeople	4.36	67.14	-0.1028	1.875	45.63
Б	Johnny	4.34	67.01	-0.0952	2.485	47.56
E	KristenAndSara	3.56	66.21	-0.0877	1.844	45.69
-	Average	4.09	66.79	-0.0952	2.068	46.29
	ArenaOfValor	-	-	-0.1059	1.809	46.49
	BasketballDrillText	-	-	-0.0999	1.802	42.99
F	SlideEditing	_	-	-0.3264	2.251	49.60
-	SlideShow	-	-	-0.1821	2.057	44.50
	Average	_	-	-0.1786	1.980	45.90

Table 4. Performance comparison between the proposed algorithm and [16].

Table 4. Cont.

Class		[16]		Proposed		
	Sequence	BDBR (%)	TS (%)	BDPSNR (dB)	BDBR (%)	TS (%)
Total Average		-	-	-0.0814	1.535	47.26
TS/BDBR		-		30.79		
Same Sequence Average		2.71	63.16	-0.0638 1.454		47.51
Same Sequence TS/BDBR		23.31		32.68		

Table 5 lists the performance comparison between the proposed algorithm and [31], in which only Class B, C, D, and E sequences are available. The testing of [31] was conducted using VTM 7.0, as indicated. The coding tools of the ISP, MIP and low-frequency non-separable transform (LFNST) are turned off for the experiments in [31]. Although LFNST is deactivated, the ISP and MIP are conditionally turned off by the proposed criterion in our experiments for comparison with [31], as shown in Table 5. The number of encoded frames is 100, which is the same as that in [31]. The performance of the proposed algorithm has the BDBRs of 1.49% and 2.12% in Class B and E sequences, respectively, which is better than 2.07% and 2.90% in [31]. The TS/BDBR of the proposed algorithm is 33.03, which is better than the 29.66 of [31].

Class	Sequence	[31]		Proposed	
Class		BDBR (%)	TS (%)	BDBR (%)	TS (%)
	BasketballDrive	3.28	59.35	2.14	52.17
P	BQTerrace	1.08	45.30	0.77	42.78
D	Cactus	1.84	52.44	1.57	50.12
	Average	2.07	52.36	1.49	48.36
	BasketballDrill	1.82	48.48	1.93	48.35
	BQMall	1.87	52.47	1.59	48.27
С	PartyScene	0.26	38.62	0.42	42.18
	RaceHorses	0.88	49.05	1.16	50.17
	Average	1.21	47.16	1.28	47.24
	BasketballPass	1.95	47.70	1.91	47.78
	BlowingBubbles	0.47	40.35	0.67	43.81
D	BQSquare	0.19	31.95	0.16	33.14
	RaceHorses	0.54	41.69	1.19	47.37
	Average	0.79	40.42	0.98	43.03
	FourPeople	2.70	57.57	1.96	49.31
E	Johnny	3.22	56.88	2.38	51.24
	KristenAndSara	2.78	55.11	2.02	49.87
	Average	2.90	56.52	2.12	50.14
Total A	Total Average		48.35	1.42	46.90
TS/BDBR		29.	66	33.0)3

Table 5. Performance comparison between the proposed algorithm and [31].

Figure 6 compares the rate-distortion curve (RD curve) between the proposed algorithm and VTM10.0 for Class A2 Parkrunning3 sequences. The curve of the proposed method is close to that of VTM10.0, and the TS of the proposed method reaches 59.75%. Figure 7 demonstrates the QT/MTT partition results of the frame in the class D BQSquare sequence. The partition results are similar regarding both the complex blocks and uniform blocks.



Figure 6. RD curve comparison between the proposed algorithm and VTM10.0 for Class A2 Parkrunning3 sequence.



Figure 7. QT/MTT partition results for the frame in Class D BQSquare video sequence (QP 37).

5. Conclusions

In this paper, we present the fast and precise decisions for intra coding tools and MTT partitions based on the variances of the numbers of VD pixels among the MTT splits for the intra coding of H.266/VVC. According to the proposed criterion, conditionally turning off the ISP and MIP intra coding tools can speed up the encoding process. RF classifiers of machine learning based on the visual perception analysis effectively skip unnecessary split directions for the BT and the TT of the nested MTT partition. Our experimental results demonstrate that the proposed algorithm provides better coding efficiency and performance than the previous studies. Conventional methods only consider fast decisions either for coding tools or QT/MTT partitions. Previous studies regarding the H.266/VVC fast coding algorithm rarely utilize perceptual redundancy. Our proposed method takes advantage of visual perception to terminate the processes of both aspects of the coding tools and MTT partitions in low-delay and random access configurations in future works.

Author Contributions: Conceptualization, Y.-H.T. and M.-J.C.; Data curation, Y.-H.T.; Funding acquisition, M.-J.C. and C.-H.Y.; Methodology, Y.-H.T. and M.-J.C.; Software, Y.-H.T.; Supervision, M.-J.C., C.-M.Y. and C.-H.Y.; Validation, C.-R.L. and M.-C.H.; Writing—original draft, Y.-H.T., C.-R.L. and M.-J.C.; Writing—review and editing, M.-C.H., C.-M.Y. and C.-H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Science and Technology, Taiwan, under grant MOST 109-2221-E-259-012-MY3.

Data Availability Statement: Publicly available datasets were analyzed in this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Bross, B.; Chen, J.; Ohm, J.R.; Sullivan, G.J.; Wang, Y.K. Developments in international video coding standardization after AVC, with an overview of versatile video coding (VVC). *Proc. IEEE* **2021**, *109*, 1463–1493. [CrossRef]
- Chen, J.; Ye, Y.; Kim, S.H. Algorithm Description for Versatile Video Coding and Test Model 10 (VTM 10); doc. JVET-S2002; Joint Video Experts Team: Teleconference, 2020.
- VVC Reference Software. Available online: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-10.0 (accessed on 13 August 2020).
- 4. Bross, B.; Wang, Y.K.; Ye, Y.; Liu, S.; Chen, J.; Sullivan, G.J.; Ohm, J.R. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 3736–3764. [CrossRef]
- Hamidouche, W.; Biatek, T.; Abdoli, M.; François, E.; Pescador, F.; Radosavljević, M.; Menard, D.; Raulet, M. Versatile video coding standard: A review from coding tools to consumers deployment. *IEEE Consum. Electron. Mag.* 2022, 11, 10–24. [CrossRef]
- 6. De-Luxán-Hernández, S.; George, V.; Ma, J.; Nguyen, T.; Schwarz, H.; Marpe, D.; Wiegand, T. *CE3: Intra Sub-Partitions Coding Mode (Tests 1.1.1 and 1.1.2)*; doc. JVET-M0102; Joint Video Experts Team: Marrakech, Morocco, 2019.
- Pfaff, J.; Stallenberger, B.; Schäfer, M.; Merkle, P.; Helle, P.; Hinz, T.; Schwarz, H.; Marpe, D.; Wiegand, T. CE3: Affine Linear Weighted Intra Prediction (CE3-4.1, CE3-4.2); doc. JVET-N0217; Joint Video Experts Team: Geneva, Switzerland, 2019.
- 8. Bross, B.; Keydel, P.; Schwarz, H.; Marpe, D.; Wiegand, T.; Zhao, L.; Zhao, X.; Li, X.; Liu, S.; Chang, Y.J.; et al. *CE3: Multiple Reference Line Intra Prediction (Test 1.1.1, 1.1.2, 1.1.3 and 1.1.4)*; doc. JVET-L0283; Joint Video Experts Team: Macao, China, 2018.
- 9. Van der Auwera, G.; Seregin, V.; Said, A.; Ramasubramonian, A.K.; Karczewicz, M. CE3: Simplified PDPC (Test 2.4.1); doc. JVET-K0063; Joint Video Experts Team: Ljubljana, Slovenia, 2018.
- 10. Fu, T.; Zhang, H.; Mu, F.; Chen, H. Fast CU partitioning algorithm for H.266/VVC intra-frame coding. In Proceedings of the IEEE International Conference on Multimedia and Expo, Shanghai, China, 8–12 July 2019.
- 11. Park, S.-h.; Kang, J.W. Context-based ternary tree decision method in versatile video coding for fast intra-coding. *IEEE Access* **2019**, *7*, 172597–172605. [CrossRef]
- 12. Zhang, Q.; Zhao, Y.; Jiang, B.; Wu, Q. Fast CU partition decision method based on Bayes and improved de-blocking filter for H.266/VVC. *IEEE Access* 2021, *9*, 70382–70391. [CrossRef]
- 13. Yang, H.; Shen, L.; Dong, X.; Ding, Q.; An, P.; Jiang, G. Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 1668–1682. [CrossRef]
- 14. Chen, M.J.; Lee, C.A.; Tsai, Y.H.; Yang, C.M.; Yeh, C.H.; Kau, L.J.; Chang, C.Y. Efficient partition decision based on visual perception and machine learning for H.266/versatile video coding. *IEEE Access* **2022**, *10*, 42141–42150. [CrossRef]
- 15. Li, T.; Xu, M.; Tang, R.; Chen, Y.; Xing, Q. DeepQTMT: A deep learning approach for fast QTMT-based CU partition of intra-mode VVC. *IEEE Trans. Image Process* 2021, *30*, 5377–5390. [CrossRef] [PubMed]
- 16. Wu, G.; Huang, Y.; Zhu, C.; Song, L.; Zhang, W. SVM based fast CU partitioning algorithm for VVC intra coding. In Proceedings of the 2021 IEEE International Symposium on Circuits and Systems, Daegu, Korea, 22–28 May 2021.
- 17. Wang, Y.; Dai, P.; Zhao, J.; Zhang, Q. Fast CU partition decision algorithm for VVC intra coding using an MET-CNN. *Electronics* **2022**, *11*, 3090. [CrossRef]
- Zouidi, N.; Kessentini, A.; Hamidouche, W.; Masmoudi, N.; Menard, D. Multitask learning based intra-mode decision framework for versatile video coding. *Electronics* 2022, 11, 4001. [CrossRef]
- 19. Taabane, I.; Menard, D.; Mansouri, A.; Ahaitouf, A. Machine learning based fast QTMTT partitioning strategy for VVenC encoder in intra coding. *Electronics* **2023**, *12*, 1338. [CrossRef]
- 20. Park, S.h.; Kang, J.W. Fast multi-type tree partitioning for versatile video coding using a lightweight neural network. *IEEE Trans. Multimed.* **2021**, *23*, 4388–4399. [CrossRef]
- 21. Zhao, J.; Dai, P.; Zhang, Q. A complexity reduction method for VVC intra prediction based on statistical analysis and SAE-CNN. *Electronics* **2021**, *10*, 3112. [CrossRef]
- 22. Zhao, T.; Huang, Y.; Feng, W.; Xu, Y.; Kwong, S. Efficient VVC intra prediction based on deep feature fusion and probability estimation. *IEEE Trans. Multimed.* 2022. *Early Access.* [CrossRef]
- 23. Peng, S.; Peng, Z.; Ren, Y.; Chen, F. Fast intra-frame coding algorithm for versatile video coding based on texture feature. In Proceedings of the 2019 IEEE International Conference on Real-time Computing and Robotics, Irkutsk, Russia, 4–9 August 2019.
- 24. Zhang, H.; Yu, L.; Li, T.; Wang, H. Fast GLCM-based intra block partition for VVC. In Proceedings of the 2021 Data Compression Conference (DCC), Snowbird, UT, USA, 23–26 March 2021; p. 382.
- 25. Shu, Z.; Li, J.; Peng, Z.; Chen, F.; Yu, M. Intra complexity control algorithm for VVC. Electronics 2022, 11, 2572. [CrossRef]
- 26. Zhang, Q.; Zhao, Y.; Jiang, B.; Huang, L.; Wei, T. Fast CU partition decision method based on texture characteristics for H.266/VVC. *IEEE Access* 2020, *8*, 203516–203524. [CrossRef]
- Cui, J.; Zhang, T.; Gu, C.; Zhang, X.; Ma, S. Gradient-based early termination of CU partition in VVC intra coding. In Proceedings
 of the 2020 Data Compression Conference, Snowbird, UT, USA, 24–27 March 2020.

- Gou, A.; Sun, H.; Katto, J.; Li, T.; Zeng, X.; Fan, Y. Fast intra mode decision for VVC based on histogram of oriented gradient. In Proceedings of the 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 27 May–1 June 2022; pp. 3028–3032.
- 29. Yoon, Y.U.; Kim, J.G. Activity-based block partitioning decision method for versatile video coding. *Electronics* 2022, *11*, 1061. [CrossRef]
- 30. Jing, Z.; Li, P.; Zhao, J.; Zhang, Q. A fast CU partition algorithm based on gradient structural similarity and texture features. *Symmetry* **2022**, *14*, 2644. [CrossRef]
- Fan, Y.; Chen, J.A.; Sun, H.; Katto, J.; Jing, M.E. A fast QTMT partition decision strategy for VVC intra prediction. *IEEE Access*. 2020, *8*, 107900–107911. [CrossRef]
- 32. Chou, C.H.; Li, Y.C. A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile. *IEEE Trans. Circuits Syst. Video Technol.* **1995**, *5*, 467–476. [CrossRef]
- 33. Bjøntegaard, G. Calculation of Average PSNR Differences between RD-Curves; doc. VCEG-M33; ITU-T Video Coding Experts Group: Austin, TX, USA, 2001.
- 34. Breiman, L. Random forests. Mach. Learn. 2001, 45, 5–32. [CrossRef]
- Li, T.; Xu, M.; Deng, X. A deep convolutional neural network approach for complexity reduction on intra-mode HEVC. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo, Hong Kong, China, 10–14 July 2017; pp. 1255–1260.
- Mercat, A.; Viitanen, M.; Vanne, J. UVG dataset: 50/120fps 4K sequences for video codec analysis and development. In Proceedings of the 11th ACM Multimedia Systems Conference, New York, NY, USA, 27 May 2020; pp. 297–302.
- 37. Bradski, G. The OpenCV Library. Dr. Dobb's J. Softw. Tools. 2000, 25, 120–125.
- Bossen, F.; Boyce, J.; Suehring, K.; Li, X.; Seregin, V. JVET Common Test Conditions and Software Reference Configurations for SDR Video; doc. JVET-N1010; Joint Video Experts Team: Geneva, Switzerland, 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.