Hindawi
Scientific Programming
Volume 2021, Article ID 3813116, 10 pages
https://doi.org/10.1155/2021/3813116

Research Article

Fast CU Size Decision Method Based on Just Noticeable Distortion and Deep Learning

Jinchao Zhao, Yihan Wang, and Qiuwen Zhang

College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

Correspondence should be addressed to Qiuwen Zhang; zhangqwen@126.com

Received 9 June 2021; Accepted 24 November 2021; Published 8 December 2021

Academic Editor: Roberto Natella

With the development of broadband networks and high-definition displays, people have higher expectations for the quality of video images, which also brings new requirements and challenges to video coding technology. Compared with H.265/High Efficiency Video Coding (HEVC), the latest video coding standard, Versatile Video Coding (VVC), can save 50%-bit rate while maintaining the same subjective quality, but it leads to extremely high encoding complexity. To decrease the complexity, a fast coding unit (CU) size decision method based on Just Noticeable Distortion (JND) and deep learning is proposed in this paper. Specifically, the hybrid JND threshold model is first designed to distinguish smooth, normal, or complex region. Then, if CU belongs to complex area, the Ultra-Spherical SVM (US-SVM) classifiers are trained for forecasting the best splitting mode. Experimental results illustrate that the proposed method can save about 52.35% coding runtime, which can realize a trade-off between the reduction of computational burden and coding efficiency compared with the latest methods.

1. Introduction

With the increasing requirements for high quality videos, the Joint Video Experts Team (JVET) developed a new generation of video coding standards based on H.265/High Efficiency Video Coding (HEVC), namely Versatile Video Coding (VVC). How to reduce the coding complexity and save coding time on the basis of ensuring the quality of video coding has become a hot issue in the current video coding field.

Since the highly dense data brings huge challenges to bandwidth and storage, and the previous generation video coding standards are insufficient to fulfill the compression capacity of the future market, therefore, the VVC has appeared. The VVC that has good network adaptability, parallel processing capability, and compression efficiency is the latest video coding standard after HEVC. In addition, the VVC is formulated for 4K/8K video, and a bit-depth is 10-bit. The VVC extends the original partition structure, intra/interprediction, filtering, transformation, quantization/scaling, and entropy coding of the HEVC. Moreover, considering the characteristics of VVC, new prediction techniques are added, such as the quad-tree with nested multitype tree (QTMT) structure and other coding tools [1]. The usage of many advanced coding techniques greatly improves the compression efficiency of VVC but leads to a significant increase in the coding computational complexity [2, 3]. For example, the computational complexity of VTM is 19 times that of HM in "all-intra" configuration [4]. Therefore, how to reduce the complexity and efficiently compress large amounts of data has become an important issue in the practical application of VVC.

The partition structure for VVC and HEVC is a significant distinction. Since quad-tree (QT) partition structure is only allowed in HEVC, the width and height of coding unit (CUs) are equal (width and height = 64, 32, 16, or 8); it means that CUs shape can only be square [5]. Furthermore, one of the linchpin characteristics in HEVC is the concept of multiple partitions, including CU, prediction unit (PU), and transform unit (TU). The multitype tree (MTT) architecture allows the asymmetric partition in VVC, which contains binary tree (BT) and ternary tree (TT) splitting. Theoretically, CU size can be any combination of 128, 64, 32, 16, 8, or 4. Additionally, due to the introduction of intra-subpartition

(ISP) technique, even smaller length (2 or 1) exists in VVC. Thus, the MTT partition structure supports more CU sizes than the QT partition structure, thereby obtaining more efficient coding performance. It is worth noting that the allowed CU sizes may be inconsistent with the theory in practical applications. The CU shape has square or rectangular shape in the coding tree structure of VVC. The coding tree units (CTUs) are first used in the QT partition structure to generate four QT leaf nodes. Then, the leaf nodes of QT are further divided by the MTT partition structure. Figure 1 shows four types of splitting in the MTT structure including horizontal BT (BT_H), vertical BT (BT_V), horizontal TT (TT_H), and vertical TT (TT_V). In most cases, the CU, PU, and TU have the same size in the QTMT partition structure. The exception occurs when the maximum supported transform size is smaller than the width or height of the color component of the CU.

The remainder of the paper is organized as follows: the related works are introduced in Section 2. Section 3 describes the proposed method. Section 4 provides the experimental results and analysis. Finally, Section 5 concludes this paper.

2. Related Works

At present, the research on the VVC for coding mainly focuses on the intraprediction, which is manifested in the early termination of the CU division. To solve this problem, many works are proposed for fast CU size decision. An early determination method for VVC is proposed in [6], which can skip redundant MTT pruning and effectively reduce TT complexity, where the TT characteristics are defined in the VVC encoding context. A fast CU splitting approach is developed in [7], which can implement early termination. Chen et al. design a novel fast CU splitting method for the performances, which can balance the performance and complexity [8]. A deep Convolutional Neural Networks (CNN) model-based fast QT partition method is developed in [9] to forecast CU splitting mode, which considerably enhances performance. A fast MTT decision method is designed in [10], which can decrease computational complexity and maintain compression performance. Specifically, the splitting decision mode can be early decided by comparing the pixel difference of subblocks (SBPD) in horizontal and vertical subblocks so as to skip some redundant splitting modes. A fast method is devised based on spatial features in [11], where spatial features are used for early termination. In [12], a novel fast CU partition method is developed based on Bayesian to encounter huge computational burden. Chen et al. present a fast intrapartition method based on variance and gradient for decreasing coding complexity [13]. In [14], an adaptive CU splitting method based on the pooling-variable CNN is presented to decrease the coding time. A lightweight and tunable quad-tree plus binary tree (QTBT) structure method is developed based on Machine Learning (ML) in [15] to decrease the coding complexity. To reduce coding complexity, Dong et al. propose a fast method for VVC including mode selection and prediction terminating, which can decrease coding runtime [16]. Yang et al. introduce a fast intramethod including the low-complexity coding tree units (CTU) structure decision and fast intramode decision with gradient descent search [17]. In our previous works [18, 19], we proposed fast intramethod based on random forest classier and Directed Acyclic Graph Support Vector Machine (DAG-SVM) to reduce the complexity while maintaining the coding efficiency. Although the above methods can reduce the coding time, these methods do not consider the impact of human visual characteristics in the encoding procedure.

The purpose of visual perception coding is to use human visual system (HVS) characteristics to eliminate information that the human eye cannot perceive as much as possible and to provide better visual perception quality with fewer bit resources. To this end, researchers have proposed a large number of visual perception coding methods. A novel discrete cosine transform-based energy-reduced JND model (ERJND) is presented in [20] for perceptual video coding. Reference [21] designs a JND-based perceptual rate control method for HEVC, which can achieve significantly improved coding performance. In [22], a JND compensation based perceptual video coding (PVC) method is designed to compress videos with better perceptual quality. A deep-learning-based Picture-Wise Just Noticeable Difference (PW-JND) prediction model is developed in [23] for image compression. A PVC method with visual saliency modulated JND model is introduced in [24]. Recently, Shen et al. have introduced a JND guided perceptually lossless coding framework [25]. Shen et al. devise an effective method to infer the JND profile based on patch-level structural visibility learning [26].

Traditional methods mainly use techniques such as intra- and interprediction and entropy coding to eliminate redundant information to improve the rate-distortion (RD) performance [27]. However, traditional video coding methods do not fully consider the characteristics of the HVS. Therefore, how to effectively use the HVS to optimize the existing coding method has important theoretical significance and application value. A fast CU partitioning method based on visual perception characteristics is designed in this paper, which comprehensively considers multiple visual factors, such as visual mode complexity, spatial contrast masking (CM), luminance adaptation, and the disordered concealment effect of the free energy principle, and forms a hybrid JND model based on these visual characteristics. Then, based on the hybrid JND model, the CUs are separated into smooth, ordinary, and complex area. If CU is divided into complex area, the CUs utilize the trained Ultra-Spherical SVM (US-SVM) US-SVM classifier to decide the best splitting mode. Only $K - 1$ US-SVMs need to be trained for VVC, thereby eliminating regions belonging to multiple categories in the decision and reducing the total number of training samples accordingly. Finally, the proposed method can reduce the coding time and complexity.

3. The Method Based on JND and Deep Learning

3.1. Hybrid JND Model. The JND model is usually utilized to depict the minimum perceptible distortion threshold of HVS, and it is an effective measure of visual redundancy of

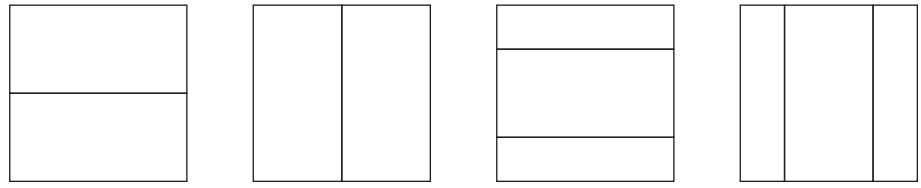


Figure 1: The partition modes of MTT structure.

video images. Generally, the following three visual factors are mainly considered when modeling the JND threshold, including luminance adaptation, that is, the masking of HVS due to changes in background brightness; CM, that is, the masking of HVS due to spatial inhomogeneity; and contrast sensitivity function, that is, the masking of HVS due to frequency changes. On the basis of related literature, comprehensive consideration of visual factors such as spatial masking effect based on visual pattern complexity and CM, luminance adaptation, and disordered concealment effect based on the principle of free energy, a hybrid JND threshold model is designed to better describe the visual distortion sensitivity of the image area.

It has been pointed out in the literature that the HVS tends to extract repeated visual content for analysis and comprehension, and it will also encode corresponding visual patterns for the content forecast as an important feature of image scene content [28]. In the regular pattern, the image scene provides a simple and intuitive element organization and arrangement mode, and its visual interaction with the human eye is relatively direct; the visual masking effect is weak at this time. In the irregular pattern, the corresponding arrangement mode and the visual interaction of the human eye are relatively complicated, and a strong visual masking effect is involved at this time [29]. It can be seen that the effective measure of the visual masking effect is the complexity of the visual pattern. Figure 2 shows an example of pattern complexity.

In the primary visual cortex, HVS shows obvious direction selectivity and the extracted direction information will be used to represent the image structure. Therefore, the direction information drawn from the image content is utilized to describe the complexity of the image mode. Studies have shown that HVS has obvious directional selectivity in the primary visual cortex. The extracted direction information will be used to represent the image structure and can describe the complexity of the image mode. Generally speaking, the complex mode usually contains a richer directional distribution, while the simple mode only contains a limited single directional distribution. Accordingly, the distribution of the gradient direction in the local image area [30] is used to indicate the corresponding visual pattern complexity (PC):

$$PC(x, y) = \sum_{k=1}^{N} \left\| H_k(x, y) \right\|_0, \tag{1}$$

where N refers to the quantized value range, H is the distribution histogram in the 5×5 neighborhood around (x, y), and $\| \cdot \|_0$ denotes the L_0 norm that is mainly used to measure the number of nonzero elements in a vector.

According to the above content, the complexity of the visual pattern of the image area can be used as an effective measure of the visual masking effect. Therefore, the combined CM effect can be used to better describe the degree of visual masking in space. Here, the spatial perception factor based on CM is used as a supplement to the complexity of the abovementioned visual mode:

$$CM(x, y) = \sqrt{\left|\mathrm{Grad}_h(x, y)\right|^2 + \left|\mathrm{Grad}_v(x, y)\right|^2}, \tag{2}$$

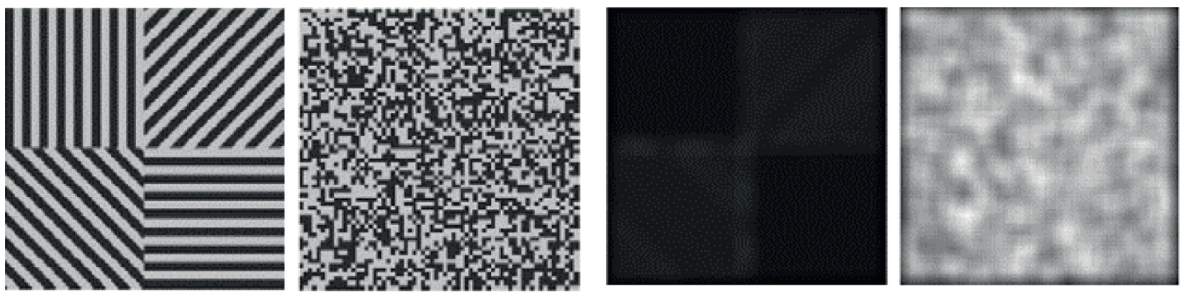
where $\mathrm{Grad}_h(x, y)$ is the gradient value of the point (x, y) in the horizontal direction and $\mathrm{Grad}_v(x, y)$ is the gradient value of the point (x, y) in the vertical directions. Generally speaking, the above two kinds of masking characteristics show different masking effects in different image scenes and have certain complementarity. For example, the contrast masking may play a major role in relatively regular edge areas; the pattern masking may become the main factor in irregular areas. Consequently, the maximum value of the two masking components in different situations is utilized as the final measurement result of the spatial visual masking (VM):

$$VM(x, y) = \max\{PC(x, y), CM(x, y)\}, \tag{3}$$

In addition, HVS usually also shows different visual sensitivity to different background brightness, which is called Luminance Adaptation (LA). The visibility thresholds are different under different brightness backgrounds. Specifically, when the background brightness is lower than 127, the visibility threshold changes as a power function with the increase of the background brightness; when the background brightness exceeds 127, the visibility threshold changes linearly with the increase of the background brightness. The visibility threshold generated by the luminance adaptation is modeled as a piecewise function based on the background brightness:

$$LA(x, y) = \begin{cases} \left(1 - \sqrt{\dfrac{B(x, y)}{127}}\right) \times 17, & B(x, y) \le 127 \\ \\ (B(x, y) - 127) \times \dfrac{3}{128} + 3, & B(x, y) > 127 \end{cases}, \tag{4}$$

where $B(x, y)$ refers to the brightness of background area of the image; it can be calculated by the average brightness of the pixels in the 5×5 neighborhood around the point (x, y), as shown in Figure 3.



(a) (b)

Figure 2: Example of pattern complexity (left: regular pattern, right: irregular pattern). (a) Input image. (b) Pattern complexity map.

1	1	1	1	1
1	2	2	2	1
1	2	0	2	1
1	2	2	2	1
1	1	1	1	1

Figure 3: Low-pass filter.

After obtaining the VM and LA and other visual characteristic components, the nonlinear additivity model for masking (NAMM) is utilized to combine these visual characteristic components to form a perceptual JND threshold model:

$$\mathrm{JND}_p(x,y) = LA(x,y) + VM(x,y) - \alpha \cdot \min\{LA(x,y), VM(x,y)\}, \tag{5}$$

where α represents the gain loss parameter caused by overlap between these visual characteristic components, which is 0.3.

HVS can accurately forecast ordered visual stimuli relatively easily based on the free energy principle and will do further analysis and understanding. However, it is difficult to accurately predict disordered information that is complex, chaotic, and uncertain. HVS usually ignores the detailed information and only extracts its main outline structure. For example, straight lines or stripes in a uniform background generally have strong certainty, so HVS can easily detect any changes on the straight line. However, there is greater uncertainty in the changes of image elements in disorderly grass and other similar scenes. At this time, HVS usually automatically ignores the details in the disordered image, so the human eye will not easily perceive the detailed grass. Therefore, another important factor that decides the JND threshold is the disordered concealment effect.

It can be seen from the above analysis that the content changes of ordered images are regular and predictable, while it is sudden and unpredictable in disordered images. Thus, the disordered image denotes the uncertainty part of the original image. If sample value of the disordered image is large, it means that the corresponding original information has a high degree of uncertainty and can hide more noise and distortion. The existing pixel domain-based JND threshold estimation model performs well and estimates more accurately in ordered areas, such as flat, edge, and ordered texture areas, but underestimates JND threshold of disordered areas, such as disordered texture area. Therefore, in response to this shortcoming, the JND threshold based on the disordered concealment effect is calculated which is similar to [31]

$$\mathrm{JND}_d(x,y) = \mu \times \left| f(x,y) - f'(x,y) \right| = \mu \cdot D(x,y), \quad f'(x,y) = \sum_k c_k \cdot f(x_k, y_k) + \varepsilon, \tag{6}$$

where $f(x,y)$ denotes the original image of video, $f'(x,y)$ represents order information in the input image, c_k denotes the k-th normalized autoregressive coefficient in the 11×11 neighborhood around the point (x,y), and ε is white noise. $D(x,y)$ refers to unordered image, and μ represents the disorder adjustment factor, with a value of 1.125.

After obtaining the JND threshold component obtained from the disordered concealment effect, the NAMM is used to combine JND_p and JND_d to form the final hybrid JND threshold model:

$$\mathrm{JND}(x,y) = \mathrm{JND}_p(x,y) + \mathrm{JND}_d(x,y) - \alpha \cdot \min\left\{\mathrm{JND}_p(x,y), \mathrm{JND}_d(x,y)\right\}, \tag{7}$$

where JND_p and JND_d denote JND threshold for unordered image and ordered image. The hybrid JND threshold model further considers the complexity of the visual pattern and the disordered concealment effect on the basis of CA and LA. It makes up for the shortcomings of traditional models that cannot effectively estimate the ordered texture and random texture of the image. Therefore, the JND threshold of the image region is utilized as an effective measure of visual distortion sensitivity.

3.2. The Proposed Method. This paper proposes a fast CU decision method based on JND and deep learning to reduce the coding complexity introduced by the MTT splitting structure. First, the hybrid JND threshold models are designed by a perceptual JND threshold model and a JND threshold based on the disordered concealment effect, which

can divide the CUs into smooth, normal, and complex area. Then, the US-SVM classifiers are utilized to decide the best splitting mode for CUs in the complex area.

Specifically, the CU splitting in VVC is regarded as a six-class classification problem (class 1: nonsplit, class 2: QT, class 3: BT_H, class 4: BT_V, class 5: TT_H, class 6: TT_V), and the number of the training US-SVM classifiers is five. To train first US-SVM classifier SVM_1, class 1 is utilized as positive training samples, and classes 2, 3, 4, 5, and 6 of training samples are used as negative training samples. By analogy, the fourth US-SVM takes class 4 as the positive sample, and classes 5 and 6 are used as the negative training samples. In order to train the fifth US-SVM classifier SVM_5, class 5 and class 6 are used as the positive samples and the negative samples, respectively. Therefore, only $K-1$ US-SVMs need to be trained, thereby eliminating regions belonging to multiple categories in the decision and reducing the total number of training samples accordingly. Finally, the proposed method can reduce the coding time. Figure 4 illustrates the overall flowchart of the proposed method.

3.2.1. The Region Classification-Based JND. According to the above analysis, the larger JND value is, the richer the regional texture information is; that is, it has higher complexity and stronger spatial masking effect. Consequently, HVS is not sensitive to noise and distortion in these areas. On the contrary, if JND value is smaller, it demonstrates that these areas are relatively flat and regular, and the organization of image elements tends to be orderly. At this time, HVS is more sensitive to noise and distortion in these areas because of the weaker visual masking effect. Thus, the JND threshold of image is used as an effective measure of visual distortion sensitivity.

It is noticed from Figure 5(b) that the higher grayscale values appear in areas with higher complexity, such as grass and other areas with more complex textures. These areas generally have higher JND threshold, the visual masking effect is more significant, and the encoding distortion will not be easily noticed by the human eye. Therefore, a smaller bitrate is allocated to obtain the same visual quality as the original one in the coding process. On the contrary, these areas with low gray values generally have a lower JND threshold, such as flat and smooth background areas. Since HVS is more sensitive to coding distortions in these areas, more bitrates need to be allocated in the encoding process to ensure that the visual quality will not be significantly affected. Figure 5 shows the original image and corresponding JND threshold distribution map.

This paper designs fast CU splitting method based on JND and deep learning to reduce the coding time, which can reduce complexity. Specifically, the JND threshold mode is utilized to divide each CU into smooth, ordinary, or complex area,

$$\begin{cases} \text{JND} \le S, & \text{smooth region} \\ S < \text{JND} < C, & \text{ordinary region} \\ \text{JND} > C, & \text{complex region} \end{cases}, \tag{8}$$

where S and C denote content weight factor, which are set to 0.15 and 0.30 based on extensive experiments, respectively.

3.2.2. The Proposed US-SVM Classifier. Many multiclass related methods have been proposed, which have some shortcomings. Recently, it is noted that the US-SVM classifier shows good characteristics in practical applications.

For K-class ($K > 2$) classification problem, the number of the US-SVMs classifiers is $K-1$. To train the first US-SVM SVM_1, class 1 sample is used as the positive training sample, and class 2, 3, ..., K training samples are used as negative training samples. To train SVM_i, class i samples are used as the positive training samples in the $i-$th US-SVM, and class $i+1$, $i+2$, ..., K training samples are used as the negative training sample. Until class $K-1$ is taken as a positive sample in $K-1$ US-SVM, and class K sample is used as a negative sample to train SVM_{K-1}. Finally, these US-SVM classifiers are utilized to decide CU partition mode. Specifically, the proposed method for VVC needs five US-SVM classifiers. To train first US-SVM classifier SVM_1, class 1 is used as positive training samples, and classes 2, 3, 4, 5, and 6 of training samples are used as negative training samples. By analogy, the fifth US-SVM SVM_5 utilizes class 5 as the positive samples, and class 6 is utilized as the negative samples. Therefore, the proposed method can eliminate regions belonging to multiple categories in the decision and reduce the total number of training samples accordingly.

Specifically, the SVM_1 is used as the root node of the binary tree, and the test samples belonging to the first class are determined. The samples that do not belong to the first class are classified by SVM_2, and SVM_5 determines the fifth class sample. Figure 6 shows the classification based on the binary tree. Specifically, since the MTT structure is introduced in VVC, the CU splitting problem is considered a six-class classification problem. Therefore, five US-SVM classifiers are trained in this paper.

3.2.3. The US-SVM Classifier Training. The US-SVM classifier model is tested utilizing the UHD test sequence. Table 1 shows the training and testing sequences. To decrease the complexity, the offline mode is utilized in the US-SVM classifier. The F-score features selecting method chooses the features of video sequences, where these features have high correlation with CU splitting. The F-score value is expressed as

$$F_i = \frac{\left(\overline{x}_i^{(+)} - \overline{x}_i\right)^2 + \left(\overline{x}_i^{(-)} - \overline{x}_i\right)^2}{1/n_+ - 1 \sum_{l=1}^{n_+} \left(x_{l,i}^{(+)} - \overline{x}_i^{(+)}\right)^2 + 1/n_- - 1 \sum_{l=1}^{n} \left(x_{l,i}^{(-)} - \overline{x}_i^{(-)}\right)^2}, \tag{9}$$

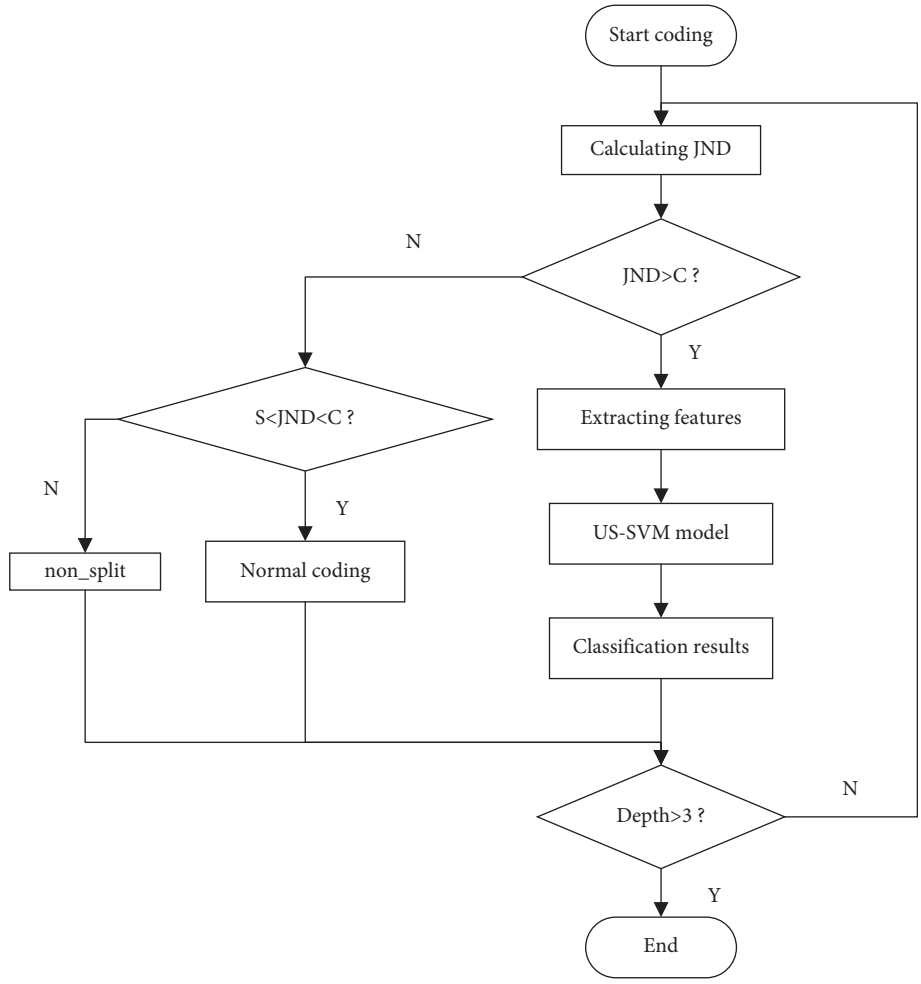


Figure 4: The overall flowchart of the proposed method.

where $\overline{x}_i$ is the average value of the i − th feature in the entire sample set, n_+ is the quantity of samples in the positive class, n_- is the quantity of samples in the negative class, $\overline{x}_i^{(+)}$ is the average value of the i − th feature in the positive sample set, $\overline{x}_i^{(-)}$ is the average value of the i − th feature in the negative sample set, $\overline{x}_{l,i}^{(+)}$ is the eigenvalues of the i − th feature of the l − th positive sample point, and $\overline{x}_{l,i}^{(-)}$ is the eigenvalues of the i − th feature of the l − th negative sample point. In order to balance the accuracy and the complexity in training US-SVM, three effective features are selected to train US-SVM through the F-score feature selection method in this paper.

The US-SVM classifiers use offline training mode. Then, the trained US-SVM classifiers are embedded in VTM10.0 to classify CU. And the classification accuracy of the US-SVM classifier is about 95.6%. Therefore, the proposed method can early predict the best CU partition mode to reduce the coding complexity with negligible BD loss.

4. Experimental Results and Analysis

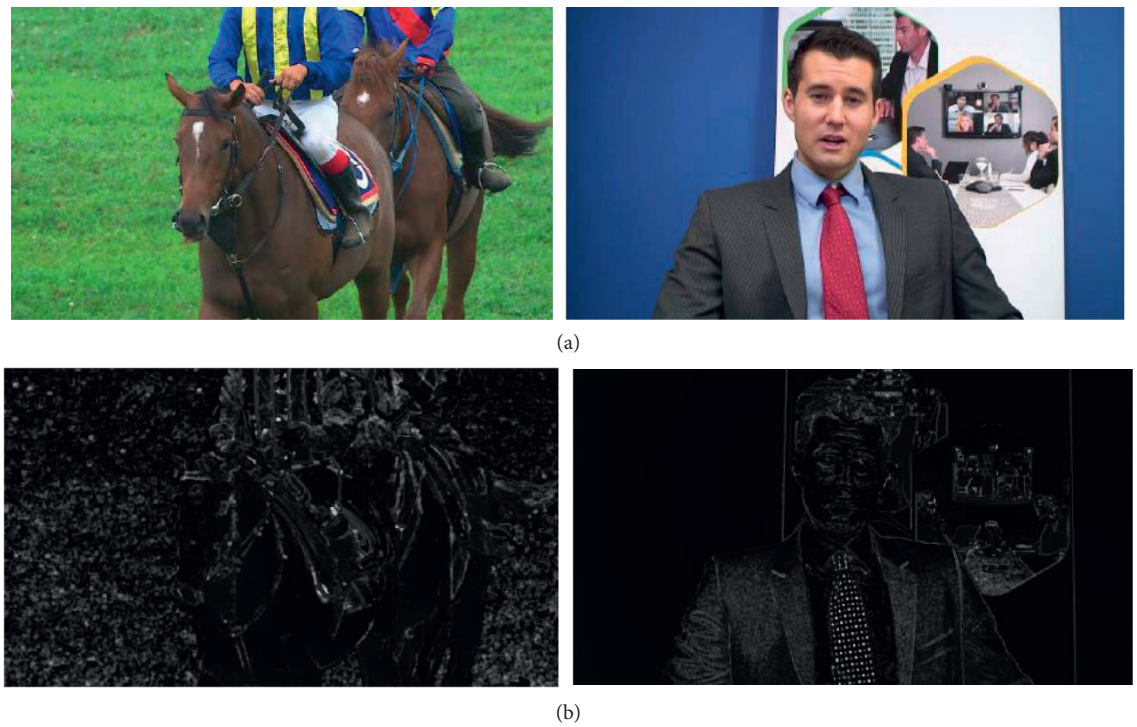
The experimental test is implemented on VTM 10.0 under "all-intra" configuration to evaluate the performance of the proposed method. The test set consists of the common test conditions (CTC) [32] sequence specified by JVET, which contains a wide range of resolutions, textures, bit depths, and motion. The Bjontegaard Delta Bitrate (BDBR) is utilized to measure the results of the proposed method [33] and average coding time saving (ACTS), where BDBR reflects the overall encoding quality and TS is used to measure the coding time saving, which is defined as

$$\text{ACTS} = \frac{T_{\text{VTM10.0}} - T_{\text{proposed}}}{T_{\text{VTM10.0}}} \times 100\%, \tag{10}$$

where $T_{\text{VTM10.0}}$ represents the coding runtime saving of anchor method that is VTM 10.0 in the proposed method and T_{proposed} is the ACTS of the proposed method. Since the different platforms are different in performance, this time does not count the time spent by the neural network.

Table 2 shows the results of the proposed method, where the sequences in Table 1 can be used since the training and testing video sequences are different for the training model. We can see that the coding runtime saving of the proposed method is about 52.35%, and the BDBR is increased by 0.99%.

Figure 7 illustrates the RD of VTM10.0 and the proposed method for two typical test videos including "*FourPeople*" and "*Kimono*". Compared with VTM 10.0, the proposed method has almost consistent RD performance with VTM 10.0.



(a)

(b)

Figure 5: The original image JND threshold map (left: RaceHorses, right: Johnny). (a) Input image. (b) JND threshold map.

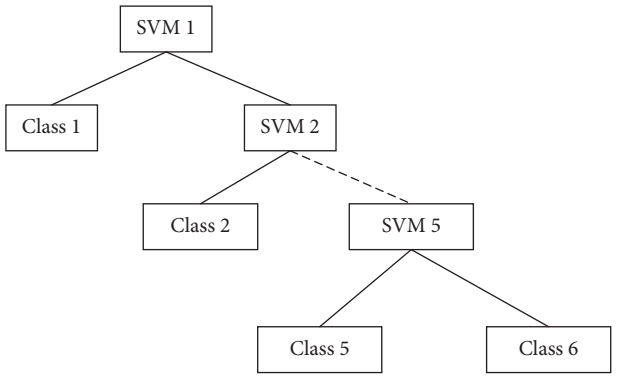


Figure 6: Classification based on binary tree.

To evaluate the performance of the proposed method, the proposed method is compared with the existing fast methods, comprising CTDM [6], FBDA [7], FIVG [13], and ACSD [14]. Tables 3 and 4 show the results of the proposed method and the existing fast methods in VVC. Table 3 demonstrates that the average BDBR of CTMD, FBDA, FIVG, and ACSD method increased to 1.06%, 1.38%, 1.38%, and 0.99%, respectively, while the average BDBR of the proposed method is only 0.99%. From Table 4, we can see that the proposed method can save more encoding time compared with CTMD, FBDA, FIVG, and ACSD methods. From Table 4, the encoding runtime savings of the CTMD, FBDA, FIVG, and ACSD are 34.32%, 29.49%, 52.16%, and 33.21%, which are less than that of the proposed method.

To see the performance advantages of the proposed method compared with the latest methods more intuitively, Figures 8 and 9 show the ACTS and BDBR increase of the proposed method and the existing fast methods in VVC. Figure 8 shows that the encoding runtime of the proposed method can reduce about 0.16–23.83% compared with CTMD, FBDA, FIVG, and ACSD method. Moreover, compared with CTMD, FBDA, and FIVG method, the proposed method can reduce BDBR about 0.07–0.39%. It is noticed that the proposed method increases the coding runtime saving and outperforms the coding performance of the existing fast methods.

Table 1: The training and testing video sequences.

	Sequences	fps	
A	Traffic	50	Training
	PeopleOnStreet	50	
B	BQTerrace	50	
	Cactus	30	
C	BQMall	50	
	BasketballDrill	60	
D	BlowingBubbles	50	
	RaceHorses	30	
4K	Bosphorus	120	
	RushHour	30	
A	PeopleOnStreet	30	Testing
	Nebuta	50	
B	BQTerrace	60	
	ParkScene	60	
C	RaceHorsesC	30	
	PartyScene	60	
D	BasketballPass	60	
	BQSquare	50	
E	Johnny	60	
	FourPeople	60	

Table 2: The results of the proposed method.

Sequences (%)		Proposed method BDBR	ACTS
Class A 2560 × 1600	PeopleOnStreet	0.96	52.13
	Traffic	0.97	50.04
	Nebuta	0.99	55.27
Class B 1920 × 1080	Kimono	1.34	51.34
	ParkScene	0.95	54.31
	BQTerrace	1.22	51.51
Class C 832 × 480	PartyScene	0.95	52.43
	RaceHorsesC	0.96	50.25
	BasketballDrill	0.89	52.42
Class D 416 × 240	BlowingBubbles	0.95	51.31
	RaceHorses	0.86	47.25
	BQSquare	0.87	53.15
Class E 1280 × 720	Johnny	0.88	55.33
	FourPeople	0.78	58.45
	KristenAndSara	1.21	50.12
	Average	0.99	52.35

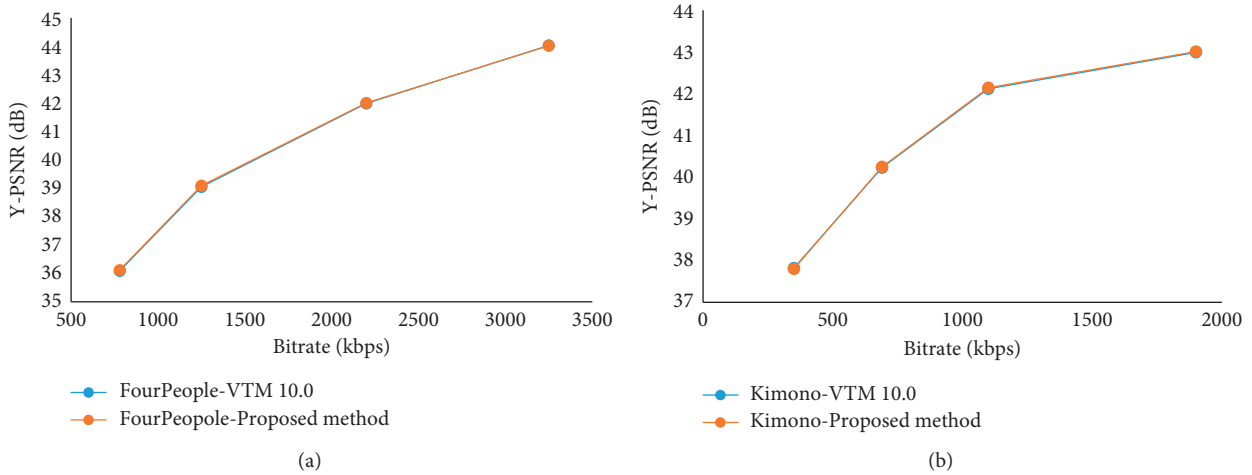


Figure 7: The RD for the proposed method and VTM 10.0. (a) The RD of "FourPeople". (b) The RD of "Kimono".

Table 3: The BDBR of the proposed method and the latest methods.

Sequences		Proposed BDBR (%)	CTDM [6] BDBR (%)	FBDA [7] BDBR (%)	FIVG [13] BDBR (%)	ACSD [14] BDBR (%)
Class B 1920 × 1080	Kimono	1.34	1.05	1.98	1.72	0.87
	ParkScene	0.95	1.11	1.38	1.28	0.83
	BQTerrace	1.22	1.00	1.19	1.16	0.95
Class C 832 × 480	PartyScene	0.95	0.76	1.05	0.28	0.55
	RaceHorsesC	0.96	0.82	2.96	0.84	0.37
	BasketballDrill	0.89	1.67	1.36	1.91	1.30
Class D 416 × 240	BlowingBubbles	0.95	0.74	0.73	0.49	0.95
	RaceHorses	0.86	0.95	1.59	0.54	0.71
	BQSquare	0.87	0.61	−0.11	0.17	0.68
Class E 1280 × 720	Johnny	0.88	1.44	1.51	3.07	1.72
	FourPeople	0.78	1.38	1.37	2.55	1.35
	KristenAndSara	1.21	1.19	1.53	2.56	1.61
Average		0.99	1.06	1.38	1.38	0.99

Table 4: The ACTS of the proposed method and the existing methods.

Sequences		Proposed ACTS (%)	CTDM [6] ACTS (%)	FBDA [7] ACTS (%)	FIVG [13] ACTS (%)	ACSD [14] ACTS (%)
Class B 1920×1080	Kimono	51.34	34.33	41.82	66.59	32.32
	ParkScene	54.31	34.56	31.60	56.28	35.41
	BQTerrace	51.51	31.07	29.47	49.44	34.50
Class C 832×480	PartyScene	52.43	35.93	35.23	41.71	31.10
	RaceHorsesC	50.25	36.06	33.89	52.07	23.63
	BasketballDrill	52.42	33.94	28.73	53.05	33.39
Class D 416×240	BlowingBubbles	51.31	35.04	21.87	43.90	33.90
	RaceHorses	47.25	35.96	31.83	44.93	31.79
	BQSquare	53.15	33.98	23.00	32.34	30.73
Class E 1280×720	Johnny	55.33	33.02	24.44	62.55	38.85
	FourPeople	58.45	35.05	26.65	62.18	38.01
	KristenAndSara	50.12	32.95	25.32	60.82	34.84
Average		53.32	34.32	29.49	52.16	33.21

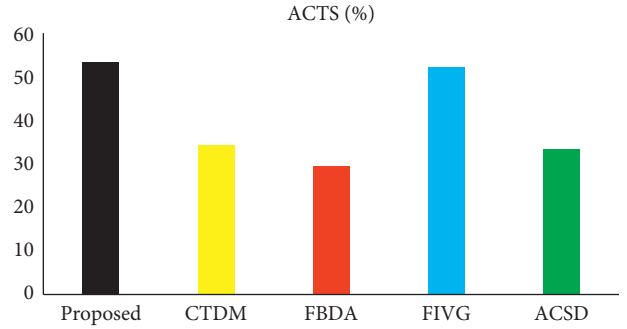


Figure 8: Coding time saving.

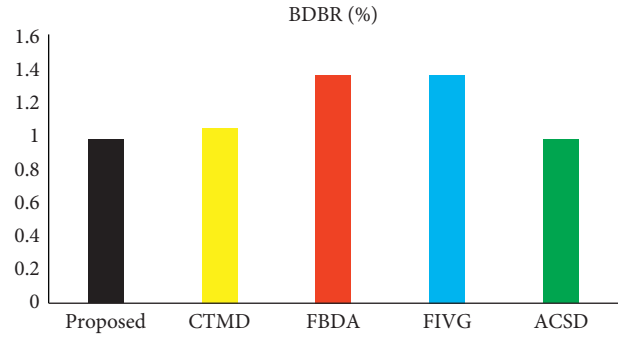


Figure 9: BDBR increase.

5. Conclusion

A fast CU decision method is introduced based on JND and the US-SVM classifiers in this paper to settle the huge complexity caused by the asymmetric splitting problem. The hybrid JND model is used to determine which region the CUs belong to. Then, the US-SVM classifiers are utilized to split CU in advance to reduce coding time. Experimental results demonstrate that the proposed method can significantly save about 52.35% coding runtime, with only 0.99% BDBR. The results may fluctuate for different videos with different resolutions, because the resolutions of the videos may have a little impact on the experimental results, where the ACTS is particularly high for the sequence such as "FourPeople" (58.45%). Moreover, the proposed method exceeds the existing fast methods and can keep the coding efficiency. We will continue searching for fast methods to reduce encoding time while maintaining encoding quality.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (nos. 61771432, 61302118, and 61702464), the Basic Research Projects of Education Department of Henan (nos. 21zx003 and 20A880004), and the Postgraduate Education Reform and Quality Improvement Project of Henan Province (no. YJS2021KC12).

References

[1] K. Misra, A. Segall, and F. Bossen, "Tools for video coding beyond HEVC: flexible partitioning, motion vector coding, l adaptive quantization, and improved," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 5, pp. 1361–1373, 2020.

[2] A. Wieckowski, T. Hinz, V. George et al., *NextSoftware: An Alternative Implementation of the Joint Exploration Model (JEM)*, ISO/IEC, Macao, CN, USA, 2017.

[3] Z. Wang, X. Meng, C. Jia et al., *Description of SDR Video Coding Technology Proposal by DJI and Peking University*, JVET-J0011, San Diego, CF, USA, 2018.

[4] V. Baroncini, J.-R. Ohm, and G. J. Sullivan, *Report of Results from the Call for Proposals on Video Compression with Capability beyond HEVC*, ISO/IEC, San Diego, CF, US, 2018.

[5] H. Yuan, C. Guo, J. Liu, X. Wang, and S. Kwong, "Motion-homogeneous-based fast transcoding method from H.264/AVC to HEVC," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1416–1430, 2017.

[6] S.-H. Park and J.-W. Kang, "Context-based ternary tree decision method in versatile video coding for fast intra coding," *IEEE Access*, vol. 7, pp. 172597–172605, 2019.

[7] N. Tang, J. Cao, F. Liang et al., "Fast CTU partition decision algorithm for VVC intra and inter coding," in *Proceedings of the 2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 361–364, IEEE, Bangkok, Thailand, November 2019.

[8] Y. Chen, L. Yu, H. Wang, T. Li, and S. Wang, "A novel fast intra mode decision for versatile video coding," *Journal of Visual Communication and Image Representation*, vol. 71, Article ID 102849, 2020.

[9] M. Amna, W. Imen, S. F. Ezahra, and A. Mohamed, "Fast intra-coding unit partition decision in H.266/FVC based on deep learning," *Journal of Real-Time Image Proceedings*, vol. 17, no. 3, pp. 1971–1981, 2020.

[10] Z. Liu, M. Dong, and M. Zhang, "A fast multi-type tree decision algorithm for VVC based on pixel difference of sub-blocks," *IEICE-Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E103, no. 6, pp. 865–859, 2020.

[11] T.-L. Lin, H.-Y. Jiang, J.-Y. Huang, and P.-C. Chang, "Fast intra coding unit partition decision in H.266/FVC based on spatial features," *Journal of Real-Time Image Processing*, vol. 17, no. 3, pp. 493–510, 2020.

[12] T. Fu, H. Zhang, F. Mu, and H. Chen, "Fast CU partitioning algorithm for H.266/VVC intra-frame coding," in *Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 55–60, IEEE, Shanghai, China, July 2019.

[13] 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," in *Proceedings of the 2019 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–4, IEEE, Sydney, Australia, December 2019.

[14] G. Tang, M. Jing, X. Zeng, and Y. Fan, "Adaptive CU split decision with pooling-variable CNN for VVC intra encoding," in *Proceedings of the 2019 IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–4, IEEE, Sydney, Australia, December 2019.

[15] T. Amestoy, A. Mercat, W. Hamidouche, D. Menard, and C. Bergeron, "Tunable VVC frame partitioning based on lightweight machine learning," *IEEE Transactions on Image Processing*, vol. 29, no. 1, pp. 1313–1328, 2020.

[16] X. Dong, L. Shen, M. Yu, and H. Yang, "Fast intra mode decision algorithm for versatile video coding," *IEEE Transactions on Multimedia*, p. 1, 2021.

[17] 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 Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1668–1682, 2020.

[18] Q. Zhang, Y. Wang, L. Huang, and B. Jiang, "Fast CU partition and intra mode decision method for H.266/VVC," *IEEE Access*, vol. 8, pp. 117539–117550, 2020.

[19] Q. Zhang, Y. Wang, L. Huang, B. Jiang, and X. Wang, "Fast CU partition decision for H.266/VVC based on the improved dag-svm classifier model," *Multimedia Systems*, vol. 5, 2020.

[20] S. Ki, S.-H. Bae, M. Kim, and H. Ko, "Learning-based just-noticeable-quantization- distortion modeling for perceptual video coding," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3178–3193, 2018.

[21] M. Zhou, X. Wei, S. Kwong, W. Jia, and B. Fang, "Just noticeable distortion-based perceptual rate control in HEVC," *IEEE Transactions on Image Processing*, vol. 29, no. 6, pp. 7603–7614, 2020.

[22] H. Wang, L. Yu, X. Tang, H. Yin, and J. Liang, "A QD&JND compensation based PVC scheme for HEVC," in *Proceedings of the 2020 Data Compression Conference (DCC)*, p. 396, March 2020.

[23] H. Liu, Y. Zhang, H. Zhang et al., "Deep learning-based picture-wise just noticeable distortion prediction model for image compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 641–656, 2020.

[24] J. Cui, R. Xiong, X. Zhang, S. Wang, and S. Ma, "Perceptual video coding based on visual saliency modulated just noticeable distortion," in *Proceedings of the 2019 Data Compression Conference (DCC)*, p. 565, March 2019.

[25] X. Shen, X. Zhang, S. Wang, S. Kwong, and G. Zhu, "Just noticeable distortion based perceptually lossless intra coding," in *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2058–2062, IEEE, Barcelona, Spain, May 2020.

[26] X. Shen, Z. Ni, W. Yang, X. Zhang, S. Wang, and S. Kwong, "Just noticeable distortion profile inference: a patch-level structural visibility learning approach," *IEEE Transactions on Image Processing*, vol. 30, pp. 26–38, 2021.

[27] H. Yuan, Q. Wang, Q. Liu, J. Huo, and P. Li, "Hybrid distortion-based rate-distortion optimization and rate control for H.265/HEVC," *IEEE Transactions on Consumer Electronics*, vol. 67, no. 2, pp. 97–106, 2021.

[28] J. Wu, W. Lin, G. Shi, L. Li, and Y. Fang, "Orientation selectivity based visual pattern for reduced-reference image quality assessment," *Information Sciences*, vol. 351, pp. 18–29, 2016.

[29] J. Wu, W. Lin, G. Shi, X. Wang, and F. Li, "Pattern masking estimation in image with structural uncertainty," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4892–4904, 2013.

[30] J. Wu, L. Li, W. Dong, G. Shi, W. Lin, and C.-C. J. Kuo, "Enhanced just noticeable difference model for images with pattern complexity," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2682–2693, 2017.

[31] J. Wu, G. Shi, W. Lin, A. Liu, and F. Qi, "Just noticeable difference estimation for images with free-energy principle," *IEEE Transactions on Multimedia*, vol. 15, no. 7, pp. 1705–1710, 2013.

[32] F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, *JVET Common Test Conditions and Software Reference Configurations for SDR Video*, JVET-M1010, San Diego, CF, USA, 2019.

[33] G. Bjontegaard, *Calculation of Average PSNR Differences between RD Curves, Document ITU-T SG16 Q6*, VCEG-M33, Austin, TX, USA, 2001.