Hindawi
Journal of Sensors
Volume 2024, Article ID 9820689, 1 page
https://doi.org/10.1155/2024/9820689

Retraction

Retracted: Intraprediction Complexity Control Algorithm Based on Visual Saliency

Journal of Sensors

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

In addition, our investigation has also shown that one or more of the following human-subject reporting requirements has not been met in this article: ethical approval by an Institutional Review Board (IRB) committee or equivalent, patient/participant consent to participate, and/or agreement to publish patient/participant details (where relevant).

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] N. Shan, W. Zhou, and Z. Duan, "Intraprediction Complexity Control Algorithm Based on Visual Saliency," *Journal of Sensors*, vol. 2022, Article ID 5069775, 13 pages, 2022.

Hindawi
Journal of Sensors
Volume 2022, Article ID 5069775, 13 pages
https://doi.org/10.1155/2022/5069775

Research Article

Intraprediction Complexity Control Algorithm Based on Visual Saliency

Nana Shan,[1] **Wei Zhou,**[2] **and Zhemin Duan**[2]

[1]*School of Information Science and Technology, Taishan University, Tai' an, 271000, China*
[2]*Department of Electronics and Information, Northwestern Polytechnical University, Xi'an, 710072, China*

Correspondence should be addressed to Nana Shan; helensnn@tsu.edu.cn

Received 24 June 2022; Revised 25 August 2022; Accepted 8 September 2022; Published 29 September 2022

Academic Editor: Yuan Li

Intraprediction is one of the most complex parts of High-Efficiency Video Coding (HEVC), because it selects the best prediction mode by calculating the cost of every Coding Unit (CU), which provides higher complexity of intracoding. Visual saliency map can show the attention regions of the human eyes, which is generated by certain static and space-time saliency detection method. By analyzing the percentage of coding time for different size CU, and the relation of visual saliency and CU depth, an intraprediction complexity control algorithm based on visual saliency is proposed in this paper. Based on the feature of the video and the target level, the saliency threshold is adapted to determine whether the current CU in the intraprediction processing should be split into smaller CUs or the division processing should be stopped early. Three samples were compared by the proposed algorithm and other algorithm, and the proposed algorithm has better performance in PSNR, BitRate, and coding time. Experimental results show that this algorithm can effectively control the coding complexity of intraprediction with minimal visual loss and can be applied to a number of scenarios, such as real-time video coding.

1. Introduction

The number of intraprediction modes in HEVC is increased up to 35, and like interprediction coding, intraprediction also use recursive algorithm to compare all prediction modes of all coding units, and then selects the optimal mode from them, which will cause high coding complexity [1].In order to use the correlation between luma and chroma, [2] presents a convolutional neural network based intrachroma prediction method and cross component linear model. The plane modeling coefficients can be predicted by the neighboring depth pixels, and a certain threshold can be compared with the prediction error [3].Support Vector Machine (SVM) was adopted to get higher precision decision-making in [4]. For different video sequences with different coding complexity, the proposed method employed a fixed decision boundary. An improved CU-level rate control algorithm was proposed in [5]. The gradient detection operator and the Hadamard transform algorithm are used to detect the texture complexity area and the transform compression area, and the scale coefficients adjust the target bits of each CTU. In order to tackle the shortcoming of traditional machine learning algorithms, reference [6] proposed the acceleration properties of different modules. It also provides Heuristic Model Oriented Framework (HMOF) algorithm, which can adapt the properties of different modules. It uses various Convolutional Neural Network (CNN) to choose the best intraprediction mode and it investigates combinations of them [7].

However, the above algorithms are based on the characteristics of intraprediction itself, which are based on the statistical results of the coding units. According to the feature of visual saliency, it also can be analyzed from the visual characteristics of the human eyes to reduce the complexity of intraprediction, which can control the complexity of intraprediction with minimal visual loss. In this paper, a complexity control algorithm of intraprediction based on visual saliency is proposed, which can reduce the encoding time within the target complexity range, and the visual loss is unnoticeable. By adapting this method, different coding strategy can be made in different application scenario, which can bring higher compression performance.

2. Intraprediction

In the spatial domain, the closer the distance between pixels, the stronger the correlation is between them, so we can use the weighted sum of several adjacent pixels to predict the current pixel. Then in the transmission, it is not necessary to transmit the actual pixel value but only the difference between the actual pixel value and the predicted value. The original pixel value can be recovered by directly adding the difference signal to the predicted value in the decoding process. Because of the strong correlation between adjacent pixels, the residual value is often very small, which can reduce the encoded information and achieve the purpose of video compression. Intraprediction is a process of removing the spatial redundancy of the video, which uses the correlation between the video images in the spatial domain to predict the current pixel according to the pixels that have been coded in the current image. It is an important compression method in video coding, especially when interprediction coding cannot be used. Intraprediction coding has become an important means to ensure the compression rate [8].

2.1. Prediction Mode. In HEVC, the prediction template for intraprediction is shown in Figure 1, where $R_{x,y}$ (shown in gray area) on the left and above represents the pixel reconstruction value of the adjacent blocks and $P_{x,y}$ (shown in white area) in the middle represents the pixel prediction value of the current block.

The prediction unit sizes supported by intraprediction are 4×4, 8×8, 16×16, 32×32, and 64×64, and each PU size includes 35 prediction modes, as shown in Table 1, which are planar mode, DC mode, and 33 angle prediction modes.

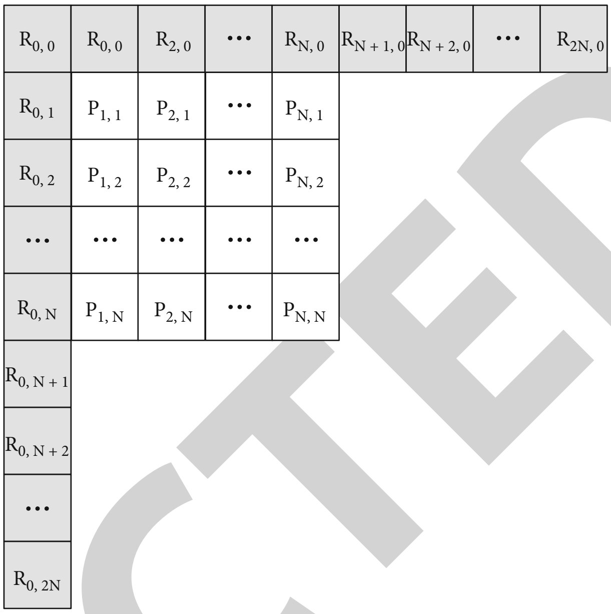
2.1.1. Planar Mode. As shown in Figure 2, the planar mode uses filters in the horizontal and vertical directions and uses the average value as the predicted value of the current pixel, which can better maintain the continuity of the image boundary.

2.1.2. DC Mode. The DC prediction is obtained by using the average value of the reference pixels on the left side and the upper side (excluding the upper left corner, the lower left side, and the upper right side) of the current pixel, so it is suitable for a flat image with a large area.

2.1.3. Angle Mode. HEVC specifies 33 angle prediction modes, mode 10 represents the horizontal direction and mode 26 represents the vertical direction, patterns 2-17 are horizontal class patterns and patterns 18-34 are vertical class patterns, as shown in Figure 3. Different modes correspond to different offset angles, and the size of the offset angle can be calculated by

$$\theta = \arctan\left(\frac{x}{32}\right). \quad (1)$$

The 33 angle prediction modes can be divided into two categories: vertical prediction and horizontal prediction. 2-17 is the horizontal prediction. When the prediction direction shifts upward, θ is positive. When the prediction direction is shifted downward, θ is negative. 18-34 is a vertical prediction, and θ is positive when the prediction direction shifts to the left; when the prediction direction is shifted to the right, θ is negative.



$R_{0,0}$	$R_{0,0}$	$R_{2,0}$	⋯	$R_{N,0}$	$R_{N+1,0}$	$R_{N+2,0}$	⋯	$R_{2N,0}$
$R_{0,1}$	$P_{1,1}$	$P_{2,1}$	⋯	$P_{N,1}$				
$R_{0,2}$	$P_{1,2}$	$P_{2,2}$	⋯	$P_{N,2}$				
⋯	⋯	⋯	⋯	⋯				
$R_{0,N}$	$P_{1,N}$	$P_{2,N}$	⋯	$P_{N,N}$				
$R_{0,N+1}$								
$R_{0,N+2}$								
⋯								
$R_{0,2N}$								

Figure 1: Intraprediction template [1].

Table 1: Number of 35 prediction modes.

No.	Mode
0	Planar mode
1	DC mode
2-34	33 angle prediction modes

2.2. Flow of Intraprediction. There are three steps in intraprediction:

2.2.1. Get Reference Pixels. As shown in Figure 1, the reference pixels of the current pixel block can be divided into five parts: left, lower left, upper left, upper, and upper right. If some reference pixels do not exist, they will be replaced by reference pixels in adjacent areas. When all reference pixels are unavailable, they will be replaced by fixed values. The calculation method is as shown:

$$R = 1 \ll (\text{BitDepth} - 1). \quad (2)$$

2.2.2. Reference Pixel Filtering. When the pixel position is as shown in Figure 1, vertical-right mode is shown in equation (3), horizontal-down mode is shown as equation (4), diagonal-down-right mode is shown in equation (5), and the filtering calculation method is

$$F_{x,0} = (R_{x-1,0} + 2R_{x,0} + R_{x+1,0} + 2) \gg 2, \quad (3)$$

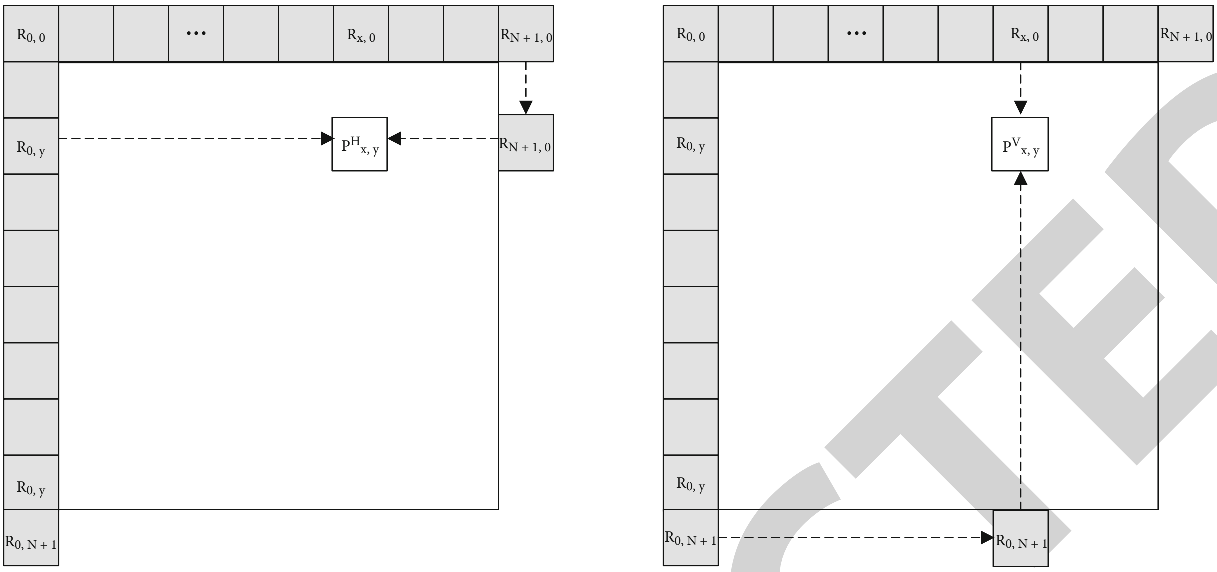


Figure 2: Planar mode [1].

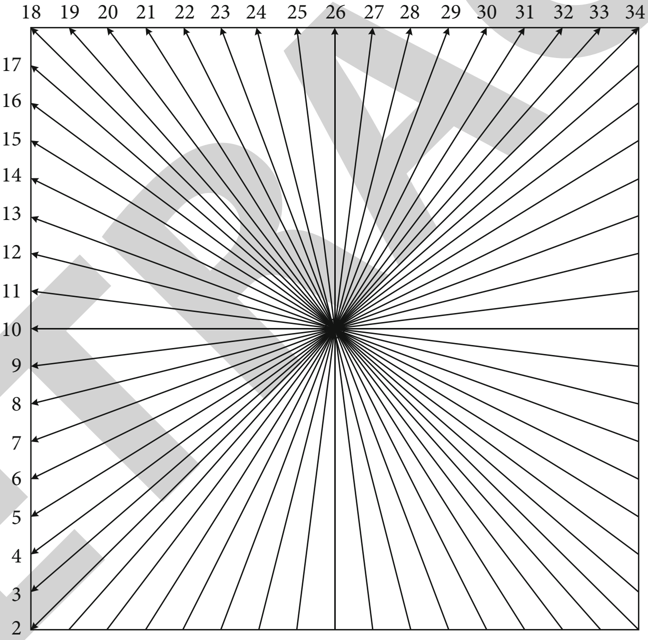


Figure 3: Angle prediction mode [1].

$$F_{0,y} = \left(R_{0,y-1} + 2R_{0,y} + R_{0,y+1} + 2\right) \gg 2, \tag{4}$$

$$F_{0,0} = (R_{0,1} + 2R_{0,0} + R_{1,0} + 2) \gg 2, \tag{5}$$

$$F_{2N,0} = (R_{2N-1,0} + 3R_{2N,0} + 2) \gg 2, \tag{6}$$

$$F_{0,2N} = (R_{0,2N-1} + 3R_{0,2N} + 2) \gg 2. \tag{7}$$

2.2.3. Calculate the Predicted Value. The calculation methods of the predicted value are different in different modes. The calculation methods of the three modes are described as follows:

(1) Planar Mode. The calculation method of the planar mode predicted pixel is shown:

$$P^{H}_{x,y} = (N - x)R_{0,y} + x\bullet R_{N+1,0}, \tag{8}$$

$$P^{V}_{x,y} = (N - y)R_{x,0} + y\bullet R_{0,N+1}, \tag{9}$$

$$P_{x,y} = \left(P^{H}_{x,y} + P^{V}_{x,y} + N\right) \gg [\log_2(N) + 1]. \tag{10}$$

(2) DC Mode.

$$\mathrm{DCAverage} = \left(\sum_{x=1}^{N} R_{x,0} + \sum_{x=1}^{N} R_{0,y} + N \right) \gg [\log_2(N) + 1], \tag{11}$$

$$P_{1,1} = (R_{1,0} + R_{0,1} + 2 \bullet \mathrm{DCAverage} + 2) \gg 2, \tag{12}$$

$$P_{x,1} = (R_{x,0} + 3 \bullet \mathrm{DCAverage} + 2) \gg 2, \tag{13}$$

$$P_{1,y} = \left(R_{0,y} + 3 \bullet \mathrm{DCAverage} + 2\right) \gg 2, \tag{14}$$

$$P_{x,y} = \mathrm{DCAverage}. \tag{15}$$

(3) Angle Mode. For angular prediction modes, each mode is offset in either the horizontal or vertical direction. First of all, the reference pixel projection needs to be mapped into a one-dimensional form according to its angle offset value, and then the predicted value of the current pixel is calculated.

3. Visual Saliency

Visual saliency refers to the ability of some elements in the scene to attract people's visual attention. This ability is due to the fact that the target has some special visual attributes, so it has a strong subjectivity. Visual saliency modeling is the process of quantifying this ability to attract attention.

3.1. Characteristics of Visual Saliency. The human visual system can quickly recognize, segment, combine, analyze, and understand objects in the visual scene. The reason why these functions can be performed so efficiently is that the human eye can only process the prominent features in the scene and ignore most of the insignificant background areas. The first stage of vision processing is to transform the scene into various feature representations, such as edges, brightness, angles, colors, and lines. The visual saliency model is based on these characteristics of the visual system [9].

3.1.1. Color Feature. Color is the most obvious contrast feature in an image or video, and areas that are inconsistent with the surrounding colors can easily become attractive places. The common description methods of color feature are directly using local color representation and also using local statistical methods such as color histogram.

3.1.2. Texture Feature. It can represent the changing characteristics of the surface of objects, such as regular decorations on buildings and stripes on animals, which generally have regular distribution. The commonly used methods of texture analysis are the statistical method, structural method, and spectral method.

3.1.3. Shape Feature. The shape of the object, including the edge and contour information of the object, is a very important aspect in the visual feature, and it is a popular research direction, and many edge detection operators have been proposed, such as Sobel operator, Canny operator, and Robert operator. These algorithms have been widely used in image or video processing, but there is still much ways for improvement.

3.1.4. Motion Feature. For video, motion information is more critical information, and in the case of fixed background, moving objects have significant characteristics.

3.2. Algorithms of Visual Saliency. The main steps of visual saliency modeling are feature detection, feature comparison to get the visual saliency map, and finally the synthesis of saliency map. Here are some common algorithms:

3.2.1. Saliency Algorithm Based on Image Sparse Representation. This algorithm first converts the format of the input image, divides the converted image of each channel into blocks and performs sparse representation, then calculates the corresponding local saliency and global saliency, and finally synthesizes the saliency map.

(1) Format Conversion

According to equation (16), the image conversion formats

$$I = R + G + B, \tag{16}$$

RG = $R - G$,

BY = $G - R + G/2 - \min(R, G)/2$.

R, G, and B represent the red, green, and blue channels of the input image, respectively, RG and BY represent the red/green and blue/yellow channels, respectively, and I represent the luminance channel.

(2) The Sparse Representation

The input image is first scaled to $2^w \times 2^w$ pixel size, with W being the size of the block. The whole image can be sparsely coded by using the least square shrinkage algorithm.

(3) Get the Saliency

$$\begin{aligned} \mathrm{SL}_l^c(x_i) &= \frac{1}{M} \sum_{j=1}^{M} \rho_{ij}^c, \\ \mathrm{SL}^c(x_i) &= \frac{1}{L} \sum_{j=1}^{M} \mathrm{SL}_l^c(x_i), \end{aligned} \tag{17}$$

$$\begin{aligned} \mathrm{SG}_l^c(x_i) &= \frac{1}{N} \sum_{j=1}^{N} \left(\rho_{ij}^c \times \frac{1}{1 + \mathrm{dist}(x_i, x_{\mathrm{center}})} \right), \\ \mathrm{SG}^c(x_i) &= \frac{1}{L} \sum_{j=1}^{M} \mathrm{SG}_l^c(x_i). \end{aligned} \tag{18}$$

(4) Synthesizes the Saliency Map

$$\begin{aligned} \mathrm{SL}(x_i) &= \sum_{c \in \{I, \mathrm{RG}, \mathrm{BY}\}} \aleph(\mathrm{SL}^c(x_i)), \\ \mathrm{SG}(x_i) &= \sum_{c \in \{I, \mathrm{RG}, \mathrm{BY}\}} \aleph(\mathrm{SG}^c(x_i)). \end{aligned} \tag{19}$$

3.2.2. Visual Saliency Detection Algorithm Based on Self-Similarity [10]. In this algorithm, the Local Steering Kernel (LSK) of each pixel in the image is used to calculate its local regression kernel matrix, and then the matrix cosine similarity between the current pixel and its surrounding pixels is calculated to obtain the saliency of the current pixel, and finally the saliency map is synthesized.

(1) Get the Saliency

$$S_i = P_r(y_i = 1|F), \tag{20}$$

$$S_i = P_r(y_i = 1|F) = \frac{p(F|y_i = 1)P_r(y_i = 1)}{p(F)}, \tag{21}$$

$$S_i = \widehat{p}(F|y_i = 1) = \frac{G_i\left(\overline{F_i} - \overline{F_i}\right)}{\sum_{j=1}^{N} G_i\left(\overline{F_i} - \overline{F_j}\right)}, \tag{22}$$

$$G_i\left(\overline{F_i} - \overline{F_j}\right) = \exp\left(\frac{-\overline{F_i} - \overline{F_j}_F^2}{2\sigma^2}\right) = \exp\left(\frac{-1 + \rho\left(F_i, F_j\right)}{\sigma^2}\right), \tag{23}$$

$$\rho_i = \sum_{l=1}^{L} \rho\left(f_i^l, f_j^l\right) \frac{f_i^l f_j^l}{F_{iF} F_{jF}}, \tag{24}$$

$$S_i = \frac{1}{\sum_{j=1}^{N} \exp\left(\left(-1 + \rho\left(F_i, F_j\right)\right)/\sigma^2\right)}. \tag{25}$$

(2) Process Color Images

The algorithm does not directly extract the saliency from each color channel and combine them but defines each color channel $C1$, $C2$, and $C3$ as a different feature matrix F_i^{c1}, F_i^{c2}, and F_i^{c3}, and then defines $F_i = [F_i^{c1}, F_i^{c2}, F_i^{c3}]$. Finally, the matrix cosine similarity calculation is adopted.

3.2.3. Detection Algorithm of Human Visual Attention Points. The algorithm gets the saliency map of the image through the following calculation. The saliency map is a gray map between 0 and 255, which represents the probability of each pixel becoming a visual attention point.

(1) Feature Extraction

The image is subsampled to generate three color components and to obtain a multiscale color feature FC_i and a multiscale local operating kernel feature FS_i.

(2) Get the Saliency

$$p(\mathrm{FC}_i) = \frac{\sum_{j=1}^{M} k\left(\mathrm{FC}_i - \mathrm{FC}_j\right)}{\sum_{i=1}^{M}\sum_{j=1}^{M} k\left(\mathrm{FC}_i - \mathrm{FC}_j\right)}, \tag{26}$$

$$p(FC_i|y_i = 1) = \frac{1}{\sum_{j=1}^{N} k\left(FC_i - FC_j\right)}, \tag{27}$$

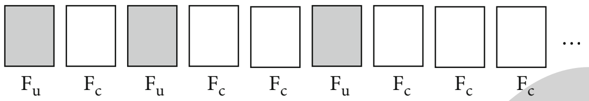


Figure 4: Angle prediction mode.

$$S_i = \frac{1}{p(F_i)} \cdot p(F_i|y_i = 1) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M} k\left(\mathrm{FC}_i - \mathrm{FC}_j\right)}{\sum_{j=1}^{M} k\left(\mathrm{FC}_i - \mathrm{FC}_j\right)} \cdot \frac{1}{\sum_{j=1}^{N} k\left(\mathrm{FC}_i - \mathrm{FC}_j\right)}. \tag{28}$$

(3) Synthesizes the Saliency Map

3.3. Application of Visual Saliency in Video Coding. The theories and methods of visual saliency are not only applicable to static images but also to video processing. The following will introduce several application algorithms of visual saliency in video.

3.3.1. Video Coding Algorithm Based on Spatiotemporal Saliency Map [11]. Firstly, the absolute feature difference of the video is calculated by the Gaussian function, and then multiple spatial saliency maps are established, and then they are combined into a global spatial saliency map, and finally the temporal visual saliency map is obtained by using the motion vector of the foreground. The spatiotemporal visual saliency map can be obtained by combining the temporal and spatial visual saliency maps.

3.3.2. Region-of-Interest-Based Video Coding Algorithm [12]. The video coding algorithm introduced in this algorithm is based on Region of Interest (ROI). High quality coding is used for regions of interest, and conversely, less saliency is used for regions of no interest. First, a saliency calculation was performed based on Itti's visual saliency model [9]. Then, the saliency factor is added to the rate-distortion optimization. And the value of saliency can be flexibly adjusted according to the degree of interest. Then, the complexity of the algorithm can be reduced by reusing some information in the video coding process. Finally, a comparison with other related perceptual coding algorithms is made.

3.3.3. Video Coding Complexity Reduction Algorithm Based on Visual Saliency [13]. Firstly, the algorithm analyzes the complexity of video coding based on two kinds of relationships: one is the relationship between the maximum coding depth and complexity, and the other is the relationship between the maximum coding depth and distortion. Then it uses visual saliency to predict the coding depth, thus reducing the amount of computation for traversal in prediction.

4. A Complexity Control Algorithm for Intraprediction Based on Visual Saliency

In this section, the proportion of coding units of different depths in the optimal mode of intraprediction of the HM-13.0 standard algorithm is analyzed; then, the proportion of the coding time of each depth CU in the whole video

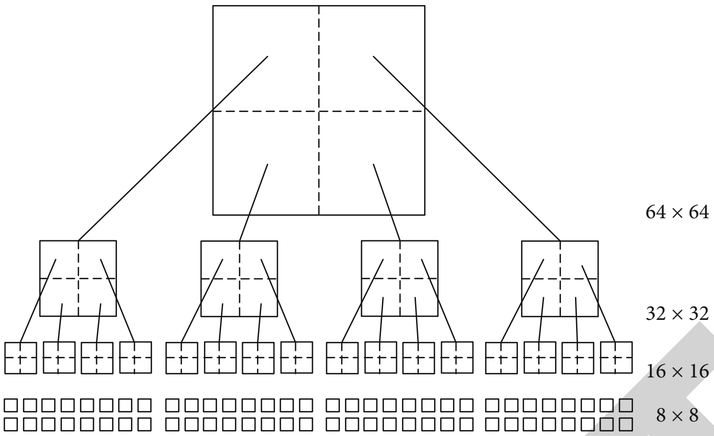


Figure 5: Quadtree structure.

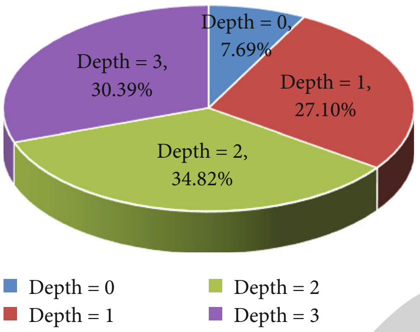


Figure 6: Percentage of different depth in the best mode.

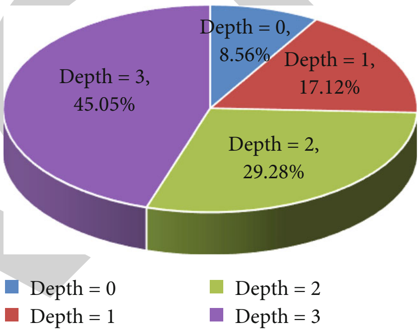


Figure 7: Percentage of different depth in the intraprediction.

coding time is counted, respectively. According to the relationship between the division of coding units in intraprediction and visual saliency, an intraprediction complexity control algorithm based on visual saliency is proposed.

4.1. Proportion of each Depth Coding Unit. Correa et al. perform complexity control by dividing each frame in the video into unlimited frames and limited frames. For unlimited frames, it is necessary to perform encoding according to a normal unit division process, and the rate-distortion cost of the encoding unit CU of each depth is calculated in a mode by mode and depth by depth traversal manner. Then the best intraprediction mode is selected by comparing the rate-distortion cost. For limited frames, the maximum CU depth of the limited frames is determined according to the previously encoded unlimited frame, and traversal of all modes and all depths is not required, so that the encoding complexity can be greatly reduced to achieve the purpose of controlling the coding complexity by setting the number of the limited frames, which is based on the requirements of the application [14].The algorithm will alternately arrange two kinds of frames according to the limitation of target complexity and the actual encoding requirements, as shown in Figure 4.

Jimenez Moreno et al. proposed an efficient Complexity Control (CC) algorithm [15], which is based on the hierarchical structure of coding units. In the process of video coding, the current coding unit needs to be further divided by considering whether the actual coding time can meet the needs of the target task; the algorithm defines the early termination condition for each size of CU. More importantly, all the parameters can be dynamically generated according to the content of the video, the configuration of the encoding, and the setting of the target complexity, so that the complexity can be effectively controlled. The above algorithm can effectively control the complexity. And algorithm in [14] based on frames. For different video sequences, each frame has its own characteristics, so it should be studied on a smaller coding unit. In [15], the reduction of coding complexity is considered from the perspective of coding time and rate-distortion cost. The visual characteristics of the human eyes are not fully utilized. This paper will analyze the relationship between visual saliency and video coding complexity and propose related algorithms. Since the 64×64 largest coding unit (LCU) in HEVC is the basic unit for video coding, the characteristics of the LCU will be analyzed first. For the quadtree structure, the rate-distortion cost of coding units before and after the division is also compared during intraprediction to determine whether the division is required, as shown in Figure 5. It is required that for each size of coding unit (85 coding units ranging in size from 64×64 to 8×8), and each intraprediction mode of each coding unit (there are 35 prediction modes in each coding unit) is adopted to rate-distortion cost calculation, so the prediction coding of intra- and interframes is the most computationally complex module in video coding. This process

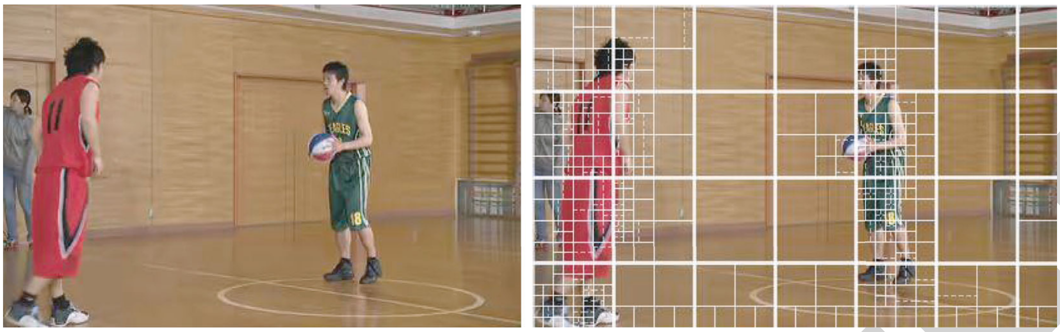


Figure 8: Division of coding units.

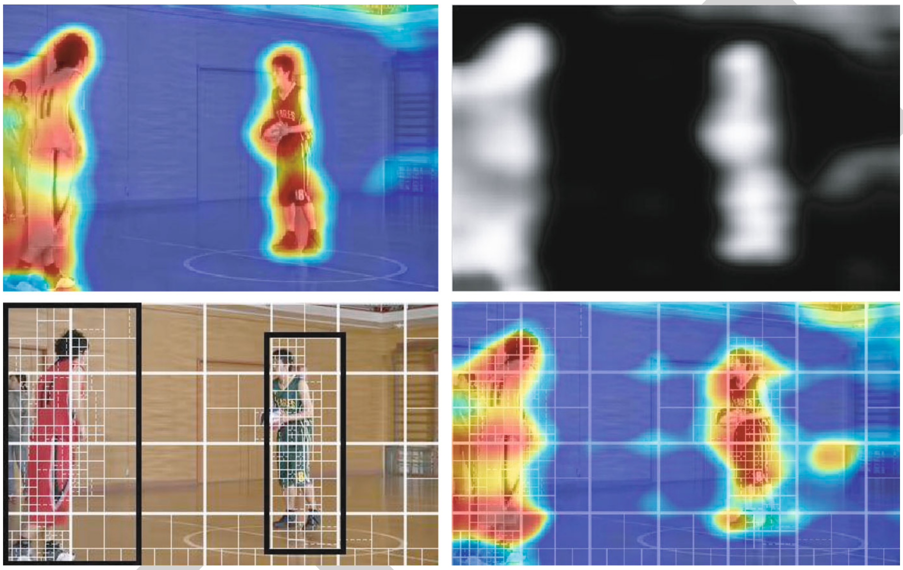


Figure 9: Division and visual saliency of basketball pass.

Table 2: Threshold of different targets.

Target	Threshold 1	Threshold 2
80%	0.12	0.14
60%	0.08	0.11
40%	0.04	0.09

is recursive traversal, regardless of the specific characteristics of the video sequence.

Select a video sequence Cactus with a resolution of 1920 × 1080 and analyze the code stream file formed after compression, then count the depth information of the optimal mode selection as shown in Figure 6. It can be seen that only about 30% of the coding units need to be divided into the maximum depth, while 70% of the coding units do not need to recurse to the maximum depth, and nearly 35% of the optimal modes have a depth of 0 or 1. Therefore, using the algorithms in Figure 6 may result in a large amount of unnecessary computation, which may reduce the overall efficiency of video coding.

4.2. Characteristics of Visual Saliency. Select test sequence Cactus with a resolution of 1920 × 1080 and count the percentage of the total time for encoding CUs of different depths in the encoding time of the whole intraprediction, as Figure 7 shows. It can be seen that the CU with a coding depth of 3 consumes more time, which is close to half of the total time of intraprediction coding, and the CU with a coding depth of 0 uses less than one tenth.

4.3. Relationship between Coding Unit Depth and Visual Saliency. Figure 8 is a certain frame in the video sequence of basketball pass of Class D with 416 × 240 resolution and its coding unit division by the HM standard algorithm. It can be seen that the area with rich motion or texture in the video has a greater depth. On the contrary, the relatively still or flat area has a smaller depth. Basketball players and basketball in the video are split into smaller units, while most of the static background, such as the ground and walls, are split into 32 × 32 coding units, and some even maintained 64 × 64 coding units.

Then generate the visual saliency map of the sequence by using the visual saliency algorithm introduced in reference [10], extract the same frame as Figure 8 to obtain its visual saliency map, and retain the regions with greater saliency. It can be seen from the figure that the greater the brightness

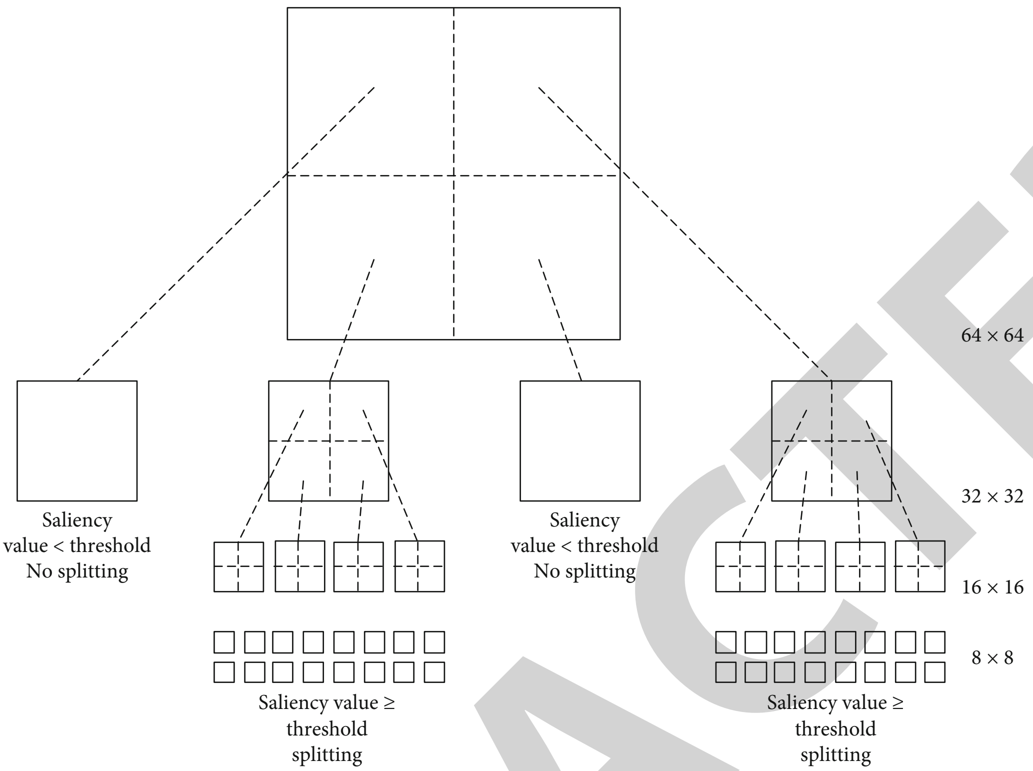

64 × 64
32 × 32
16 × 16
8 × 8
Saliency value < threshold No splitting
Saliency value < threshold No splitting
Saliency value ≥ threshold splitting
Saliency value ≥ threshold splitting

Figure 10: CU of the proposed algorithm.

of the region, the higher the saliency value, and the more it can highlight the subjective gaze target of the human eyes. From the unit division result it can be seen that the areas with high visual saliency are basically consistent with the areas with small unit division depth. As previously described, the visual characteristics of the human eyes determine that they will pay attention to some regions when watching a video, but they are not sensitive to the changes of the areas they do not pay attention to. Therefore, visual saliency can be combined with the coding process of video to reduce the complexity of video coding Figure 9.

4.4. Establishment of Model. Through the above analysis, we combine the visual saliency algorithm with the mode selection algorithm to determine the depth of the best CU, according to the saliency of the frames. If the saliency value is high, the depth of CU is larger, and the optimal mode is selected by traversing and calculating the rate-distortion cost. On the contrary, when the saliency value is low, the division with greater depth will be skipped. A larger coding module is directly adopted, so as to reduce the complexity of video coding. Firstly, the visual saliency algorithm introduced in [10] is adopted to build a saliency model and generate a visual saliency map of the video sequence. Next, threshold value 1 will be set, and when the visual saliency of the coding unit with the depth of 0 is greater than the threshold value 1, the current coding unit is split; otherwise, the depth of the current CU is 0, and the rest of the partitioning and rate-distortion cost comparisons are skipped. Threshold 2 is set in the same manner, and it is determined whether or not the 32×32 CU needs to be split. For the test sequence Cactus, the threshold values under different complexities can be obtained by counting the division results after the test. According to the statistical and the experimental results of the test sequence, the threshold is set by different targets, as shown in Table 2.

4.5. Characteristics of Visual Saliency. The division process by using the algorithm is shown in Figure 10. The dotted line represents the division operation that needs to be determined according to the threshold. If it is greater than the threshold, the division represented by the dotted line is executed. Otherwise, the partition operation is not performed.

Figure 11 shows the process flow of the algorithm during intraprediction coding, and the detailed description is as follows: (1) firstly, a saliency model of a video sequence is established, the saliency value of each pixel is obtained, and a visual saliency map is formed; (2) a mode selection process of intraprediction for a 64×64 CU, including a DC mode, a planar mode, and 33 angle prediction modes, to obtain an optimal mode of 64×64; (3) calculate an average saliency value of the 64×64 pixels and compare the average saliency value with the threshold value 1, if the average saliency value is greater than the threshold value 1, increase the depth by 1, and go to next step; otherwise, the remaining comparison and calculation steps are skipped, the division of the current CU is terminated in advance, and the depth of the optimal mode of the CU is 0; (4) a mode selection process of intraprediction for a 32×32 CU, including a DC mode, a planar mode, and 33 angle prediction modes, to obtain an optimal 32×32 mode; (5) calculate an average saliency value of 32×32

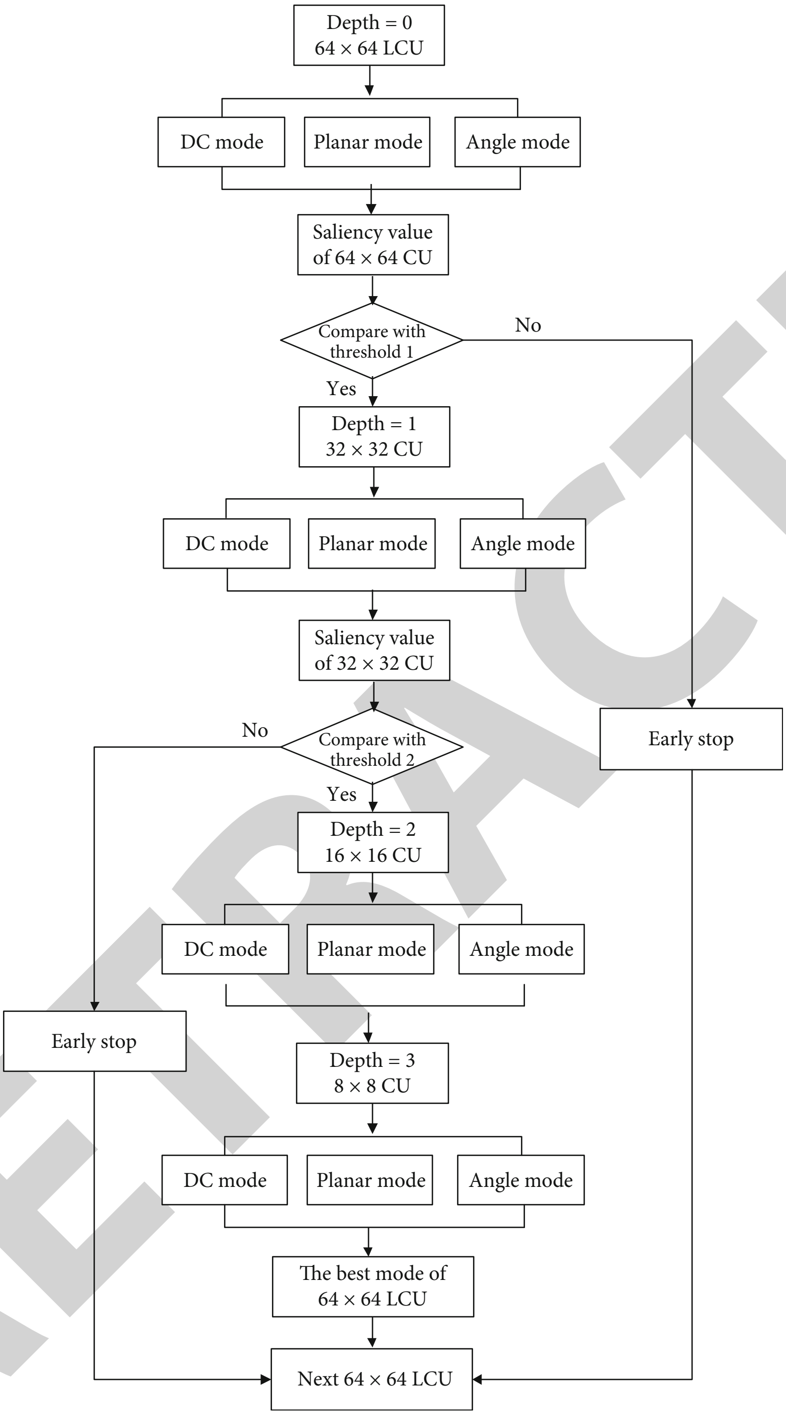


Figure 11: Division process of the proposed algorithm.

pixels and compare the average saliency value with the threshold value 2, if the average saliency value is greater than the threshold value 2, increase the depth by 1, and go to next step; otherwise, the remaining comparison and calculation steps are skipped, the division of the current CU is terminated in advance, and the depth of the optimal mode of the CU is 1; (6) a mode selection process of intraprediction for a 16×16 CU, including a DC mode, a planar mode, and 33 angle prediction modes, to obtain a 32×32 optimal mode; (7) an intraprediction mode selection process is performed on an 8×8 CU, including a DC mode, a planar mode, and 33 angle prediction modes, and an 8×8 optimal mode is obtained; (8) calculate and compare the rate-distortion cost of the $64 * 64$ CU and the sum of the rate-distortion cost of four $32 * 32$ sub-CUs, the rate-distortion costs of the $32 * 32$ CU and the sum of the rate-distortion costs of four $16 * 16$ CU sub-CUs, and the rate-distortion costs of the $16 * 16$ CU and four $8 * 8$ sub-CUs to obtain an optimal mode of a current CU; (9) The next 64×64 CU is processed.

Table 3: Result comparing.

Level	QP	BasketballDrive		Cactus		BQTerrace	
		Y-PSNR	BitRate	Y-PSNR	BitRate	Y-PSNR	BitRate
100%	22	41.11	49695.50	40.57	110334.25	42.71	179577.50
	27	39.29	21537.36	37.85	51444.30	37.79	94493.49
	32	37.82	11759.62	35.55	28035.93	34.51	51367.87
	37	35.97	6697.46	33.06	15195.26	31.53	28204.92
80%	22	41.09	49859.41	40.54	110645.62	42.65	179351.31
	27	39.29	21831.20	37.83	51957.91	37.78	94526.07
	32	37.82	11921.26	35.53	28354.69	34.50	51394.48
	37	35.96	6780.58	33.03	15327.01	31.53	28297.77
60%	22	41.04	50016.31	40.47	111627.63	42.43	182025.37
	27	39.27	22303.80	37.78	53099.25	37.69	97012.68
	32	37.80	12215.63	35.47	29018.28	34.40	52942.21
	37	35.94	6935.87	32.97	15638.85	31.43	29071.11
40%	22	40.99	50713.22	40.40	113338.02	42.24	188418.64
	27	39.24	23108.05	37.72	54625.04	37.60	102796.43
	32	37.76	12742.7	35.39	29960.68	34.23	56371.28
	37	35.88	7232.86	32.88	16117.07	31.23	30812.71

Table 4: BasketballDrive.

Coding time (%) [14]	ΔBitRate (%) [14]	ΔPSNR (dB) [14]	Coding time (%)proposed	ΔBitRate (%)proposed	ΔPSNR (dB) proposed
75%	+0.2	-0.03	77%	+0.3	-0.02
59%	+1.2	-0.21	56%	+0.7	-0.06
37%	+4.9	-0.77	39%	+2.1	-0.12

Table 5: Cactus.

Coding time (%) [14]	ΔBitRate (%) [14]	ΔPSNR (dB) [14]	Coding time (%) proposed	ΔBitRate (%) proposed	ΔPSNR (dB) proposed
76%	+0.1	-0.03	79%	+0.3	-0.03
61%	+0.5	-0.07	59%	+1.2	-0.11
41%	+5.7	-0.28	42%	+2.7	-0.17

5. Experimental Results

In order to analyze the coding performance of the algorithm, the following settings are made in the simulation experiment:

(1) the standard test platform HM-13.0 is used as a benchmark for results comparison

(2) 3 standard test sequences, BasketballDrive, BQTerrace, and Cactus, with a resolution of 1920 × 1080 in Class B were used

(3) use encoder _ intra _ main as the configuration file

(4) The objective evaluation criteria were BitRate, PSNR, and coding time

(5) The percentage of the target encoding time T_R to the standard HM-13.0 encoding time is defined as the target complexity C_R:

$$C_R = \frac{T_R}{T_{HM}} \times 100\%. \tag{29}$$

(6) The actual encoding time T_T as a percentage of that standard HM-13. 0 encoding time is defined as the encoding complexity C_T:

$$C_T = \frac{T_T}{T_{HM}} \times 100\%. \tag{30}$$

5.1. Objective Quality Assessment. Table 3 shows the test results of the algorithm on the HM-13.0 for three standard test sequences of Class B, where encoder_intra_main is used in the configuration mode, and the quantization parameters QP are set to 22, 27, 32, and 37, respectively. Under the given target complexity control, BitRate and Y-PSNR are listed, respectively. The performance of the proposed algorithm is

Table 6: BQTerrace.

Coding time (%) [14]	ΔBitRate (%) [14]	ΔPSNR (dB) [14]	Coding time (%)proposed	ΔBitRate (%)proposed	ΔPSNR (dB) proposed
72%	+1.8	-0.05	85%	+0.1	-0.06
67%	+2.1	-0.05	66%	+1.4	-0.28
43%	+2.8	-0.14	40%	+4.9	-0.48

Table 7: Average of BD-rate and BD-PSNR.

Level	BD-rate(%) [14]	BD-PSNR (dB) [14]	BD-rate(%) proposed	BD-PSNR (dB)proposed
80%	+1.279	-0.065	+1.139	-0.039
60%	+5.155	-0.223	+4.707	-0.192
40%	+11.944	-0.489	+11.073	-0.455

Table 8: BasketballDrive.

Coding time (%) [15]	ΔBitRate (%) [15]	ΔPSNR (dB) [15]	Coding time (%) proposed	ΔBitRate (%) proposed	ΔPSNR (dB) proposed
83%	+0.3	-0.01	77%	+0.3	-0.02
74%	+0.8	-0.02	69%	+0.5	-0.04

Table 9: Cactus.

Coding time (%) [15]	ΔBitRate (%) [15]	ΔPSNR (dB) [15]	Coding time (%) proposed	ΔBitRate (%) proposed	ΔPSNR (dB)proposed
85%	+0.2	-0.01	79%	+0.3	-0.03
75%	+0.8	-0.03	42%	+0.7	-0.07

Table 10: BQTerrace.

Coding time (%) [15]	ΔBitRate (%) [15]	ΔPSNR (dB) [15]	Coding time (%) proposed	ΔBitRate (%) proposed	ΔPSNR (dB) proposed
80%	+0.6	-0.02	85%	+0.1	-0.06
43%	+1.8	-0.06	71%	+0.9	-0.18

measured with luma PSNR (represented with Y-PSNR) decrease and bitrate (represented with BitRate) increase.

Tables 4–7 show the test results of the algorithm on the HM-13.0 for three standard test sequences in Class B. The encoder_intra_main is used in the configuration mode, and the quantization parameter QP is set to 22, 27, 32, and 37, respectively. The percentage of the actual encoding time in the encoding time of the HM-13.0 standard algorithm, when the target complexity control is set, is executed. The experimental results of this algorithm are compared with those of the algorithm described in reference [14] by the change of BitRate and PSNR. The better algorithm will cause less increase of BitRate and less decrease of PSNR. From these tables, the proposed algorithm is better than reference [14] with less PSNR decrease and bitrate increase, while maintaining the same coding time.

Tables 8–10 list the experimental results of this algorithm compared with the algorithm in [15] when the target complexity is set to about 80% and 70%, respectively. From these tables, the proposed algorithm is better than reference [15] with less PSNR decrease and bitrate increase, while maintaining the same coding time.

Figure 12 is R-D curves of cactus. It can be seen from the figure that the algorithm in this paper can accurately control the intracoding complexity without greatly affecting the coding performance, which are very close to the original R-D curves. The R-D curve with the target complexity of 80% is basically coincident with the HM standard algorithm and slightly deviates from the HM standard algorithm at 60% and 40%.

5.2. Subjective Quality Assessment. Figure 13(a) is a frame in the test video sequence BasketballDrive (1920 × 1080), and Figure 13(b) is its corresponding saliency map. Figures 13(c) and 13(d), respectively, show the effect comparison after the algorithm is used to divide the unit of a certain frame when the target time is set to 100%, 80%, 60%, and 40%. Figures 13(g) and 13(h) are the result of a certain frame in a test video sequence by using the algorithm for a target time of 80% and 40%, respectively. It can be seen from Figure 13(b) that in this video sequence, the human eyes mainly focus on the basketball players, but do not pay attention to the floor and walls. Therefore, the algorithm strengthens the coding and division of the regions with high visual saliency, simplifies the division of the regions with low visual saliency, and according to different target complexities. This simplified process is controlled by different thresholds, as shown in Figures 13(c) and 13(d). It can be seen from the final coding effect Figure 13 that the algorithm can effectively control the coding complexity, and at the same time, there are almost no obvious visual differences.

It can be seen from these figures that the depth of the coding unit is relatively large and the division is relatively detailed in the areas that are easy to be noticed by the human eyes. On the contrary, the depth of the region coding units to which the human eye pays less attention is small. The coding complexity is increased in a large flat area such as a floor, a wall, a river, or an area with less texture. It is not easy to improve the subjective visual quality for the human eyes, while it is helpful to improve the subjective visual quality by using more complex coding for the people and scenery that the human eyes pay attention to and the areas with more complex textures. Therefore, it is entirely possible to reduce the overall encoding complexity by reducing the number of bytes used to encode the regions with lower visual saliency. In this way, the algorithm can use less division to express the video information, so as to achieve the control of coding complexity.

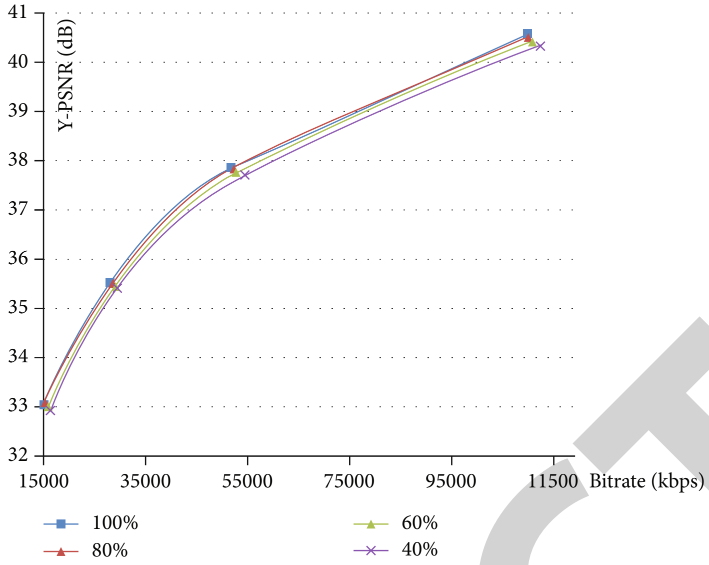
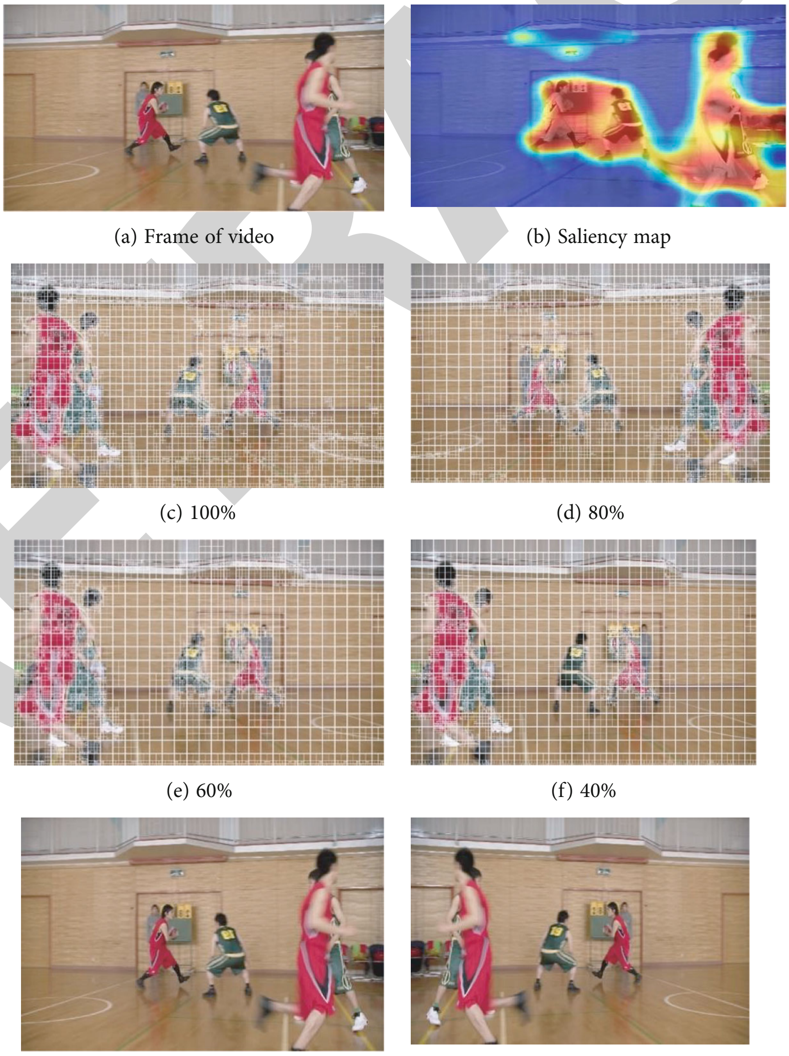


Figure 12: R-D curves of Cactus.



(a) Frame of video

(b) Saliency map

(c) 100%

(d) 80%

(e) 60%

(f) 40%

(g) 80%

(h) 40%

Figure 13: CU of BasketballDrive.

6. Conclusion

In this paper, an intraprediction unit partition algorithm based on visual saliency is proposed. Firstly, the visual saliency map of each frame of the encoded video is generated according to the visual saliency algorithm, and then the proportion of CUs with different depths in the optimal mode generated by the HM standard algorithm and the relationship between the depth of different coding units and the encoding time are analyzed. There is also the relationship between the depth of coding unit division and its visual saliency, so a unit division model based on visual saliency is established, and the threshold of division is established according to statistical data. Finally, the results of simulation experiments are given. The experimental results show that the algorithm can effectively control the encoding time under the setting of the target complexity. There is only a small degradation in the objective quality, while there is little loss in the subjective quality that is perceptible to the human eye. This method will bring higher video coding efficiency of a number of scenarios, such as real time video coding and social media video.

Data Availability

The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The work was supported by the Science and Technology Innovation Project of Tai'an, China [2021GX022].

References

[1] B. Bross, W. J. Han, G. J. Sullivan, J.-R. Ohm, T. Wiegand, and J. C. T. VC, "High efficiency video coding (HEVC) text specification draft 7," in *JCTVC 7th Meeting*, Geneva, CH, 2012.

[2] L. Zhu, Y. Zhang, S. Wang, S. Kwong, and Y. Qiao, "Deep learning-based chroma prediction for intra versatile video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 99, pp. 3168–3181, 2021.

[3] D. S. Lee, B. G. Kim, and S. K. Kwon, "Efficient depth data coding method based on plane modeling for intra prediction," *IEEE Access*, vol. 9, pp. 29153–29164, 2021.

[4] F. Pakdaman, Y. Li, M. R. Hashemi, M. Ghanbari, and M. Gabbouj, "SVM based approach for complexity control of HEVC intra coding," *Signal Processing Image Communication*, vol. 93, article 116177, 2021.

[5] Q. Li, J. Nie, and D. Yu, "Space-domain-based ctu layer rate control for HEVC," *Signal, Image and Video Processing*, vol. 15, no. 8, pp. 1873–1880, 2021.

[6] Y. Huang, L. Song, R. Xie, E. Izquierdo, and W. Zhang, "Modeling acceleration properties for flexible intra HEVC complexity control," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4454–4469, 2021.

[7] R. Yokoyama, M. Tahara, M. Takeuchi, H. Sun, and J. Katto, "CNN based optimal intra prediction mode estimation in video coding," in *2020 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1-2, Las Vegas, NV, USA, 2020.

[8] D. Wang, Y. Sun, C. Zhu, W. Li, and J. Luo, "Fast depth and mode decision in intra prediction for quality shvc," *IEEE Transactions on Image Processing*, vol. 99, 2020.

[9] J. Wang, S. Li, and Q. Zhu, "High dynamic range image compression based on visual saliency," *APSIPA Transactions on Signal and Information Processing*, vol. 9, no. 1, 2020.

[10] H. J. Seo and P. Milanfar, "Static and space-time visual saliency detection by self-resemblance," *Journal of Vision*, vol. 9, no. 12, pp. 15–27, 2009.

[11] J. Xu, X. Guo, T. Qin, C. Li, and A. Men, "A novel video saliency map detection model in compressed domain," in *MILCOM 2015-2015 IEEE Military Communications Conference*, Tampa, FL, USA, 2015.

[12] H. Hadizadeh and I. V. Bajic, "Saliency-aware video compression," *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 23, no. 1, pp. 19–33, 2014.

[13] X. Deng, M. Xu, L. Jiang, X. Sun, and Z. Wang, "Subjective-driven complexity control approach for HEVC," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 26, no. 1, pp. 91–106, 2016.

[14] G. Correa, P. Assuncao, L. Agostini, and L. S. Cruz, "Complexity control of high efficiency video encoders for power-constrained devices," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 1866–1874, 2011.

[15] A. Jimenez Moreno, E. Martinez-Enriquez, and F. Diaz-De-Maria, "Complexity control based on a fast coding unit decision method in the HEVC video coding standard," *IEEE Transactions on Multimedia*, vol. 18, no. 4, pp. 563–575, 2016.