Hindawi
Journal of Electrical and Computer Engineering
Volume 2020, Article ID 4354296, 17 pages
https://doi.org/10.1155/2020/4354296

Research Article

Optimized Rate Control Algorithm of High-Efficiency Video Coding Based on Region of Interest

Renjie Song and Yuandong Zhang

School of Computer Science, Northeast Electric Power University, Jilin 132012, China

Correspondence should be addressed to Yuandong Zhang; 1406632033@qq.com

Received 21 August 2019; Revised 18 December 2019; Accepted 29 January 2020; Published 27 February 2020

Academic Editor: Jar Ferr Yang

Aiming at the problems that the strategy of target bit allocation at the CTU layer has deviations from the human subjective observation mechanism, and the update phase of parametric model has a higher complexity in the JCTVC-K0103 rate control algorithm of ITU-T H.265/high efficiency video coding (HEVC) standard. Optimized rate control (ORC) algorithm of ITU-T H.265/HEVC based on region of interest (ROI) is proposed. Firstly, the algorithm extracts the region of interest of video frames based on time and space domains by using the improved Itti model. Then, the weight of target bits w is recalculated based on space-time domains to improve the rate control accuracy, and the target bits are distributed based on ROI by the adaptive weight algorithm once again to make the output videos more attuned with the human visual attention mechanism. Finally, the quasi-Newton algorithm is used to update the rate distortion model, which reduces the computational complexity in the update phase of the parametric model. The experimental results show that the ORC algorithm can obtain a better subjective quality in the compressed results with less bit error compared with the other two algorithms. Meanwhile, the rate distortion performance of the ORC algorithm is better on the premise of guaranteeing rate control performance.

1. Introduction

With the improvement of videos in the aspects of clarity and quality, the traditional video coding standard already cannot meet the coding requirements of high resolution videos. In order to catch up with the developed trend of high resolution videos and the requirements of coding technology, the video coding joint working group JCT-VC (Joint Collaborative Team on Video Coding) has developed a new video coding standard based on ITU-T H.264/AVC in 2010, named ITU-T H.265/High Efficiency Video Coding (HEVC) standard [1]. The rates can be reduced by more than 50% of ITU-T H.265/HEVC compared with ITU-T H.264/AVC. In the transmission process of high-resolution videos, the higher requirements are proposed for network bandwidth in order to avoid compressed coding stream in the buffer of limited bandwidth from overflow and to ensure a reasonable bit allocation strategy to generate videos with minimum loss of rate distortion performance. JCT-VC has proposed two solutions to control the rates of ITU-T H.265/HEVC, the JCTVC-H0213 rate control algorithm based on $R-Q$ model, and the JCTVC-K0103 rate control algorithm based on $R-\lambda$ model, which are the most representative [2]. The JCTVC-K0103 rate control algorithm has better rate control effects and less fluctuation of bits compared with the JCTVC-H0213 rate control algorithm. However, the JCTVC-K0103 rate control algorithm does not consider the characteristics of video content in the process of bit allocation at the CTU layer, which causes the inaccurate results of bit allocation to affect the quality of output videos and the accuracy of the rate control algorithm. At the same time, the gradient descent algorithm in the update phase of rate distortion model has high computational complexity, which increases the complexity of rate control algorithm. Therefore, optimizing the JCTVC-K0103 rate control algorithm to improve the compression performance under the premise of guaranteeing the performance of rate control algorithm has become a hot point in the coding area of videos.

Aiming at the shortcomings of K0103 rate control algorithm, many scholars have done a lot of research at home and abroad, which appears in the aspects of the complexity measurement of CTU and the computational load of parametric model. In the aspect of the complexity measurement of CTU, Guo et al. [3] proposes a method which calculates the complexity of CTU based on the pixel statistical methods, but the bits allocated will have a large error of video sequences with severe local motion. In [4], SATD is taken as the measurement of complexity, which ignores the relevant characteristics of video content. In [5], differential histograms are adopted to measure the complexity of CTU, which has the uncertainty of the selection for threshold in different video sequences. The above algorithms improve the accuracy of complexity of CTU to a certain extent, but the overall performance of rate control algorithms is not improved obviously. In the aspect of the computational load of parametric model, in [6], the rate distortion of the parametric model is improved, but the adaptability is poor when the scenes of videos are transforming, and the computational load is also large. In [7], the gradient descent method is improved in the original parametric model, but with the increment of iterations, the computational load is also increasing gradually. The above algorithms improve the performance of the parametric model to a certain extent, but the computational load reduced is not obvious.

The above algorithms are improved unilaterally from the complexity measurement of CTU and the computational load of the parametric model. The overall performance of optimized rate control algorithms is not obvious. In this paper, the rate control algorithm is optimized between the complexity measurement of CTU and weight assignments of bits at the CTU layer and the computational load of parametric model. Firstly, the improved Itti model is used to extract ROI of video frames. Then, the complexity at the CTU layer is calculated based on ROI, and the adaptive weight algorithm is used to redistribute the target bits combining with the complexity of CTU, which makes the output videos more attuned with human visual attention mechanism. Finally, the quasi-Newton method is used to update the parametric model, which reduces the computational load of the update phase of the parametric model. Therefore, the overall performance of rate control algorithm is improved.

2. Analysis of Algorithms

2.1. JCTVC-K0103 Rate Control Algorithm. The JCTVC-K0103 rate control algorithm allocates reasonable number of bits to each coding layer, given certain target rates based on the $R-\lambda$ model, to optimize the coding performance, which includes the following two steps specifically:

(1) Performing target bit allocation of GOP layer, image layer, and CTU layer hierarchically according to target bit rate

(2) The quantization parameter QP is determined by the $R-\lambda-\mathrm{QP}$ model according to target bit allocation corresponding to the hierarchy

2.1.1. Target Bit Allocation at the CTU Layer. The bit allocation of each layer in the K0103 rate control algorithm is a dynamic process. It is necessary to refer to the actual number of bits in the current coding layer when performing the allocation of target bits in the next layer and to ensure that the number of target bits allocated of current coding layer is smaller than the total number of target bits allocated. The implementation process of target bit allocation at the CTU layer is as follows.

After the number of bits is allocated at the image layer, the target bit allocation of CTU layer depends on parameter R_i, the number of target bits per frame. And equation (1) [2] of target bit allocation at the CTU layer is as follows:

$$R_{\mathrm{CTU}} = \frac{R_i - R_{\mathrm{header}} - R_{\mathrm{compic}}}{\sum_{c=k}^{N_L} W_c} W_k, \tag{1}$$

where R_{header} represents the information of data header encoded, which includes the GOP flag bit, the frame flag bit, and so on; R_{compic} represents the actual number of bits of CTU encoded in the current frame; N_L represents the total number of CTU in the current frame, $c = [1, N_L]$; W_k represents the weight of bit allocation in the kth CTU; W_c represents the weight of bit allocation of the cth CTU.

In the process of bit allocation at the CTU layer, the weight W_k of bit allocation of each CTU affects the compressed quality of videos directly. In the K0103 rate control algorithm, the weight allocated W_k can reflect the texture complexity of current CTU. Generally, CTU with high texture complexity need to be allocated more bit rates, and the value W_k should be larger at this time; CTU with low texture complexity need to be allocated less bit rates, and the value W_k should be smaller at this time. The K0103 rate control algorithm uses the mean absolute difference (MAD) to characterize the texture complexity of current CTU based on this strategy [8]. Equation (2) is as follows:

$$\mathrm{MAD} = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left| I(x, y) - I'(x, y) \right|, \tag{2}$$

where M and N represent the width and height of current CTU, respectively; $I(x, y)$ represents the pixel value of current CTU; $I'(x, y)$ represents pixel value reconstructed of current CTU.

According to the definition of MAD, the predicted value of MAD is only taken as the weight of target bit allocation in the process of target bit allocation at the CTU layer, and the accuracy is low, which affects the compressed quality of videos ultimately. Therefore, the weight of bit allocation at the CTU layer needs to be adjusted.

2.1.2. Implementation of Target Bit. The purpose of implementation of target bit is converting the value of target bit allocation into the value of quantization parameter QP, and the compression ratio and the final encoding rate are determined by QP. After the value of quantization parameter is determined, it is necessary to update the parameters and adjust the relevant parameters. The implementation process of target bits is completed at the image layer and CTU layer, and the implementation ideas are same basically. The implementation steps of target bits at the CTU layer are as follows:

The K0103 rate control algorithm uses the $R-D$ hyperbolic model to simulate the relationship between rate and distortion:

$$D = CR^{-K}, \tag{3}$$

where D represents the distortion values, C and K represent the parameters related to the characteristics of video content, and R represents the value of target bit allocation. In the rate-distortion optimization theory, the Lagrangian multiplier λ represents the absolute value of tangent slope in the rate-distortion curve, and the derivative of equation (3) is as follows:

$$\lambda = -\frac{\partial D}{\partial R} = C \cdot K \cdot R^{-K-1} = \alpha \cdot R^{\beta}, \tag{4}$$

where $\alpha = C \cdot K$, $\beta = -K - 1$, and α and β represent the parameters related to the characteristics of video content. A large number of experimental studies have shown that [9] there is a linear relationship between the quantization parameter QP and the Lagrangian multiplier λ:

$$\mathrm{QP} = 4.2005 \ln \lambda + 13.7122. \tag{5}$$

Under the premise of target rates known, the QP can be determined by adjusting α and β. In the implementation process of target bits at the image layer or CTU layer, the relationship between λ and the value of target bits satisfies:

$$\lambda = \alpha \cdot \mathrm{bpp}^{\beta}, \tag{6}$$

where bpp represents the average number of bits per pixel in the image or CTU determined by the value of target bit allocation at the image layer or CTU layer. In the determining process of QP, λ can be determined by α and β firstly, and the relationship between λ and QP determines the quantization parameter QP.

After performing encoding work of the current image or CTU, the parameters α and β are adjusted according to the actual number of bits encoded in the current image or CTU. The updating methods of parameters α and β are specifically as follows:

$$\begin{cases} \lambda_{\mathrm{com}} = \alpha_{\mathrm{old}} \cdot \left(\mathrm{bpp}'\right)^{\beta_{\mathrm{old}}}, \\ \alpha_{\mathrm{new}} = \alpha_{\mathrm{old}} + 0.1 \cdot \left(\ln \lambda_{\mathrm{old}} - \ln \lambda_{\mathrm{com}}\right) \cdot \alpha_{\mathrm{old}}, \\ \beta_{\mathrm{new}} = \beta_{\mathrm{old}} + 0.05 \cdot \left(\ln \lambda_{\mathrm{old}} - \ln \lambda_{\mathrm{com}}\right) \cdot \ln \mathrm{bpp}', \end{cases} \tag{7}$$

where λ_{com} represents the Lagrangian multiplier of image or CTU encoded, λ_{old} represents the Lagrangian multiplier of current image or CTU, α_{old} and β_{old} are the parameter values of current image or CTU, bpp$'$ represents the average number of bits per pixel in the image or CTU determined by the value of target bit allocation at the current image layer or CTU layer, and α_{new} and β_{new} represent the values updated of parameters encoded.

The K0103 rate control algorithm uses the gradient descent method to update and adjust the parameters α and β. However, the gradient descent method has a slow convergence rate, which leads to higher computational complexity for K0103 algorithm, and it is not conducive to practical application.

3. Extraction of ROI from Videos Based on Improved Itti Algorithm

To address the deficiencies of the Itti algorithm such as incomplete ROI of image and the edge blur, this paper improves the Itti algorithm to extract the ROI of video sequences based on picture contents in both spatial and temporal domains. First, the algorithm extracts texture and shape features in the spatial domain and extracts motion feature in time domain on the basis of the extraction of brightness, color, and orientation features. Then, the normalized step and the cross-scale step of Itti model are adopted to generate six single-feature salient maps, the brightness feature maps, the color feature maps, the orientation feature maps, the texture maps, the shape feature maps, and the motion feature maps. Finally, the information entropy theory is used to acquire the weights of six single-feature salient maps adaptively, and the cross-scale fusion is carried out to extract the final ROI of video sequences.

3.1. Texture Feature. The characteristics of human visual system (HVS) indicate that [10] human eyes have different attention to different regions in the image, and the texture-rich regions or the moving objects in the image are more likely to attract the attention of human eyes. The texture feature is extracted as a subfeature of the Itti model in this paper. Chiranjeevi and Sengupta [11] propose that the structural tensor matrix can represent the texture feature of image well, but the problem of inaccurate positioning of location exists. As a result of the isotropic Sobel operator having more accurate weighted coefficients of position, the isotropic Sobel operator is used to improve the structural tensor matrix to extract the texture feature of video frames, and the directional templates of isotropic Sobel operator are shown in Figure 1.

For the image $I(x, y)$, let I_x and I_y represent the horizontal gradient and vertical gradient of image, respectively. Then, the structural tensor matrix M of image is

$$M = \begin{bmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{bmatrix}, \tag{8}$$

where $I_{11} = \overline{I}_x^2$, $I_{12} = I_{21} = I_x \overline{I}_y$, and $I_{22} = \overline{I}_y^2$.

Calculating the horizontal gradient G_x and the vertical gradient G_y of isotropic Sobel operator according to Figure 1. The calculation equations are as follows:

$$\begin{aligned} G_x &= P_{i+1,j-1} + \sqrt{2} \times P_{i+1,j} + P_{i+1,j+1} - P_{i-1,j-1} - \sqrt{2} \times P_{i-1,j} \\ &\quad - P_{i-1,j+1}, \\ G_y &= P_{i-1,j+1} + \sqrt{2} \times P_{i,j+1} + P_{i+1,j+1} - P_{i-1,j-1} - \sqrt{2} \times P_{i,j-1} \\ &\quad - P_{i+1,j-1}, \end{aligned} \tag{9}$$

−1	0	1
−√2	0	√2
−1	0	1

(a)

−1	−√2	−1
0	0	0
1	√2	1

(b)

Figure 1: The directional templates of isotropic sobel operator. (a) Horizontal gradient direction. (b) Vertical gradient direction.

where G_x and G_y are replaced with I_x and I_y, respectively, and $I_{11} = G_x^2$, $I_{12} = I_{21} = G_x G_y$, and $I_{22} = G_y^2$. Since the eigenvalues λ_1 and λ_2 can represent the overall trend of grayscale in the window and the contrast in the direction of eigenvector in the structural tensor matrix, the consistency T of video frames can be calculated by λ_1 and λ_2:

$$\lambda_{1,2} = \frac{I_{11} + I_{12} \pm \sqrt{(I_{11} - I_{12})^2 + 4I_{12}^2}}{2}, \qquad (10)$$

$$T = |\lambda_1 - \lambda_2| = \sqrt{(I_{11} - I_{12})^2 + 4I_{12}^2}.$$

According to the central-peripheral difference algorithm (C-S algorithm) of the Itti model [12], the calculation formula of texture feature maps $T(c, s)$ obtained by the consistency T of video frames is

$$T(c, s) = |T(c) \ominus T(s)|, \qquad (11)$$

where $c \in \{2, 3, 4\}$ represents the center of receptive field, $s = c + \delta$ ($\delta \in \{3, 4\}$) represents the periphery of the receptive field, and $\ominus$ represents the difference operator among different scale feature maps.

The normalized function $N(\cdot)$ is used to perform cross-scale fusion of six texture feature maps to form a single-feature salient map $\overline{T}$:

$$\overline{T} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(T(c, s)). \qquad (12)$$

In summary, the texture feature salient map can be obtained.

3.2. Shape Feature. Aiming at the problem that the edges of salient maps are blur by the Itti algorithm, many scholars have done a lot of research. Long and Wu [13] adopts the improved Canny operator to extract the shape feature to locate the edges of salient maps accurately, but the computational load is large. In this paper, the boundary function is used to analyze the consistency of shape. The SUSAN corner detection algorithm is used to extract the shape feature of video frames based on the idea of corner point; the corner detection algorithm has the advantages of simple calculations, accurate positioning, and strong antinoise ability compared with the traditional edge detection algorithms and Harris corner algorithm, KLT corner algorithm, Kitchen–Rosenfeld algorithm, and so on [14].

The steps that the SUSAN corner detection algorithm extracts the shape feature as follows:

(1) Firstly, defining a graphic template containing 37 pixels to slide on the video frames and determining whether the pixel belongs to the USAN region. The discriminant is as follows:

$$c(r, r_0) = \begin{cases} 1, & |I(r) - I(r_0)| \le t, \\ 0, & |I(r) - I(r_0)| > t, \end{cases} \qquad (13)$$

where r represents the length of neighborhood pixels from the central point, r_0 represents the point of central location, and t represents the value of similarity demarcation.

In order to obtain more stable results, the similarity calculations of pixel points are performed in the following equation:

$$c(r, r_0) = e^{-((I(r) - I(r_0))/t)^6}. \qquad (14)$$

(2) Calculating the total similarity $n(r_0)$,

$$n(r_0) = \sum_{r \in D(r_0)} c(r, r_0), \qquad (15)$$

where $D(r_0)$ represents the area of graphic template centered on r_0.

(3) According to $n(r_0)$ obtained, the initial corner points are determined by using the corner response function:

$$R(r_0) = \begin{cases} g - n(r_0), & n(r_0) < g, \\ 0, & n(r_0) \ge g, \end{cases} \qquad (16)$$

where g is introduced to eliminate the effects of noise.

(4) A final set of corner points S is obtained by the operation of nonmaximum suppression among the initial corner points:

$$S = \{(x_i, y_j)\}, \quad i = 1, \ldots, M; \, j = 1, \ldots, N, \qquad (17)$$

where x_i and y_j represent the coordinates of corner points, respectively, and M and N represent the width and height of video frames, respectively.

(5) Since the boundary function can describe the border of objects, the elements which are in the set of corner points S mentioned are brought into the coordinate function $\delta(k)$, and the coordinate function is convolved with the Gaussian kernel linearly at the scale σ to obtain the boundary function $\delta(\sigma, k)$ corresponding to the set of corner points. The boundary function $\delta(\sigma, k)$ is taken as a measure of shape feature in this paper:

$$\begin{gathered} \delta(k) = \left[x(k) - O_a(k) - O_b\right], \quad k = 1, \ldots, N, \\ \delta(\sigma, k) = \left[x(k) * g(\sigma), y(k) * g(\sigma)\right], \end{gathered} \tag{18}$$

where N represents the length of border, (O_a, O_b) represents the center of regions, and $g(\sigma)$ represents the Gaussian kernel function at the scale σ.

(6) According to the C-S operation, the calculation formula of shape feature maps obtained by the boundary function $\delta(\sigma, k)$ is as follows:

$$\delta(c, s) = |\delta(c) \Theta \delta(s)|. \tag{19}$$

(7) The normalized function $N(\cdot)$ is used to perform the cross-scale fusion of six shape feature maps to form a single-feature salient map $\overline{\delta}$:

$$\overline{\delta} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(\delta(c, s)). \tag{20}$$

In summary, the shape feature salient map can be obtained.

3.3. Motion Feature. Aiming at the problem that the salient maps extracted by Itti model are incomplete, a large number of scholars have studied it. Jalink [15] proposes to improve the integrity of salient maps by using the MoSIFT algorithm to calculate the motion feature, but the MoSIFT algorithm has high computational complexity, which affects the real-time in the extracted process of salient regions. The human visual mechanism indicates [10] that the description of moving objects by human eyes is localized and the description of feature points by the SURF algorithm is also localized. Therefore, the description of motion feature by the SURF algorithm is more attuned with the human visual attention mechanism. The SURF algorithm is used to improve the MoSIFT algorithm to extract motion feature of video frames based on the idea in this paper, which reduces the computational complexity of the MoSIFT algorithm and obtains more stable motion feature. The steps of the MoSIFT algorithm are shown in Figure 2.

The steps of extracting motion feature by the improved MoSIFT algorithm are as follows:

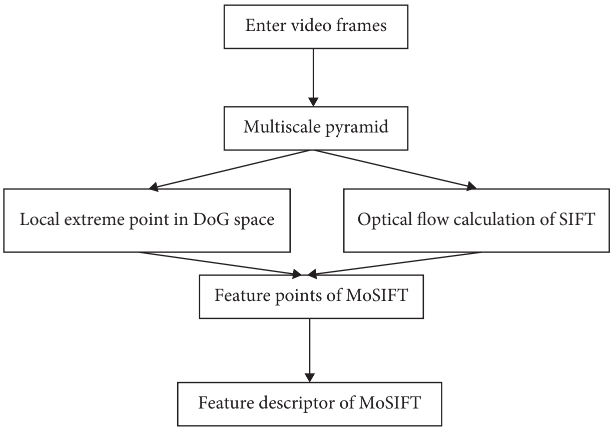

Enter video frames
Multiscale pyramid
Local extreme point in DoG space
Optical flow calculation of SIFT
Feature points of MoSIFT
Feature descriptor of MoSIFT

Figure 2: The flow chart of the MoSIFT algorithm.

(1) In the construction phase of Hessian matrix, the SURF algorithm uses Hessian matrix to extract the feature points. Since the elements which are in the Hessian matrix are calculated by second-order Taylor expansion, the computational complexity is high. Based on the theory of linear scale space (LOG), the derivation of function is equal to the convolution between the function and the derivative corresponding to the Gaussian function. And, the elements in the Hessian matrix are calculated by

$$\frac{\partial f(x, y)}{\partial x} = f(x, y) * \frac{\partial G(x, y)}{\partial x}, \tag{21}$$

where $f(x, y)$ represents a quadratic function corresponding to the image and $G(x, y)$ represents a Gaussian function corresponding to the quadratic function.

(2) The SURF algorithm is used for downsampling. Compared with the SIFT algorithm, the SURF algorithm keeps the size of frames unchanged and changes the size of filter for downsampling, which reduces the computational complexity in the process of downsampling. Each pixel processed by Hessian matrix is compared with 26 pixels in the two-dimensional image space and the neighborhood scale space, which locates the preliminary key points and filters out the key points with weak energy and error location to determine the final stable SURF feature points $D(\varphi_i, \theta)$, $i = 1, \ldots, 64$. In the determination stage of direction of feature points, the total points of horizontal and vertical Harr wavelet feature in the 60-degree fan are counted in the circular neighborhood of feature points, and the direction of the longest vector is selected as the direction of feature points by traversing the entire image. The optical flow $D(\varphi_j, \theta)$, $j = 1, \ldots, 64$ corresponding to the scale of SURF is calculated according to Figure 2.

(3) Combining the SURF feature points with the optical flow to extract the descriptors of feature points

$D(\varphi_k, \theta)$, $k = 1, \ldots, 64$, which are regarded as motion features in this paper, we get

$$D(\varphi_k, \theta) = \omega_1 \cdot D(\varphi_i, \theta) + \omega_2 \cdot D(\varphi_j, \theta), \qquad (22)$$

where ω_1 and ω_2 represent weights which are constants, and the motion feature are obtained by weighting the 64-dimensional vector of SURF and the corresponding 64-dimensional vector of optical flow. The specific description is that the values calculated by 64-dimensional Harr wavelet feature in SURF and the values calculated by the corresponding 64-dimensional Harr wavelet feature in optical flow are weighted to compose the 64-dimensional feature descriptors.

(4) According to the C-S operation, the calculation formula of motion feature maps obtained by the feature descriptors $D(\varphi_i)$ is as follows:

$$D(c, s) = |D(c) \Theta D(s)|. \qquad (23)$$

(5) The normalized function $N(\cdot)$ is used to perform the cross-scale fusion of six motion feature maps to form a single feature salient map $\overline{D}$:

$$\overline{D} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(D(c, s)). \qquad (24)$$

In summary, the motion feature salient map can be obtained.

3.4. Feature Fusion Based on Adaptive Weight. The Itti algorithm adopts the mean method to assign weights to each single feature salient map, which ignores the contribution of each feature salient map to the final salient map, which affects the overall performance of feature salient map merged. Shannon's information entropy theory [16, 17] can describe the overall statistical characteristics of source objectively, which can describe the contribution rate of the current impact factor to the whole. This paper determines the adaptive weight coefficient of each single feature salient map based on the information entropy theory, which can improve the accuracy of video frames merged.

Let a random variable of a single feature salient map is $I(A_i)$, $i = 1, \ldots, n$ and its information entropy can be represented by

$$H(X) = \sum_{i=1}^{n} P(A_i) \log_2 \frac{1}{P(A_i)}, \qquad (25)$$

where $P(A_i)$ represents the probability when removing the ith signal and $H(X)$ represents the information entropy of single feature salient map.

The values of information entropy of the brightness feature salient maps I, the color feature salient maps C, the orientation feature salient maps $\overline{O}$, the texture complexity salient maps $\overline{T}$, the shape feature salient maps δ, and the motion salient maps $\overline{D}$ obtained are calculated according to equation (25), respectively, shorted for H_1, H_2, H_3, H_4, H_5, and H_6. The adaptive weight coefficient α_i is calculated as follows:

$$\alpha_i = \frac{H_i}{H} \, (i = [1, 6]), \qquad (26)$$

where H represents the sum of H_1, H_2, H_3, H_4, H_5, and H_6, where $\sum_{i=1}^{6} H_i = 1$.

The calculation formula of the final salient map is

$$S = \alpha_1 \times N(\overline{I}) + \alpha_2 \times N(\overline{C}) + \alpha_3 \times N(\overline{O}) + \alpha_4 \times N(\overline{T}) + \alpha_5 \times N(\overline{\delta}) + \alpha_6 \times N(\overline{D}). \qquad (27)$$

In summary, the extraction of ROI of video frames can be completed, which solves the problem that the Itti model extracts the incomplete ROI and the positioning edge is blur. In order to verify the effectiveness of the above algorithm, three pictures from MSRA, the 208th frame of the tennis sequences, and the 25th frame of the basketball sequences provided by JCT-VC are used to perform experiments comparatively for the ORC algorithm, the Itti algorithm [18], the GBVS algorithm [19], the SR algorithm [20], the FT algorithm [21], the CAS algorithm [22], and the LC algorithm [23].

As shown in Figure 3, the ORC algorithm can extract the ROI in the image or video frame accurately. And, the ORC algorithm performs better in terms of the integrity of ROI extracted and the positioning accuracy of edge in the ROI compared with the other six algorithms, which proves the effectiveness of the ORC algorithm.

4. Target Bit Allocation of CTU Layer Based on Space-Time Domain

Aiming at the problem that MAD used as the weight index of CTU cannot measure the complexity of current CTU accurately in the process of target bit allocation, many scholars have studied it. Khoshnevisan and Salmasi [24] propose to take the gradient as the weight index of CTU, but ignored the contribution of influencing factors in time domain. Studies have shown that [25] the gradient of each pixel in the frame has a linear relationship with the bits allocated. The bits bpp allocated of CTU are taken as the allocated weight of the complexity of space-time domain based on ROI, the gradient T is used to measure the complexity of CTU, and the weight of bit allocation ω is redistributed at the CTU layer, which improves the accuracy of rate control algorithm. Then, the weight of bit allocation ω of CTU layer is distributed once again by the adaptive weight algorithm, which makes the output videos more attuned with human visual attention mechanism.

As a result of human eyes being sensitive to the gradient information in the image, we take gradient T as a measure of complexity of current CTU:

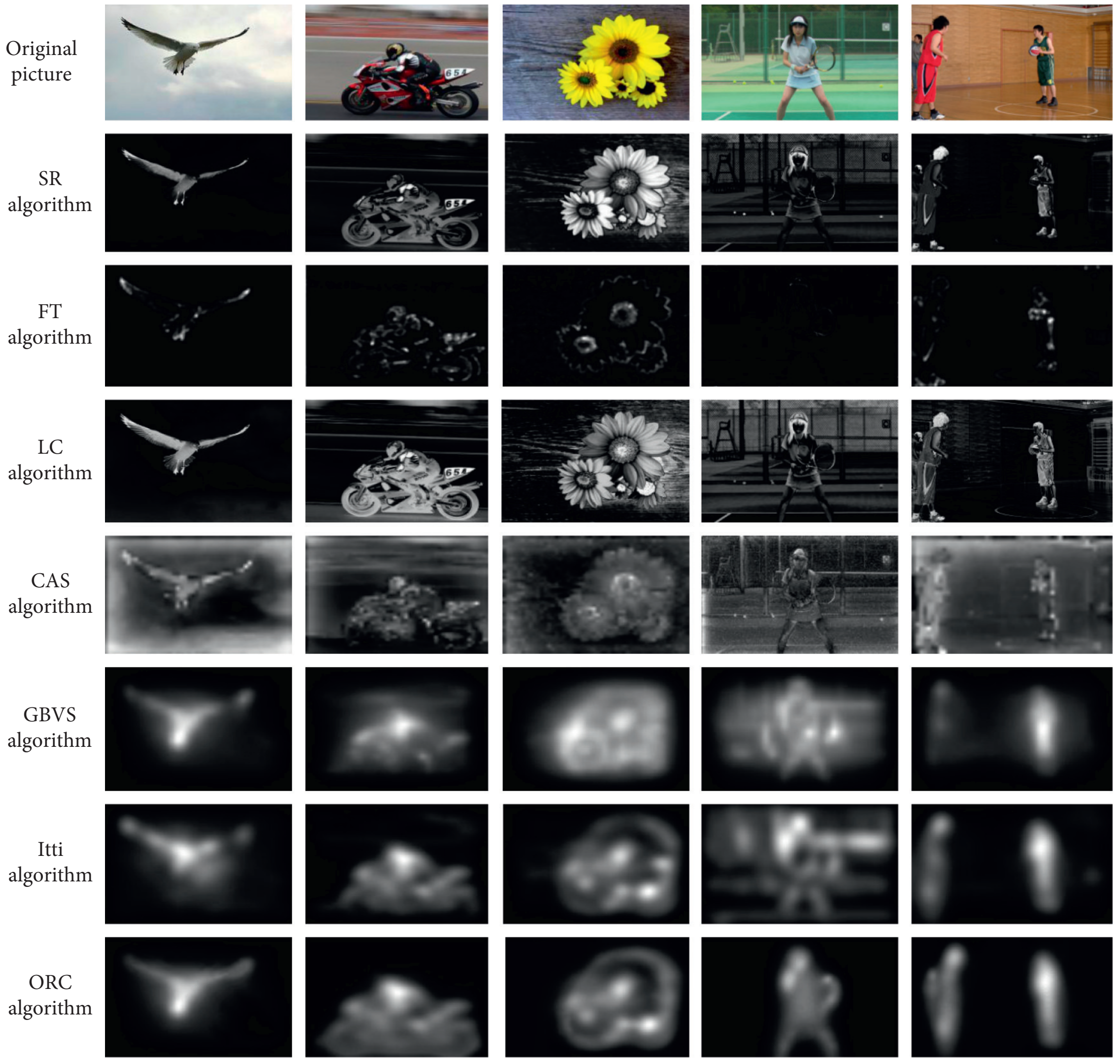


Figure 3: Experimental extraction results of ROI by seven algorithms.

$$T=\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(\left|I_{i,j}-I_{i+1,j}\right|+\left|I_{i,j}-I_{i,j+1}\right|\right)}{M\times N}, \tag{28}$$

where M and N represent the height and width of current CTU, respectively, and $I_{i,j}$ represents pixel value of the component of brightness at the position (i, j) of CTU.

In equation (6), bpp represents the average number of bits per pixel in the image or CTU determined by the value of target bit allocation at the current image layer or the CTU layer; we take bpp as the average number of bits per pixel in the CTU determined by the value of target bit allocation at the current CTU layer. The average number of bits bpp of current CTU is as follows:

$$\mathrm{bpp}=\frac{\mathrm{bpp}_{\mathrm{cf}}}{N_T}, \tag{29}$$

where $\mathrm{bpp}_{\mathrm{cf}}$ represents the total number of target bits in the current frame and N_T represents the total number of CTU in the current frame.

As a result of the CTU of current frame and the CTU of adjacent frame having the similar texture feature, the complexity of current CTU is measured by the complexity of current CTU and the complexity of CTU corresponding to the adjacent frame. The weight of bit ω_{new} allocated of current CTU is

$$\omega_{\mathrm{new}}=\mathrm{bpp}_{\mathrm{cur}}\times C_{\mathrm{cur}}+\mathrm{bpp}_{\mathrm{nei}}\times C_{\mathrm{nei}}, \tag{30}$$

where $\mathrm{bpp}_{\mathrm{cur}}$ represents the average number of bits allocated for the CTU in current frame, $\mathrm{bpp}_{\mathrm{nei}}$ represents the average number of bits allocated for the CTU in adjacent frame corresponding to the current CTU, C_{cur} represents the texture complexity obtained of current CTU according to equation (28), and C_{nei} represents the texture complexity obtained of CTU corresponding to adjacent frame according to equation (28).

The redistribution of weight of bits at the CTU layer can be achieved by following the above steps.

The human visual attention mechanism indicates that the attentiveness of human eyes from video frames is distributed in the central regions, while the attentiveness of human eyes from the peripheral regions of video frames is small [10]. We adjust the weight allocated to the target bits at the CTU layer to realize the redistribution of target bits once again based on the idea.

According to Figure 4, we determine the central coordinates (x_1, y_1) of CTU in the current frame and the central coordinates (x_2, y_2) of CTU coding in the current frame firstly, and the Manhattan distance is used to determine the weight of CTUα in the ROI and the weight of CTUβ in the RONI in the current frame. The calculation formula is as follows:

$$\begin{cases} \alpha = |x_1 - x_2| + |y_1 - y_2|, & \text{CTU} \in \text{ROI}, \\ \beta = |x_1 - x_2| + |y_1 - y_2|, & \text{CTU} \in \text{RONI}. \end{cases} \tag{31}$$

After the weights α and β are obtained, equation (32) designed is used to adjust the weight ω_{new} once again:

$$\omega_n = \omega_{\text{new}} \times \left(1 - \frac{\alpha}{\alpha + \beta}\right). \tag{32}$$

In summary, the calculation of weight of bits at the CTU layer can be completed. The more bits are allocated to ROI of high texture complexity in the compressed frames, and the less bits are allocated for RONI of low texture complexity in the compressed frames, which makes the output videos more attuned to human visual attention mechanism.

In order to verify the effectiveness of the ORC algorithm, the 25th frame of the BasketballPass sequences and the 208th frame of the tennis sequences provided by JCT-VC are used for the experiment. Figure 5 shows the experimental results of the ORC algorithm and the K0103 algorithm of the HM10.0 model.

(a) The original image of the 208th frame of Tennis sequences

(b) The experimental result of the K0103 algorithm

(c) The experimental result of the ORC algorithm

(d) The 25th frame of BasketballPass sequences

(e) The experimental result of the K0103 algorithm

(f) The experimental result of the ORC algorithm

From the perspective of vision, the subjective quality of video frames compressed by the ORC algorithm is better than the subjective quality of the K0103 algorithm in the HM10.0 model in Figure 5. The quality of videos compressed is improved by the ORC algorithm. To further illustrate the effectiveness of the ORC algorithm, the enlarged details of the 208th frame of Tennis sequences are displayed for comparison. Figures 6 and 7 show the results of ROI and RONI in enlarged details.

We can see that the subjective quality of ROI by the ORC algorithm is better than the K0103 rate control algorithm from Figure 6. The value of QP of CTU is 29 in Figure 6(a) by the code stream analysis software, Elecard HEVC analyzer, and the value of QP of CTU is 28 in Figure 6(b), which indicates that the ORC algorithm can allocate more rates to the ROI. We can see that the subjective quality of RONI by the K0103 rate control algorithm is better than the ORC algorithm from Figure 7. The value of QP of CTU is 30 in Figure 7(a) by the code stream analysis software, and the value of QP of CTU is 31 in Figure 7(b), which indicates that the ORC algorithm reduces the allocation of rates to the RONI. On the basis of above analysis, it is verified that the ORC algorithm can redistribute the rates of RONI and ROI in the video frames, which makes the output videos more attuned with human visual attention mechanism.

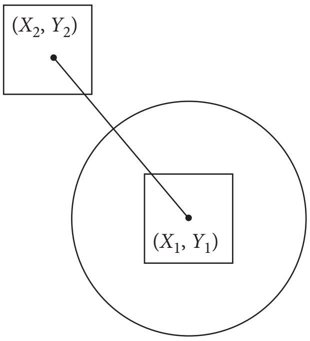


Figure 4: The schematic diagram of distance between current CTU and central CTU.

5. Update of Parametric Model Based on Quasi-Newton Method

Aiming at the JCTVC-K0103 rate control algorithm adopting the gradient descent method to update the parametric model with the problem of slow convergence speed and high computational load, Li et al. [26] propose to use the Newton method to update the parametric model, which reduces the complexity of rate control algorithm to some extent, but the calculation load is still large and the overall performance is improved limited in the updated phase of the parametric model. This paper introduces the quasi-Newton method to update the parametric model and uses the BFGS algorithm to update the positive definite matrix B_n approximated to the inverse of Hessian matrix, which reduces the computational load of the parametric model.

The quasi-Newton method is used to solve the optimization problems usually. The basic idea is to take the optimal solution of quadratic model as search direction, which obtains a new iterative point x_{n+1}, and update B_n in the each iteration. The iterative equation of quasi-Newton method is as follows:

$$x_{n+1} = x_n - \lambda H_n^{(-1)} \nabla f\left(x^{(n)}\right), \tag{33}$$

where x_n represents the nth iterative point, x_{n+1} represents the $(n+1)$th iterative point, λ represents a constant, and H_n represents a positive definite matrix approximated to the inverse of Hessian matrix. The specific implementation steps of the ORC algorithm are as follows:

(1) In the implementation process of target bits, the relationship between rate and distortion is shown in equation (3). According to equation (3), the distortion value D_1 estimated from the target bit rate and the actual coding distortion value D_2 are as follows:

$$D_1 = C_1 \times R_1^{-k_1}, \tag{34}$$

$$D_2 = C_2 \times R_2^{-k_2}, \tag{35}$$

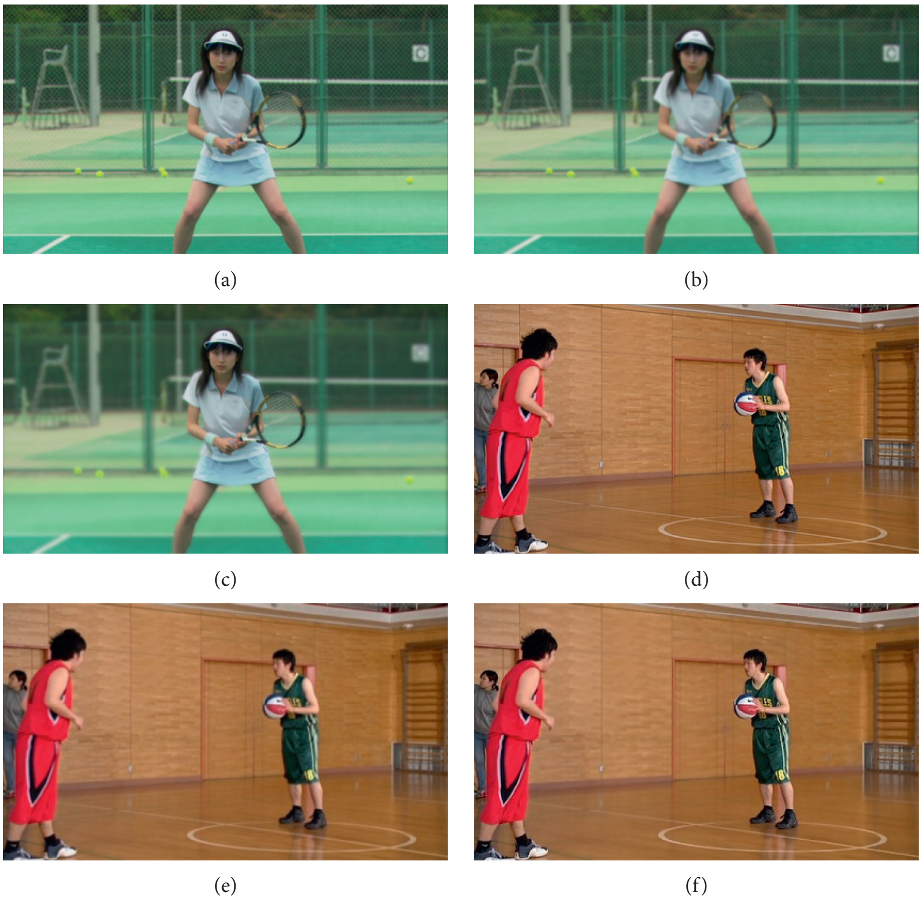

(a) (b) (c) (d) (e) (f)

Figure 5: The experimental results between the K0103 algorithm and the ORC algorithm.

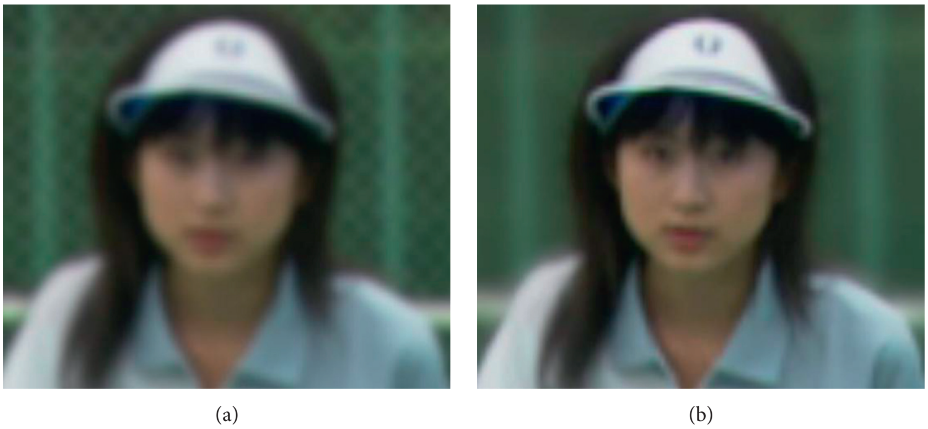

(a) (b)

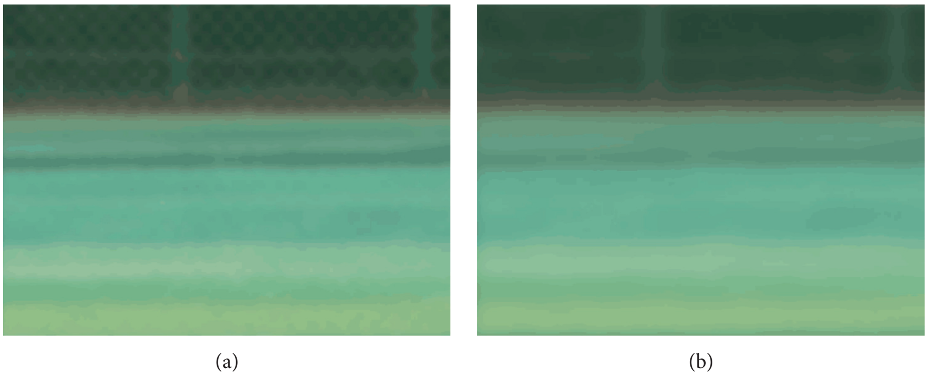
Figure 6: The comparison charts of ROI in detail. The compressed result of (a) K0103 algorithm and (b) ORC algorithm.

where C_1 and k_1 represent parameters related to the characteristics of video content before the parametric model is updated; C_2 and k_2 represent parameters related to the characteristics of video content after the parametric model is updated; R_1 and R_2 represent target bit rate and actual bit rate, respectively.

(2) Taking the logarithm of equations (34) and (35), respectively,

$$\ln D_1 = \ln C_1 - k_1 \ln R_1, \tag{36}$$

$$\ln D_2 = \ln C_2 - k_2 \ln R_2. \tag{37}$$



(a) (b)

Figure 7: The comparison charts of RONI in detail. The compressed result of (a) K0103 algorithm and (b) ORC algorithm.

The error e^2 between the value of actual coding distortion and the value of distortion estimated by the target bit rate from equations (36) and (37) are as follows:

$$\begin{aligned} e^2 &= \left(\ln D_1 - \ln D_2\right)^2, \\ e^2 &= \left(\ln C_2 - k_2 \ln R_2 - \ln C_1\right)^2, \end{aligned} \tag{38}$$

where $C = \ln C_2$ and $C_1 = D_1$. And, taking the derivative of C and k_2, respectively,

$$\begin{aligned} \frac{\partial e^2}{\partial C} &= 2 \cdot \left(C - k_2 \ln R_2 - \ln C_1\right), \\ \frac{\partial e^2}{\partial k_2} &= -2 \ln R_2 \cdot \left(C - k_2 \ln R_2 - \ln C_1\right). \end{aligned} \tag{39}$$

(3) Performing the iteration by quasi-Newton method according to equation (33), we get

$$C_2 = C_1 - 2\lambda H_n^{-1} \cdot \left(\ln C_2 - k_2 \ln R_2 - \ln C_1\right), \tag{40}$$

$$\begin{aligned} k_2 = k_1 + 2\lambda H_n^{-1} \cdot \big(&\ln C_2 \cdot \ln R_2 - k_2 \ln^2 R_2 \\ &- \ln C_1 \ln R_2\big). \end{aligned} \tag{41}$$

As we can see from equation (4), $\alpha = C \cdot k$, $\beta = -k - 1$, then

$$\begin{cases} \alpha_{\text{old}} = C_1 \cdot k_1, \\ \beta_{\text{old}} = -k_1 - 1, \end{cases} \tag{42}$$

$$\begin{cases} \alpha_{\text{new}} = C_2 \cdot k_2, \\ \beta_{\text{new}} = -k_2 - 1, \end{cases} \tag{43}$$

where α_{old} and β_{old} represent the parameters while determining the quantization parameter QP and α_{new} and β_{new} represent parameters updated. The parameters α_{new} and β_{new} updated can be obtained by simultaneous equations (40)–(43).

(4) Selecting B_n as the approximation of H_n^{-1} according to the following conditions of quasi-Newton method, we get

$$y_n = H_n \cdot \delta_n, \tag{44}$$

where $\delta_n = x_{k+1} - x_k$ and $y_n = g_{n+1} - g_n$, g_n represents the first derivative of target function at x_n. In this paper, the BFGS algorithm is used to calculate B_n, and the performance and accuracy of BFGS algorithm are higher compared with the DFP algorithm and the Broyden algorithm. The equation of B_{k+1} is as follows:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \tag{45}$$

where $s_k = x_{k+1} - x_k$, $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$, and the positive definite matrix generated satisfies the following equation:

$$\begin{aligned} \min_B \quad & \left|B - B_k\right|, \\ \text{s.t.} \quad & B = B^T, \\ & B \cdot s_k = y_k. \end{aligned} \tag{46}$$

In summary, the updated process of α and β can be achieved. In order to verify the performance of the improved algorithm compared with the K0103 rate control algorithm of the HM10.0 model, five different test sequences provided by JCT-VC with 200 frames selected are experimented, which are under the two configuration files of the type of low delay LDMmain and the type of random access RAMmain. And, the coding efficiency is measured by encoding time saved ΔT [27, 28]. The calculation equation of ΔT is as follows:

$$\Delta T = \frac{\left|\text{Time}_{\text{pro}} - \text{Time}_{\text{HM10.0}}\right|}{\text{Time}_{\text{HM10.0}}} \times 100\%, \tag{47}$$

where Time_{pro} represents encoding time of the algorithm in this paper and $\text{Time}_{\text{HM10.0}}$ represents encoding time of the K0103 rate control algorithm.

From the comparison of experimental data in Table 1, we can see that the coding time by the algorithm in this paper is lower than the coding time of K0103 rate control algorithm

Table 1: The comparison of coding time under two configuration files of LDMmain and RAMain.

Video sequences	Encoding time					
	LDMain			RAMain		
	HM10.0 (t)	Proposed (t)	ΔT (%)	HM10.0 (t)	Proposed (t)	ΔT (%)
NebutaFestival	117.54	115.69	1.85	104.38	101.51	2.87
Kimono	82.37	81.33	1.04	70.83	69.49	1.34
Slideshow	101.45	99.74	1.71	88.76	85.66	3.10
BasketballDrill	94.37	92.83	1.54	83.41	81.74	1.67
BlowingBubbles	36.42	35.26	1.16	28.93	26.51	2.42

regardless of the configuration file being LDMmain or RAMain, which reduces the computational load in the updated phase of the parametric model, and the algorithm in this paper is more effective.

6. Experiment

6.1. Experimental Parameters and Evaluation Indicators. In order to verify the validity of the algorithm in this paper, the hardware configurations are as follows: Inter(R) Core(TM) i5-3470, the main frequency 3.2 GHz, the memory 4 GB; the software configurations are as follows: Microsoft Visual Studio 2010 and OPENCV2.4.10 are used as experimental platform. Simulation was performed to verify the compression performance of the ORC algorithm in the HM10.0 model. The experimental data sets come from five different levels of video sequences provided by JCT-VC. The configuration files are LD configuration files of the IPPP coding structure. The number of experimental frames is 100 frames. In order to evaluate the compression performance of the ORC algorithm under different sequences, the component of peak signal-to-noise ratio of brightness Y − PSNR, the increment of the component of peak signal-to-noise ratio of brightness ΔPSNR, the error of rate ΔBR, the peak signal-to-noise ratio PSNR, the percentage increase of the bit rates BDBR, and the reduction of peak signal-to-noise ratio BDPSNR [29, 30] are used as indicators to measure the compression performance of the algorithm in this paper. The equations of ΔPSNR and ΔBR are as follows:

$$\Delta\mathrm{PSNR} = \mathrm{Y}-\mathrm{PSNR}_{\mathrm{Pro}} - \mathrm{Y}-\mathrm{PSNR}_{\mathrm{HM10.0}},$$
$$\Delta\mathrm{BR} = \frac{\mathrm{BR}_{\mathrm{Pro}} - \mathrm{BR}_{\mathrm{Tar}}}{\mathrm{BR}_{\mathrm{Tar}}} \times 100\%, \tag{48}$$

where $\mathrm{Y}-\mathrm{PSNR}_{\mathrm{Pro}}$ represents the component of peak signal-to-noise ratio of brightness of coded image obtained by the ORC algorithm, $\mathrm{Y}-\mathrm{PSNR}_{\mathrm{HM10.0}}$ represents the component of peak signal-to-noise ratio of brightness of coded image obtained by the K0103 rate control algorithm, $\mathrm{BR}_{\mathrm{Pro}}$ represents the actual output rate obtained by the ORC algorithm, and $\mathrm{BR}_{\mathrm{Tar}}$ represents the target bit rate.

6.2. Analysis of Experimental Results. The performance of the ORC algorithm in this paper is evaluated from two aspects: compression performance and rate control accuracy. The experimental comparison results of the ORC algorithm, the K0103 algorithm, and the algorithm in [31] are shown in Tables 2 and 3, respectively, under four sets of target bit rate, which are 500 kbps, 1 Mbps, 2 Mbps, and 4 Mbps.

From the experimental data in Tables 2 and 3, we can see that the range of ΔBR by the ORC algorithm is 0.000% ~ 0.691%, and the mean value is 0.190%. The range of ΔBR by the K0103 algorithm is 0.011% ~ 4.137%, and the mean value is 0.244%. The range of ΔBR by the algorithm in [31] is 0.009% ~ 0.824%, and the mean value is 0.220%. The output bit rates of the ORC algorithm are more in line with the target bit rate. From the perspective of rate control accuracy, the ORC algorithm can control the rate more accurately compared with the K0103 algorithm and the algorithm in [31]. At the same time, the value of Y-PSNR by the ORC algorithm is improved by 0.065 dB and 0.045 dB compared with the K0103 rate control algorithm and the algorithm in [31], respectively. From the analysis of compression performance, the ORC algorithm improves the overall quality of videos while maintaining the rate control accuracy.

In order to analyze the compression performance of the ORC algorithm specifically, BasketballPass sequences with complex texture and Kimono sequences with flat texture are selected for the experiment. Figure 8 shows the comparison results of Y-PSNR each frame among the ORC algorithm, the K0103 algorithm, and the algorithm in [31] when QP = 27.

We can analyze that the average value of Y-PSNR by the ORC algorithm is larger compared with that by the K0103 algorithm and the algorithm in [31] under two sequences of different complexities from Figure 8. The ORC algorithm can measure the texture complexity of video frames more accurately, and the compression performance gets promoted.

Table 4 shows the experimental data of PSNR in the ROI and RONI of four different sets of video sequences with 100 frames per set under different target bit rates. As we can see from Table 4, the average PSNR of the K0103 algorithm in the ROI is 36.77 and the average PSNR in the RONI is 36.72. The average PSNR of the algorithm in [31] in the ROI is 37.07, and the average PSNR in the RONI is 36.48. The average PSNR of the ORC algorithm in the ROI is 37.55, and the average PSNR in the RONI is 36.17. Compared with the K0103 algorithm and the algorithm in [31], the average values of PSNR increased to 0.78 dB and 0.48 dB, respectively, by the ORC algorithm, which means that the ORC algorithm has a definite advantage to improve the quality of videos compared with the others two algorithms. At the same time, according to the analysis of the data in Table 4, we

Table 2: Experimental comparison data between the ORC algorithm and the K0103 algorithm.

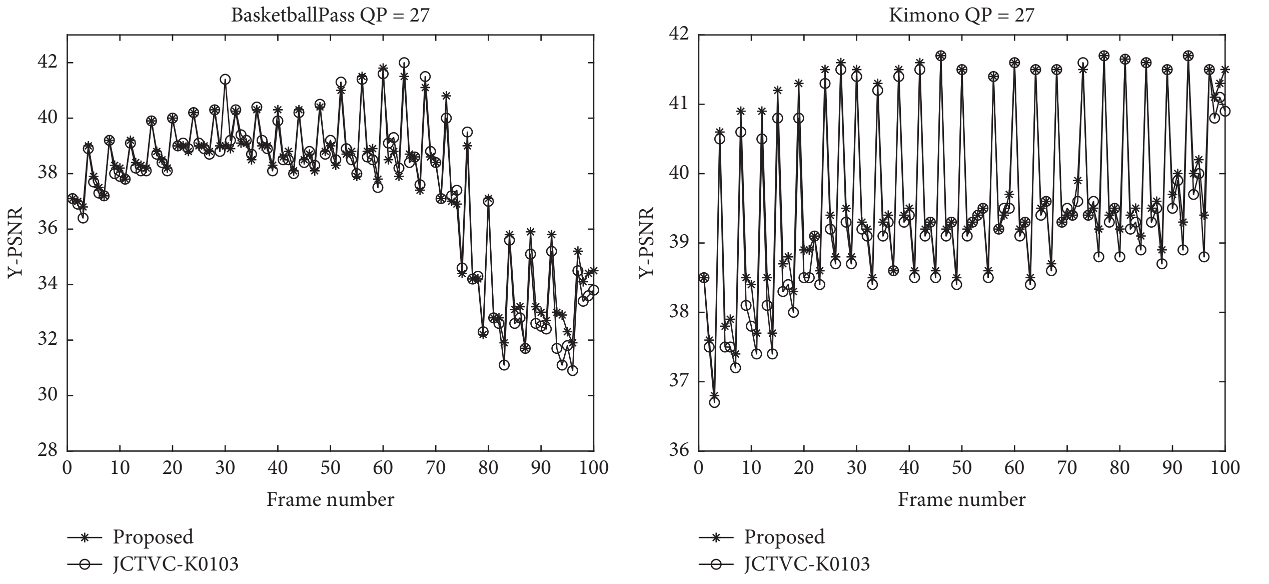
		Target bit rate (kbps)	HM10.0			ORC algorithm			ΔPSNR (dB)
			Y-PSNR (dB)	Actual coding bit rate (kbps)	ΔBR (%)	Y-PSNR (dB)	Actual coding bit rate (kbps)	ΔBR (%)	
2560 × 1600	Traffic	500	32.17	500.296	0.296	32.17	500.125	0.125	0.00
		1000	34.41	1000.014	0.014	34.44	1000.227	0.227	0.03
		2000	36.74	1999.813	0.187	36.82	2000.016	0.016	0.08
		4000	38.95	4000.054	0.054	39.02	4000.108	0.108	0.07
	PeopleOnStreet	500	36.83	500.301	0.301	36.88	500.136	0.136	0.05
		1000	38.07	1000.065	0.065	38.12	1000.225	0.225	0.05
		2000	40.24	2000.112	0.112	40.26	2000.114	0.114	0.02
		4000	42.51	4000.031	0.031	42.57	4000.107	0.107	0.06
1920 × 1080	BasketballDrive	500	29.14	500.213	0.213	29.25	500.141	0.141	0.11
		1000	31.41	999.784	0.216	31.48	1000.113	0.113	0.07
		2000	33.57	2000.074	0.074	33.59	2000.116	0.116	0.02
		4000	35.71	4000.101	0.101	35.90	4000.000	0.000	0.19
	Parkscene	500	33.27	500.186	0.186	33.35	499.847	0.153	0.08
		1000	35.46	1000.214	0.214	35.50	1000.218	0.218	0.04
		2000	37.71	2000.103	0.103	37.77	2000.236	0.236	0.06
		4000	40.35	4000.011	0.011	40.46	4000.231	0.231	0.11
1280 × 720	FourPeople	500	37.14	500.194	0.194	37.19	500.691	0.691	0.05
		1000	39.93	1000.058	0.058	39.97	1000.637	0.637	0.04
		2000	41.84	2000.196	0.196	41.92	2000.314	0.314	0.08
		4000	43.59	4000.112	0.112	43.71	4000.105	0.105	0.12
	Johnny	500	38.83	504.137	4.137	38.87	500.411	0.411	0.04
		1000	41.62	999.084	0.084	41.72	999.579	0.421	0.10
		2000	43.75	2000.253	0.253	43.82	2000.304	0.304	0.07
		4000	46.14	4000.178	0.178	46.25	4000.225	0.225	0.11
832 × 480	BasketballDrill	500	32.13	500.129	0.129	32.15	500.117	0.117	0.02
		1000	35.01	1000.095	0.095	35.05	1000.206	0.206	0.04
		2000	38.26	2000.140	0.140	38.33	2000.041	0.041	0.07
		4000	41.57	3999.213	0.187	41.63	4000.100	0.100	0.06
	BQMall	500	29.83	500.197	0.197	29.88	500.171	0.171	0.05
		1000	32.17	1000.088	0.088	32.25	1000.231	0.231	0.08
		2000	35.01	2000.244	0.244	35.13	2000.036	0.036	0.12
		4000	37.58	4000.185	0.185	37.65	4000.245	0.245	0.07
416 × 240	BasketballPass	500	36.84	500.050	0.050	36.90	500.045	0.045	0.06
		1000	40.46	1000.188	0.188	40.48	1000.321	0.321	0.02
		2000	42.97	2000.176	0.176	43.04	1999.796	0.204	0.07
		4000	45.62	4000.071	0.071	45.73	4000.113	0.113	0.11
	BQSquare	500	31.74	500.083	0.083	31.76	500.168	0.168	0.02
		1000	34.37	1000.126	0.126	34.42	1000.112	0.112	0.05
		2000	36.82	1999.765	0.235	36.85	2000.009	0.009	0.03
		4000	41.13	4000.173	0.173	41.21	4000.105	0.105	0.08
Average					0.244			0.190	0.065

Table 3: Experimental comparison data between the ORC algorithm and algorithm in [31].

		Target bit rate (kbps)	Algorithm in [31]			ORC algorithm			ΔPSNR (dB)
			Y-PSNR (dB)	Actual coding bit rate (kbps)	ΔBR (%)	Y-PSNR (dB)	Actual coding bit rate (kbps)	ΔBR (%)	
2560 × 1600	Traffic	500	32.16	500.294	0.294	32.17	500.125	0.125	0.01
		1000	34.45	1000.018	0.018	34.44	1000.227	0.227	0.01
		2000	36.78	2000.215	0.215	36.82	2000.016	0.016	0.04
		4000	38.99	4000.035	0.035	39.02	4000.108	0.108	0.03
	PeopleOnStreet	500	36.89	500.214	0.214	36.88	500.136	0.136	0.01
		1000	38.04	1000.193	0.193	38.12	1000.225	0.225	0.08
		2000	40.24	1999.245	0.756	40.26	2000.114	0.114	0.02
		4000	42.53	4000.213	0.213	42.57	4000.107	0.107	0.04

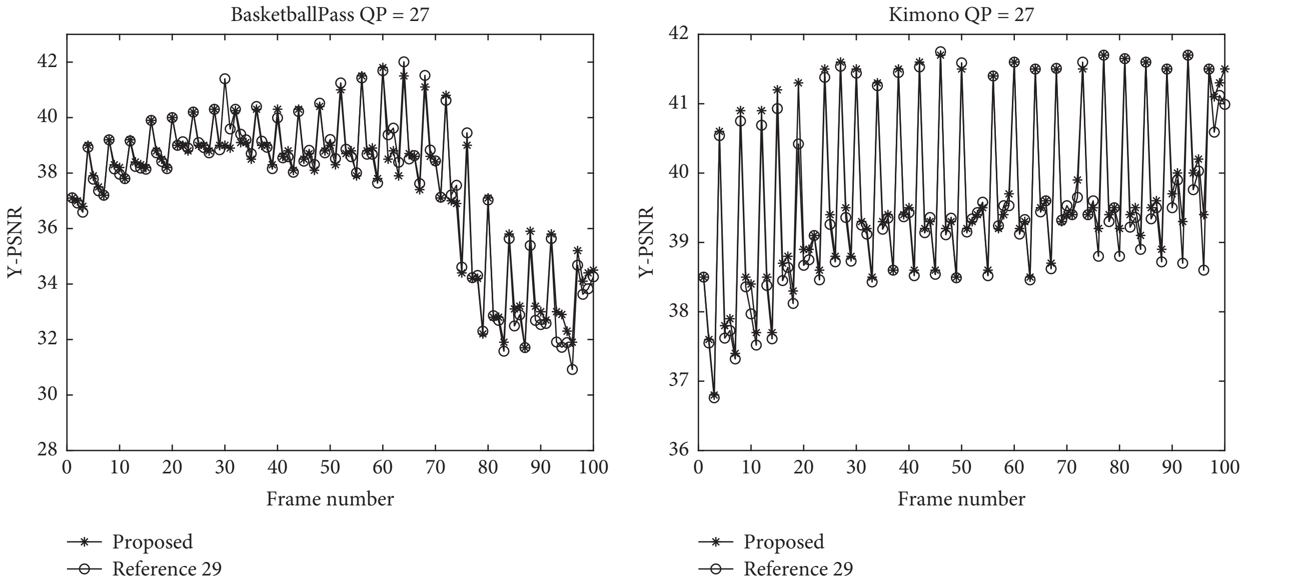
Table 3: Continued.

		Target bit rate (kbps)	Algorithm in [31]			ORC algorithm			ΔPSNR (dB)
			Y-PSNR (dB)	Actual coding bit rate (kbps)	ΔBR (%)	Y-PSNR (dB)	Actual coding bit rate (kbps)	ΔBR (%)	
1920 × 1080	BasketballDrive	500	29.16	500.312	0.312	29.25	500.141	0.141	0.09
		1000	31.43	1000.294	0.294	31.48	1000.113	0.113	0.05
		2000	33.57	2000.183	0.183	33.59	2000.116	0.116	0.02
		4000	35.81	4000.247	0.247	35.90	4000.000	0.000	0.09
	Parkscene	500	33.29	500.183	0.183	33.35	499.847	0.153	0.06
		1000	35.46	1000.216	0.216	35.50	1000.218	0.218	0.04
		2000	37.73	1999.594	0.406	37.77	2000.236	0.236	0.04
		4000	40.41	4000.214	0.214	40.46	4000.231	0.231	0.05
1280 × 720	FourPeople	500	37.15	500.204	0.204	37.19	500.691	0.691	0.04
		1000	39.95	1000.157	0.157	39.97	1000.637	0.637	0.02
		2000	41.88	2000.248	0.248	41.92	2000.314	0.314	0.04
		4000	43.65	4000.312	0.312	43.71	4000.105	0.105	0.06
	Johnny	500	38.85	504.137	4.137	38.87	500.411	0.411	0.02
		1000	41.67	999.084	0.084	41.72	999.579	0.421	0.05
		2000	43.75	2000.253	0.253	43.82	2000.304	0.304	0.07
		4000	46.17	4000.178	0.178	46.25	4000.225	0.225	0.08
832 × 480	BasketballDrill	500	32.13	500.217	0.217	32.15	500.117	0.117	0.02
		1000	35.02	1000.014	0.014	35.05	1000.206	0.206	0.03
		2000	38.28	2000.178	0.178	38.33	2000.041	0.041	0.05
		4000	41.60	4000.159	0.159	41.63	4000.100	0.100	0.03
	BQMall	500	29.82	500.217	0.217	29.88	500.171	0.171	0.06
		1000	32.18	1000.084	0.084	32.25	1000.231	0.231	0.07
		2000	35.06	2000.315	0.315	35.13	2000.036	0.036	0.07
		4000	37.59	4000.169	0.169	37.65	4000.245	0.245	0.06
416 × 240	BasketballPass	500	36.85	500.037	0.037	36.90	500.045	0.045	0.05
		1000	40.45	999.176	0.824	40.48	1000.321	0.321	0.03
		2000	42.97	2000.154	0.154	43.04	1999.796	0.204	0.07
		4000	45.68	4000.013	0.013	45.73	4000.113	0.113	0.05
	BQSquare	500	31.73	500.043	0.043	31.76	500.168	0.168	0.03
		1000	34.39	1000.115	0.115	34.42	1000.112	0.112	0.03
		2000	36.82	2000.245	0.245	36.85	2000.009	0.009	0.03
		4000	41.15	4000.169	0.169	41.21	4000.105	0.105	0.06
Average					0.220			0.190	0.045



(a) (b)

Figure 8: Continued.



(c)

(d)

Figure 8: The comparison of Y-PSNR among three algorithms. The comparison of Y-PSNR in (a) the BasketballPass sequences, (b) the Kimono sequences, (c) the BasketballPass sequences, and (d) the Kimono sequences.

Table 4: Experimental results among the ORC algorithm, the K0103 algorithm, and algorithm in [31].

		K0103 algorithm		[31] algorithm		ORC algorithm	
		ROI	RONI	ROI	RONI	ROI	RONI
BasketballDrill 1920 × 1080	500	29.62	29.60	29.88	29.48	30.01	29.49
	1000	31.99	31.95	33.20	31.88	33.31	31.91
	2000	34.01	33.95	34.91	34.19	36.24	33.93
	4000	36.18	36.10	36.35	36.12	37.55	36.19
FourPeople 1280 × 720	500	37.59	37.57	37.84	37.40	37.98	37.36
	1000	40.39	40.39	40.62	40.24	40.71	40.21
	2000	42.40	42.34	42.56	42.26	42.71	42.31
	4000	44.09	44.01	45.23	43.99	36.54	36.20
BQMall 832 × 480	500	30.63	30.61	30.87	30.47	32.04	30.44
	1000	32.77	32.71	32.97	32.63	34.11	32.67
	2000	35.63	35.55	32.77	32.51	36.99	35.59
	4000	38.02	37.94	39.15	37.91	39.31	37.93
RaceHorse 416 × 240	500	34.16	34.16	34.41	34.05	34.64	34.08
	1000	37.33	37.31	37.57	37.25	38.77	37.27
	2000	39.95	39.89	40.09	39.83	41.22	39.80
	4000	43.50	43.42	44.63	43.39	45.72	43.40
Average		36.77	36.72	37.07	36.48	37.55	36.17

can see that the range of PSNR by the K0103 algorithm between ROI and RONI is 0~0.08 dB, the range of PSNR by the algorithm in [31] between ROI and RONI is 0.23~1.24 dB, and the range of PSNR by the ORC algorithm between ROI and RONI is 0.52~2.32 dB. Compared with the K0103 algorithm and the algorithm in [31], the ORC algorithm has a larger difference of PSNR between ROI and RONI, which indicates that the ORC algorithm can make the output videos more attuned with the human visual attention mechanism.

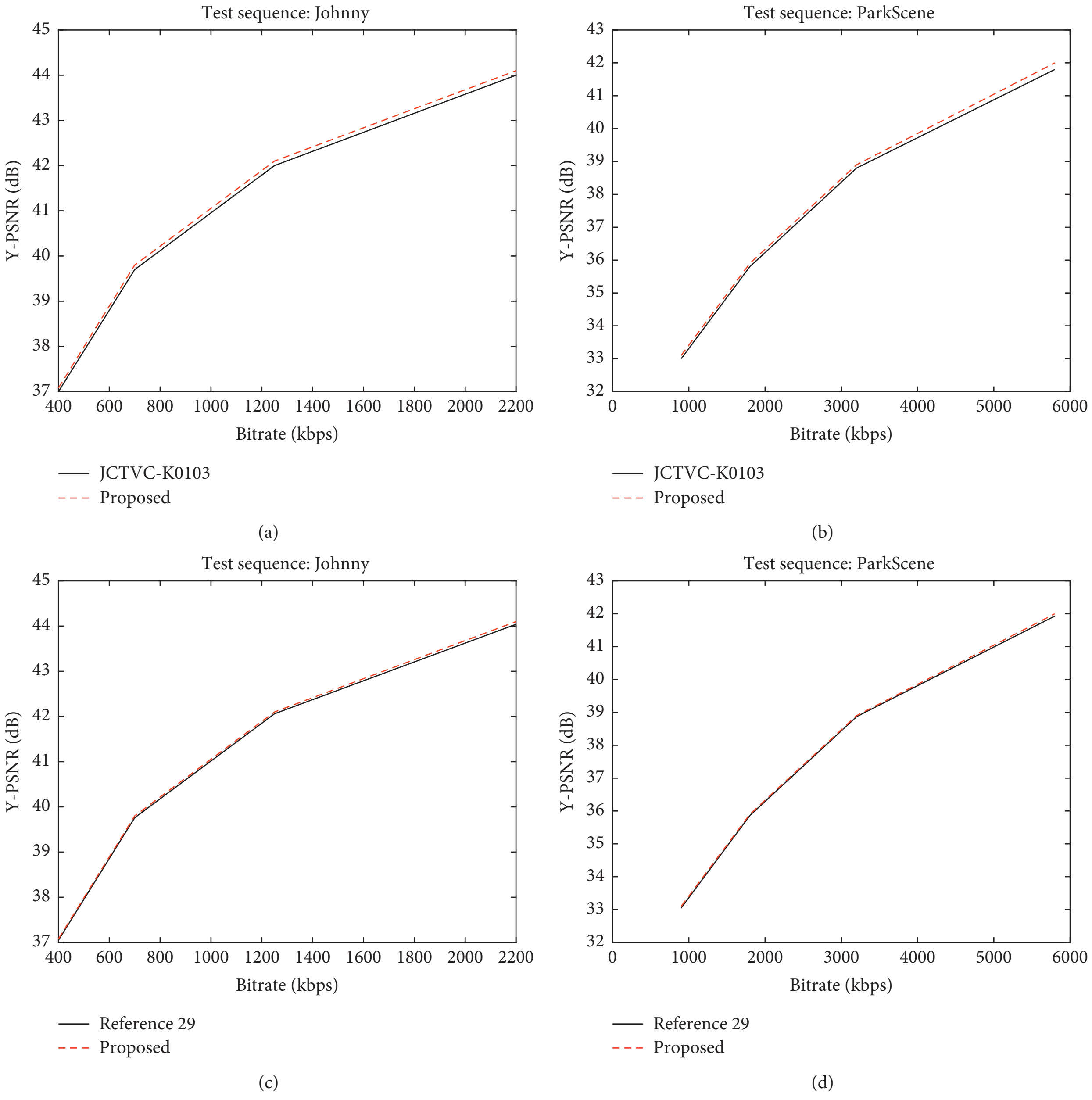
In terms of rate distortion performance, the experimental results of the ORC algorithm compared with the K0103 algorithm and the algorithm in [31] under five categories of test sequences are shown in Tables 5 and 6.

Table 5: Comparison of rate distortion performance by the ORC algorithm compared with the K0103 algorithm.

Sequences	BDBR (%)	BDPSNR (dB)
ClassA	−25.39	0.0207
ClassB	−28.21	0.0230
ClassC	−51.63	0.0469
ClassD	−50.18	0.0547
ClassE	−33.61	0.0326
Average	−37.80	0.0356

Table 6: Comparison of rate distortion performance by the ORC algorithm compared with the algorithm in [31].

Sequences	BDBR (%)	BDPSNR (dB)
ClassA	−22.65	0.0184
ClassB	−26.31	0.0215
ClassC	−48.72	0.0436
ClassD	−47.69	0.0511
ClassE	−31.43	0.0297
Average	−35.36	0.0329



(a)

(b)

(c)

(d)

Figure 9: Comparison of rate distortion performance among three algorithms. The rate-distortion curve of (a, c) Johnny sequences and (b, d) ParkScene sequences.

From the experimental comparison results in Tables 5 and 6, we can see that the increment of BDBR by the ORC algorithm is reduced by 37.80% and 35.34% on average and the loss of BDPSNR by the ORC algorithm is reduced by 0.0356 dB and 0.0329 dB on average compared with the K0103 algorithm and the algorithm in [31], respectively. The ORC algorithm maintains a better rate distortion performance compared with the K0103 algorithm and the algorithm in [31].

In order to show the rate distortion performance of the ORC algorithm more intuitively, Figure 9 shows the comparison results of the rate distortion (RD) curves among the ORC algorithm, the K0103 algorithm, and the algorithm in [31] under the Johnny sequences and the ParkScene sequences, which have different resolutions.

We can see that the rate-distortion curves of the ORC algorithm are above the rate-distortion curves of the K0103 algorithm and the algorithm in [31] from Figure 9, which indicates that the Y-PSNR of the ORC algorithm is larger compared with the K0103 algorithm and the algorithm in [31] when the rates are same, and the RD performance is more optimized. The overall performance of the ORC algorithm is better.

7. Conclusions

This paper proposes an optimized rate control algorithm of ITU-T H.265/high-efficiency video coding based on the region of interest. The algorithm improves the Itti model based on the space-time domain firstly, which extracts the ROI of video frames. And, the target bits of CTU layer are redistributed based on ROI so that the output videos are more attuned with the human visual attention mechanism. Finally, the quasi-Newton method is used to update the parametric model, which reduces the computational complexity in the updated phase of the parametric model. The experimental results show that the ORC algorithm has better compression performance and rate control accuracy than the K0103 algorithm and the algorithm in [31], which can obtain better compression results of videos.

Data Availability

The datasets available for research articles come from http://trace.eas.asu.edu/yuv/index.html and http://www.iaprtc11.org/mediawiki/index.php/MSRA_Text_Detection_500_Database_(MSRA-TD500).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] H. L. Tan, C. C. Ko, and S. Rahardja, "Fast coding quad-tree decisions using prediction residuals statistics for high efficiency video coding (HEVC)," *IEEE Transactions on Broadcasting*, vol. 62, no. 1, pp. 128–133, 2016.

[2] Z. Z. Chen and X. Pan, "An optimized rate control for low-delay H.265/HEVC," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4541–4552, 2019.

[3] Y. Guo, B. Li, S. Sun, and J. Xu, "Rate control for screen content coding in HEVC," in *Proceedings of the 2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015.

[4] B. Li, H. Li, L. Li, and J. Zhang, "λ domain rate control algorithm for high efficiency video coding," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3841–3854, 2014.

[5] Y. Gong, S. Wan, K. Yang, H. R. Wu, and Y. Liu, "Temporal-layer-motivated lambda domain picture level rate control for random-access configuration in H.265/HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 156–170, 2017.

[6] M. Zhou, X. Wei, S. Wang et al., "SSIM-based global optimization for CTU-level rate control in HEVC," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 1921–1933, 2019.

[7] M. Wang, K. N. Ngan, and H. Li, "An efficient frame-content based intra frame rate control for high efficiency video coding," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 896–900, 2015.

[8] L. P. Van, J. D. Praeter, G. V. Wallendael, S. V. Leuven, J. D. Cock, and R. K. de Walle, "Efficient bit rate transcoding for high efficiency video coding," *IEEE Transactions on Multimedia*, vol. 18, no. 3, pp. 364–378, 2016.

[9] J. He and F. Z. Yang, "Efficient frame-level bit allocation algorithm for H.265/HEVC," *IET Image Processing*, vol. 11, no. 4, pp. 245–257, 2017.

[10] J. Han, Y. Ma, B. Zhou et al., "A robust infrared small target detection algorithm based on human visual system," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 12, pp. 2168–2172, 2014.

[11] P. Chiranjeevi and S. Sengupta, "New fuzzy texture features for robust detection of moving objects," *IEEE Signal Processing Letters*, vol. 19, no. 10, pp. 603–606, 2012.

[12] D. Lai, B. Xiong, and G. Kuang, "Weak target detection in SAR images via improved itti visual saliency model," in *Proceedings of the 2017 International Conference on Frontiers of Sensors Technologies*, April 2017.

[13] X. Long and X. Wu, "Motion segmentation based on fusion of MSRF segmentation and Canny operator," *Procedia Engineering*, vol. 15, pp. 1637–1641, 2011.

[14] M. Hou and S. Xing, "Study of improving the stability of SUSAN corner detection algorithm," in *Proceedings of the 2010 International Conference on Computer Application & System Modeling*, October 2010.

[15] H. Jalink, "Fotosynthese real-time zichtbaar maken (interview met Henk Jalink)," in *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Florence, Italy, May 2014.

[16] G. Ciuperca, V. Girardin, and L. Lhote, "Computation and estimation of generalized entropy rates for denumerable Markov chains," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4026–4034, 2011.

[17] X. M. Xing, Q. Sun, and P. Y. Zhang, "Research on fault recovery and reconstruction of distribution network based on depth-first search and bacterial algorithm," *Journal of Northeast Electric Power University*, vol. 39, no. 3, pp. 38–43, 2019.

[18] X. Feng, T. Liu, D. Yang, and Y. Wang, "Saliency inspired full-reference quality metrics for packet-loss-impaired video," *IEEE Transactions on Broadcasting*, vol. 57, no. 1, pp. 81–88, 2011.

[19] Z. Zdziarski and R. Dahyot, "Extension of GBVS to 3D media," in *Proceedings of the 2014 Signal Processing & Communications Applications Conference*, April 2014.

[20] G. Zhiqi and T. Haihong, "Robust STAP algorithm based on knowledge-aided SR for airborne radar," *IET Radar Sonar & Navigation*, vol. 11, no. 2, pp. 321–329, 2017.

[21] L. Stankovic and S. Djukanovic, "Order adaptive local polynomial FT based interference rejection in spread spectrum communication systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 6, pp. 2156–2162, 2005.

[22] M. Li, M. Li, P. Zhang, Y. Wu, W. Song, and L. An, "SAR image change detection using PCANet guided by saliency

detection," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 3, pp. 402–406, 2018.

[23] Y. Yuan, D. Li, and Q. M. Meng, "Automatic polyp detection via a novel unified bottom-up and top-down saliency approach," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 4, pp. 1250–1260, 2017.

[24] L. Khoshnevisan and F. R. Salmasi, "Adaptive rate-based congestion control with weighted fairness through multi-loop gradient projection internal model controller," *IET Control Theory & Applications*, vol. 9, no. 18, pp. 2641–2647, 2015.

[25] H. M. Hu, B. Li, W. Lin, W. Li, and M.-T. Sun, "Region-based rate control for H.264/AVC for low bit-rate applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 11, pp. 1564–1576, 2012.

[26] L. Li, B. Li, D. Liu, and H. Li, "λ-domain rate control algorithm for HEVC scalable extension," *IEEE Transactions on Multimedia*, vol. 18, no. 10, pp. 2023–2039, 2016.

[27] D. Flynn, D. Marpe, M. Naccari et al., "Overview of the range extensions for the HEVC standard: tools, profiles and performance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 4–19, 2015.

[28] C. Q. Lin, H. Tong, W. F. Yu et al., "Design of intelligent state monitoring and fault handling platform for distribution network based on ubiquitous electricity internet of things," *Journal of Northeast Electric Power University*, vol. 39, no. 4, pp. 1–4, 2019.

[29] T. Zhao, Z. Wang, and C. W. Chen, "Adaptive quantization parameter cascading in HEVC hierarchical coding," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 2997–3009, 2016.

[30] J. Vanne, M. Viitanen, and T. D. Hämäläinen, "Efficient mode decision schemes for HEVC inter prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 9, pp. 1579–1593, 2014.

[31] L. Bai, L. Song, R. Xie, J. Xie, and M. Chen, "Saliency based rate control scheme for high efficiency video coding," in *Proceedings of the IEEE 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, December 2016.