An Adaptive Slice Type Decision Algorithm for HEVC Based on Local Luminance Histogram

Pengyu Liu[1,2,3](✉), Yue Zhang[1,2,3], Shanji Chen[4], Kun Duan[1,2,3], Xuan Sun[1,2,3], and Tenghe Cui[1,2,3]

[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
liupengyu@bjut.edu.cn
[2] Advanced Information Network Beijing Laboratory, Beijing 100124, China
[3] Computational Intelligence and Intelligent Systems Beijing Key Laboratory, Beijing 100124, China
[4] School of Physics and Electronic Information Engineering, Qinghai Nationalities University, Xining 810007, China

Abstract. Video frame type decision is one of the key factors affecting coding efficiency. Based on the framework of High Efficiency Video Coding (HEVC), an adaptive frame type decision algorithm based on local luminance histogram is proposed in this paper. Firstly, the local luminance histogram was calculated at the coding tree unit (CTU) level, and the difference of local luminance histogram between two frames was used to characterize the degree of inter-frame content variation. Secondly, scene-change frame is determined by comparing the degree of inter-frame content variation in the scene-change detection window, and it is defined as I frame. Finally, the Mini-GOP size is adaptively determined according to the correlation between the degree of inter-frame content variation and Mini-GOP size. GPB frame and B frame are adaptively determined for the video sequences with I frame defined. The experimental results show that, compared with the relevant algorithms in x265, the proposed algorithm can achieve efficient adaptive decision of video frame type under the premise of reducing the algorithm complexity by nearly 5%, and effectively improve the efficiency of video coding.

Keywords: High Efficiency Video Coding (HEVC) · Local luminance histogram · Scene-change detection · Slice type decision

1 Introduction

With the continuous development of multimedia technology, videos with high definition (HD), high frame rate (HFR) and multi-view point have resulted in an explosion of actual data thus presenting more formidable challenges to video coding. Compared with Advanced Video Coding (AVC), High Efficiency Video Coding (HEVC) which was published by the Joint Collaborative Team on Video Coding (JCT-VC) in 2013 has achieved

X. Wang et al. (Eds.): AICON 2021, LNICST 396, pp. 331–345, 2021.
https://doi.org/10.1007/978-3-030-90196-7_28

higher compression performance [1–3]. However, the introduction of new technologies such as partitioning structure, quad-tree coding structure for coding unit (CU) and mode decision makes the coding complexity increase significantly [3, 4]. The encoding time of HEVC is higher than H.264/AVC by 253% on average, making it impractical for implementation in multimedia applications. The open source encoder x265 based on HEVC standard is an efficient video encoder for practical industrial scenarios, which significantly reduces the encoding complexity by utilizing many accelerating methods [5].

According to the definition of the slice type of each frame in the video sequence by HEVC, frames can be divided into Intra frame (I frame), Predictive frame (P frame), Bi-directional interpolated prediction frame (B frame) and Generalized P and B frame (GPB frame). Noticeably, GPB frame that references its previous two frames replaces P frame in the low delay mode. I frame is usually determined according to the results of scene-change in the video sequence and the interval of I frame required by the business scene in encoders. And the rest of the frames in the video sequence should be decided into GPB frame or B frame. Under the premise of controllable encoding complexity, the efficient decision of slice type of video sequence can effectively improve the coding efficiency.

The variation of inter-frame scene-change changes dramatically. If the previous frame is used as the reference frame, the quality of the encoded video will inevitably decline seriously. There are three kinds of scene-change detection methods, mainly based on Motion Estimate (ME), the Rate-Distortion Cost (RD Cost) or the luminance histogram [7–13]. For the scene-change detection algorithm based on ME, Shu et al. [7] proposed a scene-change detection algorithm based on block matching, which calculated the Motion Vector Difference (MVD) between the current and the previous frame to judge whether there is a scene-change frame. Ding et al. [8] substituted the Sum of Absolute Difference (SAD) and the Sum of Absolute Transformed Difference (SATD) for the Sum of Absolute Difference (SAD) to obtain more accurate Motion Vector (MV). In addition, Lee et al. [9] proposed a scene-change detection algorithm based on optical flow field, which also relies on the calculation of MV. However, the performance of the above algorithm depends too much on the accuracy of the MV obtained by MV, which is affected by many coding parameters, resulting in limited efficiency of the scene-change detection algorithm based on motion search. Calculating the intra-frame and inter-frame RD Cost of each frame of the video sequence to decide the scene-change frame in x265 delivers excellent performance with high coding complexity. In [10], Kim et al. proposed to compare the global luminance histogram differences between the current frame and the previous one. In [11, 12], scene-change frames were judged by calculating two-dimensional histogram to extract the shape of histogram. Although significantly reduced the coding complexity, the above methods based on the global luminance histogram cannot take the changes in frames into consideration bringing about low detection accuracy and making it difficult to set the threshold.

There are mainly two kinds of methods for deciding GPB frames and B frames. The fixed GPB frame and B frame decision methods used a configurable single frame structure to determine the frame type of the entire video sequence [6, 13]. These methods are simple and suitable for most video sequences, but they cannot further improve the compression performance according to the temporal characteristics of each video sequence. The other is the adaptive GPB frame and B frame decision methods [14, 15]. Compared with the fixed methods, these methods can adaptively adjust the size of Mini Group of Picture (Mini-GOP) according to the temporal characteristics, so as to achieve higher coding efficiency. At present, fast adaptive method and method based on Viterbi have been integrated in x265. The former compared the RD Costs of different sizes of Mini-GOP to accomplish the frame type decision, and the latter completes the frame type decision through iteration based on Viterbi algorithm. Compared with the former, the algorithm based on Viterbi significantly improves the encoding efficiency, but also increases the computational complexity. Therefore, reducing the algorithm complexity under the premise of high compression efficiency is urgent to be solved.

Inspired by the low computational complexity of luminance histogram and the strong correlation of video sequences, we propose the adaptive slice type decision algorithm based on local luminance histogram. Through analyzing the degree of inter-frame content variation, our method can improve the efficiency of video coding while reducing the algorithm complexity.

2 Adaptive Frame Type Decision Algorithm

Figure 1 shows the flow of adaptive frame type decision algorithm based on local luminance histogram proposed in this paper.

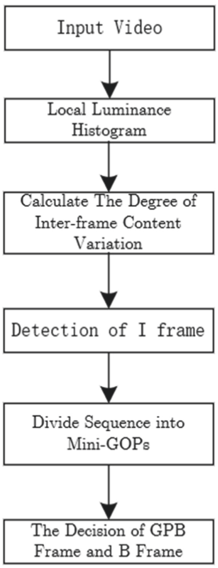

Fig. 1. The flow chart of adaptive frame type decision algorithm based on local luminance histogram

2.1 The Motion of Local Luminance Histogram

Images in the video sequence are composed of luminance component Y and chromaticity component U and V. The core of the methods based on luminance histogram is that the luminance histogram reflects the frequency of each luminance level in the image, and different frames have different luminance histogram. We can decide the type of video frames according to the luminance histogram differences.

Figure 2 shows the comparison of global luminance histogram of the scene-change in Wolf Warrior 2 [16]. We can find that the difference of the two global luminance histograms is negligible.

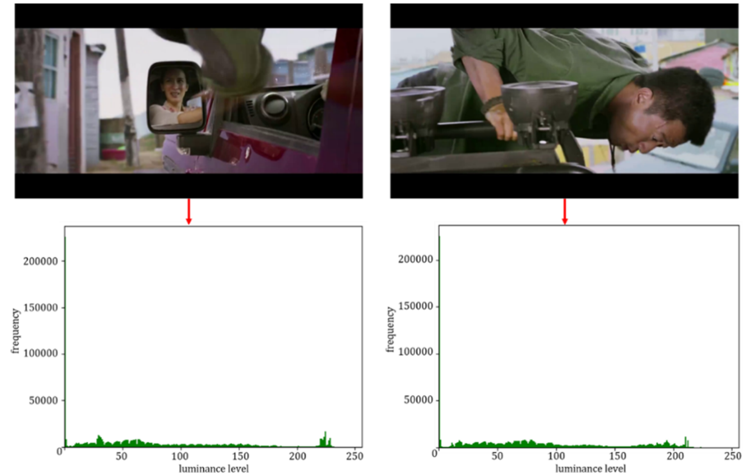

Fig. 2. The comparison of global luminance histogram of scene-change frames

Since the luminance histogram is a statistical variable, it cannot reflect the spatial information of a frame, and the spatial information is the key factor to determine the content. We propose the local luminance histogram in this paper to make up for the lack of spatial information in the global luminance histogram. This paper takes CTU as the basic unit to divide each frame of the video into blocks and obtain the corresponding local luminance histogram. Figure 3 shows the process to obtain the local luminance histogram of a frame,

$$H_n = \left\{H_{n,0,0}, H_{n,0,1}, \ldots, H_{n,i,j}, \ldots\right\}$$

$$H_{n,i,j} = \{l_0, l_1, \ldots, l_{255}\} \tag{1}$$

where H_n refers to the collection of local luminance histograms for all CTUs in *nth* frame, $H_{n,i,j}$ denotes the local luminance histogram of CTU in row *i*, column *j* in *nth* frame, l_k represents the frequency of the *kth* luminance value.

Fig. 3. Obtain local luminance histogram of each frame

Figure 4 shows the local luminance histogram of CTU at the same position in the two frames before and after the scene change in Wolf Warrior 2. Compared with the global luminance histogram, there are great differences in the brightness distribution of the local luminance histogram.

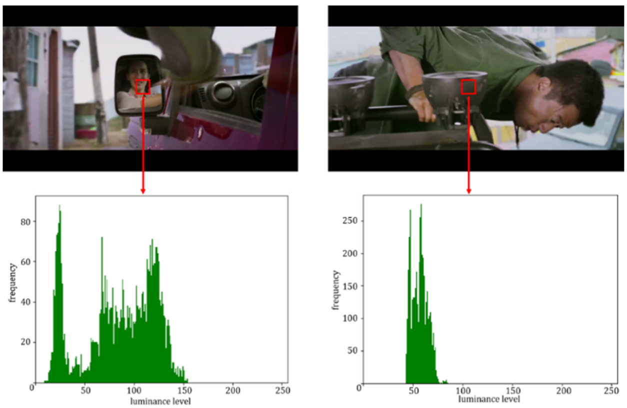

Fig. 4. The comparison of local luminance histogram of scene-change frames

We selected Kimono and three dynamic video sequences from the Internet, including Wolf Warrior 2, A Little Red Flower, Report on COVID-19 Prevention and Control in Beijing, to analyze global luminance histogram differences D_{hist} and the sum of local luminance histogram differences $D_{\mathrm{local\_hist}}$ in Table 1. We can see that $D_{\mathrm{local\_hist}}$ is always larger than D_{hist} and the difference between D_{hist} and $D_{\mathrm{local\_hist}}$ is large, which better reflects the spatial characteristics of videos. The difference between D_{hist} and $D_{\mathrm{local\_hist}}$ is calculated as follows,

$$P_D = \frac{D_{\mathrm{local\_hist}} - D_{\mathrm{hist}}}{D_{\mathrm{hist}}} \times 100\% \tag{2}$$

Table 1. The comparison of global luminance histogram differences and the sum of local luminance histogram differences

Sequences	Total frames	Scene change numbers	$P_D < 0$	$0 < P_D \leq 1\%$	$1\% < P_D < 100\%$	$P_D \geq 100\%$
Kimono1	240	1	0	0	1	0
A Little Red Flower	4089	92	0	5	40	47
Wolf Warrior 2	4423	185	0	0	95	90
Report on COVID-19 Prevention and Control in Beijing	2659	42	0	0	19	23

2.2 The Degree of Inter-frame Content Variation Between Frames

By analyzing the content characteristics of videos, it can be seen that when there is not scene change, the contents of two frames in the same scene are basically the same, and the difference of local brightness histogram is small. When scene change occurs, the contents of the two frames differ greatly, and the local brightness histogram difference between the two frames is significantly larger than that between the two frames without scene change. We determine the current CTU as a content change block by comparing the local luminance histogram difference of CTU at the same position between two frames and the size of the set threshold in this paper. The local luminance histogram difference of CTU can be calculated as follows,

$$\mathrm{diff}_{i,j} = \sum_{k=0}^{255} |H_{\mathrm{cur},i,j}[k] - H_{\mathrm{ref},i,j}[k]| \tag{3}$$

where $\mathrm{diff}_{i,j}$ is the local luminance histogram difference of CTU between the current frame and the reference one at row i, column j. $H_{\mathrm{cur},i,j}[k]$ and $H_{\mathrm{ref},i,j}[k]$ represent the local luminance histogram of CTU at row i, column j in the current frame and the reference one, respectively.

In order to avoid the influence of noises and false detections on scene-change block, this paper defines the current CTU as a scene-change block by detecting whether there are a certain number of content-changing blocks in neighboring CTUs, and calculates the degree of inter-frame content variation of the current frame based on the scene-change blocks. We use content-changing flag $F_{i,j}$ to indicate whether the block is a content-changing block. When $F_{i,j}$ is equal to one, the block is a content-changing block, otherwise it is not a content-changing block. Figure 5 shows how the four zones around CTU are currently divided.

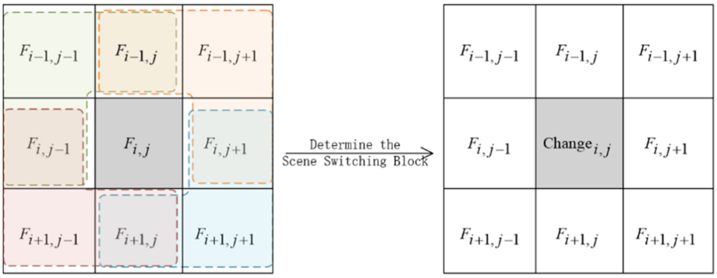

Fig. 5. Division of the area around the current block

$$\mathrm{Change}_{i,j} = \begin{cases} 1, \; (F_{i,j-1} + F_{i-1,j-1} + F_{i-1,j}) > 1 \; or \; (F_{i-1,j} + F_{i-1,j+1} + F_{i,j+1}) > 1 \\ \quad or \; (F_{i,j+1} + F_{i+1,j+1} + F_{i+1,j}) > 1 \; or \; (F_{i+1,j} + F_{i+1,j-1} + F_{i,j-1}) > 1 \\ 0, \; otherwise \end{cases} \tag{4}$$

$$\mathrm{inter\_diff} = \sum\nolimits_i \sum\nolimits_j \mathrm{Change}_{i,j} \tag{5}$$

2.3 The Decision of I Frame

In order to further improve the accuracy of our method, this paper uses the scene-change detection window as the basic unit to traverse the entire video sequence to detect I frame. We set the size of the detection window and the step as 5 according to our experience. Figure 6 shows how the detection window moves and the detection process.

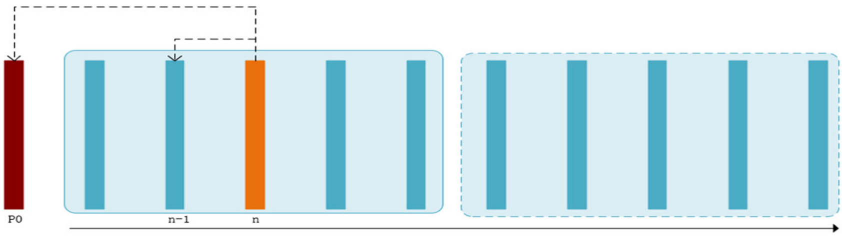

Fig. 6. The movement of the detection window and the detection process

We set the second frame of the video sequence (POC = 1) as the start of the detection window, and five frames in the detection window are detected successively from left to right. In each detection, the previous frame N − 1 of the current frame N and P0 outside the detection window are taken as the reference frames. We calculate the degree of inter-frame content variation $\mathrm{inter\_diff}_{n-1}$ and $\mathrm{inter\_diff}_{\mathrm{P0}}$ respectively, and decide whether the current frame is a keyframe according to the threshold. We set key_frame as the flag of the keyframe.

After five frames in the detection window are detected successively, there are three possible scenarios for the number of keyframes $n_{\mathrm{key\_frame}}$. When $n_{\mathrm{key\_frame}}$ equals to

0, it means that there is no scene-change frame in the current detection window. When $n_{\text{key\_frame}}$ equals to 1, this frame is I frame. When $n_{\text{key\_frame}}$ is greater than 1, we should calculate the average of the degree of content variation of all keyframes avg_diff according to the formula (6) and select the first frame that meets the formula (7) as the final scene-change frame. Figure 7 shows how to select the final scene-change frame.

$$\text{avg\_diff} = \frac{\sum \text{inter\_diff}_{\text{P0}}}{n_{key\_frame}} \tag{6}$$

$$\text{change\_frame}_n = \begin{cases} 1, \ \text{key\_frame} = 1 \text{ and inter\_diff}_{n-1} > 1.1 \times \text{avg\_diff} \\ 0, \ \text{otherwise} \end{cases} \tag{7}$$

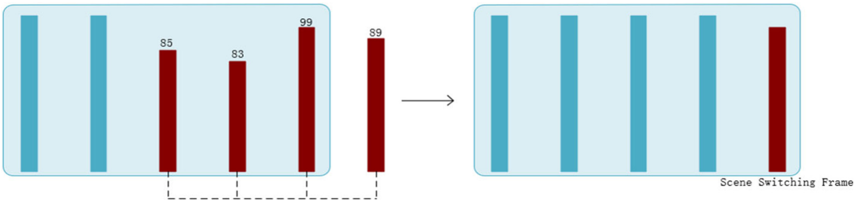

Fig. 7. The rectification of keyframes

In addition, considering the problem that frequently inserting I frames will result in a sharp increase in bit rate, this paper introduces *Flag* between Windows to smooth the density of I frames. Figure 8 shows the process of smoothing I frames.

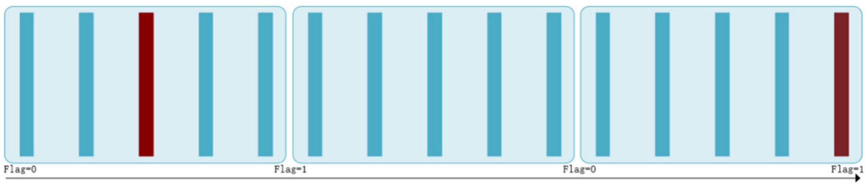

Fig. 8. Smooth I frames

Step 1: Initialize *Flag* ($Flag = 0$) and find I frame in the detection window.
Step 2: If I frame is detected and $Flag = 0$, the Flag will be changed to 1 ($Flag = 1$). Otherwise, the detected I frame is cleared and perform the next detection.
Step 3: The *Flag* will be reset when the I frame is no longer detected in the detection window.

2.4 The Decision of GPB Frame and B Frame

By analyzing the motion characteristics of the video sequence, it can be seen that the content of the video sequence with violent motion varies greatly in two successive frames, while the content of the video sequence with gentle motion is almost the same in two successive frames. If the same Mini-GOP size is used for videos with different motion characteristics, the differences in coding efficiency of two videos will be inevitably large.

On the basis of Formula 2, we use Formula 8 to calculate the overall difference of the current frame and the previous one.

$$\text{frame\_diff}_{n,n-1} = \sum\nolimits_i \sum\nolimits_j \sum_{k=0}^{255} |H_{n,i,j}[k] - H_{n-1,i,j}[k]| \tag{8}$$

The Mini-GOP size obtained by the adaptive frame type decision algorithm based on Viterbi thought in x265 is the Mini-GOP ground truth. Figure 9 shows the correlation between the Mini-GOP ground truth and $\text{frame\_diff}_{n,n-1}$. It's obvious that when $\text{frame\_diff}_{n,n-1}$ is greater than 3000, the ground truth concentrates on 1 and 2, and the corresponding GPB frame interval is 0 and 1. And when $\text{frame\_diff}_{n,n-1}$ is less than 1500, the ground truth concentrates on 4 and 5, and the corresponding GPB frame interval is 3 and 4. The correlation between the ground truth and $\text{frame\_diff}_{n,n-1}$ in the video sequence is fully explained.

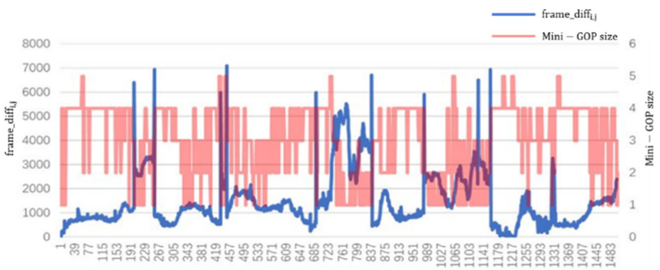

Fig. 9. The correlation between the Mini-GOP ground truth and $\text{frame\_diff}_{n,n-1}$

Firstly, our method decides the Mini-GOP size according to Algorithm 1. Traversing the video whose frame I has been determined for the first time, the video is divided into several Mini-GOPs according to the Mini-GOP size. Last frames in each Mini-GOP are set as GPB frame, and the others are set as B frame. In addition, I frame in the video sequence includes Clean Random Access (CRA) frame and Instantaneous Decoding Refresh (IDR) frame. If I frame is CRA, it will be processed as the end frame of the previous Mini-GOP according to the operation logic of GPB frame (without changing the frame type). And the first frame after CRA is the start of a new Mini-GOP. If I frame is IDR, it will be processed according to the logic at the end of the video sequence. The previous frame will be set as GPB frame and be regarded as the end frame of the previous Mini-GOP. IDR will be regarded as an independent Mini-GOP, and the first subsequent frame will be regarded as the start of a new Mini-GOP.

Alogrithm 1

```
Input: diff_i[n], Max_Size, Thr0, Thr1, Thr2, Thr3
Output: Mini_GOP_Size
j=0, flag0=false, flag1=false, flag2=false
while (j<Max_size-1) do
    if diff_i[j]>Thr2 then
        flag2=flag1=flag0=true
    else if diff_i[j]>Thr1 then
        flag1=flag0=true
    else if diff_i[j]>Thr0 then
        flag0=ture
    end if
    if (diff_i[j]>Thr3) then
        Mini_GOP_Size=j+1
        goto 25
    else if (j>1 and (diff_i[j]>Thr2 or flag2)) then
        Mini_GOP_Size=j+1
        goto 25
    else if (j>2 and (diff_i[j]>Thr1 or flag1)) then
        Mini_GOP_Size=j+1
        goto 25
    else if (j>3 and (diff_i[j]>Thr0 or flag0)) then
        Mini_GOP_Size=j+1
        goto 25
    end if
    j++
end while
Mini_GOP_Size=j+1
return
```

Secondly, the video is traversed for the second time, and the number of layers of coding structure L is determined according to Table 2. When L equals to 1, the display order is the coding order. When L is greater than 1, the video is determined according to the Formula 9 to realize the rearrangement from display order to encoding order.

Table 2. The relationship between Mini-GOP size and the number of layers of coding structure

Mini-GOP size	The number of layers L
1	1
2	2
3–5	3
6–8	4

$$
\begin{aligned}
&L = 2 \quad && M[n], M[1] - M[n-1] \\
&L = 3 \quad && M[n], M[\frac{n}{2}], M[1] - M[\frac{n}{2} - 1], M[\frac{n}{2} + 1] - M[n-1] \\
&L = 4 \quad && M[n], M[\frac{n}{2}], M[\frac{n}{4}], M[1] - M[\frac{n}{4} - 1], M[\frac{n}{4} + 1] - M[\frac{n}{2} - 1], \\
& && M[\frac{3n}{4}], M[\frac{n}{2} + 1], M[\frac{3n}{4} - 1], M[\frac{3n}{4} + 1], M[n-1]
\end{aligned} \tag{9}
$$

3 Experimental Results

In order to verify the performance of the proposed method, we compared our method with the relevant algorithms in x265 and conducted the experiments on x265 3.0 with AMD Ryzen 7 4800H with Radeon Graphics whose main frequency is 2.90 GHz and the memory is 16.0 GB. The software is Microsoft Visual Studio 2019 experimental platform.

3.1 Performance of Scene-Change Detection Method Comparison

In this subsection, correctly detected scene-change TP and wrongly detected scene-change FP were used to measure the detection accuracy. At the same time, we use the structural similarity Index Measure (SSIM) and the peak signal-to-noise ratio (PSNR) to measure the objective compression performance. And running time was used to measure the coding complexity. During the experiment, we selected 8 video sequences with different scenes switching situations, including Traffic (2560 × 1600), Kimono (1920 × 1080), FourPeople (1280 × 720), BQMall (832 × 480), RaceHorses (416 × 240), A Little Red Flower (1920 × 1056), Wolf Warrior 2 (1280 × 720), Report on COVID-19 Prevention and Control in Beijing (832 × 468). Notably, Traffic, Kimono, FourPeople, BQMall and RaceHorses are involved in JCT-VC test set, and A Little Red Flower, Wolf Warrior 2 and Report on COVID-19 Prevention and Control in Beijing are selected from the Internet [16–18]. From Table 3 to Table 5, we can find that our scene-change detection method greatly reduces the complexity of the algorithm and improves the coding performance (Table 4).

3.2 Performance of GPB Frame and B Frame Decision Method Comparison

In this subsection, we selected the coding performance and time of the fixed frame type decision algorithm with x265 as the anchor. In order to compare our algorithm with the fast frame type decision algorithm based on inter-frame rate distortion cost and the adaptive frame type decision algorithm based on Virerbi thought in x265. The encoding complexity can be investigated as follows,

$$
\Delta T = \frac{T_A - T_F}{T_F} \times 100\% \tag{10}
$$

Table 3. Performance of the proposed scene-change detection method with x265 in terms of the detection accuracy

Sequences	Scene cut numbers	x265			Proposed		
		Detection numbers	TP	FP	Detection numbers	TP	FP
Traffic	0	0	0	0	0	0	0
Kimono	1	1	1	0	1	1	0
FourPeople	0	0	0	0	0	0	0
BQMall	0	0	0	0	0	0	0
RaceHorses	0	0	0	0	0	0	0
A Little Red Flower	92	66	59	7	75	75	0
Wolf Warrior 2	185	160	154	6	125	121	4
Report on COVID-19 Prevention and Control in Beijing	42	18	16	2	35	35	0

Table 4. Performance of the proposed scene-change detection method with x265 in terms of the objective compression performance

Sequences	Resolution	BDBR-PSNR	BDBR-SSIM
Traffic	2560 × 1600	−4.79%	−3.98%
Kimono	1920 × 1080	−2.17%	−1.57%
FourPeople	1280 × 720	−1.70%	−1.14%
BQMall	832 × 480	−7.95%	−4.05%
RaceHorses	416 × 240	−1.29%	−1.67%
A Little Red Flower	1920 × 1056	−0.41%	−1.45%
Wolf Warrior 2	1280 × 720	−1.76%	−1.77%
Report on COVID-19 Prevention and Control in Beijing	832 × 468	−1.04%	−0.94%

where T_F represents the encoding time of the fixed frame type decision algorithm with x265, T_A represents the encoding time of the fast frame type decision algorithm based on inter-frame rate distortion cost, the adaptive frame type decision algorithm based on Virerbi thought in x265 or our algorithm.

During the experiment, we selected tested test sequences in the A, B, C, D, E, F classes, including Traffic, PeopleOnStreet, Cactus, Kimono, BasketballDrill, PartyScene, BasketballPass, RaceHorses, FourPeople, vidyo4, SlideShow and SlideEditing.

Table 5. Performance of the proposed scene-change detection method with x265 in terms of the coding complexity

Sequences	Resolution	x265	Proposed
Traffic	2560 × 1600	46071.3	34.13
Kimono1	1920 × 1080	24382.7	10.41
FourPeople	1280 × 720	9684.6	7.34
BQMall	832 × 480	3333.9	2.46
RaceHorses	416 × 240	1249.8	0.92

Table 6 reports that compared with the fast decision algorithm in X265, our algorithm is significantly superior in both compression efficiency and coding complexity. And compared with the adaptive frame type decision algorithm based on Viterbi in x265, our algorithm has a slight lead in compression efficiency and saves more than 5% coding time on average. In addition, considering different motion characteristics of video sequences, the compression performance of our method is significantly reduced for video sequences with obvious lens occlusion or irregular motion (such as rotation and scaling), including Cactus, Basketballpass and PartyScene. However, it has a good compression performance for gently moving or still video sequences, including Kimono, Fourpeople, SlideEditing, etc.

Through the experimental results, our method improves the compression efficiency for most videos with encoding time reducing (Table 7).

Table 6. Performance of the proposed GPB frame and B frame decision with x265 in terms of compression efficiency

Class	Sequences	Proposed		x265 fast method		x265 Viterbi method	
		BDBR-PSNR (%)	BDBR-SSIM (%)	BDBR-PSNR (%)	BDBR-SSIM (%)	BDBR-PSNR (%)	BDBR-SSIM (%)
Class A	Traffic	0.5	0.1	2.6	7.8	−1.2	−0.2
	PeopleOnStreet	−1.1	−0.8	0.1	2.6	−2.3	−0.7
Class B	Cactus	3.9	2.7	5.7	8.2	−0.6	1.1
	Kimono	−1.4	−2.2	−1.2	0.4	−2.4	−2.2
Class C	BasketballDrill	−1.3	2.4	4.2	6.7	0.1	0.8
	PartyScene	2.1	5.5	8.2	9.8	−1.3	1.2
Class D	BasketballPass	1.7	0.4	1.0	2.1	−2.1	−0.4
	RaceHorses	−1.7	−1.2	−0.2	2.6	−1.1	0.0
Class E	FourPeople	−3.8	−4.7	3.7	4.5	1.5	1.6
	vidyo4	−1.4	0.8	1.6	0.2	3.2	2.4
Class F	SlideShow	−9.8	−5.9	−0.9	4.9	−10.4	−4.5
	SlideEditing	−10.6	−6.4	−0.7	−0.3	−7.4	−8.1
Average		−1.91	−0.78	2.01	4.13	−2.00	−0.75

Table 7. Performance of the proposed GPB frame and B frame decision with x265 in terms of coding complexity

Class	Sequences	Proposed	x265 fast method	x265 Viterbi method
		ΔT (%)	ΔT (%)	ΔT (%)
Class A	Traffic	−1.1	1.6	2.5
	PeopleOnStreet	−0.5	0.5	3.2
Class B	Cactus	0.6	2.7	6.4
	Kimono	−1.6	1.6	2.4
Class C	BasketballDrill	0.8	2.4	7.6
	PartyScene	1.4	6.4	4.7
Class D	BasketballPass	−0.8	4.7	4.2
	RaceHorses	−1.4	4.3	9.5
Class E	FourPeople	−0.2	2.0	2.8
	vidyo4	−0.6	3.4	2.0
Class F	SlideShow	−0.1	3.8	5.2
	SlideEditing	−1.7	2.5	8.6
Average		−0.43	2.99	4.93

4 Conclusion

In order to improve the efficiency of HEVC and reduce the complexity, this paper takes CTU as the basic unit to partition each frame of the video sequence and obtain local luminance histogram, and proposes an adaptive frame type decision algorithm based on local luminance histogram. Firstly, the degree of inter-frame content variation is calculated according to the difference of local luminance histogram. Secondly, the scene-change detection window is used to traverse the whole video sequence, and the scene-change frame is determined by comparing the degree of inter-frame content variation. Finally, the Mini-GOP size is adaptively determined through the degree of inter-frame content variation, and the types of frames in the whole video are determined. The experimental results show that the proposed method can effectively improve the coding efficiency while reducing the computational complexity by nearly 5% compared with the relevant algorithm in x265, and realize the efficient adaptive decision of video frame type with strong robustness.

Acknowledgement. This work was supported in part by the National Key Research and Development Program of China under Grant No. 2018YFF01010100, the Beijing Natural Science Foundation under Grant No.4212001 and the Basic Research Program of Qinghai Province under Grant No. 2021-ZJ-704.

References

1. ITU–T, and ISO/IEC JCT–VC: High efficiency video coding [S]. ITU–T Rec. H.265 and ISO/IEC 23008–2 (HEVC), April 2013
2. BITMOVIN Inc.: Video developer report (2019). https://go.bitmovin.com/
3. Sullivan, G.J., Ohm, J.-R., Han, W.-J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circ. Syst. Video Technol. **22**(12), 1649–1668 (2012)
4. Correa, G., Assuncao, P., Agostini, L., et al.: Performance and computational complexity assessment of high–efficiency video encoders. IEEE Trans. Circ. Syst. Video Technol. **22**(12), 1899–1909 (2012)
5. Bossen, F., Bross, B., Suhring, K., et al.: HEVC complexity and implementation analysis. IEEE Trans. Circ. Syst. Video Technol. **22**(12), 1685–1696 (2012)
6. MulticoreWare Inc.: x265 HEVC Encoder/H.265 Video Codec. http://www.x265.org/
7. Shu, H., Chau, L.P.: A new scene change feature for video transcoding. In: International Symposium on Circuits and Systems (ISCAS 2005), Kobe, Japan, 23–26 May 2005. DBLP (2005)
8. Ding, J.R., Yang, J.F.: Adaptive group-of-pictures and scene change detection methods based on existing H.264 advanced video coding information. IET Image Process. **2**(2), 85–94 (2008)
9. Lee, J., Kim, S.J., Lee, C.S.: Effective scene change detection by using statistical analysis of optical flows. Appl. Math. Inf. Sci. **6**(1), 177–183 (2012)
10. Kim, J.R., Suh, S., Sull, S.: Fast scene change detection for personal video recorder. IEEE Trans. Consum. Electron. **49**(3), 683–688 (2003)
11. Kang, S.J.: Positional analysis-based scene-change detection algorithm. In: 2015 IEEE International Conference on Consumer Electronics, Taiwan, China, pp. 11–12 (2015)
12. Cho, S.I., Kang, S.J.: Histogram shape-based scene-change detection algorithm. IEEE Access **7**, 27662–27667 (2019)
13. VideoLAN ORG x264 The best H.264/AVC encoder. https://www.videolan.org/developers/x264.html
14. Liu, Z., Wang, L., Li, X., et al.: Optimize x265 rate control: an exploration of lookahead in frame bit allocation and slice type decision. IEEE Trans. Image Process. **28**(5), 2558–2573 (2018)
15. Forney, G.D.: The viterbi algorithm. Proc. IEEE **61**(3), 268–278 (1973)
16. "Wolf Warriors 2" clip: Leng Feng and Rachel compete with mercenaries. https://www.mgtv.com/l/100009562/4052086.html?lastp=so_result
17. "A Little Red Flower" clip: "Trilogy of Life" directed by Han Yan. https://www.mgtv.com/b/348435/10455525.html?fpa=se&lastp=so_result
18. Report on COVID-19 prevention and control in Beijing. https://www.mgtv.com/b/340731/10748501.html?fpa=se&lastp=so_result
19. Bjontegaard, G.: Calculation of average PSNR differences between RD-curves. In: 13th Video Coding Experts Group Meeting, Austin, TX, USA, pp. 290–291 (2001)