Temporal Dependent Bit Allocation Scheme for Rate Control in HEVC

Jianfeng Xie*†, Li Song*†, Rong Xie*†, Zhengyi Luo‡, Xiangwen Wang‡
*Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University
†Cooperative Medianet Innovation Center, Shanghai, China
‡School of Computer and Information Engineering, Shanghai University of Electric Power
Email: {richrd, song_li, xierong}@sjtu.edu.cn, llzzyynjupt@sina.com, wxw21st@gmail.com

***Abstract*—In this paper, we propose a temporal dependent bit allocation scheme for rate control in state-of-the-art High Efficiency Video Coding (HEVC) standard, to improve its coding performance by utilizing the temporal correlation information. In typical rate control scheme, the bit is allocated to different Coding Tree Units (CTUs) according to their complexity, which reflect their texture and motion information in some degree. However, this scheme ignores the temporal correlation between different frames, which leads to the bit allocation is not the optimal for the whole sequence. Therefore, we refine the bit allocation by taking into account temporal information, especially different coding units' distortion influence on the future frames. The result shows that under the same rate control performance, proposed scheme can effectively improve the coding quality performance, especially the Structure Similarity (SSIM) quality performance, the subjective assesment also shows very well.**

***Index Terms*—video coding, HEVC, rate control, temporal dependent, bit allocation**

I. INTRODUCTION

Rate control plays a very important role in video coding, especially in real time communication application, due to the limited bandwidth at present. Specifically, at a given bit-rate, the coded video quality should be improved as high as possible. For most of rate control algorithm, the common solution is via reasonably allocating bit to various frames and blocks. For example, MPEG-2 applies TM5 [1] rate control algorithm, MPEG-4 utilizes VM8 [2], H.264/MPEG-4 Advanced Video Coding (AVC) [3] uses the rate control algorithm specified in [4]. For HEVC, there are three main rate control schemes, which are described in [5] [6] [7], respectively.

But all of the above algorithms have a common disadvantage that in the CTU level's bit allocation, the estimated bit is based on the blocks complexity, that is Mean Absolute Difference (MAD) in most time. This allocation scheme has not considered the coding block's correlation in the temporal respect. In other words, this kind of bit allocation may only be in favor of current block, which leads to a local optimization rather than global optimization. For example, there is a coding unit that has a low complexity but a high temporal correlation. According to these existing schemes, this coding unit will be coded in a low quality way. But because of its high temporal correlation, it has a great probability to be referenced in the following several frames. So these frames' quality may suffer performance degradation. For these reasons, in this paper we designed a bit allocation scheme which incorporates temporal dependence. Specifically, when coding current unit, we take the temporal related unit into account. If it will be referenced many times, we called it has a high temporal weight, then we should allocate to it more bits. This way can effectively ensure those temporal related units' coding quality. Considering the performance of the above three rate control schemes, we use R-λ model as our benchmark in this paper.

The remainder of this paper is organized as follows. In Section II, we briefly introduce the state-of-the-art λ domain rate control model that has been used for HEVC. In Section III, the temporal dependent bit allocation scheme applying to HEVC is proposed. Section IV shows the experimental results and some necessary analysis about R-D performance and coding complexity. Section V draws the conclusion.

II. OVERVIEW OF THE R-λ RATE CONTROL MODEL

Similar with other rate control scheme, the algorithm procedure of R-λ model mainly includes two part: one is bit allocation and the other is calculation of lagrange multiplier λ and quantization parameter QP. In [7], Li builds an exponential relationship between rate and lagrange multiplier, which is modeled as

$$\lambda = \alpha \times bpp^{\beta} \tag{1}$$

where bpp indicates the bit per pixel, if the target bit of one CTU or a frame is T and number of pixels is N, then the bpp is calculated as

$$bpp = \frac{T}{N} \tag{2}$$

α and β are the model parameter, they are updated after coding one CTU or frame every time. Then QP can be determined thorough a empirical equation [7].

$$QP = 4.2005 \ln \lambda + 13.7122 \tag{3}$$

Bit allocation will be realized in three layers, which include GOP layer, picture layer and CTU layer. In GOP layer, target bit is determined by global target bitrate, frame rate and number of frames in current GOP. In picture layer, target bit is determined by the left bit in current GOP and left number

978-1-4673-9604-2/15/.00

of frames. In CTU layer, the bit is allocated to every CTU according to their weight, which can be expressed as

$$T_{CTU} = \frac{T_{Pic} - Coded_{Pic}}{\sum\limits_{NotCodedCTUs} \omega_{CTU}} \times \omega_{CurCTU} \tag{4}$$

where T_{Pic} is target bits of current picture, $Coded_{Pic}$ is the number of bits coded in the current picture. ω_{CTU} is the weight of CTU. In R-λ model, the weight of one CTU is measured by predicted MAD, which can be calculated as

$$MAD_{CTU} = \frac{1}{N} \sum_{i=0}^{N-1} |P_{org}(i) - P_{pred}(i)| \tag{5}$$

where N is the number of pixels in CTU, $P_{org}(i)$ is the pixel value of original picture, $P_{pred}(i)$ is the pixel value of predicted picture, which is achieved from the previously coded picture belonging to the same picture level. Then the weight of CTU is determined by

$$\omega_{CTU} = {MAD_{CTU}}^2 \tag{6}$$

After all level's bits have been allocated, the lagrange multiplier and the quantization parameter can be determined by equation (1) and (3). Then rate control procedure can be conducted in video coding.

III. TEMPORAL DEPENDENT RATE CONTROL SCHEME

The key issue about temporal dependent rate control is scaling the temporal weight of every coding block. The open source video coding software x264 [8] utilize the MB-tree [9] to do the similar work. But it is not so accurate, which may leads to a limited performance improvement. Here we utilize a much more accurate algorithm, source distortion propagation model [10], to measure coding block's temporal weight.

A. Source Distortion Based Temporal Propagation Chain

The source distortion propagation model refines the rate distortion optimization problem as follow

$$\min_{o_i} \left\{ \sum_{j=i}^{N} D_j + \lambda_g \cdot R_i \right\} \tag{7}$$

where o_i indicates the coding parameter of coding unit i. D_j indicates the distortion of coding unit j, which is a temporal related unit following unit i in the next several frames. N indicates that we consider N units totally. λ_g indicates the lagrange multiplier that have a global optimization for all the N units. Considering that the distortion of the current unit may spread over to parts of several different units in the next coding frames, the above approach comprehensively considers all the related units coding parameter choice rather than the current unit independently. So the final choice is more accurate for the entire sequence.

In order to measure the distortion of the future units, in this model, a temporal propagation chain is constructed as Fig 1 shows. CU_1 is the current CU which need to be encoded. The forward search finds best matched CU, CU_2, in the next frame. If the next CU across several CUs, then it use the closet CU's motion vector as its vector. Then CU_2 searches for the best matched CU CU_3, in frame 3. Again and again, we can construct the temporal propagation chain.

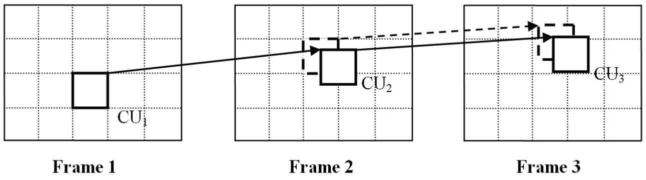


Fig. 1. Temporal propagation chain

Under the assumption of high rate, the relationship between unit distortion D_{i+1} and motion compensation predicted error D_{i+1}^{MCP} can be expressed as

$$D_{i+1} = e^{-b \cdot R_{i+1}} D_{i+1}^{MCP} \tag{8}$$

where R_{i+1} indicates the bit of unit $i+1$, b is a constant related to source distribution. D_{i+1}^{MCP} can be approximately represent as

$$\begin{aligned} D_{i+1}^{MCP} &= \left\| F_{i+1} - \hat{F}_i \right\|^2 = \left\| F_i - \hat{F}_i + F_{i+1} - F_i \right\|^2 \\ &\approx \alpha \cdot \left(\left\| F_i - \hat{F}_i \right\|^2 + \left\| F_{i+1} - F_i \right\|^2 \right) = \alpha \cdot \left(D_i + D_{i+1}^{OMCP} \right) \end{aligned} \tag{9}$$

F_{i+1} is the pixel value of unit $i+1$, $\hat{F}_i$ is the reconstructed pixel value of unit i, $D_i^{OMCP} = \|F_i - F_{i-1}\|^2$ is the source distortion using forward search on the original frames. α takes 0.94 empirically. After substituting it to expression (7), we have

$$D_{i+1} = e^{-b \cdot R_{i+1}} \cdot \alpha \cdot (D_i + D_{i+1}^{OMCP}) = \beta_{i+1} \cdot D_i + C_{i+1} \tag{10}$$

where $\beta_{i+1} = e^{-b \cdot R_{i+1}} \cdot \alpha$, $C_{i+1} = \beta_{i+1} \cdot D_{i+1}^{OMCP}$, all of them are independent to the current CTU's coding parameter choice o_i . Iteratively, we can estimate all the CTU's distortion in the chain. The k-th CTU's distortion can be expressed as

$$D_k = \beta_k \cdot \beta_{k-1} \cdot \ldots \cdot \beta_{i+1} \cdot D_i + C_k \tag{11}$$

Then expression (6) can be rewritten as

$$\min_{o_i} \left\{ \left(1 + \sum_{s=i+1}^{N} \prod_{t=i+1}^{s} \beta_t \right) D_i + \lambda_g \cdot R_i \right\} \tag{12}$$

The propagation factor is defined as

$$\kappa_i = \sum_{s=i+1}^{N} \prod_{t=i+1}^{s} \beta_t \tag{13}$$

It measures how much proportion information of current unit is spread to its temporal related units. From (7) (9), we have

$$\beta_{i+1} = \frac{D_{i+1}}{D_i + D_{i+1}^{OMCP}} = \alpha \cdot \frac{D_{i+1}}{D_{i+1}^{MCP}} \tag{14}$$

where D_{i+1}^{MCP} can be estimated through D_i and D_{i+1}^{OMCP} according to (8), D_{i+1} can be estimated through D_{i+1}^{MCP} and quantization step Q:

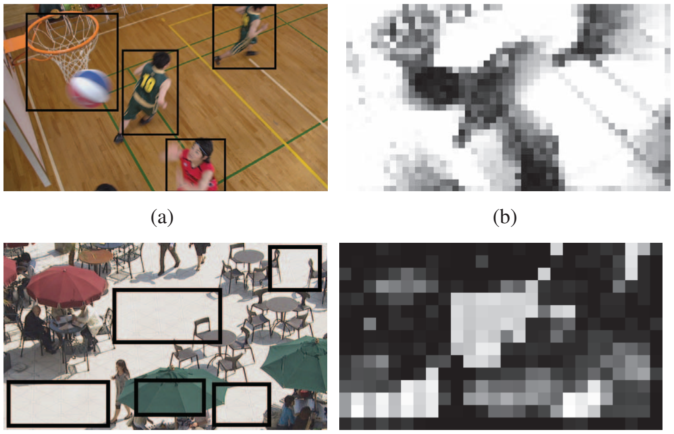
$$D_{i+1} = D_{i+1}^{MCP} \cdot F(\theta) = D_{i+1}^{MCP} \cdot F\left(\frac{\sqrt{2}Q}{\sqrt{D_{i+1}^{MCP}}}\right) \qquad (15)$$

where $F(\theta)$ have a stable distribution for most sequence and can be achieved by a lookup table. More information can be found in [10].

B. Intuitive Validation

About the effectiveness of this model, An intuitive validation is given in this subsection. In the above model, the measurement index indicating temporal weight is the propagation factor κ_i. A higher propagation factor indicates that current unit has a higher chance to be referenced in the next frames. Here we calculate every 16x16 size blocks' propagation factor, regularize them into [0, 1] and show them through an enhanced gray image.

In Fig 2, (a) and (c) is the original picture of sequence $BasketballDrill$ and $BQSquare$ respectively. (b) and (d) is their propagation factor map image. In the gray map image, a high gray scale means a high propagation factor or high temporal weight. Comparing (a) and (b), it can be found that for these black rectangle marked regions which have very high motion intensity and change very fast. They have a low chance to be referenced in the next several frames. That means they should have a relative low temporal weight. The result showing in (b) confirms this conclusion. Similarly, in (c), these regions marked by black rectangle have low motion intensity. They have a relative high temporal weight as shows in (d).


(a) (b)

(c) (d)

Fig. 2. Temporal weight map image(left: original picture, right: weight map)

C. Temporal Dependent Bit Allocation

As we discussed above, the current bit allocation scheme has not considered the temporal correlation between different frames. Using source distortion propagation model, we can incorporate coding units' temporal characteristic into the rate control frame. For the GOP layer and picture layer, the original bit allocation scheme is adopted. In the CTU layer, the temporal weight is taken into bit allocation process.

We use $\hat{T}_{CTU}$ to indicate the target bit determined by temporal weight, which can be expressed as

$$\hat{T}_{CTU} = \frac{T_{Pic} - Coded_{Pic}}{\sum\limits_{NotCodedCTUs} k_{CTU}} \times k_{CurCTU} \qquad (16)$$

where T_{Pic} is target bits of current picture, $Coded_{Pic}$ is the number of bits coded in the current picture. The temporal weight k_{CTU} is defined as

$$k_{CTU} = \kappa_{CTU} + c \qquad (17)$$

where κ_{CTU} is the propagation factor derived from temporal propagation chain. Because the temporal weight might be zero, which means the current CTU is independent to the next few frames, so temporal weight is defined as the propagation factor plus a constant c which avoids the division by zero error. Considering that the propagation factor is much larger than 1 in most of situation, so the constant c is set as 1.

On the other hand, the original complexity bit allocation is also taken into account in order to keep a stable rate. Specifically, the final target bit is set as the weighted mean of temporal weight target bit and complexity weight target bit. We use $\tilde{T}_{CTU}$ to indicate the target bit determined by complexity weight, which is calculated through equation (4). Finally, the target bit compromised between temporal weight and complexity weight can be expressed as

$$T_{CTU} = \mu \times \hat{T}_{CTU} + (1 - \mu) \times \tilde{T}_{CTU} \qquad (18)$$

where μ is a weight factor raging in [0, 1]. In our experiment, it is set to 0.5.

At the same time, in order to avoid very few extreme case that the final target bit is far from the real encoding bit, which may lead to the quality fluctuation too drastic, the following limited target bit is used in the experiment.

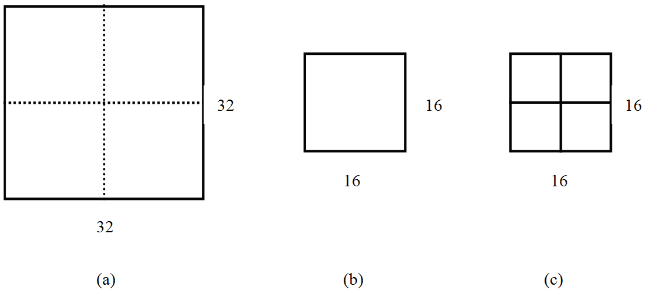
$$T_{CTU} = \begin{cases} 4\tilde{T}_{CTU} & T_{CTU} > 4\tilde{T}_{CTU} \\ T_{CTU} & 0.25\tilde{T}_{CTU} \leq T_{CTU} \leq 4\tilde{T}_{CTU} \\ 0.25\tilde{T}_{CTU} & T_{CTU} < 0.25\tilde{T}_{CTU} \end{cases} \qquad (19)$$

D. Implementation Issues

In HEVC [11], each picture is partitioned into non-overlapped CTUs that have a size of 64x64 pixels. Each CTU can be recursively split in a quadtree fashion into CUs, down to leaf CUs which usually have a minimum size of 8x8 pixels. When construct the temporal propagation chain, the coding unit size has four choices, from 64x64 size to 8x8 size. The bigger size means a lower computing complexity, but a rougher temporal weight estimation. On the contrary, smaller size means a more accurate temporal weight estimation, but a higher computing complexity. In our experiments, a trade-off size, 16x16 pixels, is chosen as the test configuration.

Another issue need to be considered is that when constructing the temporal propagation chain, every coding unit's distortion information of previous frame should be recorded.

Specifically, when calculating the temporal weight, we need to estimate the motion compensation prediction error D_{i+1}^{MCP} first. And this process will use the previous frame unit distortion as shown in equation (9). As we discussed above, the CU size may changes from 64x64 to 8x8 pixels. It seems that we cannot achieve the 16x16 size unit distortion in a straightforward manner.



(a) (b) (c)

Fig. 3. Three kind of CU splitting cases

We propose to resolve this as follows. For every CTU, its partition mode and every CU's distortion are recorded when it is encoded. Every CTU includes 16 units whose size is 16x16 pixels. For every 16x16 size unit, if the CU where it resides in has a best partition mode which is 16x16 pixels size, then we can directly achieve its distortion, as shown in Fig 3(b). If the partition mode has a size bigger than 16x16 pixels, we use the average distortion for every 16x16 unit as the estimated distortion, as shown in Fig 3(a). If the partition mode has a size smaller than 16x16 pixels, we use the sum of all the small CUs' distortion as the estimated distortion, as shown in Fig 3(c). This may not very accurate, but it's simple and effective enough for the estimation process.

IV. EXPERIMENTAL RESULTS

Experiments are conducted to test the proposed rate control scheme in terms of rate control accuracy, R-D performance. Specifically, bitrate error, PSNR and SSIM are recorded to compare with the HM16.0 anchor rate control algorithm. The benchmark used in the experiment is HM16.0 which has incorporated R-λ rate control algorithm. The encoding Group of Picture (GOP) configuration uses the default low delay P coding structure. CTU partition uses the default configuration (e.g. CTU size is 64x64 pixels, Maximum CTU depth is 4). The target bitrate is set according to HEVC call for proposal [12], which is shown in Table I. When constructing the temporal chain, we use a 17x17 full search window and set the chain length as 20 frames. And the standard test sequences in different classes provided by HEVC are adopted.

A. R-D Performance

Two popular quality metrics, PSNR and SSIM, are adopted in the experiment, where the latter is a more effective video quality metric than the former in the sense of providing a good approximation of the perceptual visual quality degradation.

Table II shows the BD-Rate of Proposed method vs. HM16.0 anchor rate control algorithm. The average BD-Rate with PSNR distortion is -1.78%, the average BD-Rate with SSIM distortion is -4.19%. In the best situation, the proposed approach can achieve a significant coding gain up to 6.46% for PSNR and 13.08% for SSIM. That means at the same video quality, proposed scheme needs much less bitrate than the anchor scheme.

TABLE I
TARGET BITRATE

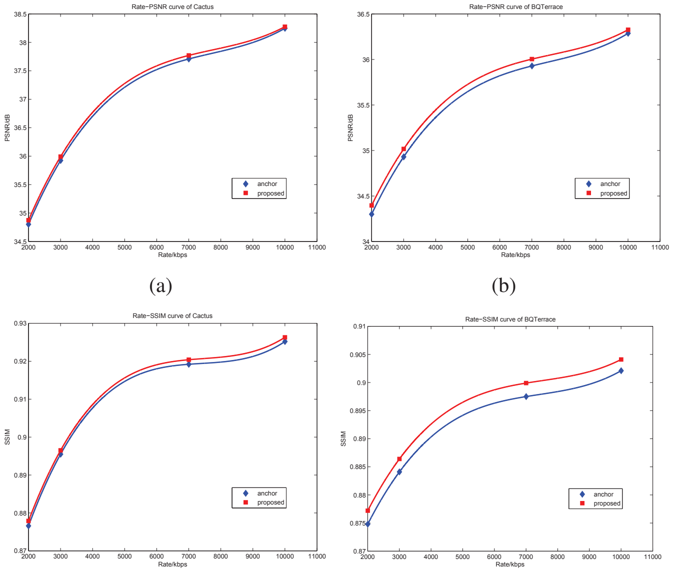
	Sequence	Target bitrate/kbps
ClassB	Kimono, ParkScene	(6000, 4000, 1600, 1000)
	Cactus, BasketballDrive, BQTerrace	(10000, 7000, 3000, 2000)
ClassC	All sequences	(2000, 1200, 512, 384)
ClassD	All sequences	(1500, 850, 384, 256)
ClassE	All sequences	(1500, 850, 384, 256)

TABLE II
BD-RATE (%) VS HM16.0 ANCHOR RC

	Sequence	BD-Rate (PSNR)	BD-Rate (SSIM)
ClassB	Kimono	-1.61%	-3.33%
	ParkScene	-3.03%	-4.32%
	Cactus	-3.03%	-3.62%
	BasketballDrive	-3.63%	-7.39%
	BQTerrace	-6.46%	-13.08%
ClassC	BasketballDrill	-0.89%	-1.14%
	BQMall	-1.67%	-3.66%
	PartyScene	-2.73%	-3.86%
	RaceHorses	-0.92%	-2.51%
ClassD	BasketballPass	-0.39%	-3.51%
	BQSquare	0.61%	-9.66%
	BlowingBubbles	-1.09%	-5.30%
	RaceHorses	-0.17%	-3.82%
ClassE	FourPeople	-2.78%	-1.79%
	Johnny	1.30%	1.63%
	KristenAndSara	-1.90%	-1.72%
Average		-1.78%	-4.19%

The Rate-Distortion curves of two sequence, $Cactus$ and $BQTerrace$, are plot in Fig 4. They visually show that both of PSNR and SSIM have gain at the same bitrate.

What should be paid especial attention to is the significant gain on SSIM. That seems kind of surprise because we have not optimized this quality metric on purpose. Through a careful observation, it can be found that for the regions with low temporal weight, they have high variance at the same time. This can be roughly validated through Fig 2. Moreover, there is a conclusion from [13], which indicates that SSIM distortion is roughly equal to the MSE distortion scaled by the inverse variance of the local region. In other word, for the textured region, higher MSE distortion can be tolerated. So this region can be encoded with a relative low quality mode to improve the entire SSIM quality. Comparing our method, for the low temporal weight region, roughly corresponding to texture region, fewer bits are allocated to them and we code them with relative low quality. So these two methods are consistent in some degree and finally lead to a significant gain on SSIM.



(a) (b)

(c) (d)

Fig. 4. R-D curve (left: $Cactus$, right: $BQTerrace$)

B. Rate Control Performance

To compare the bitrate estimation accuracy of our proposed algorithm, a mismatch ratio is defined by

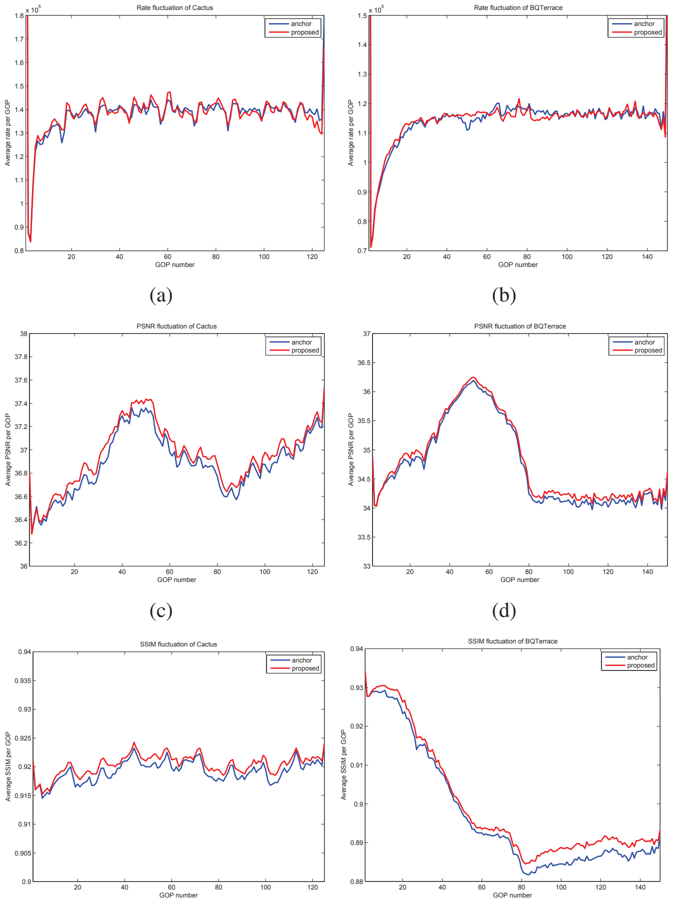
$$M\% = \frac{|R_{actual} - R_{t\arg et}|}{R_{t\arg et}} \times 100\% \tag{20}$$

where $R_{t\arg et}$ and R_{actual} denote the target bit rate and the actual bit rate of the test video sequences respectively. Table III states the bit rate mismatch comparisons between the HM anchor rate control algorithm and the proposed rate control algorithm. Every sequence's mismatch ratio is the average mismatch at four target bitrates. It can be observed that the proposed algorithm has an approximate mismatch comparing with the anchor rate control scheme. So the proposed algorithm basically keeps the same rate control performance.

TABLE III
MISMATCH (%) OF RATE CONTROL

Sequence		Anchor	Proposed
ClassB	Kimono	0.05%	0.05%
	ParkScene	0.04%	0.04%
	Cactus	0.04%	0.04%
	BasketballDrive	0.05%	0.05%
	BQTerrace	0.06%	0.06%
ClassC	BasketballDrill	0.23%	0.24%
	BQMall	0.29%	0.29%
	PartyScene	0.24%	0.24%
	RaceHorses	0.15%	0.15%
ClassD	BasketballPass	0.37%	0.38%
	BQSquare	0.41%	0.41%
	BlowingBubbles	0.35%	0.35%
	RaceHorses	0.21%	0.21%
ClassE	FourPeople	0.40%	0.40%
	Johnny	0.41%	0.41%
	KristenAndSara	0.42%	0.50%
Average		0.23%	0.24%

To illustrate the rate control performance more intuitive, Fig 5 shows the rate, PSNR and SSIM fluctuation of sequence $Cactus$ and $BQTerrace$ at 3000kbps. From (a)(b), it can be seen that the proposed scheme provides a same smooth rate fluctuation as anchor scheme. That means the rate control performance has not degraded. At the same time, the coding quality of the proposed algorithm achieves some improvement. From (c)(d), we can see that PSNR has some gain for most of frames. From (e)(f), it can be seen that SSIM achieves significant improvement. That is beneficial for video's visual quality. More detail will be shown in subsection C.



(a) (b)

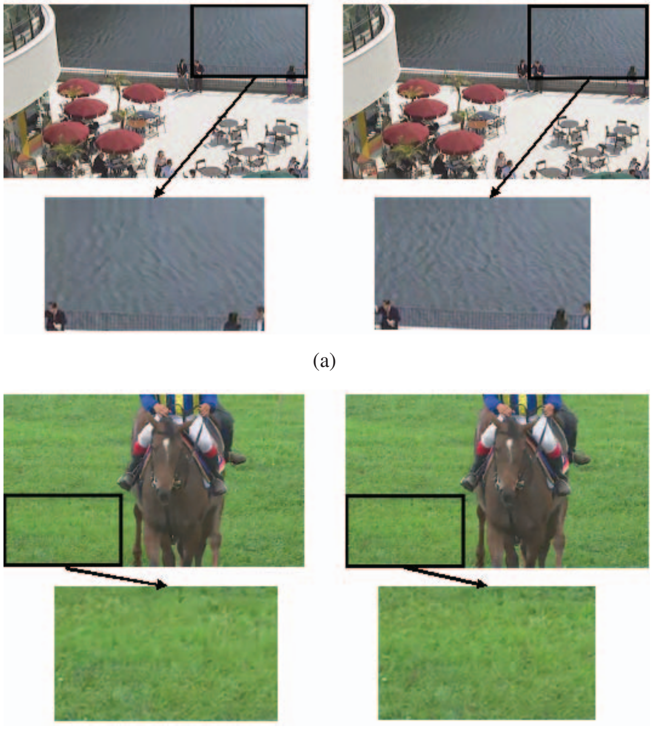
(c) (d)

(e) (f)

Fig. 5. Fluctuation of rate, PSNR and SSIM (left: $Cactus$, right: $BQTerrace$)

C. Subjective Assessment

As we have seen, the proposed scheme can significantly improve the SSIM quality. That indicates the subjective visual quality should have been refined. In this section, we select some frames to illustrate the details. Fig 6 shows the decoded picture of 346th frame of sequence $BQSqaure$ and 132th frame of sequence $RaceHorses$. For every sub-figure, the left is encoded with anchor rate control algorithm, and the right is encoded with proposed algorithm. Comparing them with each other, it can be seen that the proposed algorithm can improve the subjective quality indeed. For example, in Fig 6(a), the ripple region shows much better after using proposed algorithm. Same case can be found in Fig 6(b) on the grass region.


(a)

(b)

Fig. 6. Comparing of subjective quality (left: anchor, right: proposed)

D. Coding Complexity

About the coding complexity, we record some coding time with anchor and proposed scheme, respectively. Then time ratio is calculated with proposed time divided by anchor time. The test platform is a PC equipped with 2.70 GHz Intel(R) Xeon(R) CPU and 64G memory running Microsoft Windows 7 Professional. The comparing result is shown in Table III. From Table IV, we can see that the average time comparing anchor scheme has increased about 3.71%.

Although the increasing time is not too much, we can further decrease the encoding time with some effective algorithm. For example, the most time-consuming chain constructing can use a fast motion search replacing full search. The search block can use a low resolution block, such as 8x8 pixels block, to approximate 16x16 pixels block. Also, for those videos which have low motion strength, we can set a shorter chain length. Through these methods, more time can be saved certainly.

V. CONCLUSION

In this paper, we propose a novel temporal dependent bit allocation scheme for rate control, which refines the current rate control scheme by utilizing temporal information. Based on source distortion propagation model, we estimate every CTU's temporal weight on the next several frames. By substituting the complexity weight with temporal weight, we redefine a bit allocation scheme on CTU level. Experimental results verify the effective performance and reasonable computational complexity.

TABLE IV
COMPARING OF ENCODING TIME

Sequence		Target bitrate/kbps	Time ratio
ClassC	BasketballDrill	2000	104.82%
		1200	102.47%
		512	112.37%
		384	106.18%
	BQMall	2000	111.13%
		1200	102.51%
		512	101.59%
		384	105.07%
	PartyScene	2000	101.42%
		1200	106.54%
		512	104.95%
		384	104.17%
	RaceHorses	2000	100.79%
		1200	100.44%
		512	96.62%
		384	98.27%
Average			103.71%

ACKNOWLEDGMENT

This work was supported by NSFC (61221001, 61271221), the 111 Project (B07022) and the Shanghai Key Laboratory of Digital Media Processing and Transmissions.

REFERENCES

[1] (2014, Jul. 15). Test Model 5 [Online]. Available: http://www.mpeg.org/MPEG/MSSG/tm5/

[2] H.-J. Lee, T. Chiang, and Y.-Q. Zhang, Scalable rate control for MPEG-4 video, IEEE Trans. Circuits Syst. Video Technol., vol. 10, no. 6, pp. 878C894, Sep. 2000.

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, Overview of the H.264/AVC video coding standard, IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560C576, Jul. 2003.

[4] K.-P. Lim, G. Sullivan, and T. Wiegand, Text Description of Joint Model Reference Encoding Methods and Decoding Concealment Methods, document Rec. JVT-N046, Hong Kong, China, Jan. 2005.

[5] H. Choi, J. Nan, J. Yoo, D. Sim, and I. Bajic, Rate control based on unified RQ model for HEVC, Document JCTVC-H0213, San Jos, CA, USA, Feb 2012.

[6] Shanshe Wang, Siwei Ma, Shiqi Wang, Debin Zhao, Wen Gao, Quadratic -domain based rate control algorithm for HEVC, IEEE ICASSP2013, Apr.2013.

[7] Bin Li; Houqiang Li; Li Li; Jinlei Zhang, " Domain Rate Control Algorithm for High Efficiency Video Coding," Image Processing, IEEE Transactions on , vol.23, no.9, pp.3841,3854, Sept. 2014.

[8] x264, available at http://www.videolan.org/developers/x264.html

[9] Garrett-Glaser J. A novel macroblock-tree algorithm for high-performance optimization of dependent video coding in H. 264/AVC[J].

[10] Yang T, Zhu C, Fan X, et al. Source distortion temporal propagation model for motion compensated video coding optimization[C]//Multimedia and Expo (ICME), 2012 IEEE International Conference on. IEEE, 2012: 85-90.

[11] Sullivan G J, Ohm J, Han W J, et al. Overview of the high efficiency video coding (HEVC) standard[J]. Circuits and Systems for Video Technology, IEEE Transactions on, 2012, 22(12): 1649-1668.

[12] ITU-T Q6/16, ISO/IEC JTC1/SCZQ/WG11, VCEG-AM91 (2010) Joint Call for Proposals on Video Compression Technology, 22 January 2010, Kyoto, Japan.

[13] Yeo C, Tan H L, Tan Y H. On rate distortion optimization using SSIM[C]//Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012: 833-836.