

RESEARCH

Open Access

FMO-based H.264 frame layer rate control for low bit rate video transmission

Rhandley D Cajote¹, Supavadee Aramvith^{1*} and Yoshikazu Miyanaga²

Abstract

The use of flexible macroblock ordering (FMO) in H.264/AVC improves error resiliency at the expense of reduced coding efficiency with added overhead bits for slice headers and signalling. The trade-off is most severe at low bit rates, where header bits occupy a significant portion of the total bit budget. To better manage the rate and improve coding efficiency, we propose enhancements to the H.264/AVC frame layer rate control, which take into consideration the effects of using FMO for video transmission. In this article, we propose a new header bits model, an enhanced frame complexity measure, a bit allocation and a quantization parameter adjustment scheme. Simulation results show that the proposed improvements achieve better visual quality compared with the JM 9.2 frame layer rate control with FMO enabled using a different number of slice groups. Using FMO as an error resilient tool with better rate management is suitable in applications that have limited bandwidth and in error prone environments such as video transmission for mobile terminals.

1. Introduction

The H.264/AVC standard [1] has received much attention recently because of its high coding efficiency, error robustness and network friendly architecture. The standard was designed to address a broad class of conversational, broadcast and interactive multimedia services for both wired and wireless environments. The H.264/AVC has the biggest impact in applications where bandwidth is a limiting constraint and robustness to transmission errors is required. An application such as video transmission for mobile wireless environments is a good example where low bit rates are typical and the channel is highly prone to error.

In order to meet the target bit rates demanded by the application and to be able to maximize the video quality, the video encoder implements a rate control algorithm. Since the design of encoders is not covered by standards, designers are free to implement their own rate control algorithms to suit their particular applications.

The H.264/AVC introduces a new error resilient tool called flexible macroblock ordering (FMO) [2], available in the baseline and extended profiles. Using FMO allows flexibility in changing the encoding and transmission

order of macroblocks (MBs) on top of the normal raster scan order. This is accomplished by dividing the picture into *slice groups*, and each slice group can contain several slices. By definition, a slice is a sequence of MBs that belong to the same slice group. The MBs can then be grouped into different slice groups. The H.264/AVC standard supports seven different FMO map types and allows a maximum of eight slice groups per picture for each map type. Six map types are predefined in the standard, as described in [3]. The MB mapping can be specified in the *picture parameter sets* (PPS) with minimal overhead. The seventh map type (type 6), also called the explicit FMO type, allows full flexibility in assigning MBs to slice groups. There is no rule for specifying the slice group mapping when using the explicit map type; this specification, however, requires a higher number of overhead bits since the MB-to-slice group mapping must be specified in the PPS.

The main advantage of using FMO is the ability to contain the spatial propagation of error within the slice boundary. Since each slice is designed to be decodable independently of other slices, using FMO allows the encoder and decoder to resynchronize their states at the slice boundary in the event that there is an error in the bit stream. Using FMO also provides a way to spread the erroneous MBs within the frame and take advantage of the spatial locations of the successfully decoded MBs

* Correspondence: supavadee.a@chula.ac.th

¹Department of Electrical Engineering, Chulalongkorn University, Bangkok 10330, Thailand

Full list of author information is available at the end of the article

for better error concealment. However, using FMO for added error resiliency has some trade-offs in coding efficiency. Coding efficiency is reduced because of the restriction of intra prediction across slice boundaries. The motion vector prediction is affected because of having constrained or dispersed search space. The context adaptive variable length coding/context adaptive arithmetic coding entropy coding is also reset at the beginning of each slice. Using FMO also adds overhead bits because of slice headers and PPS bits. If the MB-to-slice group map, also referred to as an MB address map or an MBA map, is changed in every frame, then a PPS header has to be constructed and inserted in the bit stream.

In the design of the H.264 rate control, the trade-offs in using FMO have not been taken into consideration. The effect is that the target bit rate is often exceeded when the FMO is enabled, especially when the number of slice groups increases. The objective of this article is to present a new frame layer rate control enhancement scheme that takes into consideration the effects of using explicit FMO map types. The idea is to consider the number of motion vector differences in each frame to compute an enhanced mean absolute difference (MAD) measure and frame complexity measure and to develop a quantization parameter (QP) adjustment scheme for rate control.

The rest of the article is organized as follows. In Section 2, we provide background information and related studies about rate control and FMO in H.264. In Sections 3 and 4, we discuss the proposed header bits model and frame complexity measure. In Section 5, the proposed enhancements to the frame layer rate control are presented. The experimental set-up and results are discussed in Sections 6 and 7, followed by the conclusion in Section 8.

2. Related study

The effect of reduced coding efficiency and additional overhead bits when using FMO is progressively severe at low bit rates, where header bits can occupy a significantly larger portion of the total bit budget compared to the source bits. Increasing the overhead bits reduces the number of bits allocated for source coding, resulting in reduced video quality. Thus, when using FMO as an error resilient tool for video transmission at low bit rates, careful consideration of the trade-offs is essential when error rates are high and bandwidth is limited. Our approach is to consider a new header bits model that works well when FMO is enabled to allocate the header bits more efficiently. Also, we propose enhancements to the frame layer rate control to better allocate the source bits.

In order to fully utilize FMO for low bit rate video transmission, the trade-offs must be considered in the

operation of the rate control. The video encoder rate control is responsible for allocating the bits per frame for optimum performance. At low bit rates where every bit is important, the rate control performs the crucial function of mapping a QP to the target bits for each frame and at the same time maintaining good visual quality. In the existing implementation of the adaptive rate control for H.264/AVC [4], there is still some room for improvement in terms of buffer status management, target bits allocation and improved frame complexity measures. Also the trade-offs of using FMO are not taken into consideration.

Numerous studies have been done to improve the performance of H.264/AVC; for example, improvements in the H.264/AVC rate control include adopting new frame complexity measures to enhance the model-based rate control scheme in [4] that uses MAD. In [5], gradient-based complexity measures used in still images are adopted as a measure of frame complexity. The use of the MAD ratio and peak signal-to-noise ratio (PSNR)-based complexity measure has also been explored [6-8] to adjust QP and the bit allocation. In [9], a rate control technique for offline processing using a video quality metric and evolution strategy was proposed; however, this scheme is still computationally complex. In [10], a rate model for header bits is developed and a two-stage encoding process is proposed to improve the rate control. Many other studies have been done on rate control and a recent survey of these studies is provided in [11]. Although a lot of studies have been done to improve the performance of H.264/AVC rate control, very few address the issue of how to make more efficient use of FMO. In [12], a joint source-channel rate distortion analysis is used to adapt the FMO type selection for different video scenes; however, this is only applicable to the fixed FMO types in the standard and does not include the use of the explicit FMO type. In [13], the best frames to be coded with FMO are determined using rate distortion analysis with a rate constraint, but this is implemented with constant QP. In [14], bit rate reduction is accomplished by classifying MBs into two slice groups with similar transform coefficient distributions. However, using only two slice groups limits the error resiliency of FMO. In [15], MBs are classified into different FMO slice groups according to a region of interest and different QPs are assigned to each slice group.

The approach taken so far [14,15] modifies the FMO map to minimize the overhead in bits, and the rate control essentially remains the same. In this article, we take a more proactive approach by proposing enhancements to the H.264/AVC frame layer rate control regardless of the FMO mapping, using an explicit FMO map type, to better control the rate when FMO is enabled. The approach taken is similar to other studies on rate

control [6-8] where frame complexity, target bits and QP adjustment schemes are made to enhance the frame layer rate control. We take this approach further by considering the number of motion vector differences to enhance the MAD and develop a new header bits model with FMO enabled, using a different number of slice groups.

3. Proposed header bits model

Motion vectors of neighbouring MBs are often correlated because object motion can extend over large regions in the frame. In H.264/AVC, this correlation is exploited by computing a motion vector prediction from the MBs in the left, upper and upper-right locations of the current MB being encoded, since the motion vectors of these MBs are already known in a normal raster scan order. The motion vector difference between the prediction and the true motion vector of the current MB is then encoded and transmitted. However, when using FMO for the purpose of error resiliency, the MB ordering can be scattered to minimize the effect of error propagation. In most cases, neighbouring MBs are not available for inter-prediction if they belong to different slice groups. This affects the computation of the motion vector difference and hence affects the coding performance. In this article, we analyse the relationship of the motion vector difference and the number of slice groups to develop a new header bits model that performs well when FMO is enabled.

Previous studies investigated the use of motion vectors to model header bits for the purpose of rate control. In [10], the motion vectors have been used to model the number of header bits of inter-MB and intra-MB. This has been shown to be an effective and accurate model for header bits when FMO is not used. But when FMO is enabled with a different number of slice groups, the model in [10] is no longer accurate, since using FMO greatly affects the motion vector difference but not the actual motion vector.

The header bits model in [10] for inter-MB uses a two-pass encoding process, the number of motion vectors (N_{nzMVe}) and the number of non-zero motion vectors (N_{MV}) gathered from the first pass encoding as shown in (1), where γ and ω are model parameters.

$$R_{hdr,inter} = \gamma (N_{nzMVe} + \omega \times N_{MV}) \quad (1)$$

In order to address the effect on the loss of coding efficiency when using FMO because of the reduced availability of MBs for intermotion prediction, we adapt the model in (1) to model the header bits of P -frames. In this study, we also use a two-pass encoding process to gather modelling data. During the first-pass encoding process of each frame, the number of non-zero motion

vector differences, the number of motion vectors and the number of header bits are obtained for each MB in the frame.

Following the model, data are obtained from the first-pass encoding, and the model parameters are computed using linear regression analysis. The total number of non-zero motion vector differences (N_{nzMVD}), the total number of motion vectors (N_{MV}) and the number of slice groups (num_slice) for a particular frame are used to model the frame header bits (H_{Pframe}) as shown in (2), where α_1 and α_2 are model parameters. In this case, the effects of intra-MBs are not considered since the header information includes only the MB modes; they are not crucial to the accuracy of the model.

$$H_{Pframe} = \alpha_1 N_{nzMVD} + \alpha_2 (N_{MV} + \text{num_slice}) \quad (2)$$

We experimented with the use of three-model parameter, but the performance is almost the same as the two-model parameter since the number of slices is fixed throughout the video sequence. The added computational complexity of linear regression with three parameters is not justified by the improved modelling accuracy.

By using the number of non-zero motion vector differences and including the effect of slice header overhead in the prediction of the frame header bits, we were able to obtain a more accurate header model than that of given in [10]. To compare the accuracy of the two models, the R^2 parameter is computed. The R^2 is a quantity used to measure the degree of data variation from a given model [16], and is defined as (3), where Y_i and \hat{Y}_i are the actual and estimated values of data points i , respectively, and \bar{Y} is the mean.

$$R^2 = 1 - \frac{\sum_i (Y_i - \hat{Y}_i)^2}{\sum_i (Y_i - \bar{Y})^2} \quad (3)$$

when R^2 is close to 1, the model data correlate well with the actual experimental data. Several quarter common intermediate format video sequences were encoded with QP values from 8 to 40 and a frame rate of 10 fps for a total of 100 frames using different numbers of FMO slice groups. The average R^2 value is then computed. A comparison of the R^2 values between the header model in [10] using (1) and our proposed model using (2) is shown in Table 1. The column labels indicate the number of FMO slice groups, i.e. FMO using 2, 4 and 8 slice groups is designated as FMO2, FMO4 and FMO8, respectively. The proposed model has higher R^2 values compared to the model given in [10] and is shown to be better correlated with the number of header bits when FMO is used.

Table 1 Comparison of R^2 values between the models in D.K. Kwon [10] and the proposed modified header bits model using 0 (NoFMO), 2, 4, and 8 slice groups

R^2 Video	NoFMO		FMO2	
	Proposed	[10]	Proposed	[10]
Akiyo	0.798	0.785	0.806	0.774
Carphone	0.917	0.882	0.922	0.887
Claire	0.843	0.820	0.856	0.827
Foreman	0.753	0.668	0.715	0.607

R^2 Video	FMO4		FMO8	
	Proposed	[10]	Proposed	[10]
Akiyo	0.787	0.665	0.756	0.245
Carphone	0.931	0.901	0.937	0.907
Claire	0.854	0.789	0.842	0.634
Foreman	0.738	0.658	0.750	0.668

4. Proposed frame complexity measure

The current implementation of the rate control algorithm in the JM reference software follows the adaptive scheme as described in JVT-G012r [4]. There is however some limitation on the adaptive rate control algorithm and improvements have been proposed by several researchers. The adaptive rate control in [4] has two main objectives: the computation of the number of target bits and the mapping of the target bits to an appropriate QP that will be used for coding the current frame. The computation of the target bits relies on the estimation of the frame complexity using a linear MAD prediction of the previous frames. Since the prediction does not consider the complexity of the current frame to be encoded, the MAD prediction is not an accurate estimate of the frame complexity, especially in complex sequences containing a lot of motion. The mapping of the frame QP to the target bits uses a quadratic rate distortion model; the number of bits allocated for residue depends on the computed target bits and the average header bits used in the previous frames. For low bit-rate applications and complex sequences, the target and header bits are not accurately predicted. Thus, the resulting QP assignment for encoding the current frame may not be optimal. Also the design of the rate control does not consider the overhead of using FMO; hence, whenever FMO is enabled, the adaptive rate control cannot accurately meet the target bits.

Previous study on improving the frame complexity measure is based on modifying the MAD prediction. In [7,8], a more accurate frame complexity measure using the MAD ratio and PSNR-based ratio is computed based on the MAD of the previous frames. In this article, we propose to use the number of non-zero motion vector difference ratios computed from the first-pass encoding process combined with the MAD ratio to improve the estimate of the frame complexity.

We have shown previously in Section 3 that the number of non-zero motion vector differences is a useful parameter to model the header bits and that the amount of motion vector information is also correlated with the complexity of the frame and consequently the amount of bits used for the residue and motion information. Following the framework in [7,8], we compute the non-zero motion vector difference ratio ($N_{nzMVDratio,i}$) as the ratio of the number of non-zero motion vector differences ($N_{nzMVD,i}$) in the i th frame and the average non-zero motion vector difference of all previously coded frames as shown in (4).

$$N_{nzMVDratio,i} = \frac{N_{nzMVD,i}}{\frac{1}{(i-1)} \sum_{j=1}^{i-1} N_{nzMVD,j}} \quad (4)$$

The MAD ratio ($MAD_{ratio,i}$) is computed as the ratio of the predicted MAD of the current frame ($MADP_i$) to the average MAD of all previously coded P -frames in the group of pictures (GOP) using (5).

$$MAD_{ratio,i} = \frac{MADP_i}{\frac{1}{(i-1)} \sum_{j=1}^{i-1} MADP_j} \quad (5)$$

Then, the frame complexity (FC_i) measure for the i th frame is computed by combining the MAD ratio and the N_{nzMVD} ratio, as shown in (6). The model parameter β is set empirically with a value of 0.3 for complex sequences and 0.7 for simple sequences by comparing the variance of the sum of $N_{nzMVDratio}$ per frame with a threshold.

$$FC_i = \beta \cdot MAD_{ratio,i} + (1 - \beta) \cdot N_{nzMVDratio,i} \quad (6)$$

The choice of β is based on experimentation; several values of β were used to encode several video sequences. We computed the R^2 parameter between the frame complexity measure and the actual number of generated bits with different numbers of slice groups. For the Akiyo and Claire sequences, using β from 0.6 to 0.9, the highest R^2 is obtained when $\beta = 0.7$, as shown in Table 2. When $\beta < 0.6$, the computed R^2 is lower, and hence those values are not shown.

Similarly for the Carphone and Foreman sequences, using β from 0.1 to 0.4, the highest R^2 is obtained when $\beta = 0.3$ as shown in Table 3. For other values of β , the R^2 parameter is lower and hence they are not shown.

To determine a threshold value to decide when to use $\beta = 0.3$ for simple sequences and $\beta = 0.7$ for complex sequences, we computed the standard deviation of the sum of $N_{nzMVDratio}$ per frame. We determined the average of the standard deviations for all the test sequences at different rates as shown in Table 4. This average

Table 2 Comparison of R^2 values between the computed frame complexity model and the number of generated bits for different values of β using the Akiyo and Claire sequences

R^2	Beta			
	0.6	0.7	0.8	0.9
Akiyo				
NoFMO	0.899	0.902	0.902	0.890
FMO2	0.904	0.907	0.907	0.901
FMO4	0.906	0.907	0.905	0.896
FMO8	0.894	0.895	0.893	0.884
R^2	Beta			
Claire	0.6	0.7	0.8	0.9
NoFMO	0.845	0.847	0.841	0.820
FMO2	0.844	0.845	0.836	0.811
FMO4	0.824	0.823	0.815	0.790
FMO8	0.841	0.840	0.830	0.802

value is normalized by the rate, as shown in the last column of Table 5 and these are used as the threshold values.

To determine the accuracy of the frame complexity model, we compare the actual generated bits and the computed frame complexity measure using (6) for several test sequences. The Carphone sequence (complex sequence) was encoded at a fixed QP of 32, corresponding to a bit rate of approximately 48 kbps, so that the generated bits will be proportional to the frame complexity. The normalized generated bits were compared with the frame complexity measure using (6) of our modified rate control algorithm with no FMO and FMO with eight slice groups. These are shown in Figure 1a,b.

As shown in Figure 1, the computed frame complexity from (6) correlates well with the actual number of generated bits. A similar trend is observed with other test sequences with different numbers of slice groups.

Table 3 Comparison of R^2 values between the computed frame complexity model and the number of generated bits for different values of β using the Carphone and Foreman sequences

R^2	Beta			
	0.1	0.2	0.3	0.4
Carphone				
NoFMO	0.867	0.894	0.894	0.866
FMO2	0.879	0.898	0.897	0.874
FMO4	0.872	0.896	0.900	0.885
FMO8	0.884	0.892	0.897	0.884
R^2	Beta			
Foreman	0.1	0.2	0.3	0.4
NoFMO	0.701	0.691	0.639	0.519
FMO2	0.731	0.742	0.729	0.677
FMO4	0.742	0.760	0.758	0.727
FMO8	0.724	0.746	0.750	0.731

Table 4 The computed standard deviation of the sum of $N_{nzMVDratio}$ ratios at different bit rates for all test video sequences

Standard dev. of sum of $N_{nzMVDratio}$					
Rate (kbps)	Akiyo	Claire	Carphone	Foreman	Avg.
20	31.29	30.26	40.31	43.65	36.38
32	39.38	35.88	53.53	59.47	47.06
48	45.48	39.22	61.66	68.20	53.64
64	47.04	43.63	74.48	77.97	60.78
96	50.12	45.80	79.77	90.22	66.48

The average value is used as the basis of the threshold for β .

Hence, the enhanced frame complexity measure using (6) is an accurate measurement of frame complexity and can be used to adjust the QP assignment to improve the frame layer rate control.

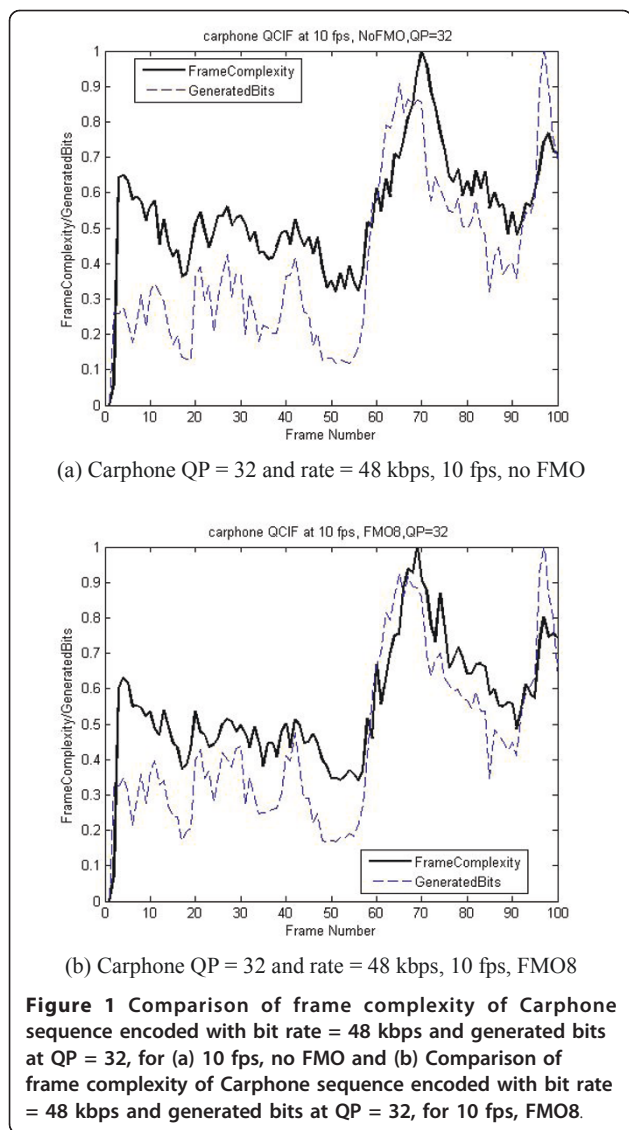
5. Proposed frame layer rate control enhancements

The purpose of rate control is to compute QP for all frames within the allowable rates. With FMO enabled, the effect on the rate control is the increased number of header bits because of PPS and slice headers, and higher buffer levels because of loss of coding efficiency as compared to not using FMO. The proposed improvements to the frame layer rate control of H.264/AVC are improved bit allocation by modifying the target bit using the frame complexity measure, enhancement of the existing MAD complexity measure, a new header bits model and adjustment of QP with FMO considerations.

It can be assumed, without loss of generality, that the GOP structure is *I*PPP..., where *I* is an intra-coded picture and *P* is a forward-predicted picture. The adaptive rate control scheme in the H.264/AVC is composed of two layers: the GOP layer rate control and the frame layer rate control. An additional basic unit layer rate control is added if the size of the basic unit is smaller than a frame. It was noted in [4] that using a bigger basic unit, a higher PSNR can be achieved with higher bit fluctuations, and using a smaller basic unit there will be smaller bit fluctuations with a slight loss in PSNR. Since we want to maximize PSNR for this study, the

Table 5 The computed normalized standard deviation of the sum of $N_{nzMVDratio}$ ratios at different bit rates for all test video sequences

Normalized standard dev. of sum of $N_{nzMVDratio}$					
Rate (kbps)	Akiyo	Claire	Carphone	Foreman	Thresh.
20	1.56	1.51	2.02	2.18	1.82
32	1.23	1.12	1.67	1.86	1.47
48	0.95	0.82	1.28	1.42	1.12
64	0.74	0.68	1.16	1.22	0.95
96	0.52	0.48	0.83	0.94	0.69



basic unit is selected as a frame so there is no need for an additional basic unit layer rate control. In addition, only the frame layer rate control is modified; the operation of the GOP layer rate control remains the same.

The operation of the GOP layer rate control is described briefly as follows. At the beginning of the GOP, the GOP layer rate control computes the total number of bits for the GOP and assigns an initial QP for the first *I*- and the first *P*-frame. For the succeeding *P*-frames, the number of remaining bits in the GOP is updated based on the generated bits of the previous frame. The details of the GOP layer rate control may be found in [4].

The operation of the frame layer adaptive rate control algorithm in H.264/AVC is composed of three parts: determining the target bits for each *P*-frame, computing the QP and adjusting the QP. The operations of each

component are discussed in the following sections, along with the proposed enhancements.

5.1. Computation of the frame layer target bits

To compute the target bits for each frame, the fluid flow traffic model is used based on linear tracking theory [17]. The number of target bits (T_{buf}) for the i th frame is computed based on the current buffer fullness (CBF), target buffer level (TBL), frame rate, and available channel bandwidth, as shown in (7).

$$T_{buf,i} = \left[\frac{b_r}{f_r} - \Gamma (CBF_{i-1} - TBL_i) \right] \quad (7)$$

In (7), b_r and f_r denote the bit rate and frame rate, respectively. The CBF and the TBL are denoted as CBF_{i-1} and TBL_i , respectively. In the JM reference software, Γ is a constant with a typical value of 0.5. The initial values for CBF_{i-1} and TBL_i are computed at the GOP layer rate control.

Target bits (T_{rem}) for the i th frame are also computed, based on the remaining bits in the GOP, as the ratio of the remaining bits in the GOP and the number of non-coded *P*-frames, $T_{rem,i} = R_i/N_i$.

To obtain better estimates of the target bits, we adjust the computation of T_{rem} to consider the frame complexity FC_i (see Section 3). We denote the modified target bits as T_{mod} as shown in (8).

$$T_{mod,i} = \begin{cases} FC_i \cdot T_{rem,i} & 0 < FC_i < 1.0 \\ 1.1 \cdot T_{rem,i} & 1.0 \leq FC_i < 1.2 \\ 1.2 \cdot T_{rem,i} & 1.2 \leq FC_i \end{cases} \quad (8)$$

The parameters in (8) are derived empirically from experiments. The idea is to set $T_{mod,i}$ to larger values for frames with higher frame complexity and to set $T_{mod,i}$ to smaller values for frames with lower frame complexity. This is done to save bits from the less complex frames and allocate more bits to more complex frames.

The total number of bits allocated for the i th frame (T_i) is computed as a weighted combination of the target bits computed from the TBL and buffer occupancy ($T_{buf,i}$) and the target bits computed from the remaining bits in the GOP ($T_{mod,i}$) as shown in (9).

$$T_i = \beta_r \cdot T_{mod,i} + (1 - \beta_r) \cdot T_{buf,i} \quad (9)$$

In (9), the typical value of β_r in the JM reference software is 0.5.

5.2. Using the proposed header bits model

In H.264 after computation of the target bits, the number of bits allocated for texture is computed by subtracting the estimate of the number of header bits from the

computed target bits. The estimate of the number of header bits is computed as the average number of header bits of previously coded P -frames. Previous studies have found that the number of header bits varies greatly from frame-to-frame and a simple average is not a good estimate of the header bits [10].

The proposed improvement to the frame layer rate control of H.264/AVC is the modification of the estimate of the header bits using the proposed header bits model, as computed using (2), to consider the effect of FMO and slice header overhead. This modification gives a more accurate estimate of the header bits and consequently makes the bit allocation for the texture bits more accurate as well. The number of bits allocated for texture ($T_{\text{txt}, i}$) is computed as shown in (10).

$$T_{\text{txt}, i} = T_i - H_{\text{Pframe}, i} \quad (10)$$

After the estimated header bits are subtracted from the computed target bits, QP for the i th frame is computed from the remaining texture bits using the quadratic rate-distortion model [14].

5.3. QP adjustment scheme using frame complexity

After computing QP using the quadratic rate-distortion model, QP is further adjusted to ± 2 of the previous QP to maintain smoothness of visual quality. This kind of adjustment is not sufficient in some cases, especially when FMO is used. We further adjust QP depending on whether the target bit is positive or negative and a lower bound is imposed on the texture bits.

When the computed number of target bits per frame is low, i.e. there is a low bit rate and a high complexity frame, there is a high probability that number of target bits will fall below zero for the succeeding frames. In this case, the QP is adjusted to be larger than 2 from the previous frames resulting in poor video quality. The effect is severe when FMO is used with eight slice groups where the number of target bits is observed to be negative most of the time, especially in complex sequences. Thus, it is important to prevent negative target bits to maintain smooth visual quality. As an improvement, we use the computed frame complexity, the buffer status, and the number of slice groups to adjust QP to maintain positive target bits for improved performance.

Depending on the amount of header bits, the remaining number of bits for texture can be too small; in this case, a lower bound is imposed on the texture bits given by (11).

$$T_{\text{texture}} = \max \left\{ T_{\text{texture}}, \frac{b_r}{\text{MINVAL}_f} \right\} \quad (11)$$

In the JM reference software, MINVAL is a constant with a typical value of 4. The QP value computed when

using the lower bound usually does not meet the target bits for the current frame; the mismatch is higher when FMO is enabled with a large number of slice groups. Thus, it is necessary to further adjust QP for such cases.

5.3.1. Negative target bits

When the frame is complex and FMO is enabled, the CBF tends to be significantly larger than the TBL. In such cases, the target bits tend to be negative, so the current buffer level must be reduced by increasing QP to maintain positive target bit levels. The amount of QP adjustment depends on the number of slice groups when FMO is used as shown in (12). The adjustments in QP are based on empirical experiment to avoid negative target bits as much as possible. Increasing the number of slice groups increases the header bits because of the slice headers, thus increasing the probability that the current buffer level is higher than the TBL. To keep the target bits positive, we increase QP by 2. In the worst case when the number of slice groups is eight, the rate increases by 12-15%; in this case, we increase QP by 3. Larger adjustments using QP + 4 can achieve tighter control over the buffer, but the drastic change in visual quality becomes annoying. Smoother visual quality and smaller PSNR deviation are maintained by making smaller adjustments in QP.

$$\text{QP} = \begin{cases} \text{QP} + 2 & \text{num_slice_grp} < 4 \\ \text{QP} + 3 & \text{otherwise} \end{cases} \quad (12)$$

5.3.2. Positive target bits

When the computed target bit is positive and the number of allocated bits for texture is greater than the minimum bound using (11), then QP is computed using the quadratic rate-distortion model [18]. To maintain smoothness of visual quality, QP is limited to within ± 2 of the current value between pictures. As an improvement, QP is further adjusted depending on the CBF, frame complexity and number of FMO slice groups as shown in (13). Since the target bits are already positive, we do not need drastic QP adjustments as in the case of negative target bits. The threshold values are set empirically based on the experiments.

$$\text{QP} = \begin{cases} \text{QP} - 1 & \left(\Gamma \cdot (\text{CBF} - \text{TBL}) < \frac{b_r}{f_r} \right) \text{ and } (\text{FC} < 0.9) \\ \text{QP} + 1 & \left(\Gamma \cdot (\text{CBF} - \text{TBL}) > \frac{b_r}{f_r} \right) \text{ and } (\text{FC} > 1.1) \\ \text{and } (\text{num_slc_grp} < 4) & \\ \text{QP} + 2 & \left(\Gamma \cdot (\text{CBF} - \text{TBL}) > \frac{b_r}{f_r} \right) \text{ and } (\text{FC} > 1.1) \\ \text{and } (\text{num_slc_grp} > 4) & \end{cases} \quad (13)$$

The idea is that if the buffer occupancy is low and the frame is not complex, then QP is reduced by 1 to improve the visual quality. If the buffer occupancy is high and the frame complexity is high, then QP is adjusted by 1 to reduce excessive buffer fill-up. Lastly,

when the buffer level is high, the frame is complex, and in the worst case the number of slice groups is 8 and QP is adjusted by 2.

5.3.3. Lower bound on texture bits

When the amount of bits allocated for texture is set to the minimum bound dictated by the bit rate and the frame rate as in (10), QP is simply adjusted by adding 2. Otherwise QP is unchanged as shown in (14).

$$QP = \begin{cases} QP + 2 & T_{\text{texture}} < \frac{b_r}{\text{MINVAL} \times f_r} \\ QP & \text{otherwise} \end{cases} \quad (14)$$

5.3.4. Frame skipping

After encoding the current frame, the number of generated bits is added to the buffer and the model parameters of the rate control are updated. If the current buffer level is above a certain threshold, then the encoder will skip encoding the incoming frame. The initial buffer size (B_s) is set at $3.0 \times (b_r / f_r)$ to simulate a typical low-bit rate and low delay application. The buffer occupancy threshold before skipping a frame is set to $0.8 \times B_s$.

6. Experimental set-up

To analyse the effectiveness of the proposed frame layer rate control enhancement, we modified the frame layer rate control of the JM 9.2 reference software and compared its performance with the original JM 9.2. FMO is enabled using the explicit FMO map type where the MBA map changes in every frame. The encoder is modified to construct and insert a PPS header into the bit stream when FMO is enabled for that sequence.

Four standard video sequences are encoded using the baseline profile at level 3.0. The video sequences are chosen such that there are sequences with low, medium and high motion content. Each frame is encoded four times with no FMO and with FMO enabled with 2, 4 and 8 slice groups. Each sequence is encoded for a total of 100 frames, a frame rate of 10 fps, and at rates of 20, 32, 48, 64 and 96 kbps, respectively. The GOP structure is *IPPP* with one reference frame. The initial QP is 40 to limit the number of bits of the initial *I*-frame.

The PSNR, PSNR standard deviation and total number of skipped frames are used to evaluate the performance

Table 6 Comparison of PSNR and PSNR standard deviation averaged over different bit rates and different numbers of FMO slice groups

Video	Avg. PSNR (dB)			Avg. PSNR std.	
	JM	Proposed	Gain	JM	Proposed
Akiyo	42.11	42.16	0.05	3.37	3.29
Claire	42.67	42.70	0.03	2.99	2.86
Carphone	33.49	33.69	0.19	3.65	3.21
Foreman	31.28	31.92	0.64	3.43	2.11

Table 7 Comparison of PSNR and PSNR standard deviations averaged over different numbers of FMO slice groups at 20 kbps bit rate

20 kbps	Avg. PSNR (dB)			Avg. PSNR std.	
	Video	JM	Proposed	Gain	JM
Akiyo	36.76	37.02	0.25	2.47	2.12
Claire	37.81	37.96	0.15	2.22	1.64
Carphone	28.67	29.24	0.57	3.88	2.70
Foreman	25.80	26.97	1.17	4.60	2.35

Video	Avg. Rate (kbps)		Total Skip	
	JM	Proposed	JM	Proposed
Akiyo	20.09	20.01	39	8
Claire	20.12	19.98	26	0
Carphone	20.30	20.07	86	6
Foreman	20.33	20.19	143	18

of the rate control algorithm compared to the existing implementation as described in [4].

7. Results

The PSNR and standard deviation are averaged at different rates using 20, 32, 48, 64 and 96 kbps and are also averaged for different numbers of FMO slice groups, i.e. no FMO and FMO with 2, 4 and 8 slice groups. The results are summarized in Table 6, and show that the proposed rate control enhancements can improve the PSNR especially for sequences with large motion such as Carphone and Foreman, where the average gain in PSNR is 0.19 and 0.64 dB, respectively. The average PSNR standard deviation is also reduced, which indicates a more stable buffer management and less fluctuation in video quality for all test sequences.

The proposed rate control enhancements perform well at bit rates of 20 and 32 kbps for sequences with medium and high motion content such as Carphone and

Table 8 Comparison of PSNR and PSNR standard deviations averaged over different numbers of FMO slice groups at 32 kbps bit rate

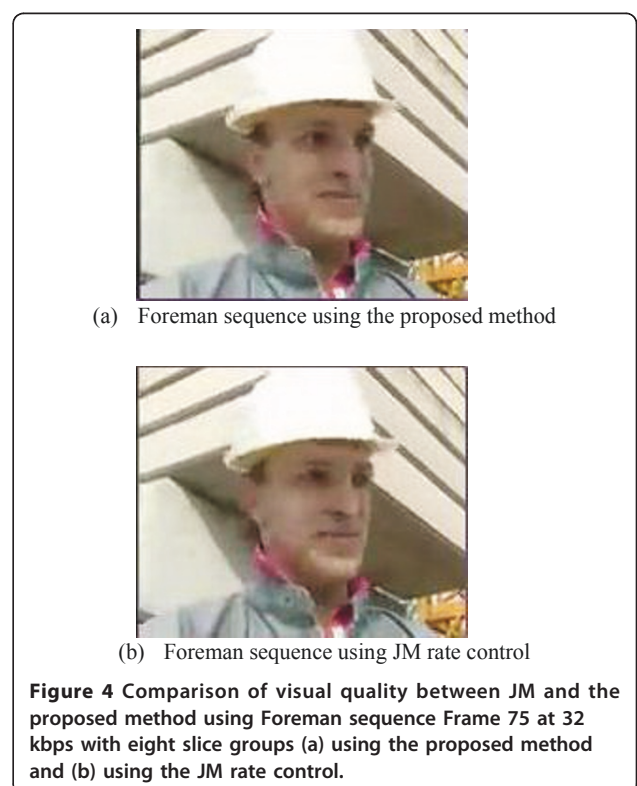
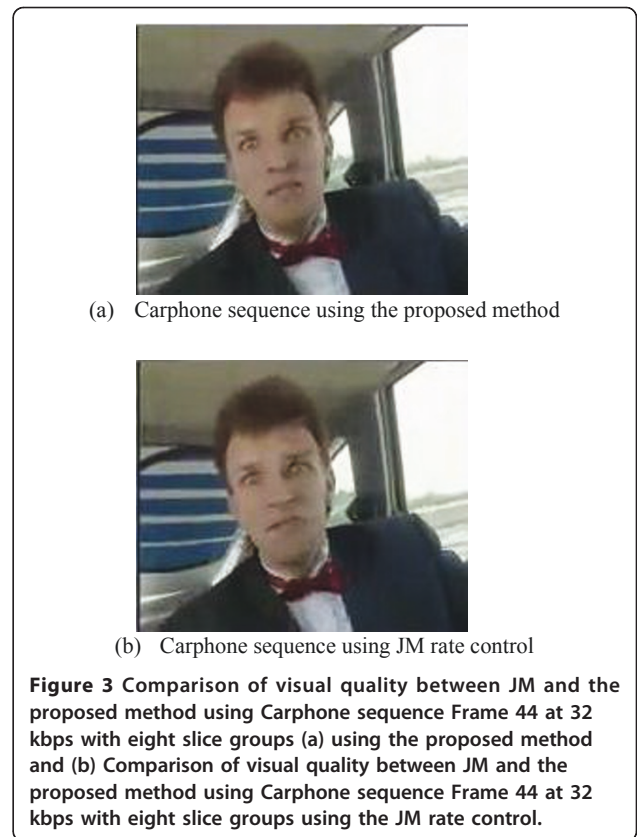
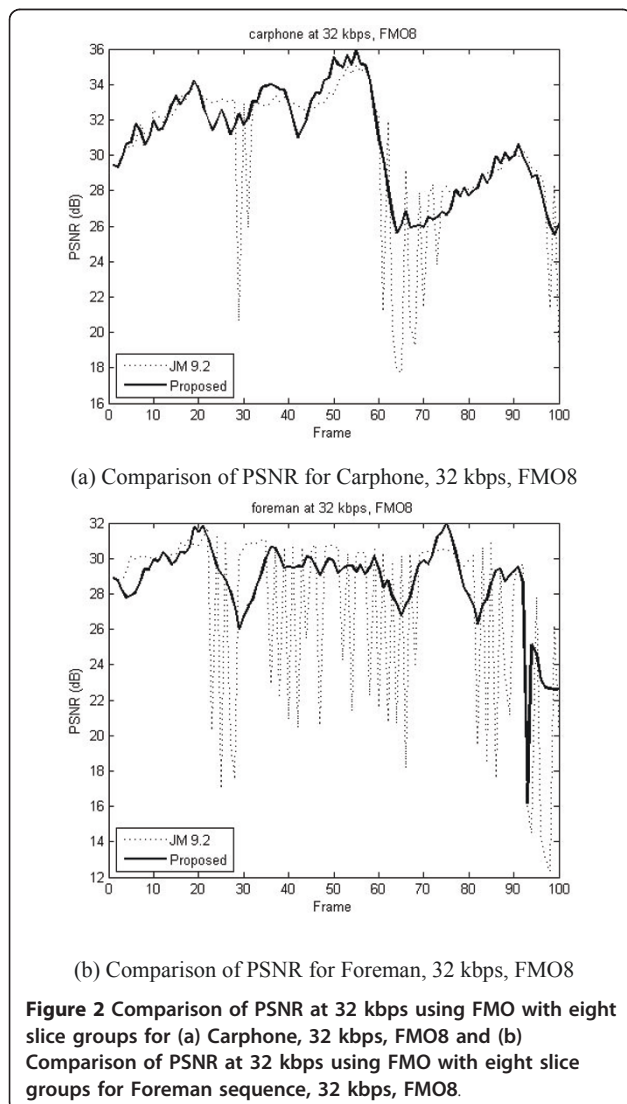
32 kbps	Avg. PSNR (dB)			Avg. PSNR std.	
	Video	JM	Proposed	Gain	JM
Akiyo	40.15	40.17	0.02	2.70	2.70
Claire	40.99	40.96	-0.03	2.36	2.29
Carphone	31.56	31.84	0.29	3.63	2.95
Foreman	28.91	30.21	1.30	4.46	1.94

Video	Avg. Rate (kbps)		Total Skip	
	JM	Proposed	JM	Proposed
Akiyo	32.00	31.97	0	0
Claire	32.06	31.98	2	0
Carphone	32.23	32.09	23	1
Foreman	32.23	32.13	77	2

Table 9 Comparison of PSNR between JM and proposed method for Foreman at different rates and different FMO slice groups

Foreman	Avg. PSNR (dB) NoFMO		Avg. PSNR (dB) FMO2	
Rate (kbps)	JM	Proposed	JM	Proposed
20	27.88	29.06	26.60	27.79
32	31.12	31.38	29.68	30.65
48	33.18	33.28	32.61	32.74
64	34.62	34.60	33.89	34.15
96	36.46	36.50	36.11	36.11

Rate (kbps)	Avg. PSNR (dB) FMO4		Avg. PSNR (dB) FMO8	
	JM	Proposed	JM	Proposed
20	25.15	26.61	23.57	24.43
32	27.67	30.09	27.17	28.73
48	32.10	32.12	29.94	31.61
64	33.78	33.83	32.84	33.36
96	35.78	35.82	35.52	35.51



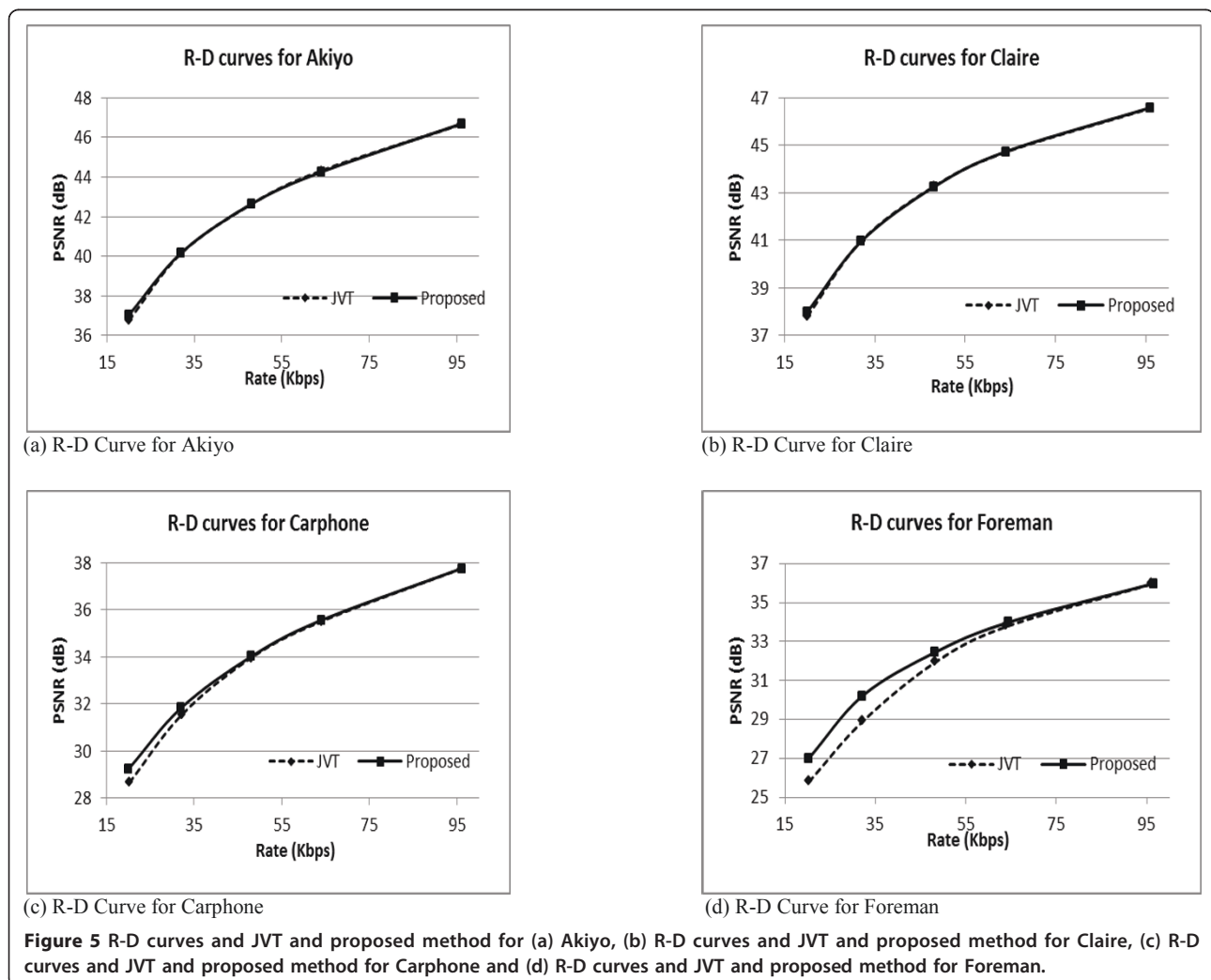
Foreman, as shown by the average PSNR and average rate in Tables 7 and 8. This is because the accuracy of the frame complexity model and header bits model depends on the motion vector difference when FMO is enabled. As an example, a comparison of the performance of the proposed rate control with the JM reference rate control at different FMO settings and at different rates for the Foreman sequence is shown in Table 9. Figure 2a,b shows the PSNR plot per frame of Carphone and Foreman sequences with FMO enabled using eight slice groups at 32 kbps. The plot shows a more stable PSNR and lower number of frames skipped compared to the JM version.

The average PSNR, average standard deviation, average generated bits and total number of skipped frames over all FMO slice group settings are shown in Tables 7 and 8 for 20 and 32 kbps, respectively. Improvements in the PSNR are most significant at low bit rates and for sequences with medium and high motion content. The

PSNR gains for sequences with low motion content, such as Akiyo and Claire, are comparable with the JM rate control. However, it should be noted that PSNR gains are achieved at a slightly lower bit rate. This means that the proposed scheme can allocate the bits more efficiently than the JM rate control. The number of frames skipped is also significantly reduced.

The results of other bit rates are not shown because of space constraints. But, the generalization can be made that at higher bit rates the gains in PSNR, standard deviation and number of skipped frames gradually decrease because the side effects of using FMO are less noticeable at higher bit rates. This is shown by comparing the rate distortion curves of the proposed rate control enhancements with the JM reference software (labelled as JVT) using the sequences under test as shown in Figure 3a-d.

To compare the subjective quality of the video sequence, Figure 4a shows the 44th frame of the



Carphone sequence with eight FMO slice groups at 32 kbps using the proposed rate control enhancements. Figure 4b shows the same frame using the JM rate control with some visible artefacts appearing around the lip area. Figure 5a,b shows the 75th frame of the Foreman sequence with eight FMO slice groups at 32 kbps using the proposed rate control enhancement and the JM rate control. In Figure 5b, some artefacts can be seen in the left eye area.

7. Conclusion

We have presented some improvements to the H.264/AVC frame layer rate control using FMO for added error resiliency. We propose a new header bits model that uses the number of motion vector differences to more accurately model the header bits. A new frame complexity measure is proposed also using the number of motion vector differences to enhance the existing MAD-based frame complexity measure. We propose some target bits modification and QP adjustment schemes considering buffer fullness, frame complexity, and number of FMO slice groups to generate a QP that better allocates the bits for encoding the current frame.

It has been shown that the implemented FMO-based frame layer enhancements generally improve the PSNR and can achieve the target bit rates more accurately compared to the current H.264/AVC rate control at bit rates of 20 and 32 kbps. A smoother video quality is achieved because of the smaller PSNR standard deviation, leading to a more stable buffer management. The number of skipped frames is also significantly reduced at low bit-rates and for high motion sequences, thus improving the overall PSNR.

For our future study, the proposed rate control scheme will be extended to cover the scenario of error-prone channels.

Acknowledgements

This research was supported in part by the Collaborative Research Project entitled Wireless Video Transmission, the JICA Project for AUN/SEED-Net, Japan, and the Thailand Research Fund, grant no. MRG4780212. The authors declare that they have no competing interests.

Author details

¹Department of Electrical Engineering, Chulalongkorn University, Bangkok 10330, Thailand ²Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan

Received: 14 December 2010 Accepted: 18 September 2011

Published: 18 September 2011

References

1. Advanced video coding for generic audiovisual services. ITU-T Rec. H.264/ISO/IEC 14496-10 (MPEG-4) AVC (2003)
2. S Wenger, M Horowitz, FMO: flexible macroblock ordering. ISO/IEC MPEG and ITU-T VCEG: JVT-C089. (May 2002)

3. Y Dhondt, P Lambert, Flexible macroblock ordering as an error resilience tool in H.264/AVC, in *5th FTW PhD Symp*, Ghent University, (December 2004)
4. Z Li, F Pan, KP Lim, G Feng, X Lin, S Rahardja, Adaptive basic unit layer rate control for JVT, in *JVT 7th meeting*, Pattaya, Thailand, (March 2003)
5. Y Zhou, Y Sun, Z Feng, S Sun, New rate-distortion modeling and efficient rate control for H.264/AVC video coding. *Signal Process. Image Commun.* **24**(5), 345–356 (2009)
6. C Lee, S Lee, Y Oh, J Kim, Cost-effective frame-layer H.264 rate control for low bit rate video, in *ICME* (2006)
7. M Jiang, N Ling, On enhancing H.264/AVC video rate control by PSNR-based frame complexity estimation. *IEEE Trans Consum. Electron.* **15**(1), 231–232 (2005)
8. M Jiang, X Yi, N Ling, Improved frame-layer rate control for H.264 using MAD ratio, in *Proceedings of the 2004 International Symposium on Circuits and Systems, ISCAS '04*, **3**, III-813-16, (23-26 May 2004)
9. SLP Yasakethu, WAC Fernando, S Adedoyin, A Kondoz, A rate control technique for offline H.264/AVC video coding using subjective quality of video. *IEEE Trans Consum Electron.* **54**(3), 1465–1472 (2008)
10. D-K Kwon, M-Y Shen, C-C Jay Kuo, Rate control for H.264 video with enhanced rate and distortion models. *IEEE Trans Circ Syst Video Technol.* **17**(5), 517–529 (2007)
11. Z Chen, KN Ngan, Recent advances in rate control for video coding. *Signal Process. Image Commun.* **22**(1), 19–38 (2007)
12. H Chen, Z Han, R Hu, R Ruan, Adaptive FMO selection strategy for error resilient H.264 coding, in *ICALIP* (2008)
13. Z Wu, JM Boyce, Optimal frame selection for H.264/AVC FMO coding, in *ICIP 2006* (October 2006)
14. LT Ha, H-S Kim, C-S Park, S-W Jung, S-J Ko, Bitrate reduction using FMO for video streaming over packet networks, in *PWASET*. **37** (January 2009)
15. AK Kannur, B Li, An enhanced rate control scheme with motion assisted slice grouping for low bit rate coding in H.264, in *ICIP 2008*, San Diego, California, (October 2008)
16. JL Devore, *Probability and Statistics for Engineering and Sciences*, 3rd ed., Pacific Grove: Brookes-Cole, (1991)
17. F Pan, Z Li, K Lim, G Feng, A study of MPEG-4 rate control scheme and its improvements. *IEEE Trans Circ Syst Video Technol.* **13**, 440–446 (2003). doi:10.1109/TCSVT.2003.811603
18. HJ Lee, T Chiang, Y-Q Zhang, Scalable rate control for MPEG-4 video. *IEEE Trans Circ Syst Video Technol.* **10**, 878–894 (2000). doi:10.1109/76.867926

doi:10.1186/1687-6180-2011-63

Cite this article as: Cajote et al.: FMO-based H.264 frame layer rate control for low bit rate video transmission. *EURASIP Journal on Advances in Signal Processing* 2011 **2011**:63.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com