Abstract—One of the target applications of the H.264/MPEG AVC codec is the transmission of video sequences whenever bandwidth and storage capacity are limited (e.g. video telephony or video conferencing over mobile channels and devices). To comply with this requirements, it is necessary to tune the encoder parameters taking into account the buffer level, channel variations, and the resulting visual quality.

This paper describes an accurate rate control algorithm that requires less memory and is less computationally complex than other rate control algorithms. In the proposed approach, the number of coded bits in each frame is accurately predicted from the percentage of null quantized coefficients (called “zeros”). Such a quantity can be directly computed from the energy of the quantized signal ($E_q$). Therefore, the coefficient statistics does not need to be stored, and the image parametrization can be performed with reduced hardware requirements. The performance of the proposed solution outperforms the JVT algorithm we used as a comparison in terms of objective and subjective visual quality.

Keywords—video coding, transform coding, ITU-T Recommendation H.264, MPEG-4 part 10, rate control, R-D parametrization, bit allocation, rate-distortion analysis.

I. INTRODUCTION

The core element in a low-cost rate control algorithm is a parametric model that makes it possible to estimate the number of coded bits in each frame. In fact, an algorithm based on a good model efficiently sets the coding parameters in order to fit the channel constraints. A performing solution is given by the rate-distortion model suggested by S.K. Mitra et al. in [1]–[3]. In these papers, the rate and distortion related to lossy image coding are functions of the percentage of null quantized coefficients in each block (called “zeros”). Experimental results [2] show that the “zeros” parametrization suits a great number of images better than previous models. Therefore, this method was successfully implemented on different transform-based coding standards and proved to be suitable both for local and frame-level analysis.

In this approach the video encoder needs to estimate the quantization parameter (called $QP$) that fits the requested target percentage of “zeros” (called $\rho_z$). Such estimation is usually based either on a histogram of the transform coefficients or on a parametric model. In our work we focused on the second solution since it was found to provide good performances and to be less demanding in terms of computational and memory requirements. In fact, the introduction of another parameter $E_q$, which approximates the energy of the quantized DCT coefficients, permits estimating the parameter $\rho_z$ with sufficient accuracy. The resulting rate control algorithm based on $(\rho, E_q)$-modelling provides good performances in terms of both video quality and control accuracy without actual pre-quantization and pre-coding.

Section II describes the “zero”-parametrization and how the bit rate of coded images can be related to the percentage of null quantized transform coefficients. In addition, this section presents a parametric model avoids storing the histogram of DCT coefficients. The percentage of “zeros” can be related to the parameter $E_q$, and consequently, it is possible to estimate the target quantization parameter with low computational requirements. A rate control algorithm based on this model is described in section III. Finally, in section IV some experimental results are presented and compared with those of the rate control algorithm developed by Joint Video Team ([4], [5]).

II. PARAMETRIC MODELS FOR H.264 COEFFICIENTS ESTIMATED THROUGH THE ACTIVITY

Most of the rate control algorithms are based on a local hyperbolic R-D model [6], where bit rate and distortion are functions of quantization step (e.g. [7]–[9]). This model requires a pre-analysis of individual frames and allows a direct control over quantization. Nevertheless, it can be inefficient in some cases. Whenever the encoder is processing pictures with varying characteristics, the uniform distribution of the bits between macroblocks produces a non-homogeneous quality in the image.

A better solution can be found through the rate-distortion model suggested by S.K. Mitra et al. in [1]–[3], where the bit rate of a coded image is function of the percentage $\rho$ of null quantized coefficients (called “zeros”). Figure 1 shows that the relationship between the overall bit rate $R$ and $\rho$ is well approximated with a straight line

$$ R(\rho) = m \cdot \rho + q $$

(1)

where the parameters $m$ and $q$ depend on the type (I, P, or B) and on the sequence.

![Fig. 1. Plots of bit rate vs. $\rho$ from the coded sequence carphone (GOP IP...P 15 frames), as $QP$ varies between 15 and 51; the plots refer to frame 0 (Intra coded) and frames 30,46 (Inter coded).](image)

Since the H.264 encoder is controlled by the quantization parameter $QP$, the “zeros” percentage has to be referred to one out of the 51 possible values for $QP$. This relation can be obtained through the probability density function (p.d.f.) of the DCT coefficients, which could be provided either by the histogram of the DCT coefficients or by a parametric model. Since the second solution requires lower computational resources and memory, our rate control algorithm uses a parametric p.d.f. of the transform coefficients in order to
compute the most appropriate quantization step corresponding to a target value of $\rho$.

According to such a model, in [10] it was shown that the percentage of “zeros” $\rho$ depends on the ratio

$$\frac{\sigma_x}{\Delta} = \sqrt{E[(x - m_x)^2]} \approx \sqrt{E[x^2]}$$

where $\sigma_x$ is the standard deviation of the DCT coefficients $x$, $x_q$ is the quantized value of $x$, and $\Delta$ is the quantization step.

Moreover, the ratio $\sigma_x/\Delta$ can be approximated in the following way

$$\frac{\sigma_x}{\Delta} = \frac{\alpha_c \Delta}{\Delta} = E_q,$$

where $\alpha_c \Delta$ is the average activity, and $\Delta$ is the average quantization step for the current frame $n$. This simplification reduces the computational requirements since the computation of $E_q$ is based on the activity parameter $\alpha_c \Delta$ of the H.264 encoding process.

According to the experimental results shown in Fig. 2, the relation between $E_q$ and $\rho$ is well approximated by a second order polynomial, i.e., $E_q \approx E_q$ with $\hat{E}_q$ given by

$$\hat{E}_q = c_0 + c_1 \cdot (1 - \rho) + c_2 \cdot (1 - \rho)^2.$$  

From Fig. 2, it appears that each plot depends both on the statistics of the image and the type of coding (I,P,B). Therefore, the values of the coefficients $c_i$ have to be adapted accordingly.

III. A “ZEROS”-BASED RATE CONTROL ALGORITHM

In the previous section it has been briefly described how the number of “zeros” can model the bit rate produced by an H.264 encoder. In this section we show how this parametrization can be used to control the amount of coded bits in order to transmit a video sequence over a channel of a given capacity.

The algorithm adopts a feedback scheme where the control is performed in different steps operating at different levels. The first control is performed at the beginning of each GOP and allocates $G_{k,0}$ bits for the $k$-th group of pictures. The second level deals with the bit rate and the coding parameters of a single picture. Finally, the quantization parameter is corrected at macroblock level in order to fit the global constraints.

In the following subsections each control level is described in detail.

A. Bit rate control at GOP level

Given the target bit rate $R_0$ and the frame rate of input video sequence $F$, the video encoder sets the overall number of bits

$$\overline{B} = \frac{R_0 \cdot N}{F},$$

where $N$ is the number of frames for each group of pictures. This value has to be corrected during the coding process because of bit allocation errors and variations of the available bandwidth.

Bit rate allocation errors are corrected through the equation

$$G_{k,0} = \delta G_{k-1} + \overline{B} = \left(B_c - \frac{B_c}{8}\right)$$

where $G_{k,0}$ represents the available bits before coding the $n$-th frame of the $k$-th GOP and $\overline{B}$ is defined in (5). $\delta G_{k-1}$ is the difference between target and effective bit usage after coding the $(k-1)$-th GOP, and the parameters $B_c$ and $B_e$ respectively refer to the buffer level and the buffer dimension. The GOP level rate control tries to keep $B_c$ as close as possible to $B_e/8$ in order to avoid underflows.

The second type of correction is carried out whenever the transmission bit rate changes, and it is given by the equation

$$G_{k,n} \leftarrow G_{k,n} + \left(R_n' - R_n\right) \cdot \frac{N - n}{F_c}$$

where $R_n'$ is the new available bit rate. These two operations make it possible to adapt the coded bit stream to channel variations.
avoiding transmission delays. However, whenever the available bandwidth is reduced too much, the algorithm starts skipping some B-coded pictures in order to allocate more bits for those frames that are used as references for motion compensation.

In order to deal with fast time-varying channels, the algorithm also considers a “micro-GOP”, i.e. a group made of an I or P-type picture and the following B-type pictures. The number of available bits for a micro-GOP is

$$G_{adj}^m = G_{adj}^{i-2} + \frac{R_b}{F} \cdot (1 + \text{number of } B)$$

where \text{number of } B is the number of B-type pictures between the other ones in the GOP structure, and \( j \) is the number of the micro-GOP in the current GOP. After the coding of each frame, \( G_{adj}^m \) is updated according to the rules presented in the following paragraphs.

B. Bit rate control at frame level

The second level of bit rate control is carried out at frame level. In the current group of pictures, the available bits \( G_{b,n} \) are shared among all the frames that are not yet coded. The number of bits assigned to each picture is computed taking into account the type of coding, the image statistics and the previous bit rate allocation errors.

The bit rate control for the \( n \)-th frame in the \( k \)-th GOP is divided into four steps. First, the target bit rate is computed according to the available number of bits in the GOP, the coding type and the characteristics of the previous images. Second, the algorithm estimates whether it is worth coding the current picture or skipping it. In the first case, the algorithm goes on the third step and computes the average QP value for the current frame. Then, the current picture is coded, and the parameters of image statistics are updated according to the coding results. If the current frame is skipped, the algorithm starts processing the following picture. In the following paragraphs each step will be presented in detail.

B.1 Computation of the target bit rate

Before coding the \( n \)-th frame in the current GOP, the algorithm estimates the target bit rate \( T_n \) at GOP level and the target bit rate \( \bar{T}_n \) at micro-GOP level

$$T_n = \beta \cdot \bar{T}_n + (1 - \beta) \cdot \bar{T}_n \quad n = 0, \ldots, N - 1$$

where

$$\bar{T}_n = K_i \cdot \frac{G_{b,n}}{K_i \cdot n_i + K_P \cdot n_P + K_B \cdot n_B} \quad i = I, P, B$$

in (10) \( n_i \) is the number of remaining i-type frames in the GOP, and \( K_i = K_{i,P} \cdot K_{P,B} \). The target bit rate \( \bar{T}_n \) is computed as a convex combination of the target average quantization step of the target bit rate because of an obsolete value of \( R_b \), and the following frames can be affected by bit starvation.

The quantity \( \bar{T}_n \) is computed through the equation

$$\bar{T}_n = \bar{T}_{n-1} + \delta B_1 + \delta B_2 - \frac{R_b}{F} \quad (13)$$

where

$$\delta B_1 = K_{[i,P]} \cdot \frac{G_{adj}^m}{K_{[i,P]} \cdot n_{adj}^m + n_{B}^m}$$

and

$$\delta B_2 = \left( \bar{T}_n - \frac{B_b}{8} \right) \cdot \frac{K_i}{K_{[i,P]} \cdot n_{adj}^m + n_{B}^m} \cdot R_b.$$ (15)

The parameter \( n_{adj}^m \) is the number of remaining i-type frames in the micro-GOP. All these parameters are updated after the coding of the current frame as it is described in the following paragraphs.

B.2 Frame skipping control

After computing the target number of bits \( T_n \) for the current frame, the rate control algorithm estimates whether skipping the current frame or not. In fact, whenever a picture is skipped, \( T_n \) bits are saved for the following frames. Therefore, frame skipping permits dealing with bit rate allocation errors and scene changes in an efficient way since the algorithm skips those frames that are not used as references for temporal prediction whenever the remaining frames in the GOP suffer from bit starvation. In this way, we avoid an excessive distortion of reference pictures that decreases the motion estimation efficiency.

In the proposed algorithm, a frame is skipped whenever the inequality

$$T_n \leq \frac{R_b}{F} \cdot \frac{N_P + N_B}{8}.$$ (16)

holds.

In addition, whenever the current picture is a B-type frame, the following condition is tested

$$G_{n,b} \leq \frac{N_P + N_B \cdot R_b}{F \cdot 8}.$$ (17)

This test allows the rate control to check whether the coder has already used a greater number of bits than expected. In this case, the current B-type frame is skipped.

B.3 Computation of the \( \bar{Q}T_{n,T}^m \)

As the H.264 encoder is driven by the quantization parameter \( QP \), the algorithm has to compute the average \( QP \) value for \( n \)-th frame from \( T_n \). According to the R-D model presented in Section II, the bit rate \( T_n \) has to be referred to a target average percentage of “zeros” \( \rho_{T,n} \) through the equation

$$\rho_{T,n} = \frac{T_n - q}{m}$$ (18)

where \( m \) and \( q \) are estimated from previously coded pictures (e.g. the \((n-1)\)-th frame). The parameters \( m \) and \( q \) are the slope and the intersect of equation (1).

From equation (4), the parameter \( E_q \) can be computed from \( \rho_{T,n} \) for a given set of coefficients \( c_i \), \( i = 0, 1, 2 \). Therefore, the target percentage of “zeros” \( \rho_{T,n} \) is related to a target quantized signal energy value \( E_{q,T} \) via (4), where the set of coefficients varies according to the coding type of the current frame. Then, the algorithm estimates the target average quantization step \( \bar{Q}_n \) as
\[ \Delta n,T = \frac{\hat{E}_{q,T}}{\Delta n_{prev}} \]  \hspace{1cm} (19)

where \( \Delta n_{prev} \) is the predicted average activity for the current frame. In our approach \( \Delta n_{prev} \) is equal to the average activity of the previous frame of the same coding type. Nevertheless, more efficient prediction scheme can be implemented.

Finally, the target average quantization step \( \Delta n,T \) is converted into an average target quantization parameter \( \frac{QP_{n,T}}{\Delta T} \) as described in [10].

B.3 Parameters update

The rate control algorithm requires estimating the quantization parameter QP corresponding to a given target percentage of “zeros”. In order to provide an accurate control over a wide range of bit rates, an adaptive approach, which requires a reduced computational load and avoids the storage of many coefficients tables as in [10], is adopted. For this purpose, an LMS-based technique proved to be satisfactory.

After coding the \( n \)-th frame, the coefficients \( c_{t,i}, t = I, P, B \) and \( i = 0, 1, 2 \), are updated in the following way. First, the estimation error of \( E_q \) is found through the equation

\[ e_{E_q} = E_q - \hat{E}_q \]

with

\[ \hat{E}_q = \sum_{i=0}^{2} c_{t,i} \cdot (1 - \rho_n)^i \]

where \( \rho_n \) is the actual percentage of “zeros” of the current frame.

Then, the appropriate set of coefficients is updated

\[ c_{t,i} \leftarrow c_{t,i} + \kappa \cdot e_{E_q} \cdot \rho_n^i \]

where \( \kappa \) is the adaptation gain of the estimator.

We kept a low \( \kappa \) value (\( \kappa = 0.01 \)) resetting \( c_{t,i} \) values whenever the relative bit allocation errors are greater than a threshold. The initial values of \( c_{t,i} \) are computed from a training set of sequences coded with constant QP.

In addition, the algorithm updates the slope \( m \) and the intersect \( q \) of equation 1 setting

\[ m \leftarrow \frac{h_n - S_n}{1 - \rho_n} \]

and

\[ q \leftarrow 0.9 \cdot q + 0.1 \cdot (h_n - m) \]

where \( S_n \) is the total number of bits produced and \( h_n \) is the number of header bits for the \( n \)-th frame in the current GOP.

As for the bit rate related parameters, the available number of bits is updated according to

\[ G_{k,n+1} = G_{k,n} - S_n \]

\[ G_{j+1,i}^{m1} = G_{j+1,i}^{m1} - S_n. \]  \hspace{1cm} (26)

In this way, the target bit rate for the following picture is modified compensating previous bit allocation errors.

The constants \( K_{I,P} \) and \( K_{P,B} \) are set to

\[ K_{I,P} = \frac{X_I}{X_P} \quad K_{P,B} = \frac{X_P}{X_B}. \]

and the complexity ratios \( K_{I,P}, K_{P,B} \) between frames of different type are computed from the parameters \( X_i, i = I, P, B \).

In order to avoid sudden changes in the complexity ratios, \( X_i \) is found through the averaging filter

\[ X_i \leftarrow \omega X_i + (1 - \omega) \bar{X}_i \]

where the input signal is the complexity

\[ \bar{X}_i = 2^{\frac{QF_n}{6}} \cdot S_n \]

The parameter \( QF_n \) is the average QP value of the whole picture, and \( S_n \) is defined in (23). The variables \( X_i, \bar{X}_i \) and their corresponding complexity ratios \( K_{I,P}, K_{P,B} \) allow the coding process of the H.264 encoder to adapt to the input video data.

C. Bit rate control at macroblock level

At macroblock level the quantization parameter is corrected according to the number of remaining bits and the percentage of “zeros”. This grants a good control both over picture quality and coded bits keeping bit rate within the given constraints and smoothing visual distortion across different macroblocks. The proposed algorithm uses the same macroblock level control reported in [10].

We adopted the RD-optimization performed by JVT encoder in order to compare our results with the rate control algorithm included in the version JM 7.6 of the encoder. Therefore, in our approach the rate control chooses the quantization parameter while the macroblock coding mode is selected minimizing the cost function

\[ J(mode, QP_m) = D(mode, QP_m) + \lambda \cdot R(mode, QP_m) \]

where \( mode = 0, \ldots 10 \) is the macroblock mode, \( QP_m \) is quantization parameter chosen for the current macroblock, \( D(m, QP_m) \) is the coding distortion, and \( R(m, QP_m) \) is the bit rate. The Lagrange multiplier \( \lambda \) is set in the following way

\[ \lambda = \lambda_0 \cdot 2^{QP/6} \]

with \( \lambda_0 = 0.85 \) for I or P-slices and \( \lambda_0 = 3.4 \) for B-slices (see [4] and [11]).

IV. EXPERIMENTAL RESULTS

In order to evaluate the “zeros” algorithm performance, we coded different sequences at various bit rates using two different rate controls. The first one is the “zeros”-based rate control, while the second one is the algorithm implemented in the Joint Model 7.6 of H.264 by the Joint Video Team (here, denoted with the label JVT [5], [12]).

The configuration parameters of the H.264 video coder are reported in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GOP structure</td>
<td>IPBBP</td>
</tr>
<tr>
<td>GOP length</td>
<td>15 and 60</td>
</tr>
<tr>
<td>Coding algorithm</td>
<td>CABAC</td>
</tr>
<tr>
<td>Search window width</td>
<td>16</td>
</tr>
<tr>
<td>MV resolution</td>
<td>1/4 pixel</td>
</tr>
<tr>
<td>Hadamard</td>
<td>enabled</td>
</tr>
<tr>
<td>Num. of reference frames</td>
<td>1</td>
</tr>
<tr>
<td>RD optimization</td>
<td>enabled</td>
</tr>
<tr>
<td>SP pictures</td>
<td>not used</td>
</tr>
<tr>
<td>Slice mode</td>
<td>not used</td>
</tr>
</tbody>
</table>

Tab. 1. Configuration parameter for the H.264 encoder

For each coded sequence we computed the bit rate and the perceptual quality represented by the PSNR. In addition, we calculated the standard deviation of this parameter \( \sigma_{PSNR} \) in order to evaluate how strongly the visual distortion varied among
different frames. In fact, strong PSNR variations affect the resulting video quality since the displayed sequence looks unnatural and visually unpleasant. A video sequence with great PSNR variations may be worse than a sequence which has a lower average video quality but limited quality variations.

At first, we coded different sequences at different bit rates. The results are reported in Fig. 3, 4, 5, 6 and in Table 2 for QCIF sequences.

Fig. 3a shows the number of allocated bits and the PSNR for each frame of the sequence salesman coded at 128 kbit/s. It is apparent that the video quality of the “zeros” algorithm is less varying even if the allocated number of bits is approximately the same (see Fig. 3b).

![Fig. 3a](image)
(a) Number of bits/frame

![Fig. 3b](image)
(b) PSNR/frame

Fig. 3. Bits/Frame and PSNR/Frame plot of 360 QCIF frames for the sequence salesman (GOP IBBP 60 frames) at 30 frame/s.

The reported data show that the “zeros”-based approach provides a better visual quality (as measured by the PSNR) with respect to the JVT algorithm. In addition, the data in Table 2 and in Fig. 4,5,6 show that the perceptual quality variation (measured by $\sigma_{PSNR}$) is smaller in the proposed algorithm. In fact, the graphs show the experimental distortion-rate curve with superimposed vertical bars that denote $\pm \sigma_{PSNR}$. The figures confirm that the proposed algorithm produces both a greater PSNR value and lower $\sigma_{PSNR}$ at all bit-rates, i.e., both a higher and smoother quality. This fact proved to be independent of the complexity of the sequence.

We performed the same analysis on CIF sequences in order to evaluate the performance of the algorithm with wider-sized pictures. Experimental results are reported in Figure 8 and confirm the previous results.

![Fig. 4](image)

Fig. 4. Distortion-Rate plot of 360 QCIF frames for the sequence foreman (GOP IBBP 15 frames) at 30 frame/s; the superimposed vertical bars denote $\pm \sigma_{PSNR}$.

![Fig. 5](image)

Fig. 5. Distortion-Rate plot of 360 QCIF frames for the sequence salesman (GOP IBBP 60 frames) at 30 frame/s; the superimposed vertical bars denote $\pm \sigma_{PSNR}$.

V. CONCLUSIONS

In this paper, we presented a parametric model that permits a tight control of the bit rate. The model is based on the relation between the number of coded bits in a frame and the parameters $\rho$ and $E_{q}$ that can be easily computed from the H.264 encoder syntax. Moreover, this model was used to implement a rate control algorithm that quickly adapts the coding parameter to the sequence characteristics and the available bandwidth.

According to the experimental results, the “zeros”-based algorithm provides better performance than JVT reference algorithm, and its application to control transmission buffer turns out to be a performing solution in terms of visual quality. In fact, the sequences that were coded using the “zeros” algorithm have higher and smoother visual quality than sequences coded with JVT.

ACKNOWLEDGMENT

This work was carried out within the Italian Ministry of Education, University and Research (MIUR) Project “FIRB PRIMO”.

![Fig. 6](image)
Fig. 7. Distortion-Rate plot of 240 QCIF frames for the sequence news (GOP IBBP 60 frames) at 30 frame/s; the superimposed vertical bars denote $\pm \sigma_{PSNR}$.

Fig. 8. Distortion-Rate plot of 120 CIF frames for the sequence salesman (GOP IBBP 60 frames) at 30 frame/s; the superimposed vertical bars denote $\pm \sigma_{PSNR}$.

REFERENCES


