A Rate Control Algorithm for the H.264 Encoder

Authors: Simone Milani*, Luca Celetto†, Gian Antonio Mian*

e-mail: {simone.milani,mian}@dei.unipd.it, luca.celetto@st.com

* Dept. of Information Engineering, University of Padova - Via Gradenigo, 6/B - 35131 Padova - Italy
† STMicroelectronics, via C. Olivetti 2, 20041 Agrate (MI), Italy

Abstract—The emerging coding standard H.264 is mainly intended for video transmission in all areas where bandwidth or storage capacity is limited (e.g., video telephony or video conferencing over mobile channels and devices). Since its main applications concern video communication over time-varying bandwidth channels, the bit rate has to be controlled with scalable algorithms that can be implemented on low resource devices. The paper describes a rate control algorithm that needs reduced memory area and complexity compared to other ones. The number of coded bits for each frame can be accurately predicted through the percentage of null quantized coefficients. It is possible to reduce the hardware requirements of the original technique through a parametric model: in this way coefficient statistics storage is not needed and the whole rate control algorithm becomes more suitable for low-resource devices.

I. INTRODUCTION

The H.264 project is born from a joined effort between two leading standard bodies, ITU-T and ISO/IEC, in order to provide an enhanced video coding standard. A joint committee, the Joint Video Team (JVT), has carried on H.264 definition since February 2002. The new standard has inherited all the features of previous video coding algorithm H.26L, developed by ITU-T Study Group 16, and has been integrated into MPEG-4 draft (Recommendation 10).

H.264 is mainly intended for video transmission in all areas where bandwidth or storage capacity is limited (e.g., video telephony or video conferencing over mobile channels and devices), supplying an enhanced coding efficiency and an improved network adaptation [4]. Experimental results show that H.264’s coding performances overcome MPEG-4 at low bit rates [5]. Since many target applications concern video transmission over time-varying bandwidth channels, we need to control bit rate with scalable algorithms that allow modifying coding parameters according to channel’s variations. Our aim is focused on developing a flexible buffer control that maximizes video quality keeping the number of bits under transmission constraints.

In the articles [8] and [9], an efficient model for the bit rate produced by a transform video coder is presented and in [9] it is shown how it can be used in different coding standards. As there is a low occurrence of bit allocation errors, rate control does not need to compensate coding errors and can adapt more quickly to varying bandwidth conditions.

This article provides a description of its implementation on emerging video coding standard H.264, reducing its hardware requirements and making it more suitable for low-resource devices.

In section II “zeros”-parametrization is described. Image’s rate can be linearly related to the percentage of null quantized transform coefficients (called $\rho$ as in [8]). In section III, a parametric model is used to avoid coefficient histogram storage. A rate control algorithm based on $\rho$-modeling is described in section IV; the quantization step is modified while coding different macroblocks according to a target percentage of “zeros”, calculated according to channel capacity. The algorithm switches between quantization step and $\rho$ domains through a coefficients histogram or a distribution function (pdf), estimated from previously encoded data.

Finally, some experimental results are compared with another rate control algorithm developed by Joint Video Team in section V. It is possible to notice that $\rho$-modeling provides better performances both in terms of quality and of rate constraints.

II. RATE DISTORTION MODELING BASED ON “ZEROS”

Most of rate control algorithms are based on hyperbolic R-D models, where bit rate and distortion are functions of quantization step (e.g. [1]). This modeling has been adopted in many control techniques as it provides simple analysis and description of images, allowing a direct control over quantization. This model suits image characteristics locally but in some cases it can be inefficient. If there is a low spatial correlation (e.g. the encoder is processing pictures with varying characteristics) the R-D model estimated from a portion of an image does not suit other regions.

A better solution can be found through rate-distortion modeling suggested by S.K. Mitra et al. in [8]–[10]. In such papers, rate and distortion related to lossy coding of images are functions of the percentage of null quantized coefficients in each block (called “zeros”). Experimental results [9] show that the “zeros” parametrization suits a great number of images more sharply than previous models. This method has been successfully implemented on different transform-based coding standards and it has been proved to be suitable both for local and frame-level analysis. As a matter of fact, its application to the emerging video coding standard H.264 is an interesting topic of investigation.

H.264 encoder implements a hybrid transform video coder with motion compensation and spatial prediction. Each input image is predicted according to either spatial or temporal correlation, estimating its pixels either from previous pictures or from neighboring pixels. Then prediction error is transformed through a multiplication-free transform derived
from a Discrete Cosine Transform applied on 4x4 blocks. The obtained coefficients are quantized and reorganized in couples of values according to a run-length coding. Then runs and levels are converted into binary strings of variable length.

After the transform operation, we can store the frequency of each coefficient in a histogram, which allows us computing the percentage of “zeros” in each frame \((\rho)\) according to

\[
\rho(\Delta) = \sum_{|a|<\Delta} p_x(a),
\]

(1)

where \(\Delta\) is the quantization step chosen for that picture and \(p_x(a)\) is the probability distribution of transform coefficients \(x\). In H.264 standard, \(\Delta\) values are fixed and identified by a “quantization parameter” (named QP) that assumes integer values in the range \([0,51]\). As can be seen from Figure 1, the picture rate can be well represented as a linear function of \(\rho\) through the equation

\[
R(\rho) = \lambda \cdot \rho + \gamma,
\]

(2)

that can be further approximated by assuming that if all coefficients are null no bits are sent. In this way we can express the bit rate by the linear equation

\[
R(\rho) = \lambda \cdot (1 - \rho).
\]

(3)

Performing the same analysis on different kinds of pictures, we obtained the same behavior, independently of the nature of the prediction (spatial, temporal or temporal bi-directional). However, coefficient statistics can differ for pictures of different type and prediction error standard deviation can also vary. We decided to keep coefficients statistics separated, according to the type of the encoded frame.

III. PARAMETRIC MODELS FOR H.264 COEFFICIENTS ESTIMATED THROUGH ACTIVITY

In order to correlate the “zeros” percentage with quantization step, the equation (1) needs to know the coefficients distribution, which can be provided either by the storage of coefficients frequencies or by a parametric model. The second solution is analyzed in this section, as it turns out to be cheaper in terms of memory requirements and computational effort.

At first we considered a generalized gaussian model [9], described by the density function (pdf)

\[
p_x(a) = \gamma \cdot e^{-\beta |a|^\alpha}; \quad (4)
\]

then we adopted a “laplacian+impulsive” distribution, described by the equation

\[
p_x(a) = \alpha' \cdot \delta(a) + \beta' \cdot e^{-\frac{\gamma}{2} |a|}, \quad (5)
\]

where \(\delta(a)\) is the Dirac impulse.

The first model provides a good estimate of coefficients statistics for I and P frames, while the second one is best-suited for B frames but can be used for the other types too (see Figure 2).

Equation (4) is defined by \(\beta\) and \(\alpha\) (since the third variable \(\gamma\) is found through normalization) where \(\alpha\) can be estimated from \(E[|x|]\) and \(\sigma_x^2 = E[(x-m_x)^2]\) according to the equation

\[
\alpha = F^{-1} \left( \frac{E[|x|]}{\sigma_x} \right), \quad (6)
\]

\(F(\cdot)\) is a monotonic increasing function defined as

\[
F(\alpha) = \frac{\Gamma \left( \frac{\alpha}{2} \right)}{\sqrt{\Gamma \left( \frac{1}{\alpha} \right) \cdot \Gamma \left( \frac{3}{2} \right)}} \quad (7)
\]

and in our implementation it was mapped in a memory table (see [11]).

For what concerns equation (5), the whole pdf is identified by \(\alpha'\) and \(\gamma'\).

The values of parameters \(\alpha, \beta\) in (4) and of parameters \(\alpha', \gamma'\) in (5) can be expressed as functions of the average activity \(\bar{a}ct\), defined as

\[
\bar{a}ct = \frac{1}{N_{MB}} \cdot \sum_{m=0}^{N_{MB}} a_{ct}(m) \quad (8)
\]

where

\[
a_{ct}(m) = \sum_{x,y=0}^{16} \mid I_m(x,y) - \hat{I}_m(x,y) \mid, \quad (9)
\]

with \(I_m(x,y)\) the original pixel of \(m\)-th macroblock at position \((x,y)\), \(\hat{I}_m(x,y)\) its prediction and \(N_{MB}\) the number of macroblocks of the frame.

The relations \(\alpha(\bar{a}ct), \beta(\bar{a}ct), \gamma'(\bar{a}ct)\) can be well-approximated through second order polynomials

\[
\alpha(\bar{a}ct) = \alpha_0(\rho) + \alpha_1(\rho) \cdot \bar{a}ct + \alpha_2(\rho) \cdot (\bar{a}ct)^2, \quad (10)
\]

\[
\beta(\bar{a}ct) = \beta_0(\rho) + \beta_1(\rho) \cdot \bar{a}ct + \beta_2(\rho) \cdot (\bar{a}ct)^2, \quad (11)
\]

\[
\gamma'(\bar{a}ct) = \gamma'_0(\rho) + \gamma'_1(\rho) \cdot \bar{a}ct + \gamma'_2(\rho) \cdot (\bar{a}ct)^2, \quad (12)
\]

while \(\alpha'(\bar{a}ct)\) can be found through the equation
The correction is performed using the formula

\[ G_{k,0} = \delta G_{k-1} + \bar{G} \]  \hspace{1cm} (13)

where \( G_{k,n} \) are the available bits before coding the \( n \)-th frame of the \( k \)-th GOP and \( \bar{G} \) has been defined in eq. (12). \( \delta G_{k-1} \) is the difference between target and effective bit usage after the coding of \( (k-1) \)-th GOP.

At picture level, before the coding of \( n \)-th frame of the current GOP, the algorithm computes a target bit rate \( T_n \) from expression

\[ T_n = K'_i \cdot \frac{G_k}{n + K_i \cdot P \cdot P + K_i \cdot P \cdot P_B + n_B} \]  \hspace{1cm} (14)

in which \( n_k \) is the number of remaining frames of \( i \) type in the GOP and \( K_{i,j} \) is the target bit ratio between an \( i \)-type coded frame and a \( j \)-type one (\( i, j = I, P, B \)). The constants \( K'_i \), \( i = I, P, B \), depends on these ratios and corresponds to

\[ K'_I = K_I \cdot P \cdot P_B \quad K'_P = K_P \cdot P_B \quad K'_B = 1. \]  \hspace{1cm} (15)

Then, the available number of bits is updated

\[ G_{k,n+1} = G_{k,n} - S_n \]  \hspace{1cm} (16)

where \( S_n \) is the actual number of bits used to code the \( n \)-th picture.

The target bit rate for the following picture is modified, compensating previous coding errors. As H.264 encoder is
driven by the quantization parameter QP, we must calculate the average QP value for \(n\)-th frame from \(T_n\). According to the R-D model presented in the previous section, we can associate to bit rate \(T_n\) an average percentage of “zeros” equal to

\[
\rho^T_n = 1 - \frac{T_n}{\lambda}
\]

(17)

where \(\lambda\) can be estimated from previously coded pictures (e.g. the \((n - 1)\)-th frame). Through the distribution \(p_x(a)\) of the previous picture, we can transform percentage \(\rho^T_n\) in \(Q_P^T_n\), which represents the estimated average quantization factor for current frame. The \(Q_P^T_n\) is clipped according to the law

\[
Q_P^T_n = \begin{cases} 
Q_P_{n-1} + 3 & \text{if } Q_P^T_n > Q_P_{n-1} + 3 \\
Q_P_{n-1} - 3 & \text{if } Q_P^T_n < Q_P_{n-1} - 3 \\
Q_P^T_n \text{ not clipped} & \text{otherwise.}
\end{cases}
\]

(18)

In this way it is possible to avoid a different quality between different picture zones, which would produce unpleasant effects.

At macroblock level the quantization parameter must be corrected. This grants a good control both over picture quality and coded bits, keeping bit rate under given constraints and smoothing coding distortion across the different macroblocks. After coding \(m\)-th MB of \(n\)-th frame, the percentage of null quantized coefficients in the previous coded \(m\) macroblocks is \(\rho^P_m\) and the number of bits used to code the picture is \(B^P_m\). According to the given target, \(B^H_n = T_n - B^P_m\) bits are left to code the remaining macroblocks: the percentage of “zeros” required to fit the constraints is equal to

\[
\rho^R_m = 1 - \frac{B^R}{\lambda} \cdot \frac{N_{MB}}{N_{MB} - m}
\]

(19)

where \(N_{MB}\) is the total number of macroblocks in each frame.

This leads to estimate the ratio \(k = \frac{\rho^R_m}{\rho^P_m}\), which affects the quantization parameter \(Q_P^T_{m+1}\) of the following macroblock according to the equation

\[
Q_P^T_{m+1} = \begin{cases} 
Q_P^T_m + 3 & \text{if } 1 + 3 \cdot \delta \leq k < +\infty \\
Q_P^T_m + 2 & \text{if } 1 + 2 \cdot \delta \leq k < 1 + 3 \cdot \delta \\
Q_P^T_m + 1 & \text{if } 1 + \delta \leq k < 1 + 2 \cdot \delta \\
Q_P^T_m & \text{if } 1 - \delta \leq k < 1 + \delta \\
Q_P^T_m - 1 & \text{if } 1 - 2 \cdot \delta \leq k < 1 - \delta \\
Q_P^T_m - 2 & \text{if } 1 - 3 \cdot \delta \leq k < 1 - 2 \cdot \delta \\
Q_P^T_m - 3 & \text{if } -\infty \leq k < 1 - 3 \cdot \delta.
\end{cases}
\]

(20)

with \(\delta\) specified in the following paragraph.

In [8] a linear law was used in order to correct \(Q_P^T_n\) value since in the H.263 encoder the relation between \(Q_P\) and quantization step \(\Delta\) can be expressed as

\[
\Delta_{H.263} = 2 \cdot Q_P
\]

As in the H.264 encoder this relation is ruled by the exponential formula

\[
\Delta = 0.67 \cdot 2^{Q_P} \quad 0 \leq Q_P \leq 51, Q_P \in \mathbb{Z},
\]

(21)

\(\delta_k\) can be estimated by

\[
\delta_k = 0.67 \cdot 2^{Q_P} / 9.
\]

(22)

In order to fit the targeted bit rates, the constant \(C\) has been set to 3000. A reduced value of \(\delta_k\) allows the encoder to react more quickly to the changes in \(\rho^P_m\).

Note that \(\delta\) is monotonically increasing with the quantization parameter \(Q_P\) as, according to equation (21), the variation of \(\Delta\) corresponding to a variation of \(Q_P\) is more relevant for higher values of the quantization step. This can bring to strong variations in bit rate and coding quality across different macroblocks. That is why it is necessary to avoid frequent variations in the choice of the quantization step, using a larger value for \(\delta\).

In order to achieve a sufficient statistic for \(\rho^P_m\), \(Q_P^T_n\) remains equal to \(Q_P^T_{m+1}\) until \(B^P_m \geq 0.1 \cdot T_n\), then the rate control at macroblock level is applied.

Moreover the use of a parametric model reduces the required memory area and accesses, since one “reading” and one “writing” operation per coefficient are required in order to estimate the frequency histogram.

It is worth noting that many parameters belonging yet to H.264 syntax (i.e. the number of quantized coefficients different from 0 and \(\alpha\) of \(m\)) are employed to control the bit rate. Therefore a few extra operations are required at macroblock level, avoiding the pre-analysis and recoding performed in other rate control algorithms (see [7]). Computational complexity is greatly reduced since no pre-coding is needed and all the analysis is performed approximately with the number of operation reported in Table 1 (where \(N_f\) is the number of coded frames and \(N_{MB}\) the number of macroblock per frames). The average values are estimated assuming that GOP is made of 60 frames with structure IPPB.

In addition it must be stressed that computational requirements remain the same independently from the type of coding adopted for each frame (I,P or B).

V. EXPERIMENTAL RESULTS

In order to evaluate the “zeros” algorithm performances, we coded different sequences with different rate control algorithms at various bit rates. In this case the reference is the algorithm described in [7] and implemented in the Joint Model 42 of H.264 by the Joint Video Team (it will be denoted JVT-E069 from the signature of document where it was presented). The bit rate is produced adopting the CAVLC lossless coder (with R-D optimization disabled).

The bit rate and PSNR vs. frame number are reported in Figures 3, 4 for sequence foreman and in Figures 5, 6 for sequence table. It must be pointed out that the sequence table is characterized by a scene change at frame no. 130. The algorithm based on “zeros” allocates more bits at this
TABLE I

<table>
<thead>
<tr>
<th>Operation</th>
<th>No. operation (I frames)</th>
<th>No. operation (P frames)</th>
<th>No. operation (B frames)</th>
<th>Average no. operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum</td>
<td>(38 \cdot N_f + 40 \cdot N_f \cdot N_{MB})</td>
<td>(38 \cdot N_f + 40 \cdot N_f \cdot N_{MB})</td>
<td>(37 \cdot N_f + 40 \cdot N_f \cdot N_{MB})</td>
<td>(37 \cdot N_f + 40 \cdot N_f \cdot N_{MB})</td>
</tr>
<tr>
<td>mult</td>
<td>(26 \cdot N_f + 4 \cdot N_f \cdot N_{MB})</td>
<td>(23 \cdot N_f + 4 \cdot N_f \cdot N_{MB})</td>
<td>(22 \cdot N_f + 4 \cdot N_f \cdot N_{MB})</td>
<td>(22 \cdot N_f + 4 \cdot N_f \cdot N_{MB})</td>
</tr>
<tr>
<td>pow(_2)</td>
<td>(N_f)</td>
<td>(4 \cdot N_f)</td>
<td>(4 \cdot N_f)</td>
<td>(4 \cdot N_f)</td>
</tr>
<tr>
<td>log(_2)</td>
<td>(3 \cdot N_f)</td>
<td>(N_f)</td>
<td>(N_f)</td>
<td>(N_f)</td>
</tr>
</tbody>
</table>

point, thus avoiding the great quality degradation provided by JVT-E069 and smoothing the PSNR.

Fig. 3. Bits per frame for the sequence foreman (360 frames coded with GOP IBBP 60 frames at 30 frame/s) with target bit rate 64 kbit/s.

Fig. 5. PSNR for the sequence table (360 frames coded with GOP IBBP 60 frames at 30 frame/s) with target bit rate 64 kbit/s.

Fig. 4. PSNR for the sequence foreman (360 frames coded with GOP IBBP 60 frames at 30 frame/s) with target bit rate 64 kbit/s.

Fig. 6. PSNR for the sequence table (360 frames coded with GOP IBBP 60 frames at 30 frame/s) with target bit rate 64 kbit/s.

TABLE II

<table>
<thead>
<tr>
<th>Rate (kbit/s)</th>
<th>24</th>
<th>32</th>
<th>48</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\varepsilon_{err}) (%)</td>
<td>0.46</td>
<td>0.02</td>
<td>-0.00</td>
</tr>
<tr>
<td>(PSNR_{ref}) (dB)</td>
<td>28.08</td>
<td>29.42</td>
<td>30.29</td>
</tr>
<tr>
<td>(s_{PSNR}) (dB)</td>
<td>2.13</td>
<td>2.11</td>
<td>2.14</td>
</tr>
<tr>
<td>(\varepsilon_{err,PT}) (%)</td>
<td>-0.62</td>
<td>-0.57</td>
<td>-0.52</td>
</tr>
<tr>
<td>(PSNR_{PT}) (dB)</td>
<td>27.27</td>
<td>28.63</td>
<td>29.66</td>
</tr>
<tr>
<td>(s_{PSNR_{PT}}) (dB)</td>
<td>2.85</td>
<td>2.90</td>
<td>2.90</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS

According to experimental results shown in the paper, the “zeros” model provides a good parametrization of coded...
pictures and its application to control transmission buffer turns out to be a performing solution, both in terms of visual quality and in terms of reduced complexity. Moreover it does not need a pre-analysis, providing a lower coding delay than other solutions (see [7]).

ACKNOWLEDGMENT

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

REFERENCES


