16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

NO-REFERENCE PSNR ESTIMATION ALGORITHM FOR H.264 ENCODED VIDEO SEQUENCES

Tomás Brandão[1,3] *and Maria Paula Queluz*[2,3]

[1] Department of Sciences and Information Technologies, ISCTE - University Institute of Lisbon, Portugal
[2] Department of Electrical and Computer Engineering, IST - Technical University of Lisbon, Portugal
[3] IT-Lisbon, Torre Norte, Piso 10, Av. Rovisco Pais, 1, 1049-001 Lisbon, Portugal
phone/fax: +(351) 218 418 454 / 218 418 472; e-mails: {tomas.brandao, paula.queluz}@lx.it.pt

ABSTRACT

This paper proposes a no-reference PSNR estimation algorithm for video sequences subject to H.264/AVC encoding. The proposed method explores statistical properties of the transformed coefficients, which can be modeled by a Cauchy or Laplace probability density function. The distribution's parameters are computed from quantized coefficient data received at the decoder, combining maximum-likelihood with linear prediction estimates. Since the proposed algorithm has no knowledge about the original sequences, it can be used as a no-reference metric for evaluating the quality of the encoded video sequences. When compared with recent state-of-the-art algorithms proposed for the same purpose, it has shown better PSNR estimation accuracy in a set of video sequences subject to different encoding rates.

1. INTRODUCTION

Due to the increasing transmission of digital video contents over broadband and wireless networks, quality monitoring of multimedia data is becoming an important matter. From a quality of experience perspective, it is desirable to evaluate content quality at the receivers. Such a system would have to deal with different sources of distortion, namely lossy encoding of media data and data transmission errors. Moreover, and since the original signals are usually not available at the receiver, quality scores should be provided without the knowledge of the original signals - *no-reference* metrics - or using very limited information about them, transmitted through a side channel - *reduced-reference* metrics.

This paper proposes an algorithm that estimates errors due to lossy compression in block-wise DCT (discrete cosine transform) based video encoding schemes. Although the proposed method is oriented to H.264/AVC [1] encoded streams, it can be easily adapted to other DCT-based video encoding schemes.

It is assumed that statistics of the transformed coefficient data can be modeled by a parametric distribution. The distribution parameters are estimated from the quantized coefficient values, which are available at the receiver, combining the maximum-likelihood (ML) parameter estimation method with a linear prediction scheme, as suggested in [2] for still images and in [3] for MPEG-2 encoded video (I frames only). The method proposed in this paper can be seen as an extension of those methods to H.264. The main innovations are the additional use of the Cauchy probability density function as a coefficient distribution model, and the ability to estimate the PSNR in inter modes (P and B frames). The final result is a PSNR estimate that is computed without the need of the original data, thus resembling a no-reference quality metric.

The performance of the proposed algorithm has been evaluated using different video sequences, subject to different encoding rates. The resulting PSNR estimates have shown greater accuracy than the ones provided by the state-of-the-art algorithms [4] and [5], which have been proposed with the same objective.

The work presented in this paper is sponsored by *FCT - financiamento plurianual*.

The paper is organized as follows: after the introduction, section 2 gives a brief overview of the H.264 standard; section 3 shows how to compute a no-reference PSNR estimate from the coefficient's distribution; section 4 is focused on the estimation of the coefficient's distribution parameters from the quantized data; experimental results are depicted in section 5, and finally, remarks and future directions are given in section 6.

2. BRIEF OVERVIEW OF THE H.264 STANDARD

2.1 Encoder and decoder schemes

A typical H.264 encoder is partially represented in figure 1(a). An input frame $\mathbf{F_{in}}$ subject to encoding is divided in 16×16 block-wise units called *macroblocks* (MBs). Each macroblock can be encoded in *intra* or *inter* mode. In *intra* mode, a prediction block $\mathbf{P}$ is computed from samples taken from the current frame, that have been previously encoded, decoded and reconstructed. In *inter* mode, $\mathbf{P}$ is computed by motion-compensated prediction from reference frame(s) $\mathbf{F_{ref}}$. The difference between $\mathbf{P}$ and the original MB pixel values $-\mathbf{D}-$ is transformed (using a block-wise transform) and quantized, resulting in a set of quantized transform coefficients $\mathbf{X}$, and the corresponding quantization indexes. These indexes are then re-ordered and entropy encoded for transmission.

As for the H.264 decoder, partially represented in figure 1(b), it receives a compressed bitstream, whose elements are entropy decoded and reordered in order to produce a set of quantized coefficient data $\mathbf{X}$. The quantized coefficients are then rescaled and inverse transformed, resulting in a residual signal $\mathbf{D'}$, which is added to the current prediction signal, $\mathbf{P}$. The decoded frame results from the sum of $\mathbf{P}$ and $\mathbf{D'}$ for all MBs.

2.2 Transform and quantization

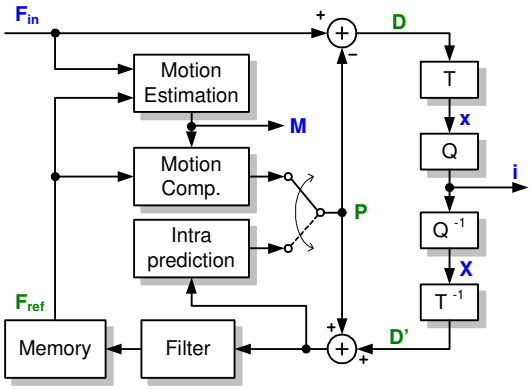
The transform operation used in H.264 is an integer approximation of the classical block-wise DCT (used in previous standards, such as JPEG and MPEG-2). The main transform block size in H.264 is 4×4, although the use of 8×8 transform is also possible in higher profiles. For simplicity, only the 4×4 size has been considered in this paper. Let $\mathbf{D}$ represent the differences between the original and the predicted image values in a 4×4 block. The transformed coefficient values $-\mathbf{x}-$ can be computed as:

$$\mathbf{x} = \mathbf{T}\mathbf{D}\mathbf{T}^{\mathrm{t}} \odot \mathbf{S}, \tag{1}$$

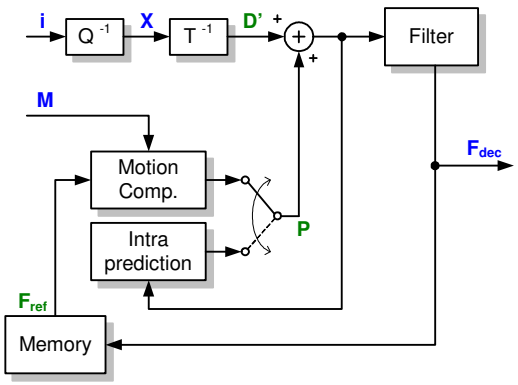
where $\odot$ represents point-by-point multiplication, $\mathbf{T}$ is the transform matrix and $\mathbf{S}$ is a post-scaling matrix, which are defined as [6]:

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}; \quad \mathbf{S} = \begin{bmatrix} \frac{1}{4} & \frac{1}{\sqrt{5}} & \frac{1}{4} & \frac{1}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} & \frac{1}{10} & \frac{1}{\sqrt{5}} & \frac{1}{10} \\ \frac{1}{4} & \frac{1}{\sqrt{5}} & \frac{1}{4} & \frac{1}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} & \frac{1}{10} & \frac{1}{\sqrt{5}} & \frac{1}{10} \end{bmatrix}.$$

In commercial encoders, the transform operation is implemented using integer arithmetic only (add and shift operations). As for the



(a) Encoder.



(b) Decoder.

Figure 1: H.264 generic encoder and decoder schemes.

$mod(QP,6)$	q_B
0	0.6250
1	0.6875
2	0.8125
3	0.8750
4	1.0000
5	1.1250

Table 1: Base quantization steps.

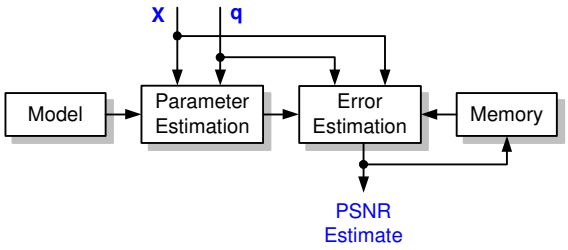


Figure 2: PSNR estimation scheme.

post-scaling operation, the reference software [7] implements it together with the quantization, using only integer operations.

The value of the quantized coefficient X_k is given by:

$$X_k = \underbrace{sign(x_k) \times \left\lfloor \frac{|x_k|}{q_k} + 1 - \alpha \right\rfloor}_{i_k} \times q_k, \tag{2}$$

where q_k is the quantization step, α is a parameter that controls the width of the dead zone around 0 and i_k represents the quantization index that is actually transmitted. In the reference software [7], $\alpha \approx 2/3$ for intra blocks and $\alpha \approx 5/6$ for inter blocks. The quantization step, q_k, can be derived from a H.264 parameter called QP, which may differ from macroblock to macroblock. The general rule to compute q_k from QP is:

$$q_k = q_B\big(mod(QP,6)\big)2^{\lfloor QP/6 \rfloor}, \tag{3}$$

where q_B is a base quantization step (see table 1) and $mod(m,n)$ is the remainder of integer division of m by n.

3. PSNR ESTIMATION

Assuming pixel values in the range of $[0 \ldots 255]$, the image PSNR is usually given by:

$$\text{PSNR}_{[\text{dB}]} = 10\log_{10}\frac{255^2}{MSE}; \qquad \text{MSE} = \frac{1}{M}\sum_{k=1}^{M}\varepsilon_k^2, \tag{4}$$

where M is the number of pixels, ε_k^2 is the squared difference between the k-th reference and distorted pixels and MSE is the mean square error. In accordance with Parceval's theorem, it is indifferent to measure the PSNR in the pixel or in the DCT domain. Thus, for the remainder of this paper, M will be the number of DCT coefficients under analysis and $\varepsilon_k^2 = (X_k - x_k)^2$ will be the squared difference between original, x_k, and quantized, X_k, coefficients.

If the original coefficient data distribution is known, it is possible to estimate the local mean square error, $\hat{\varepsilon}_k^2$, at the k-th coefficient, by observing the value of its quantized value, X_k:

$$\hat{\varepsilon}_k^2 = \int_{-\infty}^{+\infty} f_X(x|X_k)(X_k - x)^2 dx, \tag{5}$$

where $f_X(x|X_k)$ represents the distribution of the original coefficients values conditioned to the observed value of X_k. Using *Bayes rule* and considering that $P(X_k|x) = 1$ if x is in the quantization interval around X_k, $[a_k; b_k]$, and $P(X_k|x) = 0$, otherwise, (5) can be rewritten as:

$$\hat{\varepsilon}_k^2 = \int_{a_k}^{b_k} \frac{f_X(x)}{P(X_k)}(X_k - x)^2 dx = \frac{\int_{a_k}^{b_k} f_X(x)(X_k - x)^2 dx}{\int_{a_k}^{b_k} f_X(x) dx}, \tag{6}$$

where $f_X(x)$ is the original coefficient data distribution and the limits a_k and b_k are defined as:

$$\begin{cases} a_k = -\alpha q_k \\ b_k = \alpha q_k, \end{cases} \quad \text{if } X_k = 0; \qquad \begin{cases} a_k = |X_k| - (1-\alpha)q_k \\ b_k = |X_k| + \alpha q_k, \end{cases} \quad \text{if } X_k \neq 0. \tag{7}$$

From (6), it can be concluded that the squared error estimate depends on the value of the quantized coefficient X_k, on the quantization step q_k (which determines a_k and b_k) and on the coefficient distribution $f_X(x)$. X_k and q_k can be derived from the encoded bitstream, while $f_X(x)$ needs to be estimated from the available quantized data.

A no-reference PSNR estimation method can then be implemented using the scheme depicted in figure 2. The received coefficient values and the corresponding quantization steps are used for

estimating the parameter(s) of $f_X(x)$. Once $f_X(x)$ is derived, the local squared error is estimated and therefore a no-reference PSNR value can be computed. A memory block is also included, to generalize the scheme for situations where the error currently under estimation is dependent on past estimates.

4. COEFFICIENT DISTRIBUTION MODELING

The most common models found in literature for the statistical distribution of H.264 coefficient data are Cauchy [8] and Laplace [5] probability density functions. Both models rely in one parameter that needs to be estimated. This section presents a methodology to estimate the distribution's parameter, using quantized coefficient data, that as been derived for both models.

4.1 Cauchy model

The *Cauchy* probability density function (PDF) can be described by:

$$f_X(x) = \frac{1}{\pi}\frac{\beta}{\beta^2+x^2}, \tag{8}$$

where β is the distribution's parameter and x represents the coefficient's value at a given frequency position. If the original coefficient values were known, an estimate for parameter β could be computed using the *maximum-likelihood* (ML) method [9]:

$$\beta_{ML} = \arg\max_{\beta}\left\{\log\prod_{k=1}^{N} f_X(x_k)\right\}, \tag{9}$$

where the x_k is the k-th coefficient value and N is the number of coefficients. Using (8) in (9) leads to

$$\beta_{ML} = \arg\max_{\beta}\left\{\sum_{k=1}^{N}\left(\log\beta - \log(\beta^2+x_k^2)\right)\right\}. \tag{10}$$

The value of β that maximizes (10) can be computed by finding the zeros of the derivative with respect to β, which corresponds to

$$\frac{N}{\beta} - 2\sum_{k=1}^{N}\frac{\beta}{\beta^2+x_k^2} = 0. \tag{11}$$

To solve (11), *Newton-Raphson*'s root finding method was used, starting with a small value (0.1) as the initial value of β. Convergence has been achieved in all experiments. The resulting value for β_{ML} can be seen as a reference value, thus it will be addressed to as the "original" parameter value.

Now, let's suppose that only quantized data is available for estimating β, which is the case at the receiver (decoder) side. The ML method can still be used:

$$\hat{\beta}_{ML} = \arg\max_{\beta}\left\{\log\prod_{k=1}^{N} P(X_k)\right\}, \tag{12}$$

where $P(X_k)$ represents the probability of having value X_k at the quantizer's output. Assuming that the quantizer is linear with step size q_k, which may differ from block to block, and includes a dead zone around 0, controlled by parameter α, $P(X_k)$ can be written as:

$$\begin{aligned} P(X_k) &= \int_{a_k}^{b_k}\frac{1}{\pi}\frac{\beta}{\beta^2+x^2}dx \\ &= \begin{cases} \frac{2}{\pi}\tan^{-1}\left(\frac{\alpha q_k}{\beta}\right), & \text{if } X_k = 0; \\ \frac{1}{\pi}\left(\tan^{-1}\left(\frac{b_k}{\beta}\right) - \tan^{-1}\left(\frac{a_k}{\beta}\right)\right), & \text{otherwise.} \end{cases} \end{aligned} \tag{13}$$

Using (13) in (12) leads to:

$$\begin{aligned} \hat{\beta}_{ML} = \arg\max_{\beta}\Bigg\{ &\sum_{k_0=1}^{N_0}\log\left(\frac{2}{\pi}\tan^{-1}\left(\frac{\alpha q_{k_0}}{\beta}\right)\right) + \\ &\sum_{k_1=1}^{N_1}\log\frac{1}{\pi}\left(\tan^{-1}\left(\frac{b_{k_1}}{\beta}\right) - \tan^{-1}\left(\frac{a_{k_1}}{\beta}\right)\right)\Bigg\} \end{aligned} \tag{14}$$

The two summation terms in (14) correspond to the two possible cases in (13). In practice, the set of quantized coefficients X_k has been split according to those cases: quantized coefficients with zero and non-zero values, respectively. Accordingly, N_0 and N_1 represent the number of coefficients (at a given frequency), that fall in those cases. The value of β that maximizes (14) can be obtained by finding the zero of the derivative with respect to β, which corresponds to:

$$\begin{aligned} &\sum_{k_1=1}^{N_1}\frac{\frac{a_{k_1}}{\beta^2+a_{k_1}^2} - \frac{b_{k_1}}{\beta^2+b_{k_1}^2}}{\tan^{-1}\left(\frac{b_{k_1}}{\beta}\right) - \tan^{-1}\left(\frac{a_{k_1}}{\beta}\right)} - \\ &\sum_{k_0=1}^{N_0}\frac{\alpha q_{k_0}}{\tan^{-1}\left(\frac{\alpha q_{k_0}}{\beta}\right)\left((\alpha q_{k_0})^2+\beta^2\right)} = 0. \end{aligned} \tag{15}$$

If $N_0 < N$, a solution for (15) can be found numerically, using the same method as in (11). If $N_0 = N$, $\beta \to 0$, meaning that the estimated coefficient distribution is a *Dirac's delta* function centered in 0. In other words, the ML method will fail if all coefficients to zero at a given frequency are quantized to 0.

4.2 Laplace model

The coefficient's distribution for the Laplace model is described by:

$$f_X(x) = \frac{\lambda}{2}\exp(-\lambda|x|), \tag{16}$$

where λ is the distribution's parameter and x is the coefficient value.

Following a procedure similar to what has been done in subsection 4.1, an ML estimation for λ using the original coefficient data, is given by:

$$\lambda_{ML} = \frac{N}{\sum_{k=1}^{N}|x_k|}, \tag{17}$$

where N represents the number of coefficients at the given frequency and x_k is the coefficient's value.

Assuming that only quantized data is available, λ can be computed using the ML method in the same way as in (12). In this case, the probability $P(X_k)$ can be written as:

$$P(X_k) = \begin{cases} 1 - e^{-\alpha\lambda q_k}, & \text{if } X_k = 0; \\ \frac{1}{2}e^{-\lambda b_k}(e^{\lambda q_k} - 1), & \text{otherwise.} \end{cases} \tag{18}$$

Using (18) in (12), and looking for the zeros of the derivative with respect to λ will lead to:

$$\sum_{k_0=1}^{N_0}\frac{\alpha q_{k_0}}{e^{\alpha\lambda q_{k_0}} - 1} + \sum_{k_1=1}^{N_1}\left(\frac{q_{k_1}}{e^{\lambda q_{k_1}} - 1} - b_{k_1}\right) = 0, \tag{19}$$

a result that is similar to what has been derived in [3], for MPEG-2 intra frames. Once again, the solution can be found by using an iterative root finding algorithm. If $N = N_0$, $\lambda \to +\infty$, leading to a phenomena that is equivalent to the described at the end of the previous subsection, *i.e.*, the estimated distribution is a *Dirac's delta* function.

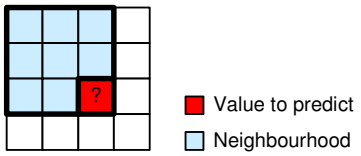


Figure 3: Neighbourhood configuration.

Frame type	Model	
	Laplace	*Cauchy*
I	0.647	**0.402**
P	**0.450**	0.618
B	**0.507**	0.522

Table 2: Mean PSNR estimation error (dB).

4.3 Improving parameter estimation using prediction

The problem of estimating the transform coefficients distribution's parameter when all coefficients have been quantized to zero can be tackled by exploring the correlation between parameter values at neighbouring DCT frequencies. For instance, when using the Cauchy model in the I frames of the H.264 encoded versions of the test sequences displayed in figure 4, the correlation between parameter values in a 4-connected neighbourhood is 0.93. Using matrix notation, a prediction value for the distribution's parameter, $\hat{p}_{pred}$ (which can be the Cauchy's "β" or the Laplace's "λ") at a given frequency can be computed as:

$$\hat{p}_{pred} = \mathbf{v}^{\mathrm{t}}\mathbf{w}, \tag{20}$$

with

$$\mathbf{v} = \begin{bmatrix} 1 \\ p_{\upsilon_1} \\ \vdots \\ p_{\upsilon_K} \end{bmatrix} \text{ and } \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_K \end{bmatrix},$$

where $\mathbf{v}$ is a vector whose elements are parameter values taken from the neighbourhood, K is the neighbourhood size and $\mathbf{w}$ is the linear weight vector. $\mathbf{w}$ can be computed using least squares, minimizing the square error between the "original" parameter values and $\hat{p}_{pred}$, in a randomly chosen training set.

The neighbourhood configuration used in the experiments is illustrated in figure 3. Since low-frequency coefficients are less vulnerable to the effects of lossy compression, its structure has been chosen with the purpose of recursively predict parameter values, starting from those frequency positions.

The linear prediction value, $\hat{p}_{pred}$, that results from (20) is then combined with $\hat{p}_{ML}$ according to

$$\hat{p}_f = r_0\hat{p}_{pred} + (1-r_0)\hat{p}_{ML}, \tag{21}$$

where $r_0 = N_0/N$ represents the rate of coefficients quantized to zero and $\hat{p}_f$ is the final estimate for the distribution's parameter. The idea is as follows: since ML estimates become more inaccurate as the rate of coefficients quantized to zero increases, more trust is given to the predictor in this case; on the other hand, if the number of coefficients quantized to zero is low, the ML estimator will most likely get accurate results, so there is no need for the predicted value.

5. RESULTS

5.1 Experimental setup

The video sequences depicted in figure 4 have been used in the experiments. All sequences are CIF format (352×288), with a frame rate of 25Hz, and have been encoded at 256, 512, and 1024 kbit/s, using the reference H.264 software [7]. A GOP-12 structure *IBBPBBP...* has been used in all encoding runs and only the 4×4 transform size was allowed. The low complexity rate-distortion optimization algorithm provided on the software has been selected for faster encoding.

According to the type of the frame subject to PSNR estimation (I, P or B), different coefficient distribution models have been used. The model that has been selected for each frame type was the one leading to the highest PSNR estimation accuracy. To evaluate this, the PSNR has been estimated using equations (4) and (5) and distribution parameters have been estimated using the original coefficient data. The resulting mean PSNR estimation error is depicted in table 2. Based on these results, the Cauchy model was selected for the I frames, while the Laplace model was selected for P and B frames.

Regardless of the selected model, parameter estimation has been improved by adding prediction to the ML estimates, as described in 4.3. The predictors have been trained with one half of the available samples and separate training procedures have been followed according to the frame type. The use of prediction lead to an improvement of 43%, 22% and 27% on the no-reference PSNR estimation error for I, P, and B frames, respectively.

The effect of *skipped* macroblocks that occur in P and B frames has also been considered using the following error compensation procedure:

$$\mathrm{MSE}_{est} = r_s\mathrm{MSE}_{ref} + (1-r_s)\mathrm{MSE}_{\varepsilon}, \tag{22}$$

where r_s is the rate of skipped MBs within the frame under analysis and MSE_{ref} is the MSE of the reference frame(s). $\mathrm{MSE}_{\varepsilon}$ is the mean square error estimate given in section 3, considering the non-skipped MBs only.

5.2 PSNR estimation results

Using the algorithm setup described in 5.1, the PSNR has been estimated and compared with its true value. Results are depicted in figures 5(a) to 5(c), separated according to the frame type. As can be observed from those plots, the proposed method is quite accurate.

For comparison purposes, two recent algorithms [4, 5] have been implemented as faithful to the original work as possible. The algorithm proposed by Ichigaya *et al.* in [4] models the coefficients distribution according to a mixture of two Laplace PDFs: one is estimated using all the quantized values, while the other is estimated using non-zero quantized values only. It has been developed for no-reference PSNR estimation of MPEG-2 encoded videos, but the main ideas proposed in [4] can be easily adapted to H.264.

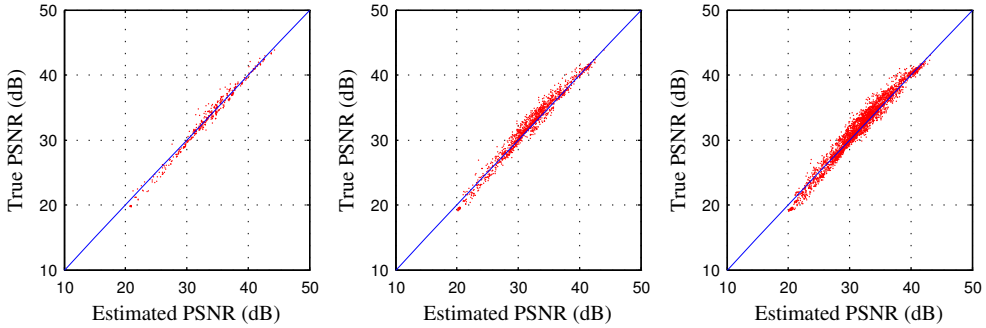
The algorithm proposed by Eden in [5] was developed for no-reference PSNR estimation on H.264 encoded videos and also uses the Laplace model. In short, this algorithm proposes a low complexity parameter estimation method and deals with the "all quantized to zero" problem by imposing empirical bounds for the parameter's value in those situations. An observation of the results depicted in [5], suggests that this method is quite effective for I-frames, but it is not very accurate for estimating the PSNR of P and B frames.

A comparison between the proposed method and our implementations of those methods is depicted on Table 3. As can be observed from the table, the method proposed in this paper shows the highest accuracy regardless of the frame type. Due to a more complex parameter estimation methodology, the proposed method is slower than the remaining ones. During the experiments, the average PSNR estimation computation time for a given frame was 36.3ms, taking about 4.5 and 6.4 times longer than Ichigaya's and Eden's algorithms, respectively (for non-optimized implementation code running on a P4@3.4GHz processor).

6. CONCLUSIONS AND FUTURE WORK

This paper proposes a no-reference PSNR estimation algorithm for H.264 encoded video sequences. When compared with previous

Figure 4: Video sequences used in the experiments. From left to right: *Coastguard*; *Foreman*; *Mobile & Calendar*; *Stephan*; *Table-tennis*; *Tempete*. All sequences have 352×288 resolution and 25 Hz frame rates.



(a) I-frames. (b) P-frames. (c) B-frames.

Figure 5: No-reference PSNR estimation *versus* true PSNR.

	I-frames			P-frames		
	Ichigaya's [4]	Eden's [5]	**proposed**	Ichigaya's [4]	Eden's [5]	**proposed**
Mean absolute error [dB]	0.94	0.98	**0.51**	1.60	1.46	**0.66**
Root mean square error [dB]	1.12	1.16	**0.66**	1.93	1.84	**0.85**
Error value at percentile 99 [dB]	2.15	2.10	**1.34**	3.70	3.72	**1.71**
Correlation	0.99	0.99	**0.99**	0.96	0.98	**0.99**
	B-frames			**All frames**		
	Ichigaya's [4]	Eden's [5]	**proposed**	Ichigaya's [4]	Eden's [5]	**proposed**
Mean absolute error [dB]	1.99	1.90	**0.67**	1.81	1.71	**0.65**
Root mean square error [dB]	2.39	2.30	**0.89**	2.20	2.12	**0.86**
Error value at percentile 99 [dB]	4.38	4.39	**1.81**	4.16	4.12	**1.77**
Correlation	0.94	0.98	**0.98**	0.94	0.98	**0.99**

Table 3: No-reference PSNR estimation error statistics.

work in this field [4,5], it has the ability to compute local errors and uses a different coefficient distribution model in the I frames. Additionally, it tackles the problem of estimating distribution parameters when all coefficients have been quantized to zero exploring correlation between parameter values at adjacent DCT frequencies. The cost is an increase in complexity, mainly due to iterative search of parameter values. Nevertheless, it showed better PSNR estimation accuracy than the reference algorithms, on the video sequences used in the experiments, justifying the increase in complexity.

Since PSNR is a rough quality metric we plan, as further work, to weight the estimated local errors according to spatio-temporal making effects, in order to obtain a video quality metric that correlates better with the human perception of quality.

REFERENCES

[1] *ITU-T H.264 - Advanced video coding for generic visual services*, March 2005.

[2] T. Brandão and M. P. Queluz, "No-reference image quality assessment based on DCT domain statistics ", *Signal Processing*, vol. 88, n. 4, pp. 822–833, April 2008.

[3] T. Brandão and M. P. Queluz, "Blind PSNR estimation of video sequences using quantized DCT coefficient data", in *Proc. of Picture Coding Symposium 2007*, Lisbon, Portugal, Nov. 2007.

[4] A. Ichigaya, M. Kurozumi, N. Hara, Y. Nishida, and E. Nakasu, "A method of estimating coding PSNR using quantized DCT coefficients", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, n. 2, pp. 251–259, Feb. 2006.

[5] A. Eden, "No-reference estimation of the coding PSNR for H.264-coded sequences", *IEEE Transactions on Consumer Electronics*, vol. 53, n. 2, pp. 667–674, May 2007.

[6] I. Richardson, *H.264 and MPEG-4 video compression*, John Wiley & Sons, 2003.

[7] *JM 12.4 - H.264 reference software*, available online at `http://iphome.hhi.de/suehring/tml/`, 2007

[8] Y. Altunbasak and N. Kamaci, "An analysis of the DCT coefficient distribution with the H.264 video coder", in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 3, pp. 177–180, May 2004.

[9] R. Duda, P. Hart and D. Stork, *Pattern Classification - 2nd Edition*, Wiley-Interscience, 2000.