# BLIND ESTIMATION OF THE QP PARAMETER IN H.264/AVC DECODED VIDEO

*M. Tagliasacchi, S. Tubaro*

Politecnico di Milano
Dipartimento di Elettronica e Informazione
20133 Milano, Italy

## ABSTRACT

No-reference video quality monitoring algorithms rely on data collected at the receiver side. Typically, these methods assume the availability of the bitstream, so that motion vectors, coding modes and prediction residuals can be readily extracted. In this paper we show that, even without the availability of the bitstream, the decoded video sequence can be reverse engineered in order to reveal part of its coding history. Specifically, we illustrate a method for blindly estimating the quantization parameter (QP) in H.264/AVC decoded video on a frame-by-frame basis. We demonstrate by means of extensive simulations the robustness of the proposed algorithm. We discuss its usefulness in the field of video quality assessment (e.g. to perform blind PSNR estimation) and we provide an outlook on video forensics tools enabled by the proposed method (e.g. to detect temporal cropping/merging).

## 1. INTRODUCTION

Video communication entails coding and transmission over error-prone networks. Therefore, the received video sequences may be a degraded versions of the original ones. User's experience might be affected by distortions introduced by lossy coding as well as by channel induced distortions. In this context, video quality assessment algorithms provide objective measures of the video quality, possibly well correlation with human perception. Most of the algorithms discussed in the literature belong to the class of full-reference methods, whereby both the original and the processed video sequence need to be available for comparison [1][2][3]. In some circumstances, e.g. when monitoring is to be performed at the receiver side, the original video is unavailable and no-reference methods need to be used instead. In the literature there are several works describing no-reference algorithms that address distortions due to lossy coding [4][5][6] and packet losses [7][8]. All these methods assume the availability of the received bitstream, so that motion vectors, coding modes and prediction residuals can be readily extracted. In some circumstances, the bitstream is unavailable, e.g. because it is encrypted or processed by third party decoders and only the pixel values of the decoded video sequence can be used.

To address this scenario we recognize that each coding operation introduces a characteristic footprint that can be detected and analyzed to trace back the coding history of the considered video sequence. Specifically, in this paper we describe a method that enables the estimation of the quantization parameter (QP) in H.264/AVC video directly from the decoded sequence in the pixel domain. Moreover, it can be effectively added as a pre-processing step in state-of-the-art no-reference algorithms [6] to estimate the PSNR of the received video sequence.

The proposed method is inspired to the recent works in the area of multimedia forensics. In [9] the authors propose a method for estimating the JPEG quantization tables from the pixel values only. In [10], the authors devise an algorithm to detect double JPEG compression, starting from the observation that DCT coefficients exhibit a characteristic statistical distribution. The same method has been applied to MPEG-2 video for tampering detection [11], but it is constrained to work on I-frames only. All the aforementioned methods can be applied to still images only but they fail when applied to video contents encoded with H.264/AVC, since they do not account for the specific spatial and temporal prediction tools enabled by the standard.

The rest of this paper is organized as follows. Section 2 briefly illustrates the H.264/AVC quantizer and a model for the statistical distribution of quantized DCT. Section 3 describes the proposed algorithm, specifying two methods for estimating the quantization parameter. Section 4 demonstrates by means of extensive experimental results the robustness of the proposed method. Section 5 concludes the paper and provides some pointers to future works in the area of multimedia forensics.

## 2. BACKGROUND

We consider the quantization of transformed prediction residuals in H.264/AVC. We explicitly refer to the $4 \times 4$ transform enabled in the baseline, extended and main profiles[1]. Let $\mathbf{E}$

[1]The principles described here also apply to the $8 \times 8$ transform of the high profile

**Table 1**. Base quantization steps

| $mod(QP,6)$ | $q_B$ |
|---|---|
| 0 | 0.625 |
| 1 | 0.6875 |
| 2 | 0.8125 |
| 3 | 0.8750 |
| 4 | 1.0000 |
| 5 | 1.125 |

denote a $4 \times 4$ block of prediction residuals. The (approximate) DCT transform $\mathbf{Y}$ computed by H.264/AVC is given by

$$\mathbf{Y} = \mathbf{Z} \odot \mathbf{S}, \quad \mathbf{Z} = \mathbf{T}\mathbf{E}\mathbf{T}^T \tag{1}$$

where $\mathbf{T}$ and $\mathbf{S}$ are defined in [12] and $\odot$ denotes element-wise multiplication. The transform operation $\mathbf{T}$ is implemented using integer arithmetic only (add and shift operations), while the post-scaling operation $\mathbf{S}$, is implemented together with the quantization, using only integer operations.

The value of the quantized coefficient $Y_j$ is given by:

$$\hat{Y_j} = I_j \times q = \text{sign}(Y_j) \left\lfloor \frac{|Y_j|}{q} + 1 - \alpha \right\rfloor \times q \tag{2}$$

where $q$ is the quantization step, $\alpha$ is a parameter that controls the width of the dead zone around 0 ($\alpha = 2/3$ for intra blocks and $\alpha = 5/6$ for inter blocks) and $I_j$ represents the quantization index that is actually transmitted. The H.264/AVC standard does not define the quantization step directly. Instead, the quantization parameter QP is defined, from which $q$ can be computed as follows

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

where $q_B$ is defined in Table 1.

According to (2), the distribution of quantized coefficients is given by

$$p(\hat{Y}; q) = \sum_k w_k \delta(\hat{Y} - kq) \tag{4}$$

As already observed in [11], if we apply the same transform/quantization process at the decoder to the received block $\tilde{\mathbf{X}}$, we can model the the result using the following distribution

$$p(\tilde{Y}; q) = \sum_k w_k N(\tilde{Y}, kq, \sigma^2) \tag{5}$$

where $N(y, \mu, \sigma^2)$ denoted a Gaussian probability density function with mean $\mu$ and variance $\sigma^2$. The model in (5) accounts for the (irreversible) rounding operations that occur when working in finite precision arithmetic.

## 3. PROPOSED ALGORITHM

The goal of the proposed algorithm is to reliably estimate the $QP$ parameter from the decoded video sequence in the pixel domain, in case the bitstream is not available. To this end, we assume that the sequence has been coded using H.264/AVC (baseline, extended or main profile) and that all macroblocks in a frame share the same $QP$. This can be the result of encoding using a fixed $QP$, or enabling any rate control algorithm that adjusts the $QP$ on a frame-by-frame basis. In the following, we describe the algorithm for P-slices, although the same principles apply to I- and B-slices.

The proposed algorithm computes the quantized motion-compensated prediction residuals in the transform domain, starting from the pixel domain values. Then, the $QP$ is estimated by means of a maximum likelihood estimation procedure that adopts the model in (5). More in detail, the algorithm works as follows. For each frame

1. Perform motion estimation to compute motion vectors for each $4 \times 4$ block. Any motion estimation algorithm can be used for this purpose. Let $(mvx^i, mvy^i)$ denote the motion vector of the $i$th $4 \times 4$ block.

2. Compute the motion-compensated prediction residuals for each $4 \times 4$ block. Let $\hat{\mathbf{X}}^i$ denote the $4 \times 4$ block in the pixel domain and $\hat{\mathbf{E}}^i$ its prediction residuals

3. Discard those block that satisfy the following condition

$$\sum_{x=1}^{4}\sum_{y=1}^{4} |\hat{\mathbf{X}}^i(x,y)|^2 < \sum_{x=1}^{4}\sum_{y=1}^{4} |\hat{\mathbf{E}}^i(x,y)|^2 \tag{6}$$

   This serves to retain only those blocks that are likely to be inter-predicted.

4. Transform the prediction residuals $\hat{\mathbf{E}}^i$ according to (1) to obtain $\hat{\mathbf{Y}}^i$.

5. Collect the transformed prediction residuals from all the retained blocks and estimate the $QP$ as

$$\hat{QP} = \arg\max_{QP} \sum_{j=1}^{N} \log p(\hat{Y}_j; q) \tag{7}$$

   where $q$ is the quantization step corresponding to $QP$ according to (3).

We notice that, when $\sigma \ll q$ as it is typically the case in practical coding scenario when $QP \geq 20$, the expression in (7) simplifies to

$$\hat{QP} = \arg\min_{QP} \sum_{j=1}^{N} \left( \min_k (\hat{Y}_j - kq)^2 - \log(w_{\hat{k}}) \right) \tag{8}$$

where $\hat{k} = \arg\min_k (\hat{Y}_j - kq)^2$. Therefore, for each quantized transform coefficient $\hat{Y}_j$ we need to determine the distance to the closest quantization reconstruction level $kq$, and select the value of $QP$ that minimizes the sum of the distances over all coefficients.

**Table 2**. Test material

| | | R [kbps] | PSNR [dB] |
|---|---|---|---|
| *Hall* | L | 125 | 31.58 |
| | M | 250 | 35.37 |
| | H | 500 | 37.89 |
| *Foreman* | L | 250 | 31.27 |
| | M | 500 | 34.85 |
| | H | 750 | 36.88 |
| *Mobile* | L | 500 | 26.49 |
| | M | 750 | 28.28 |
| | H | 1000 | 29.79 |

**Table 3**. $QP$ estimation results

| | | MeanAE | MaxAE | % |
|---|---|---|---|---|
| *Hall* | L | 0.121 | 7 | 4.3% |
| | M | 0.007 | 2 | 0.3% |
| | H | 0.014 | 2 | 0.7% |
| *Foreman* | L | 0.301 | 11 | 6.8% |
| | M | 0.05 | 7 | 1.0% |
| | H | 0.065 | 2 | 4.6% |
| *Mobile* | L | 0.021 | 6 | 0.3% |
| | M | 0 | 0 | 0.0% |
| | H | 0 | 0 | 0.0% |

In practice, the likelihood function $\sum_{j=1}^{N} \log p(\hat{Y}_j; q)$ presents more than one local maxima, corresponding to integer fractions of the the true quantization step size $q$ (i.e. at $q/2$, $q/3$, etc.) and, in some cases, the global maximum does not correspond to $q$. Therefore, when solving (8), we scan the possible values of $QP$ starting from the largest $QP$ and stopping the search as soon as we find the first local maximum. If we fail to detect a local maximum, the $QP$ of the current frame is set equal to the $QP$ of the previous frame.

## 4. EXPERIMENTAL RESULTS

We tested the proposed algorithm on three sequences at CIF spatial resolution ($352 \times 288$ pixels) and 30 frames/second, namely *Foreman*, *Hall* and *Mobile*. We encoded all sequences using the H.264/AVC reference software (JM12.1 - main profile) using an IPPP group of pictures (GOP), with GOP size equal to 15. We encoded each sequence at three target bitrates. Due to the diverse spatio-temporal characteristics of the sequences, we use a different set of target bitrates for each sequence, corresponding to low (L), medium (M) and high (H) visual quality. We enabled the the rate control algorithm embedded in the reference software [13] with a basic unit equal to 396 macroblocks, so that the $QP$ is adjusted on a frame-by-frame basis within the interval $[24, 40]$. Table 2 reports the encoding conditions used to prepare the test material indicating both the target bitrate and the Peak Signal-to-Noise Ratio (PSNR) between the original and the encoded video sequences.

Table 3 shows the results obtained for the various test conditions using the method illustrated in Section 3. We adopted two error metrics, i.e. the mean absolute error ($MeanAE$) and the maximum absolute error ($MaxAE$) defined as follows

$$MeanAE = \frac{1}{N} \sum_{n=1}^{N} |\hat{QP}(n) - QP(n)| \qquad (9)$$

$$MaxAE = \max_{n=1,..,N} |\hat{QP}(n) - QP(n)| \qquad (10)$$

We also report the percentage of frames for which $\hat{QP}(n) \neq QP(n)$. We observe that, in all cases, the proposed method accurately estimates the correct $QP$ value. The estimation accuracy generally improves at higher bitrates. This is due to the larger number of non-zero transform coefficients that contribute to evaluating (8).

Figure 1 shows, as an example, the actual and estimated QP on a frame-by-frame basis for two encoded video sequences, namely *Foreman* at 500 kbps and *Mobile* at 1000 kbps. In both cases we observe that the estimated value of $QP$, i.e. $\hat{QP}$, matches the true value. For *Foreman* we notice that in two out of the three frames for which $\hat{QP} \neq QP$, the error is equal to 6, i.e. the detected quantization step size is half of the actual value.

## 5. CONCLUSIONS AND FUTURE WORK

In this paper we propose a method for estimating the $QP$ value of H.264/AVC encoded video by exploiting only the raw pixel values, in cases when the bitstream is not available.

The method presented in this paper can be readily combined with state-of-the-art no-reference algorithms such as those in [6], which typically require the availability of the H.264/AVC bitstream. In fact, [6] requires the knowledge of the QP value and motion-compensated prediction residuals in order to compute an estimate of the PSNR between the original and the coded video sequence.

In addition, we can leverage the output of the proposed method as a pre-processing step for further video forensics analysis. Here, we envisage just a few examples of techniques that might take advantage of what is discussed in this paper.

- *Detecting the GOP structure of an encoded video sequence.* The H.264/AVC standard does not impose a regular GOP structure. Nevertheless, it is customary to configure the H.264/AVC encoder in such a way to apply a fixed structure for the whole video sequence. A straightforward extension of the proposed method to I and B slices enables to detect such a structure. If the video sequence is tampered with, e.g. by cropping/inserting frames, the GOP structure is altered.

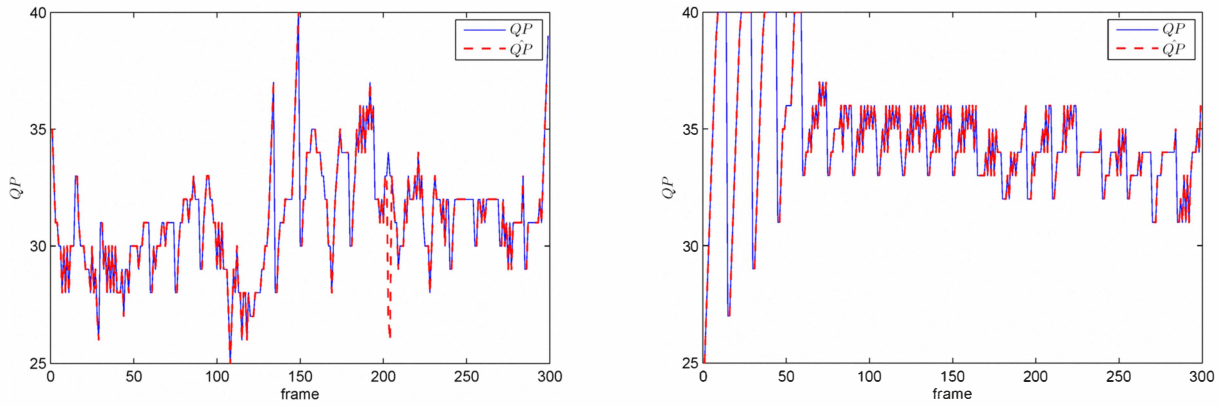


**Fig. 1**. Estimated $\hat{QP}$ for (a) *Foreman* (500 kbps); (b) *Mobile* (1000 kbps) video sequences.

- *Reconstructing the encoded motion field.* The motion vectors estimated at the decoder side might not coincide with those obtained at the decoder side. Nevertheless, once the $QP$ value is obtained, motion vectors can be refined in such a way to find a predictor whose residuals are compatible with the quantization step size. In case such a predictor cannot be found, this can potentially indicate that the video frame has been locally tampered with.
- *Detecting the rate control algorithm.* Rate control is a non-normative coding tool, e.g. it is not defined by the standard. Each encoder might implement a different rate control strategy, which might be interpreted as sort of a footprint of the specific encoder implementation. The proposed method can be extended to enable, at the decoder, to estimate the rate on a frame-by-frame level and, as a consequence, to detect the encoder that has been used.

## 6. REFERENCES


[1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error measurement to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 3, pp. 600–612, April 2004.

[2] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Trans. Broadcast Television Receivers*, , no. 3, pp. 312–322, September 2004.

[3] K. Seshadrinathan and A. C. Bovik, "Motion-based perceptual quality assessment of video," in *SPIE Conference on Human Vision and Electronic Imaging*, San Jose, CA, USA, January 2009.

[4] D. S. Turaga, Y. Chen, and J. Caviedes, "No-reference PSNR estimation for compressed pictures," *Image Communication - Special issue on Objective Video Quality Metrics*, vol. 19, no. 2, pp. 173–184, February 2004.

[5] A. Ichigaya, Y. Nishida, and E. Nakasu, "Nonreference method for estimating PSNR of MPEG-2 coded video by using DCT coefficients and picture energy," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 6, pp. 817–826, June 2008.

[6] T. Brandao and M. P. Queluz, "No-reference PSNR estimation algorithm for H.264 encoded video sequences," in *EURASIP European Signal Processing Conference*, Lausanne, Switzerland, August 2008.

[7] A. R. Reibman, V. A. Vaishmpayan, and Y. Sermadevi, "Quality monitoring of video over a packet network," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 327–334, April 2004.

[8] M. Naccari, M. Tagliasacchi, and S. Tubaro, "No-reference video quality monitoring for h.264/avc coded video," *IEEE Trans. Multimedia*, 2008.

[9] Z. Fan and R. L. de Queiroz, "Identification of bitmap compression history: Jpeg detection and quantizer estimation," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 230–235, February 2003.

[10] T. Penvy and J. Fridrich, "Detection of double-compression in JPEG images for applications in steganography," *IEEE Transactions on Information Security and Forensics*, vol. 3, no. 2, pp. 247–258, 2008.

[11] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double quantization," in *ACM Multimedia and Security Workshop*, Princeton, NJ, USA, September 2009.

[12] I Richardson, *H.264 and MPEG-4 Video Compression*, John Wiley & Sons, 2003.

[13] G. J. Sullivan, T. Wiegand, and K.-P. Lim, "Joint model reference encoding methods and decoding concealment methods," Tech. Rep. JVT-I049, Joint Video Team (JVT), September 2003.