No-reference analysis of decoded MPEG images for PSNR estimation and post-processing

Forchhammer, Søren; Li, Huiying; Andersen, Jakob Dahl

Published in:
Journal of Visual Communication and Image Representation

Link to article, DOI:
10.1016/j.jvcir.2011.01.006

Publication date:
2011

No-reference analysis of decoded MPEG images for PSNR estimation and post-processing

Søren Forchhammer, Huiying Li, and Jakob Dahl Andersen

Department of Photonics Engineering, Technical University of Denmark, Bldg. 343, DK-2800 Kgs. Lyngby, Denmark

Abstract

We propose no-reference analysis and processing of DCT (Discrete Cosine Transform) coded images based on estimation of selected MPEG parameters from the decoded video. The goal is to assess MPEG video quality and perform post-processing without access to neither the original stream nor the code stream. Solutions are presented for MPEG-2 video. A method to estimate the quantization parameters of DCT coded images and MPEG I-frames at the macro-block level is presented. The results of this analysis is used for deblocking and deringing artifact reduction and no-reference PSNR estimation without code stream access. An adaptive deringing method using texture classification is presented. On the test set, the quantization parameters in MPEG-2 I-frames are estimated with an overall accuracy of 99.9 % and the PSNR is estimated with an overall average error of 0.3 dB. The deringing and deblocking algorithms yield improvements of 0.3 dB on the MPEG-2 decoded test sequences.

Keywords: No reference PSNR estimation, quantization parameter estimation, I-frame detection, image post-processing, video post-processing, DCT, MPEG

1. Introduction

Digital TV systems of today are operated using a range of image resolutions and qualities. Furthermore different video coding schemes including MPEG-2 and MPEG-4 part 10/H.264 are used. We address the problem of

Email address: sofo@fotonik.dtu.dk (Søren Forchhammer)

analyzing decoded DCT (Discrete Cosine Transform) images with focus on processing decoded MPEG-2 video, in a heterogenous environment. The goal is to achieve high visual quality and quality assessment. The DCT is also the basis of JPEG image coding and other video coding schemes as H.261-263, MPEG-4 part 2, and proprietary schemes. We consider the problems of frame type detection, PSNR estimation, post-processing and validation based solely on the decoded images. The aim is to increase performance by processing at TV receivers. Large high resolution flat panel displays and high quality projectors of today seem to magnify coding artifacts, so these also become visible in good quality images and video and thus put increased focus on image quality and post-processing. Besides broadcast video and video distributed on cable-net, also images and video material from the Internet, storage media and consumer cameras may be displayed on a large flat panel display. Decoded MPEG video may be post-processed to attenuate the coding artifacts and thereby increase the perceived quality, and it may also be re- or transcoded for storage or further transmission. For all these tasks, adaptive algorithms utilizing MPEG parameters extracted from the coded stream have been presented [1–4]. In some cases however, the coded stream is not accessible e.g. when encryption prevents access to the MPEG stream. In other cases, it may be desirable from an architectural point of view not to access the code stream parameters. One example is receiving the decoded digital video over a High-Definition Multimedia Interface (HDMI) connection, where the MPEG stream prior to decoding was only accessible in encrypted form, e.g. due to digital rights management (DRM) protection. Furthermore the decoded video may be delivered in a different resolution than it was coded. In all these cases, it may be necessary to base the processing and analysis solely on the decoded images and video signals.

MPEG-2 has been the work horse of digital TV. Now H.264/MPEG-4 is being deployed world wide, but MPEG-2 will play a major role in digital TV for years to come, e.g. on cable-net systems and Digital Terrestrial TV. We selected MPEG-2 as a prominent example out of a large class of DCT based image and video coders co-existing with H.264/MPEG-4. We focus on estimating the important MPEG-2 parameters based on the decoded video: DCT block size and position, I-frame detection and for the I-frames, estimation of the quantization step size, which determine the distortion. We shall utilize this information to estimate the PSNR of I-frames and to guide post-processing filters for deblocking and deringing. A measure is introduced to validate that the (decoded) video stream originates from an MPEG-2 coded

stream as well as validating the parameter analysis. Thus decoded MPEG-2 I-frames may be separated from images originating from other formats e.g. decoded MPEG-4/H.264 frames. The identification of I-frames is also beneficial when transcoding MPEG streams [5], e.g. transcoding from MPEG-2 to MPEG-4/H.264.

MPEG-4/H.264 I-frames may differ from those of previous DCT based schemes by applying intra-prediction and an in-loop deblocking filter [6] as well as the smaller 4×4 DCT blocks. This means that the coding artifacts of MPEG-4/H.264 are different and less pronounced compared to those of MPEG-2 [7]. Therefore we suggest applying different post-processing for the two coding schemes and in this paper focus on MPEG-2 post-processing.

Methods for (decoder side) no reference PSNR estimation, i.e. without access to the original video, have been presented based on parameters extracted from the MPEG streams [4]. For MPEG-2 I-frames, an analysis just based on the decoded stream was presented in [8]. This was applied in an JPEG like setting with one fixed quantizer value for each I-frame.

We extend this to the general MPEG-2 case with variable quantizer values at the (16×16 luminance pixels) macroblock (MB) level. Using the estimated MPEG parameters, we shall perform postprocessing of MPEG-2 video coding artifacts originating from the DCT domain quantization and focus on the blocking and ringing artifacts, which are the major artifacts. Research on post-processing of DCT based coding as JPEG and MPEG has a long and active history, e.g. [1–3, 9–15]. To pursue a goal of deringing algorithms, which may be useful for real-time video processing, we focus on the class of relatively simple (non-iterative) spatially adaptive post-filtering. To control this postfiltering, a popular approach [2, 3, 14], is to utilize the MPEG quantization scale parameter (Q_S). The parameter is read from the code stream and we shall refer to this as being embedded post-processing. Another goal is to perform the processing without access to code stream information. Therefore, the Q_S values are estimated based on the decoded video instead. We refer to this approach as pure post-processing to distinguish it from embedded processing. A new spatially adaptive deringing filtering based on texture analysis is developed for pure post-processing.

The rest of the paper is organized as follows: Section 2 introduces notation and the MPEG-2 parameter estimation based on the decoded video for block size and position estimation, quantization step size estimation, and I-frame detection. Section 3 presents no-reference PSNR estimation for I-frames based on the estimated MPEG parameters. Section 4 describes the

use of estimated quantization values for deblocking and deringing of MPEG-2 without code stream access. A new deringing filter developed for this set-up is also introduced. Experimental results are given in Section 5 and Section 6 concludes the paper.

2. MPEG-2 parameter estimation

The discrete cosine transform is widely used in image and video coding. The quality is determined by the quantization of the DCT coefficients. We shall focus our analysis on intra decoded MPEG-2 I-frames as one example among the DCT based standards and consider estimation of MPEG-2 parameters based on the decoded video. To facilitate the MPEG analysis, we shall reconstruct the MPEG (quantized) DCT values based on the decoded pixel values. The DCT is reversible, but rounding, clipping and lack of exact specification of DCT/IDCT in the MPEG-2 specification leave uncertainties.

2.1. Notation for MPEG-2 decoding

To provide the notation for the analysis, selected parts in the decoding process of MPEG-2 I-frames are briefly described. The basic processing unit is the 16x16 pixels (luminance) macroblock (MB), which is further divided into four 8x8 DCT blocks. The DCT transformed coefficients are locally quantized specified by one quantizer scale value, Q_S, per MB.

The variable length decoder outputs the integer values, $I_Q(u, v)$, which represent the indices of the quantization interval for the DCT coefficient at frequency (u, v). Based on $I_Q(u, v)$, a DCT coefficient $F''(u, v)$ is reconstructed in conformance with [16]. For an intra MB, i.e. no motion-compensation is used, the AC coefficients, i.e. $(u, v) \neq (0, 0)$, are reconstructed with an absolute value given by

$$|F''(u,v)| = \left\lfloor \frac{|I_Q(u,v)| \times Q_M(u,v) \times Q_S}{16} \right\rfloor, \tag{1}$$

where $\lfloor\ \rfloor$ denotes the floor function and $Q_M(u, v)$ denotes the frequency dependent quantization matrix values. The four luma DCT blocks in one macroblock are quantized using the same Q_S value, but Q_S may change from one MB to the next. After $F''(u, v)$ is reconstructed at the decoder, the inverse DCT will transform $F''(u, v)$ to an inverse transformed value, which is rounded, and, if necessary clipped, to obtain reconstructed integer values

$r'(i,j)$ in the range [0, 255], for intra blocks, and thereafter $r'(i,j)$ is output as the decoded video, $d(i,j)$.

Based on the details of dequantization of coefficients before as well as rounding and clipping after the IDCT, analysis of the decoded MPEG-2 video may be established. The focus is on estimating or detecting three important (sets of) MPEG parameters: Position of I-frames, DCT blocksize and position, and quantization step sizes (in I-frames). First the DCT block boundary positions are estimated both horizontally and vertically. Based on that, the DCT is applied to each 8×8 DCT block to obtain the recalculated DCT coefficients $F'(u,v)$ as an approximation of $F''(u,v)$. (If the detected block size is not 8×8, the image shall be scaled such that DCT blocks are rescaled to 8×8, prior to the DCT transformation.) Both frame and field DCT may be applied on MBs. The MB type (frame or field MB) can be estimated by selecting the type having the minimum number of zero DCT coefficients within the MB. Thereafter, estimation of Q_M at frame level and Q_S at MB level is performed based on the recalculated DCT coefficients $F'(u,v)$ (frame DCT if the MB is evaluated to be a frame MB, otherwise field DCT). Furthermore, measures of mismatch at MB level (M_{MB}) and frame level are calculated for detection of I-frames and validation of the analysis on detected I-frames. Details of the estimation tasks are given below.

2.2. Blocksize estimation

In [17], the size of DCT blocks was estimated by calculating absolute differences between adjoining pixels, as part of blocking artifact analysis. In order to increase independence of image content, we instead calculate a difference of absolute differences, DAD, horizontally and vertically [18]. Let a, b, c, d, e, f denote the value of 6 consecutive values horizontally. Define the absolute difference $D'_{cd} = |d - c|$ and D_{cd} as the sum of D'_{cd} plus the two corresponding differences D' of the two pixels above and for the two pixels below c and d, respectively. Based on D values of neighboring pixels, an initial difference of difference is calculated as $DAD'_{cd} = 2D_{cd} - 2D_{bc} - 2D_{de} + D_{ab} + D_{ef}$, where subscripts specifies the pixels involved. This value is thereafter thresholded to form $DAD_{cd} = DAD'_{cd}$ if $3 < DAD'_{cd} < 120$ and 0 otherwise. The DAD values are thereafter projected by summation onto the horizontal and vertical axis, respectively.

2.3. Quantization parameter estimation

Given the DCT blocksize and position, the DCT coefficients may be recalculated, based on the decoded frames. The next objective is to estimate the quantization matrix, $Q_M(u,v)$, and the quantization scale factor, Q_S. For intra coded MBs, the quantized MPEG-2 DCT coefficients can (approximately) be recovered by applying an 8×8 DCT to the decoded video $d(i,j)$. Without information about motion vectors and residues, it is not tractable to recover DCT coefficients of non-intra blocks. We consider the general case where intra/non-intra frame and MB type information is not known (or uncertain). Initially all frames are processed and the proposed method treats all MBs as intra. This may be considered a hypothesis, which after the processing, is validated or rejected. The intraframe quantization step size, $\Delta(u,v)$, in (1) is a function of the DCT frequency, (u,v),

$$\Delta(u,v) = \frac{Q_S \times Q_M(u,v)}{16}. \tag{2}$$

The initial step, for recovering I-frame values of Q_S (and $Q_M(u,v)$), is to recalculate the DCT coefficients, $F'(u,v)$, and based on these estimate the product $I_Q \times Q_S \times Q_M$ (1) for each DCT coefficient (u,v). It may be noted that Q_M is fixed at frame level and Q_S is fixed at MB level. The values of I_Q, Q_S and Q_M are all integers, and in principle Q_S and $Q_M(u,v)$ are identical to a greatest common divisor (or a divisor of this) [8]. For variable Q_S, this applies for $Q_M(u,v)$ at frame level for each frequency, (u,v) and for Q_S at MB level across the frequencies for given $Q_M(u,v)$. Based on the recalculated DCT coefficients, $F'(u,v)$, we will estimate the integer product of $I_Q \times Q_S \times Q_M$. However, errors may occur due to the non-linear processing of rounding (and clipping) after the inverse DCT transformation and the integer division in the decoder.

As the greatest common divisor operator is highly non-linear, we shall also consider a measure expressing the likelihood of a potential MPEG-2 DCT value, conforming with (1), given the recalculated F' based on the data observed, i.e. the decoded MPEG. Empirically, we have noted that the distribution of the rounding error, $x = F'' - F'$, can be approximated by a (zero mean) Laplacian distribution,

$$f(x) = (2\lambda)^{-1} e^{-|x|/\lambda}. \tag{3}$$

Assuming this distribution, the estimation (or validation) of Q_S and Q_M may be expressed as a maximum likelihood problem by maximizing a product of

terms given by (3). For given frequency (u, v), λ and candidate values of $x = F'' - F'$, taking the logarithm of the likelihood (3), the optimization may be expressed as finding the minimum of a sum of log-likelihood terms of the form

$$\min_{n,\Delta(u,v)} (|F'(u,v) - n\Delta(u,v)|), \tag{4}$$

where n is an integer and $\Delta(u, v)$ is given by Q_S and $Q_M(u, v)$ (2). The argument is the distance between the recalculated DCT value $F'(u, v)$ and the closest reconstructed value, F'', of the MPEG-2 decoder (1).

Based on the quite different notions of a greatest common denominator and this likelihood expression, practical and robust estimation schemes are presented below. The estimation process starts with Q_M estimation because Q_M is fixed for the whole frame, thus providing more statistics.

2.3.1. Quantization matrix estimation (Q_M)

The MPEG-2 default intra Q_M [16] is widely used in many applications. This was confirmed by analysis of some MPEG-2 streams captured from on-air transmission, where one other intra Q_M matrix was observed. Restricting the analysis to a limited set of candidate Q_M matrices, the problem is simplified to that of identifying (and validating) one of the candidates. The MPEG-2 default intra quantization matrix, Q_M, as well as the other intra Q_M observed in the on-air analysis constitute our candidate set.

As noted, Q_M is a sequence (or frame) level parameter set. The coefficients of the four DCT blocks within the same MB are quantized using the same value of Q_S. For each MB and candidate Q_M and a given Q_S, a *mismatch* value, M_{MB}, is introduced and defined by

$$M_{MB}(Q_M, Q_S) = \sum_{(u,v)\in MB} |round\left(\frac{F'(u,v) \times 16}{Q_M(u,v) \times Q_S}\right) - \frac{F'(u,v) \times 16}{Q_M(u,v) \times Q_S}|. \tag{5}$$

For each MB and candidate Q_M, the minimum value of (5) with respect to Q_S is found and the minimal values for each MB are summed up over all MBs. Finally the Q_M with the overall minimum mismatch is selected,

$$\hat{Q}_M = Q_M(p) \ : \ \arg \min_{p\in\{Q_M\}} \sum_{MB} \min_{q \ \in \ \{Q_S\}} (M_{MB}(p, q)).$$

The Q_S values selected in the inner sum are only used for selecting Q_M. Once Q_M is selected a more accurate algorithm is applied to estimate the Q_S values.

The distance in (5) is given by the log-likelihood expression (4) normalized by the quantization step size to make the mismatch value less dependent of Q_S and Q_M. This will favor the lower frequencies, which generally dominate for typical sequences and quantization matrices.

2.3.2. Quantizer scale estimation (Q_S)

For JPEG images and for MPEG-2 intra frames operated in a JPEG-like fashion with fixed Q_S, the decoded DCT coefficients $F''(u,v)$ for each frequency (u,v) will be distributed on integer multiples of the fixed quantization step size [8] (with integer division truncation shift S) as the quantization matrix is also fixed for (u,v). For given $Q_M(u,v)$ the values reflect the integer product $(|I_Q| \times Q_S)$.

We shall extend this to the general case of MPEG-2 allowing for variable Q_S at MB level, as is widely used, e.g. due to rate control. For a single MB, Q_S can be estimated as the greatest common divisor (gcd) of $(|I_Q| \times Q_S)$. The reconstructed values of $(|I_Q| \times Q_S)$ from the decoded video may be in error due to the decoder rounding (error), the integer division truncation error, $0 \leq S < 1$, and possibly other processing steps. As the gcd operator is highly non-linear, meaning that one wrong input value can lead to a completely wrong estimate, a more robust algorithm was developed. Let $E_r(u,v)$ denote the spatial rounding error transformed back to the DCT domain. In order to combine the error-prone values, F' with the notion of gcd, a maximum error, E_{max} is defined such that most of the time the correct F'' satisfies $|F'' - F'| \leq E_{max}$. This may be combined with (1) leading to an integer upper bound, $((|I_Q| \times Q_S)_{up}$ (and an integer lower bound) for $(|I_Q| \times Q_S)$. For example for $Q_M \geq 16$, we have:

$$\left\lceil \frac{16(|F'| - E_{max})}{Q_M} \right\rceil \leq |I_Q| \times Q_S \leq \left\lceil \frac{16(|F'| + E_{max})}{Q_M} \right\rceil . \qquad (6)$$

E_{max} was determined experimentally by fitting a Laplacian distribution (3) to the $E_r(u,v)$ for each DCT coefficient, and selecting E_{max} as the 99% level of the cumulative density function. The expression above bounds the value of $(|I_Q| \times Q_S)$ for one specific coefficient. For a single MB, Q_S is fixed and we may upper bound Q_S, by $Q_S^{up} = min((I_Q \times Q_S)_{up})$, where the minimum is taken over the frequencies (u,v) for all the non-zero DCT coefficients.
Below we present the proposed fuzzy gcd Q_S estimation algorithm.

The set $\{Q_S\}$ of potential Q_S values are given by the MPEG-2 Q_S table defined by MPEG-2 [16]. This shall constitute the candidate set.

The Q_S estimation algorithm. For each MB do

1. For all the AC DCT values, $F'(u,v)$ $(u,v) \neq (0,0)$, within the current MB, calculate $F_{QS}(u,v) = |F'(u,v)| \times 16/Q_M$, and $(|I_Q(u,v)| \times Q_S)_{up}$.
2. Round $F_{QS}(u,v)$ to the nearest even integer value $K(u,v)$.
3. Set all $K(u,v)$ less than 4 to zero. (All DC values are set to 0.)
4. Calculate the Q_S upper bound Q_S^{up} by $min((I_Q \times Q_S)_{up})$ (min is over the non-zero DCT coefficients).
5. For $j \in \{Q_S\}$ and $j_{min} = 4 \leq j \leq Q_S^{up}$:

$$\hat{Q}_S = \arg\max_j [N_1(j) + N_2(j)] \tag{7}$$

   where $N_1(Q_S)$ is the number of DCT coefficients for which $K(u,v) = Q_S$, and $N_2(Q_S)$ is the number of $K(u,v)$, which are divisible by Q_S.
6. For MBs, for which all AC coefficients have the value 0, the steps above do not provide a result. Instead, the estimated $\hat{Q}_S$ value from the previous MB is used for the current MB.

The term $N_1(j) + N_2(j)$, in the main expression (7), favors the greatest common divisor over smaller common divisors. As the DCT values are well modeled by the Laplace distribution, small DCT values are more probable than higher values. Especially $I = 1$ is the most probable non-zero coefficient. Thus the term N_1 puts extra emphasis on what corresponds to $I = 1$ and should provide a good chance of preventing that smaller erroneous values will influence the result. Step 6 is motivated by the fact that MPEG-2 (rate control) has a bias towards maintaining the same Q_S value from MB to MB (or changing Q_S in small steps).

In some cases, the decoded images may have been scaled to a different resolution after decoding, e.g. images from an HDMI connection. In these cases, the block size algorithm (Sect. 2.2) may be used to detect the scaling factor assuming an MPEG-2 8×8 block size in the coded stream prior to the scaling. Rescaling the images back to the resolution corresponding to 8×8 block resolution allows to run the algorithm above to estimate the Q_S values. Based on experiments, we increase the threshold value in Step 3 from 4 to 6 when analyzing rescaled images and increase the interval of width $\pm E_{max}$ to $\pm 2E_{max}$ in (6) to make the algorithm robust towards distortion in the scalings.

2.4. Validation and I-frame detection

I-frame detection in MPEG-2 has been proposed based on the number of zero-valued DCT coefficients in each frame [5]. An adaptive threshold was applied to the number of zero coefficients to detect I-frames. The specific algorithm in [5] is limited in that it uses a preset value for the (approximate) GOP length and an initial value for the decision threshold and thereafter processes the values over multiple frames.

Here a novel I-frame detection and validation method is proposed. The decisions are made at frame level and therefore the detection may also be applied to adaptive GOP structures as well as providing fast detection. A frame level mismatch measure M_F, based on averaging the macroblock mismatch measures M_{MB} (5) within the frame, is introduced

$$M_F = \frac{1}{N_F} \sum_{MB} M_{MB}(\hat{Q}_M, \hat{Q}_S), \tag{8}$$

where N_F is the total number of AC coefficients, $F'(u, v)$. Thus M_F measures the mismatch per AC coefficient. When the Q_S values of the frame are correctly estimated, small values of M_F are obtained (as the distances in (5)). For the P- and B-frame types, the motion compensated contributions will lead to many misleading contributions and due to the use of a (fuzzy) gcd approach, generally these errors will lead to smaller estimated Q_S values. A threshold, T_{MF}, is applied to each M_F value. Figure 1 depicts M_F values as a function of estimated Q_S values for I-frames and others (P- and B-frames), respectively. Defining the threshold $T_{MF}(Q_S)$ as a simple function of the average estimated $\hat{Q}_S$ value leads to a clear cut decision. A third-order polynomial solution

$$T_{MF}(Q_S) = a + bQ_S + cQ_S^2 + dQ_S^3 \tag{9}$$

is chosen. If M_F is below the threshold, $T_{MF}(\hat{Q}_S)$, the frame is detected as an I-frame. Figure 1 shows a clear separation between MPEG-2 I-frames on one hand and P-/B-frames on the other hand.

Experiments were also conducted on decoded MPEG-4/H.264 images of the same original test sequences providing M_F values in the range of 0.0496 and 0.174, with an average of 0.114. For the MPEG-2 decoded I-frames the maximum value of M_F was only 0.0235 (Fig. 1), i.e. less than half the minimum value for the MPEG-4 images and thus very clearly distinguishing MPEG-2 I-frames from MPEG-4/H-264 frames.

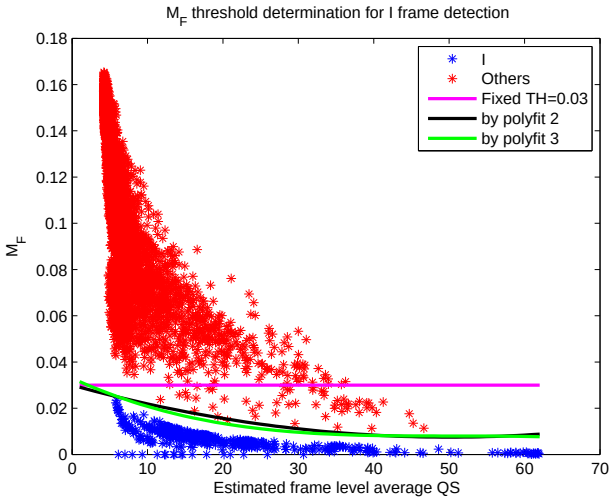

Figure 1: Mismatch values M_F for I-frames and other frames (P and B) as a function of estimated Q_S and the selected 3rd order polynomial threshold function (polyfit 3).

3. PSNR estimation based on decoded video

A number of papers have addressed no-reference quality estimation by PSNR (peak-signal-to-noise-ratio) based on MPEG-2 information of e.g. quantization parameters read from the code stream [4, 19].

For I-frames, the image quality in terms of PSNR can be estimated without code stream access, using the quantization step size values, Q_S, estimated at MB level (Section 2). In [8], no reference PSNR estimation based on decoded frames was presented for I-frames, which were MPEG-2 coded, using a JPEG like setting with a fixed Q_S. The value of Q_S was estimated at frame level based on a frequency analysis of the DCT coefficients for each frequency (u, v), restricted to the setting that Q_S would be constant over a frame. In most video applications rate control or rate shaping is applied leading to a variation of Q_S at MB level and we extend PSNR estimation to this case based on the MB level Q_S value estimation presented in Section 2. Below we revisit the no-reference techniques of [4, 8, 19, 20] to select a number of methods, which are later combined with the estimated values of Q_S at MB level and thereafter tested (Section 5.3). Due to Parseval's equation, the mean square error (mse) for the image data may be calculated in the DCT domain. Common for the papers is that the distribution of the DCT coefficients is modeled by a Laplacian distribution (3) prior to quantization. In [4, 19, 20], the estimates are based on DCT coefficient index values read from

the stream. In [8], as in our case, the estimates are based on a reconstruction of the quantized DCT coefficients from the decoded video. The estimation process involves estimating the λ parameter (3) of the (zero-mean) Laplacian distribution for the different frequencies and Q_S values, and thereafter combining this with the quantization scheme to express an expected value of the PSNR. Two estimation problems are noticed: 1) The Laplacian distribution is a good approximation but not perfect and 2) in some cases there may be a lack of (non-zero) observations, e.g. at high frequencies.

3.1. Estimating the mean square error

Expressions were derived in [8] for calculating the overall contributions to the MPEG-2 mean square error for the DC coefficient ε^2_{DC} and the AC coefficients ε^2_{AC}. Assuming a (zero-mean) Laplacian distribution (3) and integrating the mse over each quantization interval for given λ, the mse for each AC frequency was derived as:

$$\varepsilon^2_{AC} = 2\lambda^2 - \frac{2\lambda\Delta e^{-\alpha/\lambda}e^{-\Delta/2\lambda}}{1-e^{-\Delta/\lambda}}[\frac{\alpha}{\lambda}+1], \tag{10}$$

where Δ is the quantization step size as in (2), α is the shift factor in the MPEG-2 quantization scheme, and λ is the Laplacian parameter for the AC coefficient. The values of Δ, α and λ can vary for different AC frequencies (u, v), leading to a dependency on Q_M and Q_S.

The average distortion at frame level was estimated in [8] by inserting the estimated (fixed) Q_S and the estimated value of λ for each coefficient in (10). The quantization of the DC coefficient is, independently of Q_S, controlled by the parameter intra DC precision (IDP) from which the expected contribution to the distortion, ε_{DC} [8] is derived. The contributions of each coefficient are summed for the overall distortion estimate,

$$\hat{D} = \frac{1}{64}[\varepsilon^2_{DC} + \sum_{i=1}^{63}\varepsilon^2_{AC}(i)], \tag{11}$$

where i is an index of the frequencies (u, v) of the 63 AC coefficients.

Usually Q_S will vary over the MBs within one frame due to rate control. To handle the general case, a weighted sum over contributions for each frequency and quantizer step size is calculated [4]. Following this approach, the distortion expression (11) is generalized here. First the distortion contribution is calculated for each Q_S value for frequency (u, v) indexed i, using

Q_S and $Q_M(i)$ in (10) to achieve $\varepsilon^2_{AC}(Q_S, i)$. Thereafter a weighted average based on occurrences of Q_S values yields,

$$\hat{D} = \frac{\varepsilon^2_{DC} \times B_v \times B_h + \sum_{Q_S} \sum_i \varepsilon^2_{AC}(Q_S, i) \times N_{AC}(Q_S, i)}{B_v \times B_h + \sum_{Q_S} \sum_{Q_M} N_{AC}(Q_S, i)} \quad (12)$$

where B_v and B_h are the number of DCT blocks vertically and horizontally, respectively, and $N_{AC}(Q_S, i)$ is the number of AC coefficients for the (Q_S, i) combination.

3.2. Estimating the Laplace parameter

Each contribution to the distortion, $\varepsilon^2_{AC}(Q_S, i)$, given by (10) and used in (11) or (12), depends on the parameter of the Laplacian distribution, λ, estimated based on the received data. Estimation may be based on the variance [4], modified versions of second moment statistics [4, 8] or the number of zero coefficients [20].

In the basic version presented in [4], the standard deviation of the reconstructed DCT coefficients is used to estimate the Laplacian parameter, i.e. without taking the effect of quantization into account at this point. In [8], λ was estimated for each DCT coefficient based on analysis of the second moment for each of the first 24 AC coefficients (in zigzag scan order) taking the quantization effect into account. For the higher frequencies, this was modified by assuming $2\lambda^2$ as the variance. Also in [4], a compensated version is introduced by weighting the distribution based on the standard deviation with a distribution estimated based on non-zero coefficients. Both of these solutions deviate from the model to compensate for inaccuracies. The reasons may be that using the second moment is not very robust, e.g. for heavy tail distributions and to take the effects of quantization partially into account [4].

Estimating λ based on the number of zero coefficients (NZ) [20, 21] provides a faster and simpler solution, which readily matches the use of decoder side statistics and provides robustness towards outliers and heavy tail distributions. In [20] this was used for no-reference PSNR estimation of H.264 sequences. We apply this approach to MPEG-2 decoded sequences. Let $p_0 = N_0/N$ denote the ratio of the (in our case estimated) number of zero coefficients, N_0, i.e. the number of coefficients reconstructed to lie in the dead-zone interval $[-\Delta/2 - \alpha, \Delta/2 + \alpha]$, and the total number of coefficients, N.

Integrating a Laplacian distribution over the deadzone interval and equating this expression to p_0 leads to the parameter estimate,

$$\lambda(Q_S, i) = -\frac{\Delta/2 + \alpha}{ln(1 - p_0(Q_S, i))}, \tag{13}$$

where $p_0(Q_S, i)$ is the ratio of the number of zero coefficients over all the coefficients quantized by Q_S and $Q_M(i)$, and Δ is given by Q_S and $Q_M(i)$. When calculating the expressions (10) and (13), α is determined as in [8], i.e. it is obtained from the MPEG-2 TM5.

In [19] and [22], the performance is improved using a more complex maximum likelihood scheme for estimating λ. Furthermore cross coefficient correlation is utilized requiring a training phase. This and other techniques requiring training to adjust parameters are not further pursued here.

In [4], varying Q_S is considered and $\lambda(Q_S, i)$ is calculated for each Q_S and frequency, indexed by i. This was motivated by the observation that rate control schemes often select similar Q_S values for parts of the image having similar local statistics. Thus $\lambda(Q_S, i)$ may capture some of the local variations for better modeling. On the other hand, spreading the statistics for one frequency (u, v) on multiple values of Q_S may lead to an increase in the variance of the estimates of $\lambda(Q_S, i)$. We consider the alternative of using a weighted average, $\lambda_{avg}(i)$, of the $\lambda(Q_S, i)$ values for each $i = (u, v)$ based on the occurrence counts $N(Q_S, i)$ and using this one value of λ for all values of Q_S for a given i.

The use of second moment statistics or NZ combined with using $\lambda(Q_S, i)$ or $\lambda_{avg}(i)$ leads to four (slightly) different estimators (M1-M4) of the PSNR for the case of varying Q_S at MB level. In all four cases, the estimated λ values are used to calculate $\varepsilon^2_{AC}(i)$ or $\varepsilon^2_{AC}(Q_S, i)$ by (10) and the total distortion by (12). Methods M1 and M3 estimate λ based on zero coefficients (13). M1 estimates the $\lambda(Q_S, i)$ values by applying (13) for each $(Q_S, Q_M(i))$. Method M3 calculates the average $\lambda_{avg}(i)$ over Q_S for each frequency. Methods M2 and M4 estimate the λ values using second moment statistics as in [8]. M2 calculates a weighted average $\lambda_{avg}(i)$, while M4 uses $\lambda(Q_S, i)$. The performance of these methods are evaluated experimentally in Section 5.

4. Post-processing of decoded MPEG-2

In this section, the estimated MPEG parameters are used to guide post-processing, in order to provide what we call pure post-processing, i.e. without

access to the MPEG stream. Blocking and ringing artifacts are the major artifacts and when visible they are very annoying. This work will mainly discuss how to attenuate these two types of artifacts. We follow the popular approach of spatially adaptive post-filtering [2, 3, 14], where the region of artifacts are detected by analyzing the local activity of blocks and the quantizer parameter Q_S is used to control the post-processing. This approach provides a flexible solution of limited computational complexity.

Due to complexity constraints derived from a perspective of real-time video processing, other more complex approaches [1] have been ruled out. This includes methods of image restoration and enhancement methods, as projection onto convex sets (POCS) [9, 11], constrained least squares, and maximum a posteriori, maximum likelihood and anisotropic diffusion. Furthermore, we shall also refrain from heavy usage of DCT domain information and processing as in [9, 11, 23]. Motion-compensation or temporal techniques also impose increased complexity, e.g. [10, 12, 13, 15]. For MPEG-4/H.264 post-processing was presented in [15], with focus on motion-compensated filters for low-bit rate video (PSNR from 20-30 dB). Restraining the post-processing from temporal techniques, e.g. using motion vectors, does reduce the control of temporal artifacts as flickering [12, 15, 25].

We will focus on simple spatially adaptive filters, omit motion-compensation and utilize the analysis of decoded MPEG video presented in Sections 2 and 3 to estimate parameters for controlling the post-processing: The DCT block size and DCT block boundary positions are used to localize MPEG blocking artifacts and potential areas for ringing artifacts. First the I-frame detection is applied. For the detected I-frames, the quantization parameter Q_S is estimated (Section 2) and used to control the deblocking and deringing filters. Post-processing as deblocking and deringing are low-pass of nature, so visually it is desirable to focus on the areas where the artifacts are potentially visible and avoid low-pass filtering edges and texture. The MPEG-2 post-processing filters are applied to the video frames individually and apart from the control information (average Q_S of most recent I-frame) treats each frame as an independent image. After estimating Q_S the deringing filter is applied followed by the deblocking filter horizontally and vertically.

Whereas many embedded post-processing filters rely on the local Q_S value, we utilize the average Q_S value of the latest I-frame as a measure of the quality for the rest of the GOP. The motivation is that the quality of the I-frame will propagate to the P- and B-frames of the GOP and related to this, it is assumed that the MPEG quality within a GOP is roughly constant.

4.1. Deblocking

The deblocking filter is based on the deblocking filter in MPEG-4 part2 [26, 27], but here transformed to MPEG-2. Similar deblocking filters are reported in [2, 3, 14, 28, 29]. Adaptive deblocking is performed along the 8×8 DCT block boundaries horizontally and vertically using two modes based on local texture. Stronger and longer filters, called DC offset mode filtering, are applied on very smooth regions; weaker and shorter filters, called default mode filtering, are applied in the other regions. The values of Q_S are used as thresholds at MB level for both DC mode filtering and default mode filtering [26, 28]. In [14, 29] an intermediate region was introduced. We introduce two minor changes compared to [26]. Let v_1-v_8 denote 8 pixels across the boundary, and MAX and MIN the maximum and minimum value of these, respectively. 1) More post-processing is performed by filtering the intermediate regions using the default mode filter (instead of not filtering): In low activity areas, the DC mode filter is applied for $MAX - MIN < 2Q_S$ and no filtering otherwise [26]. In our filter, the default filter is applied instead of no filtering. 2) The default mode decision of actually changing a given block boundary pixel is slightly modified [18]: Define $a_{3,0} = 2v_3 - 5v_4 + 5v_5 - 2v_6$. In [26], deblocking is considered if $|a_{3,0}| < 8Q_S$, this condition is replaced by $|a_{3,0}| < MIN(5Q_S + 32, 160)$. In our pure post-processing, I-frames are processed using the estimated MB based Q_S. For P- and B-frames, the average Q_S from the previous I-frame is used for the rest of the GOP.

4.2. Deringing

The high frequency distortion from the DCT quantization causes spatial domain oscillation near high-contrast edges. The spatially adaptive post-processing filters in [2, 15, 27] included deringing. Cartoons are mainly composed of big uniform areas separated by edges, which will easily lead to clearly visible ringing [10, 13]. In natural images, the visibility of ringing artifacts is suppressed in texture regions due to masking effects. To utilize this masking and maintain the sharpness of the texture our deringing method uses texture classification in the vicinity of sharp edges.

For the deringing, simple image texture analysis techniques are chosen due to a perspective of real-time implementation on HDTV material and a desire for robustness towards errors in the Q_S estimation, where errors may occur on a few MB for I-frames. On P- and B-frames, we rely on the (average) estimated Q_S value of the last I-frame. Compared with [10, 13],

the texture analysis is used to preserve the sharpness of the texture. The overall structure of the deringing filter is shown in Fig. 2.

The biggest challenge of the texture classification is to distinguish real image texture from artifacts, as ringing artifacts also constitute a kind of texture. For increasing values of Q_S, the original image texture gets more blurred and the level of artifacts increases. Therefore for high Q_S values, the local variance will be dominated by the ringing artifacts. The threshold for the texture classification should therefore be influenced by the (in our case estimated) Q_S values.

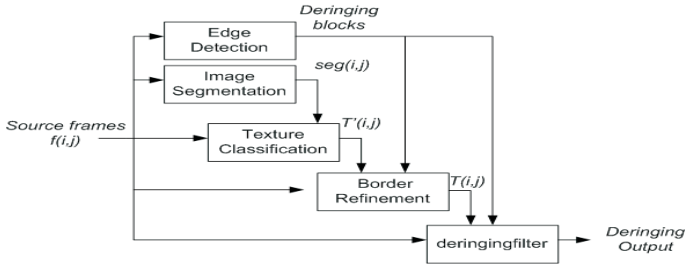

Figure 2: The overall structure of deringing filter.

4.2.1. Edge detection

Canny's method [30] was chosen for edge detection, as it has the advantages of a low detection error rate, good edge localization and only single pixel edge width. The result is a binary image where the positions of edges are marked by '1', and the other positions are marked by '0'. The *deringing blocks* are defined by the DCT blocks, which have at least one edge pixel marked by a '1', i.e. a detected edge.

4.2.2. Image segmentation

The method applied here is a simple partial segmentation, which maps the image onto homogeneous regions with respect to brightness. Let $f(i,j)$ denote the image brightness at position (i,j). First the image is divided into 16 (4×4) parts and the segmentation is performed on each part using iterative threshold selection [30]. The threshold set $\{T_{seg}\}$ initially contains 16 thresholds sorted in an increasing order. Thereafter $T_{seg}(m)$ is deleted, if

$$T_{seg}(m) - T_{seg}(m-1) < \frac{T_{seg}(16) - T_{seg}(1)}{25}. \tag{14}$$

The value of 25 has been selected based on experimental segmentation results to find a good balance between the number of regions and the activity

level within a texture region. The resulting number of valid thresholds is n ($0 < n \leq 16$). The segmentation index at pixel position (i, j) is denoted as $seg(i, j)$, $(1 \leq seg(i, j) \leq n + 1)$.

4.2.3. Texture classification

A local texture activity $E(i, j)$ is defined by:

$$E(i,j) = min\left(\sum_{(k,l)\in\mathcal{N}_{i,j}} |f(i,j) - f(k,l)| \times \delta(k,l), 40\right),$$

where $\mathcal{N}_{i,j}$ defines the neighborhood given by the 4-neighbors, i.e. the four pixels on the top, down, right and left side of the current pixel position (i, j) and the binary function $\delta(k, l) = 0$ if $seg(i, j) = seg(k, l)$ and 1 otherwise. The individual contributions to the sum are clamped to 40 to prevent single large values from dominating E. In order to avoid sudden changes of $E(i, j)$ and increase the reliability, a smoothing step is applied: $I(i, j)$ is obtained by convolving the local energy E with the mask F, $I = E * F$, where the mask F is a 5×5 all 1 filter. To distinguish image values in areas of high texture from ringing texture, the texture activity of the pixel at position (i, j) is compared to a threshold to obtain a binary mask,

$$T'(i,j) = \begin{cases} 1 & \text{if } I(i,j) < 120 + Q_S \\ 0 & otherwise \end{cases} \tag{15}$$

4.2.4. Smooth area border refinement

The aim of the proposed deringing is to focus on smooth areas close to sharp edges. But the regions near sharp edges are easily misclassified to belong to a texture area because of two issues: Ringing artifacts increase the local variance and high local texture energy is propagated to the nearby areas after the smoothing process. A conditional dilation operation is performed in the smooth area near the edges detected by the Canny filter. A 5×5 dilation mask is used, and the surviving center pixels are classified as smooth pixels. Based on the basic dilation, two additional conditions are checked. The first condition requires that the absolute difference in intensity between the center pixel and the extended pixel has to be below a threshold value equal to Q_S. The second condition requires that the difference of segmentation labels of the center pixel and the extended pixel is not larger than 2. The resulting output is an updated binary texture mask, $T(i, j)$.

4.2.5. Deringing filter

Deringing is desired in regions given by pixels of smooth regions within the *deringing blocks* defined by the edge detection (Section 4.2.1). The adaptive deringing is performed as follows starting from a simple low-pass filter. Let m and n be integers and let $h(m,n)$ denote a fixed deringing filter mask, centered at $h(0,0)$, with non-zero coefficients given by

$$h(m,n) = \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}$$

for $(m,n) \in \{-1,0,1\}^2$. Thresholding $h(m,n)$ gives an adaptive deringing mask

$$\hat{h}(k-i,l-j) = \begin{cases} h(k-i,l-j) & \text{if } B(k,l) = 1 \\ 0 & otherwise \end{cases}$$

where k and l are integers and $B(k,l)$ is a boolean factor, which is true when

$$(T(k,l) = 1) \wedge (|f(i,j) - f(k,l)| < 1.5Q'_S), \tag{16}$$

where Q'_S is given by $Q'_S = k_1 Q_S + k_2$, where k_1 is a scaling and k_2 is an offset parameter. Experimentally, k_1 was set to 0.3 and k_2 was set to 6 for P- and B-frames and 0 for I-frames. $\hat{h}(k-i,l-j)$ is thereafter normalized and used as the deringing filter.

4.3. Complexity

The methods presented have been prototype implemented in FPGA technology [31, 32] introducing only slight modifications of the algorithms as presented. This has validated the feasibility for real-time processing of full HD (1080×1920). The only significant algorithmic change was in the image segmentation (Sect. 4.2.2) in the deringing. Since the FPGA memory could only store about 1/8th of an HDTV frame, statistics was collected for each frame and then used for calculating the threshold values for the segmentation in (14) for the next frame. In this way the frames may be processed in slices.

5. Experimental results

The estimation of quantization parameters and subsequent I-frame detection, PSNR estimation and post-processing has been tested. We used six SD progressive test sequences for testing these algorithms. Four of the test

sequences, CITY, CREW, HARBOUR, and SOCCER, have 300 frames with a resolution of 720×576. ICE has 240 frames at 704×576 and FRIES has 220 frames at 720×576. The sequences were MPEG-2 coded using the TM5 rate control [33] at constant bitrates of 2M, 3M and 4Mbits/s using MPEG-2 default intra Q_M and frame MB processing, thus leading to 18 decoded sequences. The MPEG GOP length was (N=) 12 and there were 2 B-frames (M=3) between the P-frames. For initial testing and training only the first 100 frames of the 4 test sequences (CITY, SOCCER, ICE and CREW) were used, i.e. there were 12 decoded sequences in this short test set.

5.1. Quantization parameter estimation results

The MPEG-2 parameters were estimated using the Y component only. Two candidate intra Q_M matrices were used as mentioned previously. The correct Q_M matrix was identified for all I-frames. Since Q_S can vary from MB to MB, MB-level Q_S estimation was performed. For the 18 MPEG-2 decoded sequences, very accurate estimates of Q_S in I-frames were achieved by the Q_S algorithm (Table 1). The overall average of I-frame Q_S values read from the MPEG stream (MQ_S) is 20.99 and the average estimated Q_S (EQ_S) value is 21.01, i.e. the accuracy is 99.9 %.

To test the performance on video scaled to a different resolution, the 12 decoded initial test sequences (100 frames) were up-scaled to 1080p, the block size was identified (Sect. 2.2) and thereafter the images were down-scaled using cubic interpolation. The same MPEG-2 analysis scheme was applied to these re-scaled sequences but with slightly different parameters as explained in Sect. 2.3.2. An accurate Q_S estimation is also achieved in this case with an overall average Q_S of 15.35, compared to the actual average Q_S of 15.23. Histograms of estimated Q_S based on the rescaled images and the actual values of Q_S are depicted in Fig. 3.

5.2. Results on I-frame detection and validation

The I-frame detection algorithm based on thresholding (8) the mismatch measure, M_F, was tested on the 18 decoded full length test sequences concatenated into one video stream (4980 frames in all). The parameters of the threshold function (9) were based on segmenting the M_F values of the test data also displayed in Fig. 1. A fit of the third order polynomial threshold function (9) to these data lead to the coefficients: $a = 0.033, b = -0.0015, c = 3 \times 10^{-5}$ and $d = 2 \times 10^{-7}$.

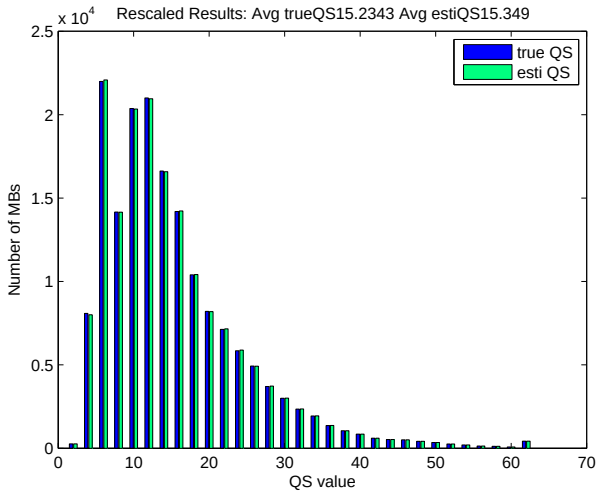

Figure 3: Histogram of estimated and actual Q_S values for the scaled test sequences.

This new method based on mismatch was compared with (our implementation of) I-frame detection based on zero coefficients [5] (Fig. 4). Figures 4 and 5 show the I-frame estimation results for the decoded 4 Mbit/s sequences, which constitute the last third of the concatenated sequence. I-frames are marked by bars (*) and the other frames marked by +. The frames, which are assigned a wrong frame type are marked by squares □. On Fig. 5, there is a very clear separation of I-frames, which is not the case for zero coefficient I-frame detection (Fig. 4). The accuracy over all 18 sequences is 99.17% for zero coefficient analysis, while using the mismatch measure allows a 100% correct segmentation just based on frame based decisions. These results also indicate that the mismatch measure indeed gives a validation of the analysis performed. Considering the complexity, both methods use an 8×8 DCT. A simple implementation of the steps in our algorithm after the DCT (including Q_S estimation) showed a complexity comparable to that of an efficient 2D-DCT implementation [31]. If only I-frame detection is required, the very good separation based on the mismatch measure suggests that it should be sufficient to process a smaller portion of blocks for faster processing.

5.3. Results on PSNR estimation of decoded I-frames

The estimation of PSNR (Section 3) was applied to the I-frames of the 18 decoded test sequences see Tables 1-2 (CITY2 refers to CITY at 2M.). Based on the Q_S estimation introduced in Section 2.3, the PSNR estimates (EQ_S) were calculated for three of the methods introduced in Section 3 (Table 1).

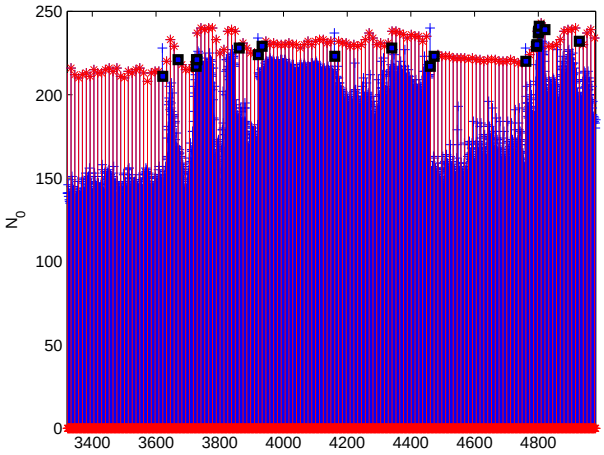

Figure 4: Number of zero coefficients, N_0, per frame for the six 4 Mbit/s sequences. I-frames are marked $*$, P- and B-frames $+$ and erroneous decisions $\Box$.

All three methods, M1-M3, calculate the distortion using (12) but differ in the calculation of λ. Method M1 applies $\lambda(Q_S, i)$ estimation based on the number of zero coefficients (13). Method M3 initially calculates $\lambda(Q_S, i)$ using zero coefficients but thereafter averages over the Q_S values for each frequency to obtain $\lambda_{avg}(i)$. Method M2 also applies averaging, but over initial estimates using second moment statistics [8] for each Q_S value.

In Table 1, I-frame PSNR estimation results based on the actual MPEG quantization step size values (MQ_S) are also given for comparison. The average is calculated over all the I-frames for each of the 18 decoded sequences. Estimation error values are also given, measured by the average absolute difference (in dB) per I-frame (adf) for each sequence. The main conclusion is that virtually the same results are obtained using the estimated and the actual values of Q_S for the estimators tested. For all three methods, the average differences are for most sequences less than 0.1 db and the average absolute differences over the sequences range from 0.00 to 0.03 dB.

Table 2 summarizes the sequence mean average error values and standard deviation for methods M1-M4 for estimated Q_S (EQ_S). The results obtained by the method of [4] are given for comparison. Here $\lambda(Q_S, i)$ is calculated based directly on the standard deviation of the (reconstructed) DCT values for each Q_S and frequency, i. Finally these λ values were averaged to also obtain λ_{avg}. For the last two methods the MPEG stream values were used (MQ_S). Method M1 and M3 give the best results, so using λ estimation based

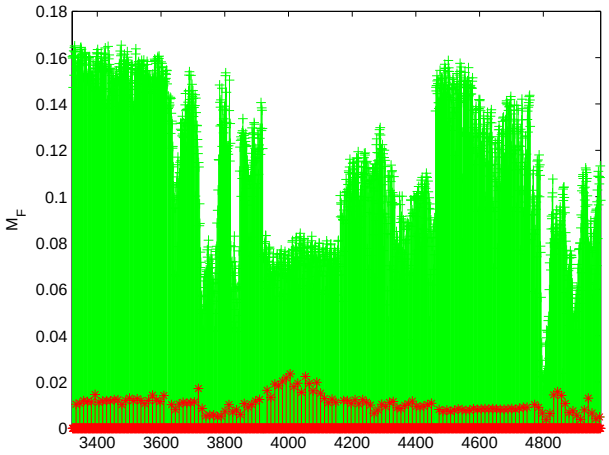

Figure 5: Mismatch measure, M_F, per frame for the six 4 Mbit/s sequences. I-frames are marked $*$, P- and B-frames $+$.

on zero coefficients (13) is best in this test. M1 and M3 give very similar results and the results are also similar to those presented in [20] based on zero coefficient λ estimation for I-frames of H.264 coded 720p50 HDTV. The method of [4] (SD) follows with an average error of 0.633 dB. The results are comparable to the I-frame results for SDTV (avg. error -0.718 dB, std. dev. 0.747) reported in [4] using the variance based λ estimate. Modifying this by averaging λ (SD λ_{avg}) did not improve the PSNR estimates. Using the second moment technique of [8] provides the least precise results with a high standard deviation and average errors of -0.982 dB and -1.180 dB, respectively.

There could be various ways to improve the results, e.g. as M1/M3 have negative average errors and the method of [4] has a positive average error, these could be weighted to achieve a better result. Also the α value (or other parameters) could be adjusted experimentally as in [20] or more advanced training based methods as in [22] be considered. We note that the simple zero coefficient based estimates provided good results. The average errors in the order of 0.3 dB for I-frames compares favorable e.g. with results in [4].

The PSNR estimation based on the estimated Q_S values was also tested for the I-frames of the decoded SD sequences scaled to HD. These scaled sequences were considered as the input to the analysis and as the first step analyzed by the block size detection algorithm and thereafter scaled back to SD. Results for the shorter test set (12 sequences of 100 frames) are given

in Table 3 for M1-M3 in form of average values at 2M, 3M and 4Mbits/s. Again a good match is achieved. Here Method M3 is best followed by M1. The per frame average absolute errors are 0.56 dB and 0.78 dB, respectively. A bit surprising, M2 actually performs better than on the sequences without scaling, 1.03 dB per frame error and only 0.07 dB deviation over the test set. The rescaling does lead to estimation errors in the Q_S estimation and discrepancies in performance using the original (or SDTV estimated) values, but still the resulting PSNR estimates are close to the actual PSNR values for I-frames.

Table 1: I-frame Q_S and PSNR estimation results for M1, M2 and M3 (Full length sequences).

	MQ_S	EQ_S	PSNR	M1 (NZ,$\lambda(Q_S,i)$)			M2 (2nd,λ_{avg})			M3 (NZ,λ_{avg})		
				$PSNR$ MQ_S	$PSNR$ (EQ_S)	adf (EQ_S)	$PSNR$ (MQ_S)	$PSNR$ (EQ_S)	adf (EQ_S)	$PSNR$ (MQ_S)	$PSNR$ (EQ_S)	adf (EQ_S)
CITY2	23.40	23.40	32.20	32.26	32.27	0.10	31.98	31.98	0.25	32.33	32.32	0.14
SOC2	46.31	46.03	32.14	31.67	31.71	0.81	28.42	28.45	3.75	31.52	31.50	1.01
ICE2	12.18	12.16	39.27	37.48	37.48	1.79	37.87	37.87	1.40	37.87	37.92	1.35
CREW2	29.83	30.40	34.91	34.48	34.39	0.52	31.59	31.56	3.35	33.63	33.56	1.35
HAR2	35.76	35.78	30.80	31.43	31.43	0.64	29.29	29.29	1.51	31.10	31.10	0.30
FRI2	47.80	46.90	32.69	30.72	30.77	1.95	27.75	27.79	4.93	30.16	30.09	2.63
Average 2M	**32.55**	**32.44**	**33.67**	**33.01**	**33.01**	**0.97**	**31.15**	**31.16**	**2.53**	**32.77**	**32.75**	**1.13**
CITY3	16.07	16.07	34.29	34.27	34.27	0.08	35.04	35.04	0.75	34.45	34.45	0.16
SOC3	20.69	20.69	34.90	35.31	35.31	0.49	33.97	33.98	1.40	35.18	35.17	0.53
ICE3	7.96	8.03	41.05	39.68	39.64	1.41	41.21	41.21	0.45	40.41	40.40	0.65
CREW3	15.34	15.37	37.41	37.13	37.13	0.30	36.14	36.14	1.27	37.26	37.25	0.29
HAR3	23.35	23.35	33.06	33.66	33.66	0.60	32.40	32.40	0.66	33.44	33.44	0.37
FRI3	22.14	22.55	35.32	34.91	34.89	0.62	33.68	33.68	2.01	34.81	34.74	0.83
Average 3M	**17.59**	**17.68**	**36.01**	**35.83**	**35.82**	**0.58**	**35.41**	**35.41**	**1.09**	**35.92**	**35.91**	**0.47**
CITY4	12.76	12.76	35.64	35.61	35.61	0.09	36.96	36.96	1.32	35.84	35.84	0.20
SOC4	14.82	14.79	36.55	36.80	36.81	0.37	36.39	36.40	1.15	36.84	36.84	0.42
ICE4	6.18	6.58	42.21	41.15	40.82	1.39	43.31	43.28	1.08	41.93	41.78	0.43
CREW4	11.43	11.43	38.67	38.49	38.49	0.20	38.39	38.39	0.46	38.80	38.80	0.21
HAR4	17.69	17.69	34.59	35.16	35.16	0.57	34.55	34.55	0.12	35.00	35.00	0.41
FRI4	14.06	14.15	36.98	36.93	36.91	0.38	36.82	36.82	1.03	37.15	37.13	0.40
Average 4M	**12.82**	**12.90**	**37.44**	**37.35**	**37.30**	**0.50**	**37.74**	**37.73**	**0.86**	**37.59**	**37.57**	**0.35**
Average all	**20.99**	**21.01**	**35.70**	**35.40**	**35.37**	**0.68**	**34.76**	**34.77**	**1.49**	**35.43**	**35.41**	**0.65**

Table 2: Evaluation of PSNR estimation for M1-M4 (EQ_S) and λ estimation by standard deviation (SD) (MQ_S) using [4] and λ_{avg}. (Same test data as Table 1)

	M1	M2	M3	M4	SD [4]	SD λ_{avg}
Average estimation error	-0.331	-0.939	-0.298	-1.180	0.633	0.895
Standard deviation of error	0.850	3.906	0.950	3.865	0.936	0.858

Table 3: I-frame PSNR estimation results for rescaled sequences (100 frames).

	MQ_S	EQ_S	PSNR	M1			M2			M3		
				$PSNR$ (MQ_S)	$PSNR$ (EQ_S)	*adf* (EQ_S)	$PSNR$ (MQ_S)	$PSNR$ (EQ_S)	*adf* (EQ_S)	$PSNR$ (MQ_S)	$PSNR$ (EQ_S)	*adf* (EQ_S)
Average 2M	21.61	21.44	34.79	34.07	34.23	0.92	33.48	33.50	1.33	34.08	34.02	0.92
Average 3M	13.71	13.88	36.88	36.35	36.31	0.75	37.03	37.03	0.72	36.61	36.56	0.46
Average 4M	10.68	11.03	38.14	37.75	37.60	0.68	39.08	39.06	1.03	38.12	38.03	0.31
Average all	**15.33**	**15.45**	**36.60**	**36.06**	**36.04**	**0.78**	**36.53**	**36.53**	**1.03**	**36.27**	**36.20**	**0.56**

Thus, the main conclusion of these experiments is that the differences in estimates based on the estimated and the actual Q_S estimates are very small. It is also noted that the best results are achieved using the number of zero coefficients for estimating λ. As we shall see, the estimated values of Q_S on I-frames may also be useful when (post-)processing the rest of the GOP.

5.4. Post-processing results

The deblocking and deringing filters presented in Section 4 based on estimated Q_S values were applied to the full test set (Table 4). The decoded test set covers a wide range of PSNR values ranging from 30 dB to 41.5 dB. Motivated by the observation that large flat panel displays seem to magnify coding artifacts, we focus on video of good quality. The quantization parameters were estimated on I-frames and thereafter the average estimated Q_S for each I-frame was used for the rest of the GOP (EQ_S). A simple optimization of the deringing filter was introduced by the mapping from Q_S to Q'_S in the threshold of the deringing filter (16). The sequences were also visually inspected on a 50" inch plasma screen.

Comparing the overall average (Avg) of the combined filter with deblocking and deringing, EQ_S, with that of the directly decoded (Dec), an overall average improvement of 0.29 dB was measured on the test set for the pure post-processing (Table 4). Furthermore it may be noted that the performance is robust in the sense that PSNR improvements were achieved on all sequences also when evaluated by picture type. To evaluate the effect of adaptively estimating the Q_S value, the pure post-processing was also executed with one fixed value for Q_S selected as the average Q_S over all sequences ($Q_S = 21$) (denoted Fixed in Table 4). Also here replacing Q_S with Q'_S gave an improvement in the deringing (16). In this case, the average improvement was 0.16 dB, i.e. only half of the improvement (measured in dB) of the adaptive version. Furthermore, the PSNR improvement was not robust when using the fixed Q_S.

The combined (com) postprocessing scheme presented was also tested using the MPEG stream (MQ_S) Q_S instead of estimated values for I-frames and the average of these for B- and P-frames. The results are virtually identical to those obtained with estimated Q_S (EQ_S) (Table 4).

Table 4: Post-processing results (PSNR) for deblocking (db), deringing (dr) and combined (com).

	Dec	EQ_S			MQ_S	Fixed
		db	dr	com	com	com
CITY2	32.36	*32.42*	32.34	32.37	32.37	32.34
SOC2	31.71	*31.97*	31.79	31.95	31.95	31.93
ICE2	38.61	38.85	38.78	*38.95*	*38.95*	38.91
CREW2	33.79	34.15	33.90	*34.16*	*34.16*	34.15
HAR2	30.01	*30.31*	30.00	30.24	30.24	30.19
FRI2	31.61	32.14	31.79	*32.22*	*32.22*	32.13
Average 2M	**33.02**	**33.31**	**33.10**	**33.32**	**33.32**	**33.28**
CITY3	34.07	*34.13*	34.06	34.10	34.10	33.97
SOC3	34.00	*34.25*	34.06	34.22	34.22	34.15
ICE3	40.39	40.60	40.55	*40.68*	40.67	40.38
CREW3	35.96	36.32	36.07	*36.34*	*36.34*	36.29
HAR3	31.96	*32.31*	31.94	32.23	32.23	32.11
FRI3	33.97	34.43	34.11	*34.49*	*34.49*	34.47
Average 3M	**35.06**	**35.34**	**35.13**	**35.34**	**35.34**	**35.23**
CITY4	35.16	*35.22*	35.16	35.20	35.20	34.98
SOC4	35.47	*35.74*	35.52	35.72	35.71	35.52
ICE4	41.48	41.68	41.58	*41.72*	*41.72*	41.16
CREW4	37.13	37.48	37.22	*37.50*	*37.50*	37.38
HAR4	33.33	*33.69*	33.32	33.62	33.62	33.41
FRI4	35.64	36.03	35.76	*36.08*	*36.08*	36.05
Average 4M	**36.37**	**36.64**	**36.43**	**36.64**	**36.64**	**36.42**
Average all	**34.81**	**35.10**	**34.89**	**35.10**	**35.10**	**34.97**

Table 5: Average post-processing improvement in PSNR (dB) by frame type.

Frame	EQ_S Post-proc.	Kim [2] scaled	KSK [3] estimated Q_S
I	0.41	0.02	0.21
P	0.41	0.02	0.37
B	0.22	0.01	0.14
All	0.29	0.01	0.20

Our post-processing approach has been inspired by post-processing filters reading the quantization parameter values from the code stream, e.g. [2, 3, 26, 27]. We compared the performance with some of these. The deblocking and deringing filters [26, 27] in MPEG-4 part2 Momusys were embedded in an MPEG-2 decoder, i.e. reading the quantization parameters from the code

stream. Comparing the performance of MPEG-4 part2 and the same post-processing embedded in MPEG-2, we got similar relative PSNR performance but the relative results were slightly worse at high bitrates when embedded in MPEG-2. On the full test set, embedding the post-processing in MPEG-2 lead to an improvement of 0.08 dB for deblocking and a decrease both for deringing and combined.

Two additional embedded schemes [2, 3] using MPEG stream Q_S values were also implemented and tested both using MPEG stream values and modified to use estimated values EQ_S. Table 5 reports performance by frame type. The deblocking and deringing methods introduced in [2] for low-bit rate H.263 both use quantization values and some values of the decoded AC coefficients. We implemented the filters using (reconstructed) AC values and MPEG stream quantization values. The direct implementation reduced PSNR, but did also reduce blocking and ringing visually. The filters were thereafter modified and optimized by scaling the Q_S values. Using a scaling factor of 0.3, it was possible to get a marginal gain overall of 0.01-0.02 dB on the full test set (Table 5).

The deblocking method introduced in [3] mainly adapts to the quantization parameter values, but it also performs minor adaption due to information of motion vectors and skip blocks. The motion vectors are used to define three classes of motion activity. The method was developed for MPEG-2 sequences and improvements of 0.2-0.4 dB were reported for I- and P-frames and 0.1-0.3 dB for B-frames [3]. We implemented a simplified version fixing the motion level to the highest level of motion. Using MPEG stream Q_S values in this version gave good results on I and P, but no improvement on B-frames. Changing to estimated Q_S values at MB level on I-frames and using an average of these for subsequent P- and B-frames actually gave an improvement and a good performance (Table 5) on all frame types in line with that of [3] leading to an overall improvement of average PSNR of 0.20 dB. In comparison, deblocking as well as deblocking combined with deringing gave an improvement of 0.29 dB using our pure post-processing. The good performance indicates that while the MB level quantization parameters are important for I-frames, they play a smaller role on P- and B-frames, where it is a combination of the reference frame quality and quantization of residuals, which determines the quality.

Visual inspection of the test images on the 50" plasma screen, as well as a simple pair comparison test, confirmed the improvement expressed by PSNR. All three methods provide an overall visual improvement in terms of reducing

blocking and ringing. A simplified version of a Pair Comparison (PC) test [34] was conducted on the sequences CREW, ICE and CITY at 2M and 3M with 16 subjects evaluating the four test clips from the three postprocessing methods and the decoded MPEG. The clips were paired in all six combinations, displayed side-by-side and played back simultaneously and compared on an ordinal scale {0, 1, 2}, scoring 2 points for the preferred sequence or 1 point each if they were judged to be of equal quality. Our method (EQ_S) was preferred in the pair comparison test overall with a mean score of 1.25 (out of 2) and it also scored best for each of the three test sequences. Both blocking and deringing are suppressed and only slight smoothing is observed. Compared with [2] better blocking, deringing and combined effects are obtained by our method. Also more details are preserved. Both our method (EQ_S) and the deblocking of [3] do a good job at attenuating the artifacts with a minor sacrifice of sharpness. Comparing the two closely, our assessment is that visually EQ_S provides slightly better deblocking, though often the impression of deblocking is similar. The results are illustrated in Fig. 6. Here a small part of a P-frame is depicted as post-processed by EQ_S and for comparison as post-processed by the embedded quantizer based version of [3], as well as the MPEG-2 decoded and the original versions. This visual impression is in line with the general assessment above. Our EQ_S version attenuates ringing artifacts notable along the black line in the center of the image and the overall sharpness is slightly better than that of [3].

6. Conclusions

Analysis and post-processing of block-based DCT decoded images, without code stream access, was presented. For the general case of MPEG-2 coding with variable quantization, the I-frame quantization parameters were estimated at macroblock (MB) level with an overall accuracy of 99.9 %. As an integral part of the analysis, I-frames were detected and validated, thus validating that the decoded image indeed came from an MPEG-2 stream and not e.g. an H.264 stream. Based on the analysis, the PSNR of MPEG-2 I-frames was estimated and adaptive deringing and deblocking performed. The average estimated I-frame quantization value, Q_S, was also used to guide the post-processing of the P- and B-frames for the rest of the GOP for processing the decoded video without access to the MPEG coded stream. Using the estimated Q_S leads to virtually the same results as reading the Q_S from the I-frame code stream. On the MPEG-2 decoded test set, an average im-

Figure 6: Part of P-frame from test sequence Crew. Upper left) Original. Upper right) MPEG-2 decoded (36.02 dB). Lower left) Deblocking and deringing using estimated Q_S values (EQ_S) (36.38 dB). Lower right) Embedded deblocking using code stream Q_S values (36.45 dB).

provement of 0.29 dB was achieved by the post-processing. Blocking and ringing artifacts are efficiently attenuated with no or very limited reduction in sharpness.

Role of funding. This work was supported in part by the Danish Strategic Research Council under the NABIIT programme.

Acknowledgement. We would like to thank Bang & Olufsen, especially Jesper Meldgaard Pedersen for fruitful discussions and collaboration.

References

[1] M.-Y. Shen and C.-C. J. Kuo, "Review of postprocessing techniques for compression artifact removal," *Journal Visual Communication Image*

Representation, vol. 9, pp. 1–14, Mar. 1998.

[2] C. Kim, "Adaptive post-filtering for reducing blocking and ringing artifacts in low bit-rate video coding," *Signal Processing: Image Communication.*, vol. 17, pp. 525–535, 2002.

[3] Do-Kyoung Kwon, M.-Y. Shen and C.-C. J. Kuo, "An improved adaptive deblocking filter for MPEG video decoder," *Image Video Communications Processing.*, pp. 702–712, 2005.

[4] A. Ichigaya, M. Kurozumi, N. Hara, Y. Nishida, and E. Nakasu, "A method of estimating coding PSNR using quantized DCT coefficients," *IEEE Trans. Circ. Syst. Video Tech.*, pp. 251–259, Feb. 2006.

[5] M.J.Knee and I.J.Poole, "Analysis of compression decoded video image sequences," *US Patent 006895049B1*, 2005.

[6] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, "Adaptive deblocking filter ," *IEEE Trans. Circ. Syst. Video Tech.*, pp. 614–619, July 2003.

[7] T. Wolff, H-H. Ho, J. M. Foley, and S. K. Mitra, "Modeling subjectively perceived annoyance of H.264/AVC video as a function of perceived artifact strength," *Signal Processing: Image Communication.*, vol. 90, pp. 80–92, 2010.

[8] D. S. Turaga, Y. Chen, and J. Caviedes, "No reference PSNR estimation for compressed pictures," *Signal Processing: Image Communication*, vol. 19, pp. 173–184, 2004.

[9] R. Rosenholtz and A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circ. Syst. Video Tech.*, vol. 2, pp. 91–95, Mar. 1992.

[10] S. H. Oguz, Y. H. Hu, and T. Q. Nguyen, "Image coding ringing artifact reduction using morphological post-filtering," *IEEE Second Workshop Multimedia Signal Processing*, pp. 628–633, 1998.

[11] B. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Multiframe blocking-artifact reduction for transform-coded video," *IEEE Trans. Circ. Syst. Video Tech.*, pp. 276–282, Apr. 2002.

[12] B. Martins and S. Forchhammer, "A unified approach to restoration, deinterlacing, and resolution enhancement in decoding MPEG-2 video," *IEEE Trans. Circ. Syst. Video Tech.*, pp. 803–811, Sept. 2002.

[13] G. Wang, T-T. Wong, and P-A. Heng, "Deringing cartoons by image analogies," *ACM Trans. Graphics*, vol. 25, pp. 1360–1379, 2006.

[14] Y-Y. Chen, Y-W. Chang, and W-C. Yen, "Design a deblocking filter with three separate modes in DCT-based coding," *Journal Visual Communication Image Representation*, vol. 19, pp. 231–244, 2008.

[15] D. T. Võ, T. Q. Nguyen, S. Yea, and A. Vetro, "Adaptive fuzzy filtering for artifact reduction in compressed images and video," *IEEE Trans. Image Proc.*, pp. 1166–1178, June 2009.

[16] "MPEG-2 Video Standard, Information Technology-Generic Coding of Moving Pictures and Associated Audio information, ISO/IEC 13818-2," 2000.

[17] W. Fischer, *Digital Television, a Practical Guide for Engineers*, Springer-Verlag, Berlin, 2004.

[18] H. Li, *MPEG Decoded Video Analysis and Postprocessing*, PhD. diss. Technical University of Denmark, 2009.

[19] T. Brandao and M. P. Queluz, "Blind PSNR estimation of video sequences using quantized DCT coefficient data," *Picture Coding Symp., Lisbon, Portugal*, 2007.

[20] A. Eden, "No-reference estimation of the coding PSNR for H.264-coded sequences," *IEEE Trans. Consumer Elec.*, vol. 53, pp. 667–674, May 2007.

[21] Z. He and S. Mitra, "A unified rate-distortion analysis framework for transform coding," *IEEE Trans. Circ. Syst. Video Tech.*, pp. 1221–1236, Dec. 2001.

[22] T. Brandao and M. P. Queluz, "No-reference PSNR estimation algorithm for H.264 encoded video sequences," *16th European Sign. Proc. Conf. (EUSIPCO 2008), Lausanne, Switzerland*, Aug. 2008.

[23] S. Liu, and A. C. Bovik, "Efficient DCT-domain blind measurement and reduction of blocking artifact," *IEEE Trans. Image Proc.*, pp. 1139–1149, Dec. 2002.

[24] A. Z. Averbuch, A. S. Schlar, and D. L. Donoho "Deblocking of block-transformed compressed images using weighted sums of symmetrically aligned pixels," *IEEE Trans. Image Proc.*, pp. 200–212, Feb. 2005.

[25] K. Virk, H. Li, and S. Forchhammer, "Reduced complexity MPEG-2 video post-processing for HD display," *IEEE Int'l Conf Multimedia Expo*, pp. 769–772, 2008.

[26] "MPEG-4 Video Standard, Part 2, Information Technology - Coding of Audio-Visual Objects - Part2: Visual," 1999.

[27] "MPEG-4 Part 2 software, MoMuSys-FDIS-VI.0-990812 reference software".

[28] S. D. Kim, J. Yi, H. M Kim, and, J. B. Ra, "A deblocking filter with two separate modes in block-based video coding," *IEEE Trans. Circ. Syst. Video Tech.*, vol. 9, pp. 156–160, 1999.

[29] S-C. Tai, Y-Y. Chen, and S-F. Sheu, "Deblocking filter for low bit rate MPEG-4 video," *IEEE Trans. Circ. Syst. Video Tech.*, vol. 15, pp. 733–741, 2005.

[30] W. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision, 2nd Ed.*, PWS Publishing, 1998.

[31] M. Petricca, H. Li, S. Forchhammer, A. Nannarelli, M. Re, Marco, J. D. Andersen, and G. C. Cardarilli, "Hardware implementation of real-time MPEG analysis and deblocking for video enhancement," *Proc. of 43rd Asilomar Conf. Signals, Systems, Computers*, 2009.

[32] D. Sannino, *Design of a FPGA based HDTV postprocessor*, M. Sc. Thesis, Technical University of Denmark, 2010.

[33] www.mpeg.org/MPEG/MSSG.

[34] ITU-T Rec. P.910 "Subjective video quality assessment methods for multimedia applications, " Geneva, Switzerland, 2008.