Signal Processing: *Image Communication* 27 (2012) 985–997

Contents lists available at SciVerse ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

A generic post-deblocking filter for block based image compression algorithms

Nelson C. Francisco [a,b,c,*], Nuno M.M. Rodrigues [a,b], Eduardo A.B. da Silva [c,d], Sérgio M.M. de Faria [a,b]

[a] *Instituto de Telecomunicações, Apartado 4163, 2411-901 Leiria, Portugal*
[b] *ESTG, Inst. Pol. Leiria, Apartado 4163, 2411-901 Leiria, Portugal*
[c] *PEE/COPPE, Univ. Fed. Rio de Janeiro, CP 68504, 21945-970 RJ, Brazil*
[d] *DEL/Poli, Univ. Fed. Rio de Janeiro, CP 68504, 21945-970 RJ, Brazil*

ARTICLE INFO

Article history:
Received 23 November 2011
Accepted 22 May 2012
Available online 1 June 2012

Keywords:
Deblocking filter
Total variation
Image coding
Pattern matching

ABSTRACT

In this paper we propose a new post-processing deblocking technique that is independent of the compression method used to encode the image. The development of this filter was motivated by the use of Multidimensional Multiscale Parser (MMP) algorithm, a generic lossy and lossless compression method. Since it employs an adaptive block size, it presents some impairments when using the deblocking techniques presented in the literature. This led us to the development of a new and more generic deblocking method, based on total variation and adaptive bilateral filtering.

The proposed method was evaluated not only for still images, but also for video sequences, encoded using pattern matching and transform based compression methods. For all cases, both the objective and subjective quality of the reconstructed images were improved, showing the versatility of the proposed technique.

1. Introduction

Block-based coding algorithms are widely used for image and video compression. They partition the input image into fixed size blocks that are sequentially processed, using transform coding, quadtree decomposition, vector quantization or other compression techniques. Several standards, from JPEG [1] to H.264/AVC [2], are examples of block-based encoders.

Despite the high compression efficiency achieved by some of these algorithms, the visual quality of the compressed images are often affected by blocking artifacts, resulting from the discontinuities induced in the block boundaries, specially at high compression ratios. Several approaches have been proposed in the literature in order to attenuate these artifacts, such as adaptive spatial filtering [3,4], wavelet-based filtering [5], transform-domain methods [6,7] or interactive methods [8], just to name a few.

Some of these deblocking techniques have been developed to work as in loop-filters, such as [9] the deblocking filter adopted by the standard H.264/AVC [2]. However, the use of loop filters requires that every compliant decoder must replicate the filtering procedure, in order to stay synchronized with the encoder. This can be an inconvenient, as a decoder would loose the flexibility to switch off the deblocking filter in order to trade-off visual quality for computational complexity. Post-deblocking methods have been proposed to overcome this drawback [10,11]. In this case, the filtering procedure is only performed after the decoding process is finished, thus not interfering with the encoder/decoder synchronization.

* Corresponding author at: Instituto de Telecomunicações, Apartado 4163, 2411-901 Leiria, Portugal. Tel.: +351 244820300; fax: +351 244820310.
E-mail addresses: ncarreira@lps.ufrj.br (N.C. Francisco), nuno.rodrigues@co.it.pt (N.M.M. Rodrigues), eduardo@lps.ufrj.br (E.A.B. da Silva), sergio.faria@co.it.pt (S.M.M. de Faria).

0923-5965/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.image.2012.05.005

Traditionally, these post-processing strategies tend to be less efficient than the in-loop filters, as they are not able to exploit all the information available in both the encoding and decoding process that helps to locate blocking artifacts and avoid filtering unwanted regions.

For the upcoming HEVC coding standard [12], a new filter architecture [13] was proposed, combining an in-loop deblocking filter and a post-processing Wiener filter. The in-loop filter reduces the blocking artifacts, while the Wiener filter is a well-known linear filter that can guarantee the objective quality optimized restoration in images degraded during the compression process by Gaussian noise, blurring or distortion. A unified architecture for both filters is proposed in [13], but it results once more in an in-loop filter that, despite its high efficiency, is still not able to present the advantages of post-deblocking methods.

In this paper we propose a new versatile post-deblocking filter that is able to achieve a performance comparable to the ones of state-of-the-art in-loop methods when used in still images and video sequences compressed with several block-based compression algorithms, such as the Multidimensional Multiscale Parser (MMP) algorithm [14], H.264/AVC [2], JPEG [1] and HEVC [12].

This paper is organized as follows: in Section 2 we present some related work that motivated the development of the proposed method; Section 3 describes the new algorithm used for mapping and classifying the blocking artifacts, as well as the adaptive filter used to reduce those artifacts. Experimental results are shown in Section 4, and Section 5 concludes the paper.

2. Related work

The development of the proposed post-processing deblocking algorithm was motivated by the use of a generic lossy pattern matching data compression method, known as the Multidimensional Multiscale Parser (MMP) algorithm [14]. MMP has been successfully used for lossy and lossless compression of several types of data sources, including still images [15] and video sequences [16]. It is based on the approximation of input data segments of variable dimensions using codevectors stored in an adaptive multiscale dictionary. As the dictionary is updated with previously compressed patterns, MMP can be seen as a combination of the Lempel–Ziv methods (LZ) [17–19] and vector quantization (VQ) [20]. Furthermore, it allows scale adaptive pattern matching, a feature that distinguishes it from the previous algorithms, and delivers a high degree of adaptability.

The use of variable sized patterns in MMP restricts the use of most existing deblocking methods, as they were developed to be employed with fixed size block-based algorithms, such as JPEG [1] and H.264/AVC [2]. In such cases, the location of the blocking artifacts is highly correlated with the border areas of the transformed blocks, and consequently depends mostly on the block dimensions. This extra information is exploited by some deblocking methods, such as the ones in [9,11,21]. It is usually employed to help classifiers to locate regions that need to be deblocked, avoiding the risk of filtering unwanted regions. Unlike these algorithms, the multiscale matching used in MMP may introduce artifacts at any location along the reconstructed image.

A similar situation occurs on motion compensated frames from encoded video sequences. Although the location of blocking artifacts on the Intra coded frames is predictable, motion compensation may replicate these artifacts to any location on an Inter-coded frame if no deblocking method is applied before performing the motion compensation. As a result, post-processing deblocking methods for Inter-coded frames should be able of efficiently locate blocking artifacts away from block boundaries. The method proposed in [10] addresses this issue by using the motion vector information in order to identify the regions of the motion compensated frames that used possibly blocked locations of the reference frames. Therefore, it cannot be considered a pure post-deblocking technique, as in addition to the decoded video, it needs information provided by the decoded bitstream. As a consequence, this technique is specific for H.264/AVC, and will not work for other algorithms that use a different encoding scheme.

In [22], a bilateral adaptive filter was proposed for the MMP algorithm that achieved satisfactory results when used in a non-predictive coding scheme. However, that method showed considerable limitations when used with predictive MMP-based algorithms, that present state-of-the-art results for natural image coding [23]. Additionally, this method is also algorithm specific, since it needs information present on an MMP bitstream.

Based on the above, we see that both block-based video encoders and MMP with a predictive scheme would benefit from a versatile post-deblocking method. In the sequel we describe such method.

3. The deblocking filter

In this section, we describe the proposed deblocking method. It is based on the use of a space-variant finite impulse response (FIR) filter, with an adaptive number of coefficients. Prior to the filtering stage, the input image is analyzed, in order to define the strength of the filter that should be used in each region. This results in a filtering map, that indicates the length of the filter's support that will be applied to each pixel or block in the image. Higher activity regions will have a shorter support length associated, while smooth areas will use a longer filter support.

3.1. Adaptive deblocking filtering for MMP

The RD control algorithm used in MMP only considers the distortion and the rate of the encoded data, without taking into account the block borders' continuity, which is the main source of blocking artifacts. As blocks at different scales are concatenated, these artifacts are likely to appear in any location of the reconstructed image, unlike the case of transform based methods, where blocking artifacts only arise in predetermined locations, along a grid defined by the size of the block transform.

Let us define an image reconstructed with MMP, $\hat{\mathbf{X}}$, as

$$\hat{\mathbf{X}}(x,y) = \sum_{k=0}^{K-1} \hat{\mathbf{X}}_k^{l_k}(x-x_k, y-y_k), \quad (1)$$

i.e., the concatenation of K non-overlapping blocks of scale l_k, $\hat{\mathbf{X}}_k^{l_k}$, each one located on position (x_k,y_k) of the reconstructed image. One can notice that each block $\hat{\mathbf{X}}_k^{l_k}$ also results from previous concatenations of J other elementary blocks, through the dictionary update process. Defining these elementary blocks as $\mathcal{D}_0{}_j^{l_j}$, where l_j represents the original scale and (u_j,v_j) represent the position of the elementary block inside $\hat{\mathbf{X}}_k^{l_k}$, we obtain

$$\hat{\mathbf{X}}_k^{l_k}(x,y) = \sum_{j=0}^{J-1} \mathcal{D}_0{}_j^{l_j}(x-u_j, y-v_j). \quad (2)$$

In this equation, one may identify the border regions of each pair of adjacent basic blocks, that correspond to the most probable location for discontinuities in the decoded image, that may introduce blocking artifacts.

In [22], a deblocking method was proposed for MMP, that stores the information regarding the original scale of each elementary block that composes each codeword, in order to locate all the existing boundaries. These boundaries correspond to the regions where the deblocking artifacts are most likely to appear, and this information is used to generate a map that defines the length of the filter's support for each region. This is done by imposing that blocks, $\mathcal{D}_0{}_j^{l_j}$, at larger scales, that generally correspond to smoother areas of the image, should be filtered more aggressively, while blocks with small values of the scale l_j, corresponding to more detailed image areas, should not be subjected to strong filtering, in order to preserve its details.

A space-variant filter is then used for the deblocking process. The control of the support's length adjusts the filter's strength, according to the detail level of the region being deblocked. This avoids the appearance of blurring artifacts, that are frequently caused by the use of deblocking techniques. Fig. 1 presents a one-dimensional representation of a reconstructed portion of the image, resulting from the concatenation of three basic blocks, $(\mathcal{D}_0^{l_0}\ \mathcal{D}_1^{l_1}\ \mathcal{D}_2^{l_2})$, each from a different scale: l_0, l_1 and l_2, respectively. At each filtered pixel, represented in the figure by a vertical arrow, the kernel support of the deblocking filter is set according to the scale l_k used for its representation.

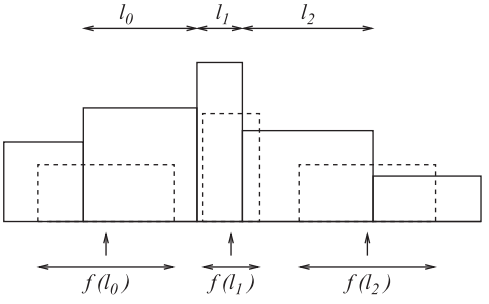

Fig. 1. The deblocking process employs an adaptive support for the FIR filters used in the deblocking.

This method proved to be successful in several cases. Nevertheless, some problems were observed when it was used in a predictive-based coding scheme. Accurate predictions result in low energy residues even for regions presenting high activity, that tend to be efficiently coded using blocks from larger scales. As a result, some highly detailed regions would be improperly considered as smooth and filtered with a large aggressive filter. This may introduce a considerable degradation on the image's detail, forcing the decrease of the overall strength of the filter (one single strength is used for all the image), and thus, limiting its effectiveness. Also, the tracking of the information about the original scale of the basic units that compose each codeword is also a cumbersome task. Furthermore, perhaps the most important disadvantage of this method is that it is only appropriate for the MMP algorithm, since it needs segmentation information obtainable from the MMP decoding process.

3.2. Generalization to other image encoders

In order to overcome the limitations described for the method from [22], we propose a new mapping approach based on a total variation analysis of the reconstructed image. The new mapping procedure starts by considering that the image was initially segmented into blocks of $N \times M$ pixels. For each block, the total variation of each of its rows and columns is determined respectively by

$$\mathcal{A}_j^v = \sum_{i=1}^{N-1} |\hat{\mathbf{X}}_{(i+1,j)} - \hat{\mathbf{X}}_{(i,j)}|, \quad \mathcal{A}_i^h = \sum_{j=1}^{M-1} |\hat{\mathbf{X}}_{(i,j+1)} - \hat{\mathbf{X}}_{(i,j)}|. \quad (3)$$

Each region is vertically or horizontally segmented if any of the above quantities exceeds a given threshold τ. With this approach, regions presenting a high activity are successively segmented, resulting in small areas that will correspond to narrower filter supports. In contrast, smooth regions will not be segmented, that will correspond to wider filter supports, associated to larger blocks.

It is important to notice that the value of τ has a limited impact on the final performance of the deblocking algorithm. A high value for τ results in fewer segmentations, and consequently, on a larger filter's supports than those obtained using a smaller value for τ. However, these larger supports can be compensated adjusting the shape of the filter, in order to reduce the weight, or even neglect, the impact of distant samples on the filter's support. In other words, the value of τ can be fixed, as this procedure only need to establish a comparative classification of the regions with different variation intensity, with the deblocking strength been controlled through the shape of the filter used.

Fig. 2 shows the filtering map generated for the image Lena coded with MMP at two different bitrates, using $\tau = 32$. Lighter areas correspond to regions that will use larger filter supports, while darker regions correspond to regions that will use narrower filter supports. It is important to notice that not only the proposed algorithm was effective in capturing the image structure for both cases, but also it revealed an intrinsic ability to adapt to the different compression ratios. The map for the image

Fig. 2. Image Lena 512 × 512 coded with MMP at 0.128 bpp (top) and 1.125 bpp (bottom), with the respective generated filter support maps using $\tau = 32$.

coded at a lower bitrate has a lighter tone, that corresponds, on average, to wider supports for the deblocking filter. This is so because as the reconstructions are heavily quantized and larger blocks tend to be used, the sum of the gradient tends to be low in these regions, corresponding to the need for strong filtering.

It is also important to notice that this approach is based on the information present in the reconstructed image only, and is thus independent of the encoding algorithm used to generate it. As a result, the proposed method overcomes the problem of misclassification of well predicted detailed regions when a predictive coding is used.

Furthermore, when applied to MMP, it avoids the need for keeping a record of all original scales of the basic units used for each block performed, as was done in [22], resulting in a more effective and less cumbersome algorithm. Note that one of the advantages of the new scale analysis scheme is that it enables the use of the adaptive deblocking method for images encoded with any algorithm.

3.3. Adapting shape and support for the deblocking kernel

For the algorithms proposed in [22], several kernels with various support lengths were tested. In [22], experimental results showed that Gaussian kernels are effective in increasing the PSNR value of the decoded image, as well as in reducing the blocking artifacts. Thus, Gaussian kernels were also adopted for the proposed method, with the same l_k+1 samples filter length, where l_k refers the segment support. The filter strength is then controlled by adjusting Gaussian's variance, producing filter kernels with different shapes. Considering a Gaussian filter, with variance $\sigma^2 = \alpha L$ and length L, we can express its impulse response (IR) as

$$g_L(n) = e^{-(n-(L-1)/2)^2/2(\alpha L)^2}, \quad (4)$$

with $n = 0, 1, \ldots, L-1$. By varying the parameter α, one adapts the IR of the filter by adjusting the variance of the Gaussian function. The IR of the filter may range from almost rectangular to highly peaked Gaussians, for different lengths. Furthermore, when α tends to zero, the filter's IR tends to a single impulse, and the deblocking effect is switched off for those cases where filtering is not beneficial.

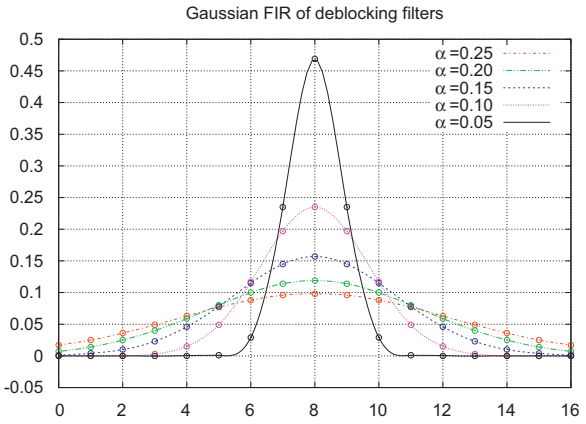

Fig. 3. Adaptive FIR of the filters used in the deblocking.

Fig. 3 represents the shape of a 17 tap filter for the several values of parameter α.

It has been observed that the use of different IR lengths on adjacent blocks at different scales and very different intensity values may create an artifact that was identified in [22]. The referred case is represented in Fig. 4, where a wide dark block *A* is concatenated with two bright blocks: one narrow block *B* followed by one wide block *C*. When blocks *A* and *B* are filtered, a smooth transition appears, that eliminates the blocking effect in the *AB* border. When the block *C* is filtered, the pixels near the *BC* border will suffer from the influence of some of the dark pixels of block A, because the filter has a very wide support region, resulting in the appearance of a dark "valley" on the *BC* border.

This problem was solved by constraining the filter's support to the pixels of the present block and to those from its adjacent neighbors. For the example of Fig. 4, the length of the filter applied to *C* block's pixels that are near

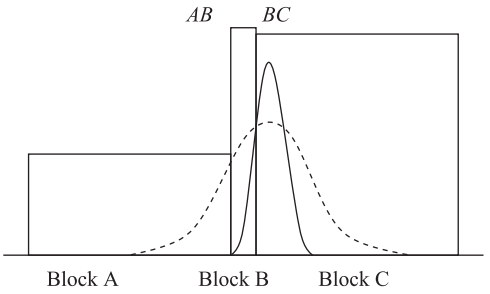

Fig. 4. A case where the concatenation of blocks with different supports and pixel intensities causes the appearance of an image artifact.

to the *BC* border is reduced, so that the left most pixel of the support is always a pixel from block *B*. This corresponds to use the filter represented by the solid line, instead of the original represented by the dashed line.

Additionally, a feature similar to that used by the H.264/AVC adaptive deblocking filter [9] was also adopted, in order to avoid natural edges' filtering. The difference between border pixels from adjacent blocks is monitored, and the filter is switched off every time this difference exceeds a pre-established step intensity threshold *s*.

3.4. Selection of the filtering parameters

The filtering parameters α (Gaussian filter variance), s (step intensity threshold at image edges) and τ (segmentation threshold) must be set in the decoder to perform the deblocking task. In [22], the parameters' values were exhaustively optimized at the encoder side in order to maximize the objective quality, and appended at the end of the encoded bit-stream. This introduced a marginal additional computational complexity and a negligible overhead, but changed the structure of the encoded bit-stream. Consequently, this approach restricts the use of the deblocking process on standard encoders, such as JPEG and H.264/AVC, that have normalized bitstream formats.

In order to address this problem, we developed a version of the proposed deblocking method that avoids the transmission of the filter parameters, by estimating their values at the decoder side. The parameter estimation is supported by the high correlation observed between the amount of blocking artifacts and some of the statistical characteristics presented by the encoded image.

The relation between τ and the shape of the used filter was already mentioned in Section 3.2. For the case of Gaussian kernels, the use of a large value for τ, that results on larger filter supports, can be compensated using a lower value for α. This corresponds to a highly peaked Gaussian that results on a filtering effect similar to the one obtained using a shorter support and a larger value for α. For that reason, the value of τ can be fixed without significant losses on the method's performance, with the deblocking effect being exclusively controlled through the adjustment of the α parameter.

Experimental tests shown that the performance of the algorithm is considerably more affected by α than by the step intensity threshold s. Thus, we started by studying the relationship between the optimal α and the statistical characteristics of the input images. The parameter s would then be responsible for the method's fine tuning.

Fixing the parameters $\tau = 32$ and $s = 100$, a large set of test images was post-processed with the proposed method, using several values of α, in order to determine the one which maximizes the PSNR gain for each case. The test set included a large number of images with different characteristics, compressed at a wide range of compression ratios and using different coding algorithms, such as MMP, H.264/AVC (compressed as still images and video sequences) and JPEG. Thus, it was possible to evaluate the behavior of the proposed method for a wide range of applications.

For each case, several statistical characteristics of each image were simultaneously calculated, in order to determine their correlation with the optimal value of α. This analysis included the average support size that resulted from the mapping procedure, the standard deviation of the distribution of the support lengths, the average variation between neighbor pixels and the standard deviation of the distribution of this variation. We observed that the optimal value of α increases with the average of the filter's support length and decreases with the value of the average variation between neighbor pixels, as expected.

We found the average support length to be a simple and effective estimator for the optimal value for α. Plotting the optimal value of α *vs.* the product of the average support lengths in both the vertical and horizontal directions, we found the equation

$$\alpha = 0.0035 \times v_{\text{size}_{\text{avg}}} \times h_{\text{size}_{\text{avg}}}, \tag{5}$$

where $v_{\text{size}_{\text{avg}}}$ and $h_{\text{size}_{\text{avg}}}$ represent the average length of the filter's support obtained by the mapping procedure, in order to present a good fit to the distribution. Images presenting a low average support length usually have many high frequency details, and cannot be subjected to strong filtering. On the other hand, if the mapping procedure results in a high average support length, the image tends to have a smooth nature, and can be subjected to stronger filtering. We have observed that, in general, the product of the average length on both directions allows to obtain better results than the use of a separate optimization for each direction. The combination of features from both directions makes the algorithm less sensitive to cases where the image characteristics tend to differ from the adopted model. In order to avoid excessive blurring on the reconstructed images, the maximum value of α was limited to $\alpha = 0.21$.

Despite the good results observed for most of the tested images, the model presented some limitations when applied to scanned text images, or, more generally, images presenting a large amount of high frequency details very concentrated in some regions. The short supports associated to detailed regions were in these cases counterbalanced by the long supports from the background regions, and Eq. (5) resulted in too aggressive filtering for these particular cases. Thus, it has been found advantageous to switch off the filter when the standard deviation of the distribution of variations between neighbor pixels (σ_{size_v} and σ_{size_h}) in each direction is high. We found appropriate to

switch the filter off when the product of the standard deviations exceeds the product of the average support lengths by a given threshold, that is, when

$$\frac{\sigma_{\text{size}_v} \times \sigma_{\text{size}_h}}{v_{\text{size}_{\text{avg}}} \times h_{\text{size}_{\text{avg}}}} > 25. \quad (6)$$

In order to preserve natural edges on deblocked images, the value of s is adapted in accordance to the strength α found for the filter. If a large α was adopted, the image needs strong deblocking, and the threshold to switch-off the filter must also be high. If α is low, then the image has a large amount of high frequency details, and the value of s needs to be decreased in order to avoid filtering natural edges. The value of s is related to α through the equation

$$s = 50 + 250\alpha. \quad (7)$$

Using Eqs. (5) and (7), the proposed method is able to adapt the filtering parameters for each frame of a given video sequences only based on its local characteristics.

The impact of the initial block dimensions on the objective quality gains was also evaluated for the proposed method. It can be seen that the use of large blocks does not affect negatively the performance of the algorithm, as these blocks tend to be segmented every time they are not advantageous. However, blocks with more than 16 pixels in length are rarely used, and even when the mapping procedure produces such large blocks, there are no quality gains associated to their use. They only contribute to increasing the overall computational complexity of the algorithm. On the other hand, using small initial blocks restricts the maximum smoothing power achievable by the filter, and consequently the maximum gains that the method can achieve. We verified that 16×16 blocks are the best compromise for most situations, even for images compressed with HEVC using 64×64 blocks. Therefore, we adopted 16×16 blocks as a default parameter for the method, without a significant impact in its performance.

3.5. Computational complexity

The computational complexity of the proposed method depends mainly on two procedures: the creation of the filter map and the deblocking process itself.

The mapping procedure has a very low computational complexity, when compared with the filtering process. It only requires to subtract each pixel to its previous neighbor, accumulate the result, and compare the accumulated value with the threshold τ, in order to decide whether to segment or not the current block. This can be regarded as a linear function $O(n)$, that depends on n, the number of pixels on the entire image. Only integer operations need to be performed.

The filtering process is a direct implementation of bilateral filters, whose computational complexity is also a linear function $O(n \cdot m)$, that depends from n, the number of pixels in the image, and m, the number of pixels used in the filter support, as the deblocked image is obtained through the convolution of the input image with the deblocking kernel. As variable filter support lengths are used, the computational complexity is maximized by the case where the maximum block size is used for all the pixels on the image.

Consequently, the resultant computational complexity is comparable to those from the methods from [9,4], and significantly lower to that from [13], that performs an adaptive multipass deblocking. The computational complexity is also considerably lower than that of interactive methods, such as the method presented in [8].

4. Experimental results

The performance of the proposed method was evaluated not only for still images, but also for video sequences, through comparison with several state-of-the-art block based encoders.

4.1. Still image deblocking

In our experiments, the performance of the proposed method was evaluated not only for images encoded using MMP, but also using two popular transform-based algorithms: JPEG and H.264/AVC, in order to demonstrate its versatility.

For images compressed using MMP, the proposed method was compared with the method proposed in [22]. As MMP is not a normative encoder, the filter parameters may be optimized by the encoder and transmitted to the decoder, in order to maximize the PSNR of the reconstruction. With this approach, the best objective quality gains are always achieved, and we have the guarantee that the deblocking filter never degrades the image's PSNR.

Consistent objective image quality gains, as well as more effective deblocking effect were obtained, when compared with the method presented in [22]. The improved mapping procedure used to estimate the block dimensions proposed in this paper eliminates the effects of erroneous consideration of accurately predicted blocks as smooth blocks observed in [22]. This avoids the exaggerate blurring of some detailed regions, with impact on the PSNR value of the filtered image. Furthermore, the new mapping procedure allows the use of a stronger deblocking in smooth regions, without degrading image details.

The comparative objective results are summarized in Table 1, while Figs. 5 and 6 present a subjective comparison between the two methods.[1] It can be seen that the proposed method achieves higher PSNR gains than the method from [22], for all cases. Additionally, it can be seen in Figs. 5 and 6 that the blocking artifacts are more effectively attenuated in both images, resulting in a better perceptual quality for the reconstructed image. High frequency details, like the ones on the headscarf from image Barbara, are successfully preserved by the proposed method.

The inefficiency of the method from [22] becomes evident in Fig. 6. The high frequency patterns from the

[1] The images used on the experimental tests are available for download at http://www.lps.ufrj.br/profs/eduardo/MMP.

headscarf tend to be efficiently predicted, and coded using relatively large blocks. The mapping used did not reflected the high frequency present in these regions, and the patterns tend to be considerably blurred. As a result, the deblocking filtering is disabled, in order to avoid the image's PSNR degradation.

It is also important to notice that the adaptability of the proposed method allows it to disable the deblocking filter for non-smooth images, such as text documents (PP1205), thus preventing highly annoying smoothing effects.

Table 1
Results for MMP coded images (dB).

Lena				
Rate (bpp)	0.128	0.315	0.442	0.600
No filter	31.38	35.54	37.11	38.44
Original [22]	31.49	35.59	37.15	38.45
Proposed	**31.67**	**35.68**	**37.21**	**38.48**
Peppers				
Rate (bpp)	0.128	0.291	0.427	0.626
No filter	31.40	34.68	35.91	37.10
Original [22]	31.51	34.71	35.92	37.10
Proposed	**31.73**	**34.77**	**35.95**	**37.11**
Barbara				
Rate (bpp)	0.197	0.316	0.432	0.574
No filter	27.26	30.18	32.39	34.43
Original [22]	27.26	30.18	32.39	34.43
Proposed	**27.38**	**30.31**	**32.51**	**34.52**
PP1205				
Rate (bpp)	0.378	0.576	0.765	1.005
No filter	27.13	31.08	34.54	37.67
Original [22]	27.13	31.08	34.54	37.67
Proposed	**27.14**	**31.09**	**34.54**	**37.67**

The versatility of the proposed method was evaluated by comparing it with the in-loop filter of H.264/AVC [9]. The images were encoded with JM 18.2 reference software, with and without the use of the in-loop filter. The non-filtered images were then subjected to a post-filtering with the proposed method. In order to preserve the compliance with the H.264/AVC standard bit-stream, the default values for α, s and τ proposed in Section 3.4 were used.

Figs. 7 and 8 present a subjective comparison between the proposed method and the in-loop filter of H.264/AVC [9]. In the reconstructions obtained with the in-loop filter disabled (Figs. 7a and 8a), blocking artifacts are quite obvious at such compression ratios. It is also interesting to notice that the deblocking artifacts have a different distribution than the ones presented by Figs. 5a and 6a, respectively. This happens because, unlike MMP, H.264/AVC only uses a limited set of pre-established block sizes. This results in the appearance of blocking artifacts in less regions, all at predictable locations (the grid that defines those transform block's boundaries), but that tend to be more pronounced at similar compression ratios.

In Figs. 7b and 8b, it can be seen that the in-loop filter used by H.264/AVC [9] is effective in reducing the deblocking artifacts, at the cost of blurring some details. The reconstruction obtained using the proposed method, presented in Figs. 7c and 8c, respectively, shows at least an equivalent perceptual quality, with a marginal objective performance advantage in most cases, with all the advantages of using a post-processing filter instead of an

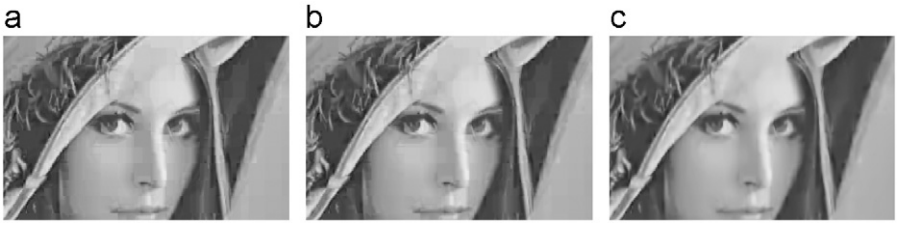

Fig. 5. A detail of image Lena 512×512, encoded with MMP at 0.128 bpp. (a) No deblocking 31.38 dB. (b) Method from [22] 31.49 dB (+0.11 dB). (c) Proposed method 31.67 dB (+0.29 dB).

Fig. 6. A detail of image Barbara 512×512, encoded with MMP at 0.316 bpp. (a) No deblocking 30.18 dB. (b) Method from [22] 30.18 dB (+0.00 dB). (c) Proposed method 30.31 dB (+0.13 dB).

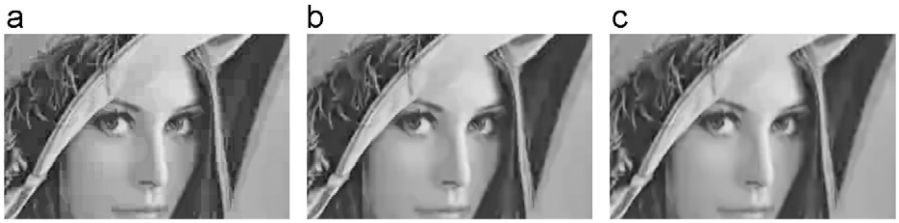

Fig. 7. A detail of image Lena 512 × 512, encoded with H.264/AVC at 0.113 bpp. (a) In-loop deblocking [9] disabled 30.75 dB. (b) In-loop deblocking [9] enabled 31.10 dB (+0.35 dB). (c) Proposed method 31.11 dB (+0.36 dB).

Fig. 8. A detail of image Barbara 512 × 512, encoded with H.264/AVC at 0.321 bpp. (a) In-loop deblocking [9] disabled 29.72 dB; (b) In-loop deblocking [9] enabled 29.87 dB (+0.15 dB). (c) Proposed method 29.84 dB (+0.12 dB).

Table 2
Results for H.264/AVC coded images (dB).

Lena				
Rate (bpp)	0.128	0.260	0.475	0.601
No filter	31.28	34.48	37.20	38.27
Original [9]	31.62	34.67	37.24	38.27
Proposed	**31.63**	**34.72**	**37.31**	**38.31**
Peppers				
Rate (bpp)	0.144	0.249	0.472	0.677
No filter	31.62	33.77	35.89	37.09
Original [9]	**32.02**	33.99	35.90	37.02
Proposed	31.98	**33.99**	**35.95**	**37.11**
Barbara				
Rate (bpp)	0.156	0.321	0.407	0.567
No filter	26.36	29.72	31.13	33.33
Original [9]	26.54	**29.87**	**31.28**	**33.45**
Proposed	**26.59**	29.84	31.25	33.42
PP1205				
Rate (bpp)	0.310	0.586	0.807	1.022
No filter	23.95	27.61	30.37	32.91
Original [9]	**24.05**	**27.75**	**30.55**	**33.09**
Proposed	24.03	27.65	30.35	32.91

Table 3
Results for JPEG coded images (dB).

Lena				
Rate (bpp)	0.16	0.19	0.22	0.25
No filter	26.46	28.24	29.47	30.41
Method from [4]	**27.83**	**29.55**	**30.61**	**31.42**
Proposed	27.59	29.32	30.46	31.29
Barbara				
Rate (bpp)	0.20	0.25	0.30	0.38
No filter	23.49	24.49	25.19	26.33
Method from [4]	**24.39**	**25.26**	**25.89**	**26.86**
Proposed	24.18	25.03	25.52	26.42
Peppers				
Rate (bpp)	0.16	0.19	0.22	0.23
No filter	25.59	27.32	28.39	29.17
Method from [4]	**27.33**	**28.99**	**29.89**	**30.54**
Proposed	26.64	28.14	29.10	29.74

in-loop filter. Furthermore, the use of the pre-established parameters' values result in a fully compliant algorithm.

The objective results presented in Table 2, for four different images with different natures, demonstrate that the proposed method is able, in several cases, to outperform the objective quality achieved by the H.264/AVC in-loop filter.

The proposed method was also tested using images encoded with JPEG. Significant quality improvements were also achieved in this case, as seen in Table 3. In this case, the proposed method was not able to outperform the method from [4]. One should note, however, that the method from [4] is specifically optimized for JPEG, since it takes advantage of the knowledge regarding the possible location of artifacts (JPEG uses a fixed 8 ×8 transform) and the artifact strength (using information from the image's quantization table), unlike the proposed method that does not make any assumption about the way the image was encoded. However, the proposed method is still able to achieve a significant objective and perceptual quality improvement for these cases.

In Figs. 9 and 10, we present the objective results obtained by both methods, for images Lena and Barbara. Blocking artifacts are evident in both the original reconstructions (Figs. 9a and 10a) at such compression ratios. From Figs. 9b and 10b, it can be seen that the method from [4] is able to significantly reduce the amount of blocking artifacts, increasing the perceptual quality of the

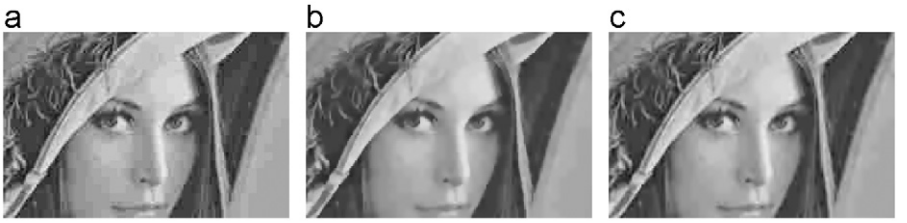

Fig. 9. A detail of image Lena 512 × 512, encoded with JPEG at 0.245 bpp. (a) Original 30.41 dB. (b) Method from [4] 31.42 dB (+1.01 dB). (c) Proposed method 31.29 dB (+0.88 dB).

Fig. 10. A detail of image Barbara 512 × 512, encoded with JPEG at 0.377 bpp. (a) Original 26.62 dB. (b) Method from [4] 27.13 dB (+0.51 dB). (c) Proposed method 26.69 dB (+0.08 dB).

reconstructed images. From Figs. 9c and 10c, it can be seen that despite the lower quality gain, the proposed method is also able to significantly reduce the amount of blocking artifacts, specially on images with a low pass nature, such as image Lena. For image Barbara, the reduction of blocking artifacts are not so strong, but the method was still able to increase the perceptual quality for this image.

Fig. 11 summarizes the objective results achieved by the proposed method, when used to filter four different images, compressed using the three tested encoders. In order to illustrate the performance of the proposed method for a wide range of image types, the results are presented for images with different levels of details. Images Lena, Goldhill and Barbara are natural images presenting, respectively, low, medium and high levels of detail. Image PP1205 results from a scanned text document, and presents a large amount of high frequency transitions, in order to evaluate how the method performs in extreme conditions.

The figure shows the PSNR gain achieved by the proposed method, over the non-filtered reconstruction. For H.264/AVC and JPEG, the gain is presented using the pre-determined parameters' values, but also the optimal values, that are obtained by testing all possible values, in order to evaluate the impact of the proposed approximation. It can be seen that the PSNR gain obtained using the pre-determined values is close to that obtained using the optimal parameters at high compression ratios. This corresponds to the case where a strong filtering is most needed, as blocking artifacts are in these cases more pronounced. The difference tends to increase for highly detailed images, because the default parameters were defined using a conservative approach, in order to avoid applying a too aggressive filtering, which would introduce blurring on high detailed regions.

An interesting phenomenon can be observed in Fig. 11d. Contrarily to the tendency observed for the other images, the PSNR gain for image PP1205, compressed using JPEG, increases when the compression ratio decreases, for the bitrate range presented in the figure. This happens because the PSNR achieved using JPEG at such compression ratios is low for this highly detailed image, with almost all details being degraded. Consequently, the deblocking filter does not have sufficient information to increase the general image's quality. When the compression ratio decreases, the amount of detail increases, and the filter becomes able to more significantly increases the PSNR gain. This tendency is however inverted when the reconstruction achieves a high level of detail, and the filtering process becomes no more beneficial. The gain then starts to decrease, in accordance to the other results. This inflexion point occurs at approximately 1.4 bpp, for the case of image PP1205.

4.2. Video sequences deblocking

The proposed method was also evaluated for video sequences deblocking. The objective was to access its performance when applied to Inter-coded frames. This approach imposes the additional challenge of locating blocking artifacts on Inter-frames, as motion compensation using non-deblocked references may introduce blocking artifacts in any location of the reconstructed images. This is different from the case of Intra-frames, where these artifacts only appear at the block boundaries. Additionally, disabling the in-loop filter causes the Inter-frames to be encoded using non-deblocked references, which reduces the probability of finding good matches for the blocks during motion estimation (ME). As a result, the motion compensation residue energy increases, decreasing the compression efficiency of Inter-frames and consequently the overall performance of the compression algorithm. For these reasons, achieving competitive results using a post-processing deblocking algorithm requires higher quality gains than that of an equivalent in-loop method, in order to compensate the lower ME performance.

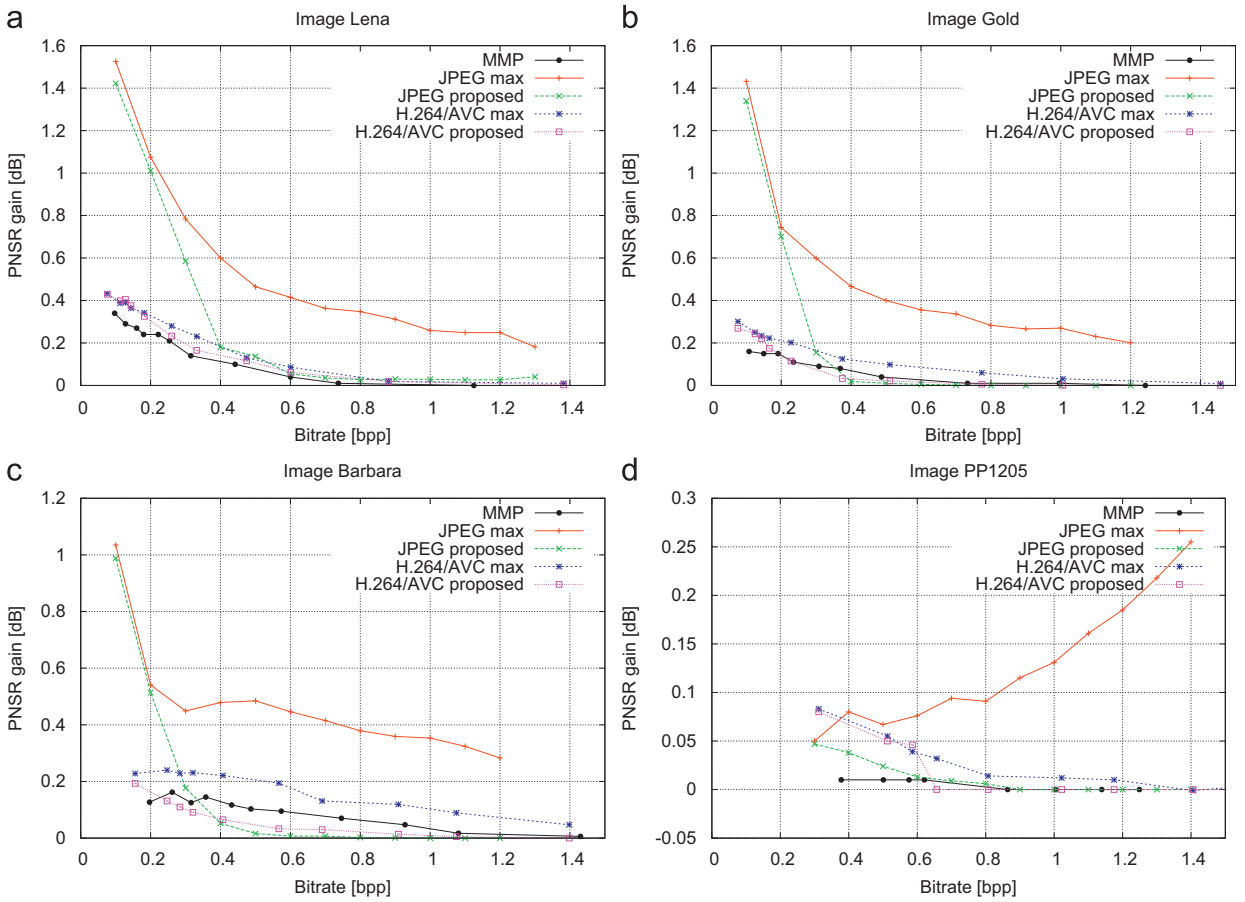

Fig. 11. Comparative results for the images Lena, Goldhill, Barbara and PP1205 (512×512).

Table 4 summarizes the results obtained for the first 128 frames of three high definition (1920×1080 pixels) well known test sequences.[2] These results were generated using JM18.2 H.264/AVC reference software, operating at high profile, either enabling or disabling the in-loop deblocking filter [9]. The non-filtered sequences were then subjected to post-filtering with the proposed method, using the parameter estimation proposed in Section 3.4. Consequently, the bitrate presented by the non-filtered sequence must be also considered for the sequence reconstructed with the proposed method. A set of commonly used parameters was adopted for these experiments, namely a GOP size of 15 frames, with an IBBPBBP pattern at a standard frame-rate of 30 fps. For ME, the Fast Full Search algorithm was adopted, with a 32 pixels search range and five reference frames. Only the PSNR values of the Luma component are presented, as references being representative to the overall results.

Unlike the case of still images, that are compressed as Intra-frames, the difference in the ME efficiency contributes to a significant difference between the achieved bitrates with the various filters, making a direct comparison of the results difficult. In order to improve this comparison, we computed the Bjøntegaard delta (BD) PSNR [24] between the two sets of results. The BD-PSNR is a metric computed as the average PSNR gain in the interval of overlapping bitrates of both sets of results. Consequently, this measure provides a reliable indication of which encoder performs better, on average, in that interval.

As it can be seen from Table 4, the post-filtering using the proposed method is able to significantly increase the average objective quality of the reconstructed video sequences, achieving global results close to those obtained enabling the H.264/AVC in-loop filter [9]. For the sequence Rush Hour, the average PSNR as a BD-PSNR decreases of up to 0.85 dB when the H.264/AVC in-loop filter is disabled, but the proposed method is able to reduce the performance gap to just 0.16 dB, which represents an average PSNR gain of 0.69 dB on the interval of the tested bitrates. Furthermore, the PSNR gains are approximately constant for all types of frames, as is shown in Fig. 12. This figure presents the PSNR both for the original (non-filtered) and post-processed first 45 frames from sequence Rush Hour, encoded using QP 43-45 (for I/P and B frames). These results demonstrate that the proposed method is able to accurately identify

[2] The used test video sequences are available for download at http://media.xiph.org/video/derf/.

Table 4
Results for H.264/AVC coded video sequences (dB).

QP (I/P–B)	**In-Loop [9] ON**		**In-Loop [9] OFF**		**Proposed**	**BD-PSNR** (dB)
	Bitrate (kbps)	**PSNR** (dB)	**Bitrate** (dB)	**PSNR** (dB)	**PSNR** (dB)	
Rush Hour						
48–50	272.46	30.62	288.24	29.99	30.69 (+0.70)	−0.16
43–45	478.65	33.62	500.79	32.92	33.67 (+0.75)	
38–40	865.48	36.38	903.19	35.73	36.42 (+0.69)	
33–35	1579.27	38.76	1636.38	38.23	38.80 (+0.57)	
Pedestrian						
48–50	409.69	28.68	420.79	28.18	28.75 (+0.56)	−0.14
43–45	711.64	31.93	730.58	31.43	32.01 (+0.58)	
38–40	1216.63	34.89	1243.96	34.44	34.83 (+0.39)	
33–35	2080.58	37.43	2107.30	37.09	37.17 (+0.08)	
Blue Sky						
48–50	572.47	26.74	583.90	26.40	26.71 (+0.30)	−0.11
43–45	912.33	30.25	924.21	29.99	30.28 (+0.29)	
38–40	1557.29	33.77	1566.44	33.54	33.73 (+0.19)	
33–35	2737.99	37.10	2740.06	36.90	36.82 (−0.08)	

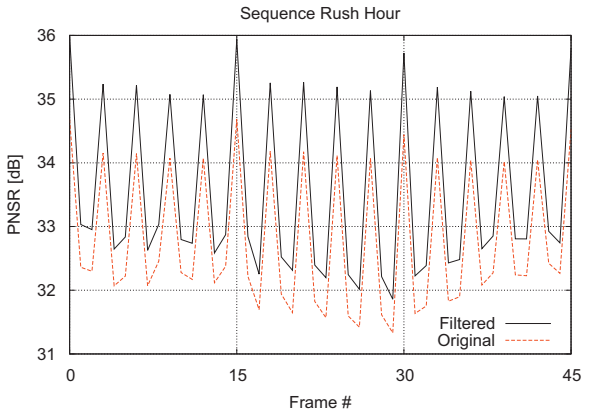

Fig. 12. PSNR of the first 45 frames from sequence Rush Hour, compressed using QP 43–45, with the H.264/AVC in-loop filter disabled, and same 45 frames post-deblocked using the proposed method.

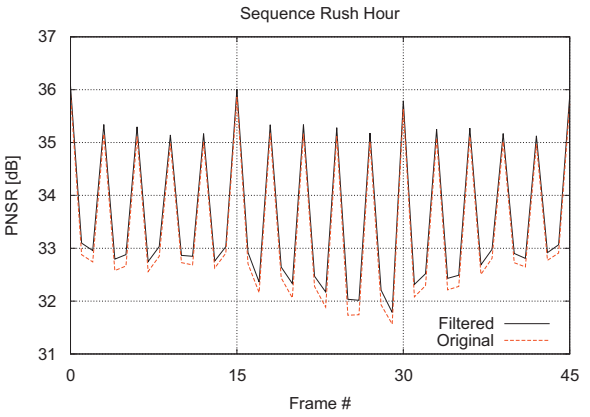

Fig. 13. PSNR of the first 45 frames from sequence Rush Hour, compressed using QP 43–45, with the H.264/AVC in-loop filter disabled only for B frames, and same 45 frames post-deblocked using the proposed method.

blocking artifacts also on Inter-frames, enhancing both the subjective and objective quality for all frames from the video sequence, independently of the coding tools used in their compression.

Motivated by these observations, we performed another experimental test, where the proposed post-deblocking method was used and the H.264/AVC in-loop filter was only disabled for non-reference frames. With this approach, there is no performance degradation on ME, allowing a more direct comparison between the gains resulting from the proposed method and those from the H.264/AVC in-loop filter. The result for the same 45 frames of sequence Rush Hour compressed using QP 43-45 is presented in Fig. 13.

In this case, the proposed method was able to outperform the H.264/AVC in-loop filter, with a BD-PSNR gain of 0.10 dB, when applied to all frames of the video sequence. Fig. 13 shows that the proposed method was not only able to improve the objective quality of the Inter-frames where the in-loop filter was not applied, but also to increase the objective quality of the reference frames that were already filtered using the in-loop filter. In this case, the increase of the PSNR is not significant, but the details and the objective quality are preserved. This was also the case for when the proposed method was applied to video sequences encoded with the H.264/AVC in-loop filter enabled for all frames, with gains up to 0.08 dB on the average PSNR. This can be important while post-processing a given video sequence, as it works regardless of the use or not of the loop filter, corroborating the fact that the proposed post-deblocking filter does not require any knowledge regarding the way it was encoded.

It is important to notice that, unlike the case of the H.264/AVC in-loop filter [9], where its in-loop nature imposes that both the encoder and the decoder must perform the same filtering tasks in order to avoid drift, the proposed method only requires the decoder to filter the reconstructed video sequence. Additionally, the filter can be switched on and off for arbitrary frames. It can be switched off when the computational resources are in

large demand, and can be switched back on when more computational resources are available. Such adaptability is not possible with the techniques from [9], where disabling the filters at some point of the decoding process results on the loss of synchronism between the encoder and the decoder.

The proposed method was also used to deblock video sequences encoded with the upcoming, highly efficient HEVC video coding standard [12]. For that purpose, we used the HM5.1 reference software, disabling both the in-loop and the ALF deblocking filter [13]. The unfiltered sequences were then subjected to a post-processing using the proposed algorithm, and the results are compared with those obtained by HEVC with both filters enabled.

The main objective of these tests was to evaluate the performance of the proposed method for the upcoming video standard, that uses a new set of coding tools, such as 64×64 unit blocks *vs.* the 16×16 blocks used on H.264/AVC and 8×8 blocks from JPEG. Some default parameters for HM5.1 were used in the experiments, such as a hierarchical-B (with an intra-period of 8) configuration. A gradual 1 QP increment was used for Inter-frames at higher levels. Motion estimation used the EPZS algorithm, with a 64 pixels search range.

The results are summarized in Table 5, for the same video sequences used to evaluate the deblocking in H.264/AVC (Table 4), in order to allow a direct comparison between the deblocking performance for both video codecs. As in H.264/AVC, disabling the filters has a significant impact on the compression efficiency of the algorithm. PSNR losses of up to 0.5 dB can be observed in some cases. Despite being outperformed by the HEVC filtering tools, the proposed method was able to significantly enhance the objective quality of the reconstructions in most cases, with increases of up to 0.28 dB in the average PSNR of the reconstructed sequences. This demonstrates once more the versatility of the proposed post-processing algorithm. Additionally, the subjective quality of the video signal was globally increased, with a considerable reduction of blocking artifacts.

These results demonstrate that, despite not being as efficient as the highly complex and algorithm-specific HEVC deblocking filters [13], the proposed method is still able to present a consistent performance when applied to signals encoded using this algorithm. This corroborates the high versatility of the proposed method and its independence relatively to the encoding tools used to compress the input images. Furthermore, the higher performance presented by [13] comes at the expense of a higher computational complexity resultant from the multipass adaptive filter. Such as in the case of the H.264/AVC in-loop filter, activating HEVC filters [13] imposes that both the encoder and the decoder need to perform this task, in order for them to remain synchronized, avoiding drift. Therefore, it is also not possible to switch the HEVC filters on and off arbitrarily in the decoder, according to the availability of computational resources.

5. Conclusions

In this paper we present an image post-deblocking scheme based on adaptive bilateral filters. The proposed method performs a total variation analysis of the encoded images to build a map that defines the filter's shape and length for each region on the image. Regions with low total variation are assumed to have a smooth nature and are strongly filtered, using filters with wide support regions to eliminate the artifacts in the blocks' boundaries. Regions with high total variation are assumed to contain a high level of detail, and are only softly filtered. The ability to reduce the length of the filter's support region or even to disable the filtering minimizes the blurring effect caused by filtering these regions.

Unlike other approaches, the proposed technique is universal, not being specifically tailored for any type of codec, being applicable both to still images and video sequences. This is confirmed by the objective and subjective image quality gains that have been observed for several tested codecs, namely MMP, JPEG, H.264/AVC and the upcoming standard HEVC. Additionally, the method is

Table 5
Results for HEVC coded video sequences (dB).

QP (I/P–B)	**In-Loop [13] ON**		**In-Loop [13] OFF**		**Proposed**	
	Bitrate (kbps)	**PSNR** (dB)	**Bitrate** (dB)	**PSNR** (dB)	**PSNR** (dB)	**BD-PSNR** (dB)
Rush Hour						
48	214.14	34.41	216.27	34.00	34.27 (+0.26)	−0.20
43	404.23	36.69	410.19	36.25	36.53 (+0.28)	
38	785.83	38.75	798.08	38.32	38.60 (+0.28)	
33	1655.04	40.67	1683.73	40.28	40.53 (+0.25)	
Pedestrian						
48	322.32	32.68	321.74	32.25	32.49 (+0.24)	−0.20
43	576.87	35.20	575.34	34.76	35.00 (+0.24)	
38	1040.16	37.50	1036.70	37.11	37.28 (+0.17)	
33	2003.40	39.68	1995.11	39.36	39.39 (+0.02)	
Blue Sky						
48	414.47	32.10	422.10	31.52	31.54 (+0.02)	−0.67
43	715.92	35.01	727.25	34.45	34.44 (−0.01)	
38	1261.90	37.80	1284.25	37.26	37.20 (−0.05)	
33	2327.27	40.41	2369.07	39.95	39.78 (−0.17)	

a post-processing technique, that does not impose the transmission of any side information, resulting in a fully compliant algorithm regardless of the codec used to compress the image.

Acknowledgments

This project was funded by FCT - "Fundação para a Ciência e Tecnologia", Portugal, under the Grant SFRH/BD/45460/2008, and Project COMUVI (PTDC/EEA-TEL/099387/2008).

References

[1] W. Pennebaker, J. Mitchel, JPEG: Still Image Data Compression Standard, Van Nostrand Reinhold, 1993.
[2] J.V.T.J. of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11, I.-T.S.Q.6), Draft of Version 4 of H.264/AVC (ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 part 10) Advanced Video Coding), 2005.
[3] T. Jarske, P. Haavisto, I. Defe'e, Post-filtering methods for reducing blocking effects from coded images, in: IEEE 1994 International Conference on Consumer Electronics, 1994. Digest of Technical Papers, 1994, pp. 218–219. http://dx.doi.org/10.1109/ICCE.1994.582234.
[4] V. Nath, D. Hazarika, A. Mahanta, Blocking artifacts reduction using adaptive bilateral filtering, in: 2010 International Conference on Signal Processing and Communications (SPCOM), Bangalore, India, 2010, pp. 1–5. http://dx.doi.org/10.1109/SPCOM.2010.5560517.
[5] Z. Xiong, M. Orchard, Y.-Q. Zhang, A deblocking algorithm for JPEG compressed images using overcomplete wavelet representations, IEEE Transactions on Circuits and Systems for Video Technology 7 (2) (1997) 433–437, http://dx.doi.org/10.1109/76.564123.
[6] T. Chen, H. Wu, B. Qiu, Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts, IEEE Transactions on Circuits and Systems for Video Technology 11 (5) (2001) 594–602, http://dx.doi.org/10.1109/76.920189.
[7] J. Xu, S. Zheng, X. Yang, Adaptive video-blocking artifact removal in discrete Hadamard transform domain, Optical Engineering 45(8) (2006). http://dx.doi.org/10.1117/1.2280609.
[8] A. Zakhor, Iterative procedures for reduction of blocking effects in transform image coding, IEEE Transactions on Circuits and Systems for Video Technology 2 (1) (1992) 91–95, http://dx.doi.org/10.1109/76.134377.
[9] P. List, A. Joch, J. Lainema, G. Bjontegaard, M. Karczewicz, Adaptive deblocking filter, IEEE Transactions on Circuits and Systems for Video Technology 13 (7) (2003) 614–619, http://dx.doi.org/10.1109/TCSVT.2003.815175.
[10] Y.-M. Huang, J.-J. Leou, M.-H. Cheng, A post deblocking filter for H.264 video, in: Proceedings of 16th International Conference on Computer Communications and Networks, 2007. ICCCN 2007, Honolulu, Hawaii, 2007, pp. 1137–1142. http://dx.doi.org/10.1109/ICCCN.2007.4317972.
[11] H.-S. Kong, A. Vetro, H. Sun, Edge map guided adaptive post-filter for blocking and ringing artifacts removal, in: Proceedings of the 2004 International Symposium on Circuits and Systems, 2004. ISCAS '04, Vancouver, Canada, vol. 3, 2004, pp. III-929-32. http://dx.doi.org/10.1109/ISCAS.2004.1328900.
[12] G.J. Sullivan, J.-R. Ohm, Recent developments in standardization of high efficiency video coding (HEVC), in: Proceedings of the SPIE, vol. 7798, 2010.
[13] Y. Liu, Unified loop filter for video compression, IEEE Transactions on Circuits and Systems for Video Technology 20 (10) (2010) 1378–1382, http://dx.doi.org/10.1109/TCSVT.2010.2077570.
[14] M.B. de Carvalho, E.A.B. da Silva, W.A. Finamore, Multidimensional signal compression using multiscale recurrent patterns, Signal Processing 82 (11) (2002) 1559–1580, http://dx.doi.org/10.1016/S0165-1684(02)00302-X.
[15] N. Francisco, N. Rodrigues, E. da Silva, M. de Carvalho, S. de Faria, V. Silva, Scanned compound document encoding using multiscale recurrent patterns, IEEE Transactions on Image Processing 19 (10) (2010) 2712–2724, http://dx.doi.org/10.1109/TIP.2010.2049181.
[16] N. Rodrigues, E. da Silva, M. de Carvalho, S. de Faria, V. da Silva, Improving H.264/AVC inter compression with multiscale recurrent patterns, in: 2006 IEEE International Conference on Image Processing, Atlanta, USA, 2006, pp. 1353–1356. http://dx.doi.org/10.1109/ICIP.2006.312585.
[17] J. Ziv, A. Lempel, A universal algorithm for sequential data compression, IEEE Transactions on Information Theory 23 (3) (1977) 337–343, http://dx.doi.org/10.1109/TIT.1977.1055714.
[18] J. Ziv, A. Lempel, Compression of individual sequences via variable-rate coding, IEEE Transactions on Information Theory 24 (5) (1978) 530–536, http://dx.doi.org/10.1109/TIT.1978.1055934.
[19] W. Finamore, M. de Carvalho, Lossy Lempel–Ziv on subband coding of images, in: 1994 IEEE International Symposium on Information Theory, Trondheim, Norway, 1994, p. 415. http://dx.doi.org/10.1109/ISIT.1994.395030.
[20] A. Gersho, R. Gray, Vector Quantization and Signal Compression, Springer, 1991.
[21] W. Dai, L. Liu, T. Tran, Adaptive block-based image coding with pre-/post-filtering, in: Proceedings of the Data Compression Conference, DCC 2005, Snowbird, USA, 2005, pp. 73–82. http://dx.doi.org/10.1109/DCC.2005.11.
[22] N. Rodrigues, E. da Silva, M. de Carvalho, S. de Faria, V. Silva, Improving multiscale recurrent pattern image coding with deblocking filtering, in: 2006 International Conference on Signal Processing and Multimedia Applications, Setubal, Portugal, 2006.
[23] N. Rodrigues, E. da Silva, M. de Carvalho, S. de Faria, V. da Silva, On dictionary adaptation for recurrent pattern image coding, IEEE Transactions on Image Processing 17 (9) (2008) 1640–1653, http://dx.doi.org/10.1109/TIP.2008.2001392.
[24] G. Bjøntegaard, Calculation of average PSNR differences between RD-curves, ITU-T SG 16 Q.6 VCEG, Doc. VCEG-M33, 2001.