# Improving Multiscale Recurrent Pattern Image Coding with Deblocking Filtering

Nuno M.M. Rodrigues[1,2], Eduardo A.B. da Silva[3], Murilo B. de Carvalho[4], Sérgio M.M. de Faria[1,2], and Vitor M.M. da Silva[1,5]

[1] Instituto de Telecomunicações, Portugal
[2] ESTG, Instituto Politécnico Leiria, Portugal
[3] PEE/COPPE/DEL/Poli, Univ. Fed. Rio de Janeiro, Brazil
[4] TET/CTC, Univ. Fed. Fluminense, Brazil
[5] DEEC, Universidade de Coimbra, Portugal
nuno.rodrigues@co.it.pt, eduardo@lps.ufrj.br
murilo@telecom.uff.br, sergio.faria@co.it.pt,
vitor.silva@co.it.pt

**Abstract.** The Multidimensional Multiscale Parser (MMP) algorithm is an image encoder that approximates the image blocks by using recurrent patterns, from an adaptive dictionary, at different scales. This encoder performs well for a large range of image data. However, images encoded with MMP suffer from blocking artifacts. This paper presents the design of a deblocking filter that improves the performance the MMP. We present the results of our research, that aims to increase the performance of MMP, particularly for smooth images, without causing quality losses for other image types, where its performance is already up to 5 dB better than that of top transform based encoders. For smooth images, the proposed filter introduces relevant perceptual quality gains by efficiently eliminating the blocking effects, without introducing the usual blurring artifacts. Besides this, we show that, unlike traditional deblocking algorithms, the proposed method also improves the objective quality of the decoded image, achieving PSNR gains of up to about 0.3 dB. With such gains, MMP reaches an almost equivalent performance to that of the state-of-the-art image encoders (equal to that of JPEG2000 for higher compression ratios), for smooth images, while maintaining its gains for non-smooth images. In fact, for all image types, the proposed method provides significant perceptual improvements, without sacrificing the PSNR performance.

**Keywords:** Multidimensional Multiscale Parser, MMP-Intra, Deblocking Filter, Image Coding.

## 1 Introduction

The success of the current state-of-the-art transform-quantisation based encoders results from their excellent performance in the compression of natural images. Nevertheless, the relative performance of these encoders decreases noticeably when we deviate from the smoothness assumption, as is the case for images like text, compound (text and graphics), computer generated, texture, medical, among others. Indeed, it is a well known fact that most of the encoders that achieve top results for these image classes have poor performances for smooth images.

J. Filipe and M.S. Obaidat (Eds.): ICETE 2006, CCIS 9, pp. 224–236, 2008.

The Multidimensional Multiscale Parser (MMP) [1] is a lossy multidimensional signal encoder, that, unlike most state-of-the-art image encoders, is not based on the transform-quantisation paradigm. It is a multiscale recurrent pattern matching method, that uses an adaptive dictionary for approximating blocks of the original signal.

Using the same pattern matching paradigm, a new image encoding method, that combines MMP with the prediction techniques of H.264/AVC [3], was proposed in [4]. MMP-Intra is able to achieve quality gains over the original MMP algorithm for all image types, but particularly for smooth images, where the performance of MMP is inferior to that of the top transform-quantisation based encoders. Experimental results show that, when combined with convenient dictionary design techniques, the rate distortion (RD) performance of MMP-Intra becomes only marginally inferior (about 0.2 to 0.5 dB) to that of the JPEG2000 [6] and H.264/AVC *high* profile [3] image encoders, for the coding of smooth images [5]. For other types of images, MMP-Intra consistently maintains its excellent performance, achieving gains over standardised state-of-the-art encoders that range from 1 to 5 dB.

MMP-Intra, as MMP, uses the concatenation of the approximations of the original image blocks, at different scales. This process introduces blocking artifacts in the decoded image, that are particularly evident for higher compression ratios.

This paper presents a new deblocking scheme for MMP-Intra, that improves the performance of this image encoder for smooth images, without compromising its compression performance for other image types. The proposed method is based on a deblocking method, originally proposed for MMP and a matching pursuit based multiscale algorithm [1][2], but introduces new adaptive features, that allow it to optimise the perceptual results, as well as the objective performance of the encoded image.

The experimental results presented in this paper demonstrate that when the new method is combined with proper strategies to control the deblocking filter's parameters, it is able to consistently improve the objective results for smooth images, achieving gains that go up to about 0.3 dB. For smooth images, these gains in PSNR correspond to obvious improvements in the perceptual quality, resulting from the reduction of the blocking effects introduced by the encoding process. For non smooth images, like text and compound images, the new filtering strength control procedure is able to attenuate, or even eliminate, the smoothing effects of the deblocking process, that result in a loss of objective quality.

In the next section we briefly present the MMP and MMP-Intra image encoding methods. Section 3 describes a recently proposed dictionary design technique and explains its importance in increasing the performance of the MMP-Intra encoder. Section 4 presents the new deblocking strategies proposed in this paper and is followed by section 5 where the experimental results of this method are presented . Section 6 ends the paper with some closing remarks and conclusions.

## 2 Image Coding with MMP

A brief discussion of the application of the MMP and MMP-Intra algorithms to image coding is presented in this section. More information about these methods can be found respectively in [1] and [4].

### 2.1 The MMP Algorithm

MMP is an multiscale approximate pattern matching algorithm. It approximates an original square image block, or its successive binary segmentations, using a vector from an *adaptive dictionary* $\mathcal{D}$. Scale transformations are used to adapt the dimensions of blocks with different sizes. The successively segmented blocks, $\mathbf{X}^l$, are represented by a binary segmentation tree, where each original square block is segmented first in the vertical, then in the horizontal direction. The superscript $l$ means that the block $\mathbf{X}^l$ belongs to $scale\ l$ or $level\ l$ of the segmentation tree (with dimensions $(2^{\lfloor \frac{l+1}{2} \rfloor} \times 2^{\lfloor \frac{l}{2} \rfloor})$).

A simple definition of the MMP algorithm can be given by the following main steps. For each block of the original image, $\mathbf{X}^l$:

1. find the dictionary element $\mathbf{S}_i^l$ that minimises the Lagrangian cost function of the approximation, given by: $J(\mathcal{T}) = D(\mathbf{X}^l, \mathbf{S}_i^l) + \lambda R(\mathbf{S}_i^l)$, where $D(.)$ is the sum of square differences (SSD) function and $R(.)$ is the rate needed to encode the approximation;
2. parse the original block into two blocks, $\mathbf{X}_1^{l-1}$ and $\mathbf{X}_2^{l-1}$, with half the pixels of the original block;
3. apply the algorithm recursively to $\mathbf{X}_1^{l-1}$ and $\mathbf{X}_2^{l-1}$, until level 0 is reached;
4. based on the values of the cost functions determined in the previous steps, decide whether to segment the original block or not;
5. if the block should not be segmented, use vector $\mathbf{S}_i^l$ of the dictionary to approximate $\mathbf{X}^l$;
6. else
   (a) create a new vector $\mathbf{S}_{new}^l$ from the *concatenation* of the vectors used to approximate each half of the original block: $\mathbf{X}_1^{l-1}$ and $\mathbf{X}_2^{l-1}$;
   (b) use $\mathbf{S}_{new}^l$ to approximate $\mathbf{S}^l$;
   (c) use $\mathbf{S}_{new}^l$ to *update* the dictionary, making it available to encode future blocks of the image.

This algorithm results in a binary segmentation tree that represents each original image block. This tree, represented in figure 1, is encoded using a top-bottom preorder approach. In the final bit-stream, each leaf is encoded using a binary symbol '1' and followed by an index, that identifies the vector of the dictionary that should be used to approximate the corresponding sub-block. Each tree node is encoded using the binary symbol '0'. The string of symbols that represents the segmentation tree is encoded using an adaptive arithmetic encoder.

Unlike conventional vector quantisation (VQ) algorithms, MMP uses *approximate block matching with scales* and an *adaptive dictionary*.

Every concatenation of two dictionary blocks of level $l-1$ results in a new block, that corresponds to a pattern that did not exist in the dictionary and is used to update it, becoming available to encode future blocks of the image, independently of their size. This updating procedure efficiently adapts the dictionary, by using only information that can be inferred by the decoder, since it is based exclusively in the encoded segmentation flags and dictionary indexes.

MMP uses a separable scale transformation $T_N^M$ to adjust the vectors' sizes before attempting to match them, allowing for the matching of vectors of different dimensions.

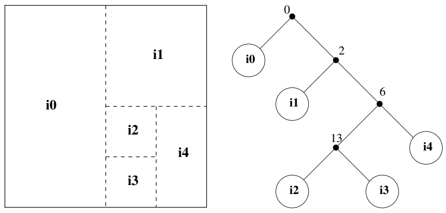



**Fig. 1.** Segmentation of a block and corresponding binary tree: the root corresponds to a original 4×4 block (level 4), while nodes $i_2$ and $i_3$ (1×1 blocks) belong to level 0

For example, in order to approximate an original block $\mathbf{X}^l$ using one block $\mathbf{S}^k$ of a different scale of the dictionary, MMP first determines $\mathbf{S}^l = T_k^l[\mathbf{S}]$. Detailed information about the use of scale transformations in MMP is presented in [1].

### 2.2 The MMP-Intra Algorithm

MMP-Intra combines the original MMP algorithm with predictive coding. For each original block, $\mathbf{X}^l$, MMP-Intra determines a prediction block, $\mathbf{P}^l_m$, using previously encoded image pixels and then it determines a residue block, given by $\mathbf{R}^l_m = \mathbf{X}^l - \mathbf{P}^l_m$. This residue block is then encoded using MMP.

MMP-Intra uses essentially the same prediction modes defined by H.264/AVC for Intra coded blocks [3][4]. Intra prediction is also used *hierarchically* for blocks of dimensions 16×16 down to 4×4 (corresponding to levels 8 to 4 of the segmentation tree). By the use of the Lagrangian RD cost function, the encoder jointly optimises the block prediction and the MMP residue encoding, determining the best trade-off between the prediction accuracy and the additional overhead introduced by the prediction data.

MMP-Intra encodes some additional information for the block prediction, namely the used prediction mode, $m$, and the block size used for the prediction step. This information is used by the decoder to determine the same prediction block, $\mathbf{P}^l_m$, that was used in the encoder. This block is added to the decoded residual block, $\hat{\mathbf{R}}^l_m$, in order to reconstruct the decoded image block, given by $\hat{\mathbf{X}}^l = \mathbf{P}^l_m + \hat{\mathbf{R}}^l_m$. Details about MMP-Intra can be found in [4].

## 3 Efficient Dictionary Design for MMP-Intra

MMP-Intra, as MMP, uses an initial dictionary consisting of a few blocks with constant value. This highly sparse initial dictionary is very inefficient, but the updating procedure quickly adapts its blocks to the original images' patterns, by introducing new blocks, $\mathbf{S}^l_{new}$, created by the concatenation of two vectors of level $l-1$ of the dictionary.

Experimental studies have shown that the final number of blocks for each level of the dictionary is, by far, much larger than the total number of blocks that are actually used. This difference grows with the target bit-rate, but can be observed for different image types and target compression ratios. The exaggerate growth of the dictionary has the disadvantage of increasing the dictionary's indexes' entropy, compromising the method's performance.

In [5], a new algorithm was proposed to limit the dictionary growth, that introduces a "minimum distance condition" between any two vectors of each level of the dictionary. This process avoids that new vectors, very close to those already available in the dictionary space, are used to update the dictionary, by using a new test condition in the dictionary update procedure. With this new algorithm, a new block of level $l$, $\mathbf{S}^l_{new}$, is only used to update the dictionary if its minimum distortion, in relation to the blocks already available in the dictionary, is not inferior to a given threshold $d$.

The optimum value for $d$ is a function of the target bit-rate and therefore of the parameter $\lambda$, and must be carefully chosen. If this value is too small, the aim of controlling the dictionary growth will not be achieved, and if it is too large, the dictionary will lose its efficiency in approximating the images' patterns. A simple expression for $d(\lambda)$ (see eq. 1) was determined by the use of a test image set, and allows the encoder to automatically achieve a close to optimum RD relation, for any given target bit-rate. Further details on how this equation was determined can be found in [5].

$$d(\lambda) = \begin{cases} 5 & if \quad \lambda \leq 15; \\ 10 & if \quad 15 < \lambda \leq 50; \\ 20 & otherwise. \end{cases} \tag{1}$$

In [5], the authors also show that the dictionary's indexes can be more efficiently encoded by using a context adaptive arithmetic encoder. The dictionary indexes are divided into groups, according to a context criterion, that, for MMP-Intra, is the original scale of the block. Instead of using just one symbol to encode a dictionary index, each index is transmitted using one context symbol followed by an index, that chooses among the elements of the corresponding segment. This carefully chosen segmentation criterion further explores the statistical dependencies of the MMP symbols, generating gains in the arithmetic coding module.

## 4 The Deblocking Filter

The MMP-Intra algorithm uses the concatenation of several approximations of the image blocks, at different scales. For each approximation, $\hat{\mathbf{X}}^l$, the RD control algorithm only controls the distortion for the image block and makes no consideration regarding the continuity in the border of the blocks. This introduces blocking artifacts in the reconstructed image, that originate from the discontinuities in the block boundaries.

In this work we present the results of our investigation in deblocking techniques that increase both the objective as the perceptual quality of the MMP-Intra's reconstructed image. We use an adaptive space-variant finite impulse response (FIR) filter to attenuate the blocks' borders discontinuities.

Let $\hat{\mathbf{X}}$ be the reconstructed signal. $\hat{\mathbf{X}}$ can be regarded as the concatenation of several blocks, $\hat{\mathbf{X}}^{l_k}_k$, that represent the algorithm's approximation of the various adjacent areas of the image. The $K$ blocks $\hat{\mathbf{X}}^{l_k}_k$, used in the approximation, have no overlapping areas and different block sizes, given by $(2^{\lfloor \frac{l_k+1}{2} \rfloor} \times 2^{\lfloor \frac{l_k}{2} \rfloor})$. The decoded image can thus be represented as

$$\hat{\mathbf{X}} = \sum_{k=0}^{K-1} \hat{\mathbf{X}}_k^{l_k}(x - x_k, y - y_k). \tag{2}$$

In equation 2, each block $\hat{\mathbf{X}}_k^{l_k}$ corresponds to a dictionary block, of scale $l_k$, that was created by the MMP dictionary update process. This means that each of these blocks can be further decomposed into their basic components, $\mathcal{D}_0$, each belonging to the original dictionary, i.e.

$$\hat{\mathbf{X}}_k^{l_k} = \sum_{j=0}^{J-1} \mathcal{D}_{0j}^{l_j}. \tag{3}$$

The blocks $\mathcal{D}_{0j}^{l_j}$ can be regarded as the basic "building units" that were used by the MMP-Intra encoder and the $l_j$ values represent the scale that was used by the encoder to represent each area of the image. The border points between each of these blocks correspond to the most probable areas for discontinuities in the decoded image.

In this work we apply a running bi-dimensional FIR filter to the reconstructed image, $\hat{\mathbf{X}}$. The filter's kernel dimensions are successively adapted to the scale of the original dictionary block that was used to approximate the area of the image that is currently being deblocked.

Blocks $\mathcal{D}_{0j}^{l_j}$ of large scales have larger support regions, meaning that the corresponding area of the image is smoother, while blocks with small values of $l_j$ are used in more detailed image areas. The used space-variant filter has the ability to adapt its support, and smoothing strength, to the dimensions of each image segment, $\mathcal{D}_{0j}^{l_j}$, being considered.

Figure 2 has a unidimensional representation of a reconstructed portion of the image, that was approximated by the concatenation of three basic blocks, $(\mathcal{D}_0^{l_0} \quad \mathcal{D}_1^{l_1} \quad \mathcal{D}_2^{l_2})$, with different scales: $l_0$, $l_1$ and $l_2$. At each filtered pixel, represented in the figure by the arrow, the kernel support of the deblocking filter is set according to the scale $l_k$.

This process is similar to the one proposed in [2], that uses a running average filter and sets the kernel support at each point to $l_k + 1$. This filter is known for its highly smoothing effect, but the support adaptation process controls its strength according to

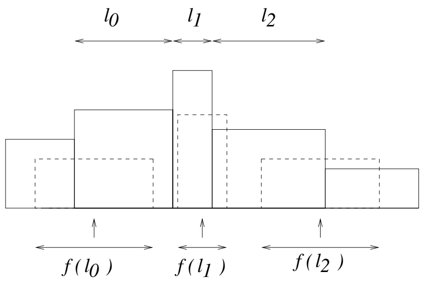



**Fig. 2.** The deblocking process uses an adaptive support for the FIR of the filters used in the deblocking

the detail level of the region that is being deblocked. This prevents some of the blurring artifacts that are usually caused by the use of too powerful deblocking techniques, but the original filtering process still results in a reduction in objective quality for smooth images. Another disadvantage of the original process is that it introduces highly disturbing blurring artifacts in non smooth images, resulting in a severe decrease in the final values of PSNR. This fact limits the applicability of this filter, because there is no practical way of avoiding the blurring of images that do not need deblocking.

In our work, we have adapted this deblocking process to MMP-Intra and developed it. This investigation resulted in a more efficient, highly adaptive, deblocking filter, that has some important advantages over the original method, namely:

- it uses a Gaussian kernel with optimised shape and support for each image, that adapts the deblocking strength to the image features resulting in perceptual, as well as, *objective* quality gains;
- the kernel shape optimisation means that the filtering strength is automatically adjusted and can be set to an arbitrarily low power. This means that, for non low-pass images, the new process automatically eliminates the highly annoying blurring effects and the corresponding PSNR losses;
- the new method considers the dimensions of the neighbouring blocks as well as those of the block being filtered, eliminating some artifacts that were introduced by the original method;
- the proposed algorithm monitors the differences in the frontiers' pixels' intensities, in order to avoid smoothing steep variations that were present in the original image and do not correspond to blocking artifacts.

### 4.1 Adapting Shape and Support for the Deblocking Kernel

In our investigation we tested different kernels with various support regions for the deblocking filter. Experimental results showed that the use of Gaussian kernels, instead of the original rectangular filter, produces gains in the PSNR value of the decoded image, as well as the desired effect of eliminating the blocking artifacts.

These tests also demonstrated that the quality of the deblocked image strongly depends on the dimensions of the support region of the used filter. In the original method, this support is set to $l_k + 1$. We varied this value and discovered that it is optimal for the running average filter, but that this is not the case when we use a Gaussian kernel.

Instead of adjusting the support region of the Gaussian kernel, we set the filter length at the same $l_k + 1$ samples used in the original method, but adjust the Gaussian's variance, producing filter kernels with different shapes. Consider a Gaussian filter, with variance $\sigma^2$ and length $L$, with an impulse response (IR) given by:

$$g_L(n) = e^{-\frac{\left(n-\frac{L-1}{2}\right)^2}{2.\sigma^2}}, \tag{4}$$

with $n = 0, 1, ..., L - 1$. We controlled the shape of the filter by changing a filter parameter $\alpha$, that controls the variance of the Gaussian, by using the expression:

$$g_L(n) = e^{-\frac{\left(n-\frac{L-1}{2}\right)^2}{2.(\alpha.L)^2}}, \tag{5}$$

to determine the filters' IR.

Figure 3 represents the shape of a 17 tap filter for the several values of parameter $\alpha$ represented in the legend. This figure clearly demonstrates the explored relation between the filters' shape and their approximate support. By varying the value of the filter's $\alpha$ parameter, one is able to efficiently adjust its IR from an almost rectangular filter, with a support region $l_k + 1$, to a Gaussian filter with different lengths. In the limit, when $\alpha$ tends to zero, the IR of the filter becomes a simple impulse, deactivating the deblocking effect for those cases were it is not beneficial.

The value of the parameter $\alpha$ is controlled by the MMP-Intra encoder. At the end of the encoding process, the MMP-Intra encoder tests the deblocking process using different values for the $\alpha$ parameter. It is then able to determine the value that maximises the PSNR of the reconstructed image. The value of $\alpha$ is then appended at the end of the encoded bit-stream, by using a 3 bit code, that corresponds respectively to the 8 possible values for $\alpha$: $\{0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40\}$. This introduces a marginal additional computational cost in the encoder, as well as an additional rate overhead, that is equally negligible.

### 4.2 Eliminating the Artifacts Introduced by Deblocking

The original method only considers the dimensions of the block currently being filtered to set the filter support. In our investigation we noticed that this fact introduces an unexpected artifact, when there exists the concatenation of wide and short blocks, with very different intensity values.

This case is represented in figure 4, where a wide dark block $A$ is concatenated with two bright blocks: one narrow block $B$ followed by one wide block $C$. When we filter blocks $A$ and $B$, a smooth transition appears, that eliminates the blocking effect in the $AB$ border. When the block $C$ is filtered, because the used filter has a very wide support region, the pixels near the $BC$ border will suffer from the influence of some of the dark pixels of block A. This causes a dark "valley" to appear in the $BC$ border, that introduces a visible artifact in the deblocked image.

In order to avoid these artifacts, the new method controls the filter length so that the deblocking filter never takes in consideration pixels that are not from the present block

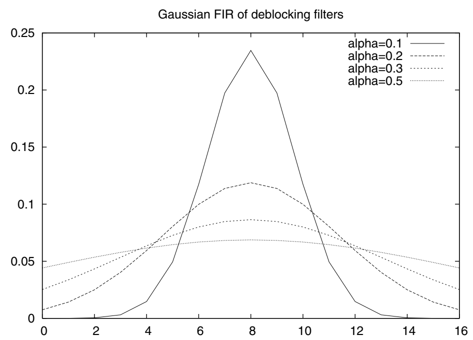



**Fig. 3.** Adaptive FIR of the filters used in the deblocking

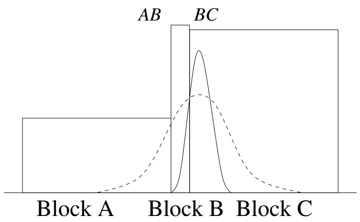



**Fig. 4.** A case were the concatenation of blocks with different supports and pixel intensities causes the appearance of an image artifact, after the deblocking filtering

or its adjacent neighbours. In the example of figure 4, the length of the filter used in the $C$ block's pixels that are near the $BC$ border is controlled, so that the left most pixel that is used in the deblocking is always the first (left most) pixel of block $B$, eliminating the described artifact. In figure 4, this means that the new method uses the filter represented by the solid line, instead of the original one, represented by the dashed line.

Another artifact caused by the original method is the introduction of smooth transitions in regions of the image that originally have very steep transitions from low to high pixel intensity values (or vice versa). The proposed algorithm monitors the differences in the frontiers' pixels' intensities, in order to avoid filtering steep variations that do not correspond to blocking artifacts. This is again controlled by the encoder, using an adaptive method.

The proposed method uses a step intensity threshold, $s$, that corresponds to the maximum intensity difference between the two border pixels, that still allows for the filtering to occur. This process is represented in figure 5, where two blocks $A$ and $B$ with very different intensity values are concatenated. In this case, the $AB$ border is only filtered if the absolute difference between the border pixels is inferior to the defined value for $s$, i.e., $|A_k - B_0| < s$.

The value of $s$ is again chosen in order to maximise the PSNR value for the particular image that is being deblocked. The encoder tests a set of different step values and transmits the code corresponding to the chosen value. A three bit code is again used to represent the eight possible values for $s$, belonging to the set $\{0, 16, 32, 64, 96, 128, 192, 255\}$, where $s = 0$ corresponds to never filtering the borders and $s = 255$ corresponds to the case where all blocks are filtered.

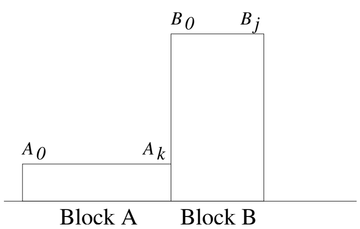



**Fig. 5.** A case where a steep variation in pixel intensities is a feature of the original image

## 5 Experimental Results

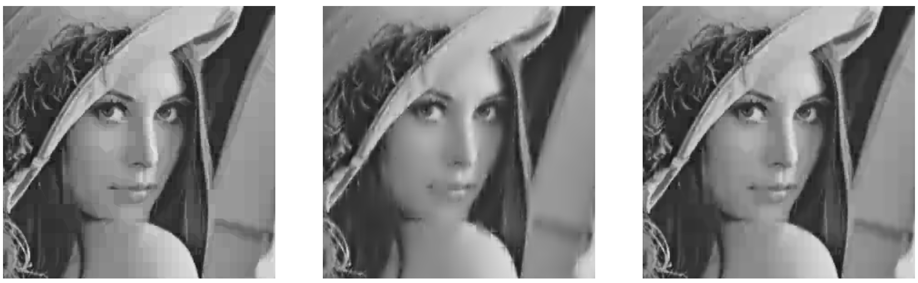
Experimental tests were performed using the proposed and original deblocking methods. Figure 6 presents a detail of image LENA 512, encoded using the described MMP-Intra algorithm without using any deblocking technique and compares it with the results of using the the deblocking technique from [2], and the new deblocking technique, proposed in this paper.

Perceptually, we can observe that the new deblocking filter is able to efficiently eliminate the blocking artifacts in Lena's face and hat, without compromising the image quality at regions with high detail, like Lena's hair and her hat's feathers. In this case, the used filter has $\alpha = 0.10$ and $s = 255$.

When compared with the original deblocking method, we can observe that the smoothing effect introduced by the proposed method is not as strong, avoiding the introduction of some blurring artifacts that are noticeable in the image of figure 6 b), specially in the areas with finer details.

In figure 6 b) we can also observe the first type of artifacts, explained in section 4.2. They appear in Lena's shoulder, where the previously described "dark valleys" are easy to observe. We can see that the proposed method efficiently eliminates these artifacts.

Figure 6 b) also shows that the original method, developed originally for the MMP encoder, suffers from a unexpectedly high performance loss, when used with MMP-Intra. Because MMP-Intra uses predictive coding, the used dictionary blocks approximate *residue* patterns. In some cases, where the prediction step is particularly efficient, some detailed areas are approximated by large, smooth, residue blocks added with detailed prediction blocks. In this case, the deblocking process uses a wide filter to deblock an image area that is not necessarily smooth. When this happens, the use of the original deblocking method originates serious artifacts, like the one observed in Lena's lip. Even when this fact is not as obvious as in the presented case, we can generally say that this factor seriously compromises the performance of the original method, when applied to MMP-Intra, resulting in a severe reduction in the PSNR results. However, due to its adaptability, the proposed method does not seem to suffer from this disturbing factor.



(a) No deblocking (30.92 dB)

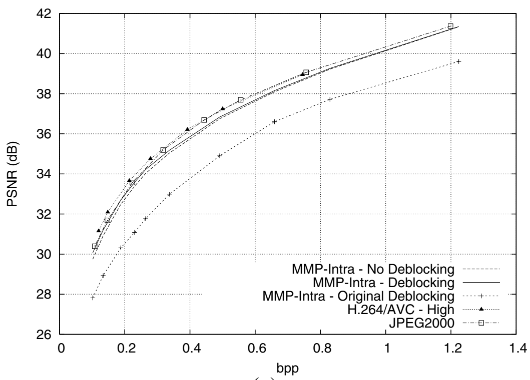
(b) Original deblocking (28.93 dB)

(c) New deblocking (31.21 dB)

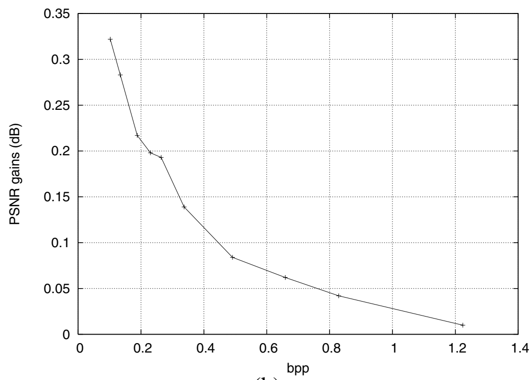
**Fig. 6.** A detail of image Lena 512, encoded with MMP-Intra at 0.135 bpp

Figure 7 a) shows the objective quality results for image Lena 512, for the MMP-Intra method with no deblocking and with the two tested deblocking techniques. Figure 7 b) highlights the PSNR quality gains introduced by the deblocking filter, that are more relevant for higher compression ratios, where the blocking artifacts are more noticeable. These gains go up to more than 0.3 dB, allowing for the PSNR results of MMP-Intra for image Lena to come even closer to the ones of top state-of-the-art transform-quantisation based encoders, like JPEG2000 [6] and H.264/AVC, [3], shown in figure 7 a). In fact, we can see that, for low bit-rates, the proposed method allows for MMP-Intra to achieve equivalent results to those of the JPEG2000 algorithm.

We also performed experimental tests using non smooth images, like text image PP1205 and compound (text and grayscale) image PP1209. Images PP1205 and PP1209 were scanned, respectively, from pages 1205 and 1209 of the *IEEE Transactions on Image Processing*, volume 9, number 7, July 2000 and are available for download at *http://www.estg.ipleiria.pt/∼nuno/MMP/*. These tests showed that the proposed kernel adaptation algorithm eliminates the highly disturbing blurring artifacts introduced when

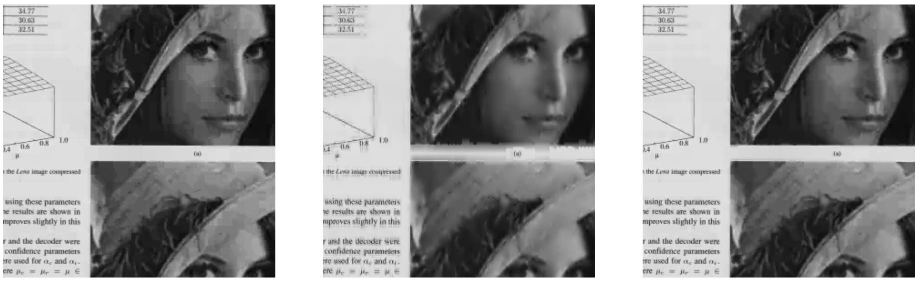



(a)




(b)

**Fig. 7.** a) Objective quality results for image Lena; b) PSNR gains of the new method, when compared with MMP-Intra with no deblocking, using $\alpha = 0.10$, $s = 255$



(a) No deblocking (32.23 dB) (b) Original deblocking (27.13 dB) (c) New deblocking (32.04 dB)

**Fig. 8.** A detail of compound image PP1209, encoded with MMP-Intra at 0.61 bpp

the original deblocking techniques are applied to these images. This can be confirmed in figure 8, where the perceptual results for compound image PP1209 are presented.

Figure 8 c) also shows that the use of the new strategies as a simple post processing deblocking algorithm, allows for a deblocking effect that improves the subjective quality of the decoded image, at the cost of a slight reduction in the PSNR value. In addition, it shows that the second type of artifacts introduced by the original method, that introduce a smoothing ramp in areas of the image that originally had an abrupt variation, is efficiently eliminated by the proposed algorithm (in this example, the value of $s$ was set to 32).

Figure 9 shows the PSNR results for compound image PP1209, for the case presented in figure 8, where the deblocking process allows for an increased perceptual image quality, at the cost of a small reduction of the PSNR value. As we can see, even in this case the objective quality achived by the MMP-Intra encoder is still about 1 dB better than that of the H.264/AVC encoder and 2 dB better than that of the JPEG2000 encoder.

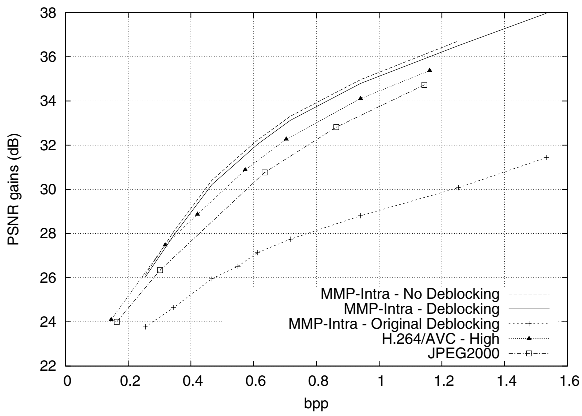



**Fig. 9.** Objective quality results for image PP1209 (adaptive deblocking filter used $\alpha = 0.10$ and $s = 32$)

## 6 Conclusions

In this paper we present a new adaptive deblocking technique that allows for improvements in both perceptual and objective quality, for the MMP-Intra image encoding algorithm. This method uses a space-variant FIR filter with an adaptive shape and support Gaussian impulse response. The filter parameters are automatically controlled in order to maximise the objective quality for smooth images and eliminate the disturbing blurring artifacts for non smooth images, like text and graphics.

The use of the new deblocking techniques achieves one of the main objectives in the on going research of multiscale recurrent pattern image encoders: finding ways to improve the algorithm's performance for smooth images, without compromising its excellent performance for non low-pass images, like text and graphics. Experimental results have shown that, for smooth images, the proposed techniques allows for coding gains that go up to 0.3 dB for low bit-rates, where the blocking artifacts are more noticeable, achieving the same objective quality as the JPEG2000 algorithm. Nevertheless, for non low-pass images, like text and graphics, the proposed method introduces no losses, allowing MMP-Intra to maintain its 1 to 5 db advantage over the state-of-the-art image encoders.

## References


1. de Carvalho, M., da Silva, E., Finamore, W.: Multidimensional signal compression using multiscale recurrent patterns. Elsevier Signal Processing (82), 1559–1580 (2002)
2. de Carvalho, M., Lima, D.M., da Silva, E., Finamore, W.: Universal multi-scale matching pursuits algorithm with reduced blocking effect. In: IEEE International Conference on Image Processing (2000)
3. Joint Video Team (JVT), ISO/IEC MPEG & ITU-T VCEG, I.J..I.T.S.Q.: 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)
4. Rodrigues, N.M.M., da Silva, E.A.B., de Carvalho, M.B., de Faria, S.M.M., Silva, V.M.M.: Universal image coding using multiscale recurrent patterns and prediction. In: IEEE International Conference on Image Processing (2005)
5. Rodrigues, N.M.M., da Silva, E.A.B., de Carvalho, M.B., de Faria, S.M.M., Silva, V.M.M., Pinagé, F.: Efficient dictionary design for multiscale recurrent patterns image coding. In: ISCAS 2006 IEEE International Symposium on Circuits and Systems (2006)
6. Taubman, D.S., Marcelin, M.: JPEG2000: Image Compression Fundamentals, Standards and Practice. Kluwer Academic Publishers, Dordrecht (2001)