Image Predictive Coding using Multiscale Recurrent Patterns

Nuno Rodrigues[1,2], Eduardo da Silva[3], Murilo de Carvalho[4], Sérgio de Faria[1,2], Vítor Silva[1,5]

[1]Instituto de Telecomunicações, Portugal; [2]ESTG, Instituto Polit´ecnico de Leiria, Portugal; [3]PEE/COPPE/DEL/EE, Univ. Fed. do Rio de Janeiro, Brazil; [4]TET/CTC, Univ. Fed. Fluminense, Brazil; [5]DEEC, Univ. de Coimbra, Portugal.

e-mails: nuno.rodrigues@co.it.pt, eduardo@lps.ufrj.br, murilo@telecom.uff.br, sergio.faria, vitor.silva@co.it.pt.

Abstract **– The multidimensional multiscale parser (MMP) algorithm was originally proposed as a multidimensional signal lossy compression method. This method uses a set of patterns stored in a dictionary, that are scaled to approximate each variable-length segment of a multidimensional signal. Thus, the input signal is encoded using only a set of flags, that define the segmentation of the original message, and the dictionary indexes that encode each of these vectors. MMP has proved to be very effective for encoding image data, achieving, for certain classes of images, like texts and graphics, relevant gains over state-of-the-art algorithms, like SPIHT and JPEG2000 [1].**

In this paper we present a new method, based on the MMP algorithm, that allies the use of multiscale recurrent patterns with predictive coding techniques. Experimental results show that this new method is able to achieve relevant gains over the original MMP for smooth images, while maintaining better results than state-of-the-art encoders, like JPEG2000, for text and compound images.

I. Introduction

In spite of the well known intrinsic limitations of the transformed-based encoding methods, this class of algorithms is generally considered as state-of-the-art, both in image and video compression. These methods assume that the image has a low-pass nature and expect most of the transform coefficients for the higher frequencies to be negligible or of little importance. This is then exploited by using coarse quantisation or by simply ignoring these frequency components.

When used to encode images with frequency distributions other then low-pass, like text and graphics, the efficiency of these methods deteriorates noticeably.

The method we present, not only does not rely on the transformation-quantisation-encoding paradigm, but it doesn't make any assumptions about the nature of the image being coded. It is built upon an algorithm that was recently proposed for image coding, called Multidimensional Multiscale Parser (MMP) [1], because it uses an adaptive dictionary of vectors to approximate variable-length input vectors. These vectors result from recursively parsing an original input block of the message. Scaling transformations are used to adapt the size of each element of the dictionary into the size of the block segment that is being considered.

The parsing of the original message block originates a segmentation tree that represents the encoded block. This tree is optimised using an R-D optimisation algorithm and then encoded using an adaptive arithmetic encoder.

Further developments of the base method [2] used an intermediate dictionary, that allowed the encoding process to be a replica of the optimisation process.

In this paper we introduce a new development to this method, called PC-MMP (Predictive Coded Multidimensional Multiscale Parser). This new method combines predictive coding techniques, like those used in H.264 Intra-coding prediction [3], with MMP. Experimental results show that this new method is able to achieve coding gains over the original MMP.

In the next section the MMP algorithm for image coding is described. Section III presents the PC-MMP method, discussing the joint use of Intra-like prediction schemes and MMP. In section IV some experimental results are shown and section V presents the conclusions.

II. The MMP Algorithm

The MMP algorithm was initially proposed as a generic lossy data compression method, but was promptly applied to image coding. Since then, several other applications have been successfully developed, like electrocardiogram data encoding and stereoscopic image encoding [4]. In this section we'll describe the most important aspects of the MMP algorithm applied to the encoding of images. An exhaustive description of the method can be found in [1].

MMP is based on approximations of data segments (in this case image blocks), using words of an adaptive dictionary $\mathcal{D}$ at different scales. For each block in the image, X^l, of size m×n, the algorithm first searches the dictionary for the element S_i^l that minimises the squared error between X^l and S_i^l, $d(X^l, S_i^l)$. The superscript l means that the block X^l belongs to $level\ l$ of the segmentation tree, that corresponds to a known block size.

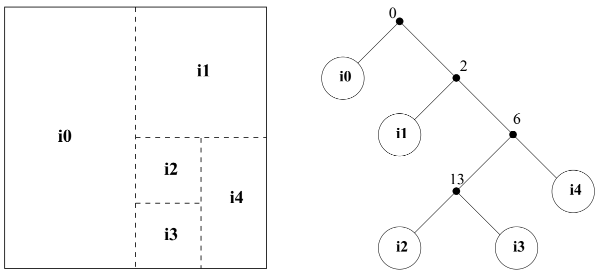
The algorithm then segments the original block into two blocks, X_1^{l-1} and X_2^{l-1}, with half the pixels of the original block, and searches the dictionary for the elements $S_{i_1}^{l-1}$ and $S_{i_2}^{l-1}$, that minimise $d(X_1^{l-1}, S_{i_1}^{l-1})$ and $d(X_2^{l-1}, S_{i_2}^{l-1})$, respectively.

In its simpler version, MMP decides whether to segment the original block or not, based on the previously determined distortion values. If the algorithm decides that the block should not be segmented, then the original block is approximated by one word of the dictionary (S_i^l). If MMP decides to segment the block, the original block is approximated by the concatenation of two smaller words of the dictionary ($S_{i_1}^{l-1}$ and $S_{i_2}^{l-1}$).

This process is repeated recursively until l=0, which corresponds to a block size of 1×1 pixels, and doesn't allow further segmentation.

This recursion results in a binary segmentation tree, that represents the consecutive decisions to segment or not to

The first author would like to thank the support of *Fundação Calouste Gulbenkian*, for the grant of a "bolsa de curta duração", that helped to support this work.



(a) (b)

Figure 1 - Example of segmentation and coding of an image block: a) segmentation; b) binary tree used for coding the block.

segment each sub-block, and a set of indexes, that identify the words of the dictionary that should be used to approximate each of the sub-blocks that correspond to the leafs of the tree.

Figure 1 represents the segmentation of an example block and the segmentation tree that MMP uses to encode it. In this example, $i_0 \ldots i_4$ are the indexes that were chosen to encode each of the sub-blocks.

Each node of the binary segmentation tree represents a segmentation and each leaf represents a sub-block that is not segmented, i.e., it's approximated using one word of the dictionary. A binary flag '0' is used to represent a block segmentation. Flag '1' indicates that the current sub-block shouldn't be segmented and is always followed by the index that was used to encode it, except for block segments of level 0 (blocks of size 1×1). When the segmentation reaches level 0, each pixel can't be further divided and in this case it's not necessary to transmit the flag '1'. This means that for each block of level 0, only the dictionary index is transmitted.

The binary segmentation tree is encoded using a top-down approach. For each node, the sub-tree that corresponds to the left branch is first encoded, followed by the right branch sub-tree. Considering once again the example presented in figure 1, and assuming that the original block size is 4×4, this block would be encoded using the following string of symbols, that are encoded using adaptive arithmetic encoding,

$$0 \quad 1 \quad i_0 \quad 0 \quad 1 \quad i_1 \quad 0 \quad 0 \quad i_2 \quad i_3 \quad 1 \quad i_4.$$

A. Approximate block matching with scales

Unlike conventional vector quantisation (VQ) algorithms, MMP uses *approximate block matching with scales*. This block matching approach is an extension of the ordinary approximate pattern matching, in the sense that it allows the matching of vectors of different lengths.

In order to do this, MMP uses a scale transformation T_N^M to adjust the vectors' sizes, before trying to match them. For example, if we want to approximate an original block X^l, of level l and size $m^l \times n^l$, using one block S^k of the dictionary, with size $m^k \times n^k$, MMP first determines $S^l = T_k^l[S]$, a scaled version of S that has the same size of X. In other words, S^l is a version of S at level (or *scale*) l. Detailed information about the used transformations and their application in MMP is presented in [1].

In [1] the authors present a comparative study on the performance of VQ methods, whose dictionary is built from blocks of the input signal, compared to that of VQ methods whose dictionary is built using scaled versions of input blocks. The presented results give the indication that the use of dictionaries with scales can be advantageous.

B. The adaptive dictionary

MMP uses an adaptive dictionary that is updated while the data is encoded, with blocks that were used to approximate the original image data. As it has been seen, each input block of image data, X, is parsed into a set of non-overlapping segments, which are approximated by blocks of the dictionary that were scaled to the dimension of each of them.

When a given data block, X^l, is segmented, instead of being approximated by a single word of the dictionary, this block is approximated by the concatenation of the words used to approximate each of it's halves. Returning once again to the example of figure 1, the block of level 1, that corresponds to node 13 of the binary tree, was segmented. This means that each of it's halves will be encoded separately, using indexes i_2 and i_3 of level 0, respectively. After MMP encodes these two blocks, a new block $S^1_{i_2:i_3}$, resulting from the concatenation of the blocks $S^0_{i_2}$ and $S^0_{i_3}$ is used to update the dictionary.

Every time a block is approximated by the concatenation of two dictionary blocks, of any given level, the resulting block is used to update the dictionary, becoming available to encode the next blocks of the image, independently of their size.

This updating procedure for the dictionary uses only information that can be inferred exclusively from the encoded segmentation flags and dictionary indexes. This means that the decoder is able to keep an exact copy of the dictionary used by the encoder, using no further information.

C. R-D optimisation

The R-D optimisation is done using the *Lagrangian multiplier*, λ. This allows the algorithm to weight both the minimum distortion achieved by the approximation of one block X^l (encoded with a segmentation tree $\mathcal{T}$), $D(\mathcal{T})$, and the rate needed to encode $\mathcal{T}$, $R(\mathcal{T})$.

The problem of minimising the rate $R(\mathcal{T})$ for a given maximum distortion D (or alternatively, determining the segmentation tree that allows a minimum distortion for a given maximum rate), is solved by minimising the value of a cost function $J(\mathcal{T})$, given by:

$$J(\mathcal{T}) = D(\mathcal{T}) + \lambda R(\mathcal{T}). \tag{1}$$

The decision to segment or not a block corresponding to a node j in the segmentation tree, having two leaf nodes $2j+1$ and $2j+2$ is then made evaluating the function:

$$J(n_j) \leq J(\mathcal{T}_{2j+1}) + J(\mathcal{T}_{2j+2}) + \lambda R_F(0), \tag{2}$$

where $J(n_j)$ is the cost of encoding the node n_j with only one block of the dictionary, $J(\mathcal{T}_{2j+1})$ and $J(\mathcal{T}_{2j+2})$ are the costs of encoding each of the left and right sub-trees and

$R_F(0)$ is the rate required to encode the segmentation flag '0'.

Because the costs of each tree node are not independent (they are in fact coupled by the dictionary updating procedure), some further considerations must be made for this optimisation procedure. Recently, a new, simpler, method of R-D optimisation of MMP, using an intermediate dictionary, was presented in [2]. This method is called MMP-RDI and produces very similar results, compared with full search MMP with RD optimisation, using a much simpler algorithm. From now on, this algorithm will be used as reference for MMP.

III. The PC-MMP Algorithm

Predictive coding with multiscale multidimensional parsing (PC-MMP) was developed based on the MMP-RDI algorithm presented before. In this new algorithm we use predictive methods, based on the techniques for intra-frame prediction present in the H.264 standard [3], to generate a predicted signal, that is encoded using MMP-RDI.

The use of predictive coding changes the patterns that are encoded by MMP. When prediction is successful, the blocks that are encoded tend to be more uniform. This favours the adaptation of the dictionary and improves approximation of the encoded blocks, resulting in a more efficient method.

In PC-MMP, the predictive coding of one image block refers to neighbouring samples of previously-coded blocks, which are to the left and/or above the block to be predicted. For each of the considered prediction modes, if the prediction pixels used by that mode are available, PC-MMP first determines the prediction block and the respective residue values. The block with the residue pixels is then encoded using MMP-RDI. All the coding decisions follow the same rules presented in section II, in order to determine the final distortion and rate needed to encode the residue block corresponding to each of the prediction modes. The algorithm then decides which of the prediction modes should be used, based on R-D considerations that follow an adapted version of the R-D optimisation algorithm presented earlier.

Because PC-MMP uses a number of different prediction modes, the prediction mode that was chosen for each block must be transmitted to the decoder. This prediction mode flag is encoded using adaptive arithmetic coding and transmitted before the segmentation tree used to approximate the residue block. The decoder is then able to determine the corresponding prediction block, based on the same previously-coded neighbour pixel blocks, and decode the approximation of the residue block. The decoder then simply adds the decoded residue block with the prediction pixels, resulting in the decoded image block.

The encoding algorithm used by PC-MMP can thus be described as follows:

For each m×n block of the original image, X^l:

1. Determine which prediction modes can be used to predict this block, based on the available prediction pixels.
2. For each of the available prediction modes, m_i:
   - i Determine the corresponding prediction block, P_i, and residue block, Q_i.
   - ii Encode Q_i using MMP-RDI, calculate the approximated (decoded) residue block $\hat{Q}_i$ and determine the segmentation tree used in the approximation, $\mathcal{T}_{m_i}$.
   - iii Determine the decoded image block, $\hat{X}_i = P_i + \hat{Q}_i$, the corresponding distortion $D(\hat{Q}_i) = d(X, \hat{X}_i)$ and the rate, $R(\hat{Q}_i)$ needed to encode the block approximation.
   - iv Calculate the R-D cost of the approximation, $J(m_i)$, given by:
     $$J(m_i) = D(\hat{Q}_i) + \lambda R(\hat{Q}_i) + \lambda R_F(m_i),$$
     where $R_F(m_i)$ is the rate used to encode the flag corresponding to mode m_i.
3. Determine m_b, the mode with the minimum R-D cost value.
4. Encode and transmit the prediction mode flag for m_b.
5. Encode and transmit the segmentation flags and dictionary indexes of $\mathcal{T}_{m_i}$.

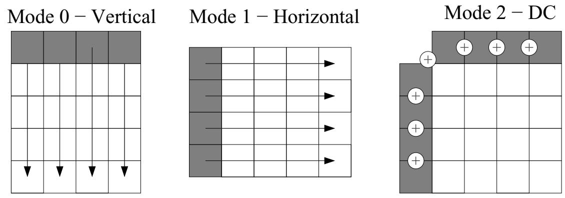


Figure 2 - Three of the prediction modes: Vertical, Horizontal and DC

IV. Experimental Results

The algorithm presented in the previous section was implemented and some experimental tests were performed. We used PC-MMP with image blocks of size 16×16 (level 8). This block size was chosen as a compromise between good prediction performance (achieved with small block sizes) and additional overhead for the prediction mode flags. This consideration is important, because PC-MMP transmits one extra prediction mode flag for each block, compared with standard MMP-RDI.

When used to encode prediction error blocks, MMP uses an initial dictionary in the scale 1×1 (level 0) with the integer values in the range [-255; 255]. The initial dictionaries for the remaining levels are obtained from this one by scale transformation. The scale transformation and dictionary update procedures are the same as the ones described in [1].

In the first implementation of PC-MMP, we used the four prediction modes defined in H.264 for $Intra\_16 \times 16$ coding. These four prediction modes, *Vertical, Horizontal, DC* and *Plane* prediction, were originally defined for image blocks of size 16×16, and were used in PC-MMP exactly as defined in the standard. The Vertical, Horizontal, and DC prediction modes are represented in figure 2, for blocks 4×4. We also implemented PC-MMP using the nine prediction modes defined by H.264 for $Intra\_4 \times 4$ coded blocks. In this case, the prediction methods were adapted in a straightforward way, in order to allow their use with larger block sizes. Additional details about all used prediction modes can be found in [3].

Figures 3 to 5 present the experimental results for test im-

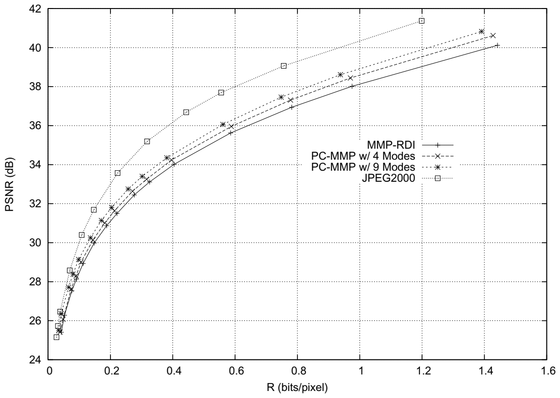


Figure 3 - Experimental results for image lena 512×512.

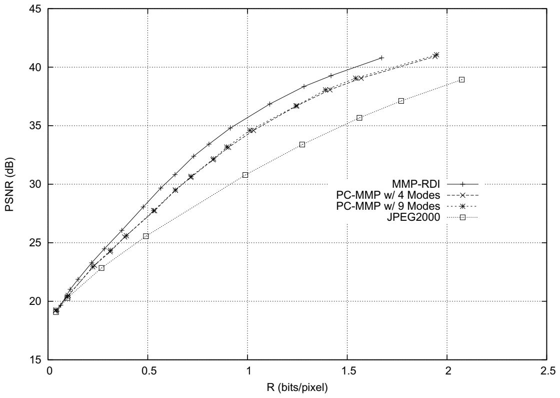


Figure 4 - Experimental results for image PP1205 512×512.

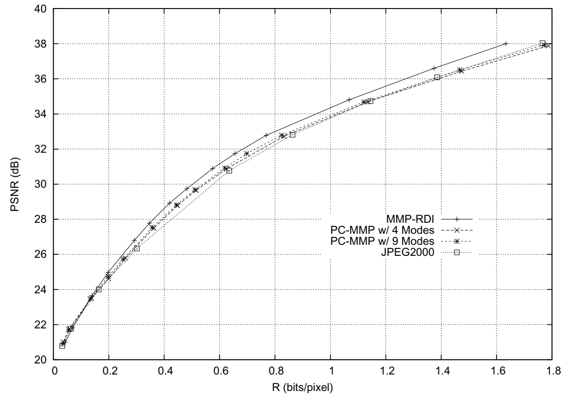


Figure 5 - Experimental results for image PP1209 512×512.

ages LENA, PP1205 and PP1209. PP1205 is a text image and PP1029 is a compound (text and grayscale) image. They 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 [5]. These images were encoded using the MMP-RDI algorithm, JPEG2000 [6] and the new PC-MMP, using the two sets of prediction modes described above.

Figure 3 clearly shows that the use of predictive schemes, associated with MMP, result in an better objective quality, when encoding smooth grayscale images. In these cases, the quality gain is more evident for smaller compression ratios, where the use of PC-MMP allows an increment of about 1 dB over the original MMP-RDI, but is still worse than JPEG2000. It is also clear that the use of nine prediction modes improves the quality of the coded image, compared with the results of PC-MMP with four prediction modes.

For text and mixed images, like PP1025 and PP1209, the use of simple intra prediction schemes results in a small loss of quality. Simple intra prediction schemes, like the ones used in these tests have some difficulty in the prediction of text images. Because of this, the blocks with the prediction residues, that are encoded by MMP, aren't as uniform as in the case of smooth images. Because PC-MMP has the additional overhead associated with the transmission of the mode flags, this can result in a small decrease in quality. In spite this, for these images PC-MMP is still consistently better than state-of-the-art JPEG2000 [6].

These results show that PC-MMP is able to improve the quality of MMP for smooth images, while maintaining very good results for text and compound images, when compared with state-of-the-art encoders.

At this time, new prediction schemes are being tested with MMP. These methods are expected to increase the performance of PC-MMP both for smooth and text images, in order to allow PC-MMP to maintain the excellent results of MMP for text images, while improving MMP performance for smooth images, when compared with transform-based encoders.

V. CONCLUSIONS

In this paper we presented a new image encoding method, that uses predictive coding (namely, the prediction methods defined in the H.264 standard for intra frames) and is based on the multiscale multidimensional parser (MMP) algorithm [1]. These results show that PC-MMP is able to improve the quality of MMP for smooth images, while maintaining very good results when used with text and compound images, when compared with state-of-the-art encoders. These results are encouraging, and further studies continue to be developed.

REFERENCES

[1] M. de Carvalho, E. da Silva, and W. Finamore, "Multidimensional signal compression using multiscale recurrent patterns", *Elsevier Signal Processing*, , no. 82, pp. 1559–1580, November 2002.

[2] Eddie de Lima Filho, "Compressão de imagens utilizando recorrência de padrões multiescalas com crit´erio de continuidade inter-blocos", Master's thesis, COPPE - Universidade Federal do Rio de Janeiro, Abril 2004, *http://www.lps.ufrj.br/profs/eduardo/teses/eddie-filho-msc.pdf*.

[3] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, "Draft itu-t recommendation and fi nal draft international standard of joint video specifi cation (itu-t rec. h.264/iso/iec 14 496-10 avc", JVT-G050 2003.

[4] M. H. V. Duarte, M. B. Carvalho, E. A. B. da Silva, C. Pagliari, and G. V. Mendona, "Stereo image coding using multiescale recirrent patterns", *IEEE International Conference on Image Processing, Rochester, New York*, pp. II661–664, September 2002.

[5] http://www.estg.ipleiria.pt/∼nuno/MMP/.

[6] D. S. Taubman and M.W. Marcelin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, 2001.