Multiscale Recurrent Pattern Predictive Image Coding with Template Matching

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

[1]Instituto de Telecomunicações, Portugal; [2]ESTG, Instituto Politécnico 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, sergio.faria, vitor.silva@co.it.pt, eduardo@lps.ufrj.br, murilo@telecom.uff.br.

***Abstract* – In this paper we describe the latest developments on the recently proposed MMP-Intra image coding algorithm. MMP-Intra combines predictive coding techniques with a new image coding paradigm, based on the Multidimensional Multiscale Parser (MMP) algorithm.**

The use of new prediction techniques, based on template matching strategies, is discussed in this text. This method improves consistently the results of MMP-Intra, specially for smooth images.

The experimental results presented in the latest sections of this paper demonstrate that MMP-Intra is able to achieve important coding gains for non-smooth images, when compared with state-of-the-art, transform-based methods, with no relevant performance losses when used for smooth images.

I. Introduction

The study of new coding paradigms as an alternative to the traditional methods based on image transforms, quantisation and entropy coding has recently shown some very interesting results. In this context, a new coding paradigm based on recurrent pattern matching using an adaptive dictionary with scale transformations was proposed in [1]. The described algorithm, named Multidimensional Multiscale Parser (MMP), achieves good results for non-smooth images, but presents some performance losses for smooth images, when compared with standard image encoders.

A primary version of a new image coding method based on MMP has been presented [2], that aimed at improving its performance for smooth images, without compromising its excellent results for text and compound images. The first results of this method, named Predictive Coded MMP (PC-MMP), were able to improve on the performance of MMP for smooth images but presented some losses for other image types.

Since then some important improvements were made in the original algorithm of [2]: adaptive block size prediction was developed; new dictionary update strategies were employed and more efficient encoding strategies were studied, resulting in a highly efficient coding scheme, named MMP-Intra.

In this paper we present both a brief summary of the improvements of our work since [2], as well as the most recent developments of MMP-Intra: the use of template matching based prediction methods, that results in consistent gains in encoding performance, mainly for smooth images.

Section II presents a brief description of the original MMP algorithm and of the PC-MMP method. Section III briefly describes the most important developments introduced by our investigation, that resulted in a new image coding algorithm, that we refer to as MMP-Intra. Section IV presents the use of a new prediction technique in MMP-Intra, based on template matching strategies. The experimental results of the new proposed method are presented in section V, where the investigated methods are compared with the most important state-of-the-art encoding methods: JPEG2000 and the H.264/AVC *High* profile intra encoder. Finally, section VI presents some conclusions of this work.

II. The MMP and PC-MMP algorithms

This section presents a brief description of the MMP algorithm applied to image coding and the first version of MMP with predictive coding, PC-MMP.

A. The MMP algorithm.

MMP encodes each original image block by approximating it with a vector from an *adaptive dictionary* $\mathcal{D}$. This is done using *different scales*, meaning that blocks of different dimensions can be approximated by this procedure. These dimensions correspond to successive binary segmentations of an original square block, first in the vertical, then in the horizontal direction. The superscript l means that the block X^l belongs to $level\ l$ of the segmentation tree (with dimensions $(2^{\lfloor\frac{l+1}{2}\rfloor} \times 2^{\lfloor\frac{l}{2}\rfloor})$).

MMP can be summarised by the following main steps:
For each block of the original image, X^l:

1. find the dictionary element S_i^l that minimises the Lagrangian cost function of the approximation, given by: $J(\mathcal{T}) = D(X^l, S_i^l) + \lambda R(S_i^l)$, 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, X_1^{l-1} and X_2^{l-1}, with half the pixels of the original block;
3. apply the algorithm recursively to X_1^{l-1} and 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 S_i^l of the dictionary to approximate X^l;
6. else
   (a) create a new vector S_{new}^l from the *concatenation* of the vectors used to approximate each half of the original block: X_1^{l-1} and X_2^{l-1};
   (b) use S_{new}^l to approximate X^l;
   (c) use S_{new}^l to *update* the dictionary, making it available to encode future blocks of the image.

When applied recursively, this algorithm generates a binary segmentation tree for each original block, that is encoded using two binary flags ('0' for the tree nodes, or block segmentations and '1' for tree leafs, or unsegmented blocks).

This binary tree is encoded using a top-bottom preorder approach. In the final bit-stream, each leaf flag is followed by an index, that identifies the vector of the dictionary that should be used to approximate the corresponding sub-block. These items are encoded using an adaptive arithmetic encoder.

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

The use of an adaptive dictionary is illustrated by the final step of the previous algorithm. Every segmentation of a block from level l originates the concatenation of two dictionary blocks of level $l-1$. The resulting block is used to update the dictionary, becoming available to encode future blocks of the image, independently of their size. This updating procedure for the dictionary uses only information that can be inferred by the decoder, since it uses exclusively the encoded segmentation flags and dictionary indexes. This means that no extra overhead is used for dictionary updating.

Block matching with scales allows the matching of vectors of different lengths. In order to do this, MMP uses a separable scale transformation T_N^M to adjust the vectors' sizes before attempting to match them. For example, in order to approximate an original block X^l using one block S^k of a different scale of the dictionary, MMP first determines $S^l = T_k^l[S]$. Detailed information about the use of scale transformations in MMP is presented in [1].

B. The PC-MMP algorithm.

PC-MMP combines the original MMP algorithm with intra-frame prediction techniques, like the ones used in the H.264/AVC standard [3]. A straightforward algorithm is used:

For each original fixed-sized image block, X:

1. for each prediction mode m:
   (a) determine the prediction block, P_m, and the respective residue values, Q_m;
   (b) encode the residual block Q_m using MMP;
   (c) determine the mode, M, that minimises the Lagrangian cost function for the current block;
2. encode the best prediction mode for the current block;
3. encode the MMP data for the residual block Q_M.

PC-MMP uses fixed blocks of 16×16 pixels and the same prediction modes used by H.264/AVC intra coded blocks.

III. The MMP-Intra Algorithm

The simple prediction scheme used by PC-MMP revealed some inefficiency when used with non smooth images, that leads to some quality losses in these cases. Since the introduction of this algorithm further investigation was conducted in order to find convenient ways to:

- improve the prediction efficiency;
- reduce the entropy of the dictionary indexes' symbols, by limiting the redundancy among dictionary elements;
- improve the dictionary adaptation process;
- increase the efficiency of the arithmetic encoding using adaptive contexts.

This investigation originated a new method, that we refer to as MMP-Intra. The main techniques developed for each of the previous topics are briefly described in this section.

The *prediction process* used by MMP-Intra is different from the one of PC-MMP in two aspects: MMP-Intra uses a adaptive block size prediction, with blocks of dimensions 16×16 down to 4×4. The Lagrangian R-D cost function, allows the encoder to optimise the prediction step, determining the best trade-off between the prediction accuracy and the additional overhead introduced by the prediction data. MMP-Intra also uses a new prediction mode: the most frequent value (MFV) among those pixels used for the prediction is used instead of the DC mode, that uses the average value of those pixels. Experiments have shown that for text and graphic images, the use of the MFV for the prediction has the advantage of creating prediction error blocks consistently centred around zero, enhancing the overall coding efficiency. In addition, for smooth images, the use of the MFV instead of DC prediction has no effect on the performance [4].

New and more *efficient dictionary design strategies* were also developed. MMP-Intra uses an initial dictionary consisting of a few blocks with constant value. This highly sparse initial dictionary is obviously not efficient for coding images, but the updating procedure quickly adapts the patterns in the dictionary to the typical patterns that are being encoded.

Experimental tests 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 in the coding. This discrepancy grows with the target bit rate and such observations are valid across images and target bit rates, indicating that there is a high redundancy among dictionary elements, that compromises the coding efficiency.

In order to reduce this redundancy a test condition was added to the dictionary update procedure, to ensure that the average quadratic distortion between each new block of level l, B^l_{new}, and the ones already available in the dictionary is not inferior to a given threshold d. Details about this technique can be found in [5].

In [6] a new technique to improve the dictionary adaptation process of the MMP-Intra based on enhanced updating techniques is described. The main idea behind this procedure is to provide the MMP dictionary with a richer set of patterns, that continue to be related to the originally created block and increase the approximation power of the dictionary. Experimental results have showed that, when combined with the redundancy control procedure, this technique achieves consistent quality gains for all image types.

In [5] a new context adaptive arithmetic encoder for the dictionary indexes is also described. With this method, the dictionary indexes are divided into groups, according to a context criterion, that, for MMP-Intra, is the original scale of the block. A different histogram is kept for each group and used in the arithmetic coding of its symbols, in order to further exploit their statistical dependencies, generating gains in the arithmetic coding module.

IV. MMP-Intra with Template Matching

As was previously mentioned, the MMP-Intra algorithm uses a set of prediction modes inspired by the ones defined in the H.264/AVC video encoder. These modes, that determine a prediction block based on the reconstructed val-

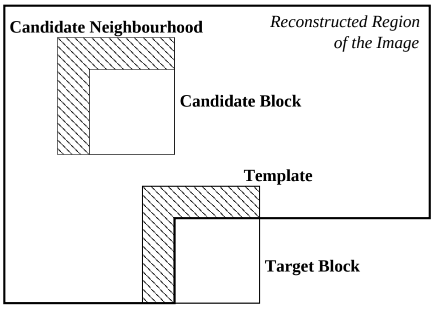

Figure 1 - Intra prediction using template matching.

ues of the neighbour pixels of the target block, have proved to be generally efficient. Nevertheless, two problems were observed: first, they loose efficiency in predicting blocks with complex texture and second, pixels of the current target block that are positioned far from the surrounding pixels, and thus are less correlated with the prediction values, are usually poorly predicted [7].

Template matching (TM) was originally proposed as a texture synthesis algorithm [8], but has been successfully applied to the slightly different goal of spatial prediction in a video encoding framework. Previous works describe advantageous implementations of this method as inter [9] and intra [7] frame prediction schemes, in a H.264/AVC video encoder.

A. Template Matching Prediction

When using template matching for intra prediction, the current block, B, is estimated by comparing its spatial neighbourhood, $N(B)$, with all the neighbourhoods in the reconstructed area of the image. $N(B)$ is called the *template* of B. The block of the reconstructed area of the image, with the most similar neighbourhood to the template is assigned as the prediction block of B (figure 1). A sum of absolute difference (SAD) is used to access the best match between the candidate neighbourhood and the template.

Template matching prediction can be related with a motion compensation using previously reconstructed pixels from within the intra picture, but with the advantage of not needing additional overhead associated with the motion vectors. The same template matching prediction step is performed in the encoder and the decoder, that uses the same search area of the reconstructed image to determine the same prediction block used by the encoder.

B. The use of Template Matching in MMP-Intra

In the original method proposed in [7], a fixed block size of 4×4 is used, but the template matching prediction is performed for four 2×2 target sub-blocks. The predictor block for the target 4×4 block is then made up from the concatenation of the four best match 2×2 candidate sub-blocks. Unlike H.264/AVC, MMP-Intra uses an adaptive block size for prediction, with blocks that range from 4×4 to 16×16. Several tests were performed for MMP-Intra regarding the use of sub-blocks for the template matching step. These experiments showed that the best results for MMP-Intra are achieved when no sub-blocks are used, *i.e.*, when the used template corresponds to the entire predicted block, independently of its size.

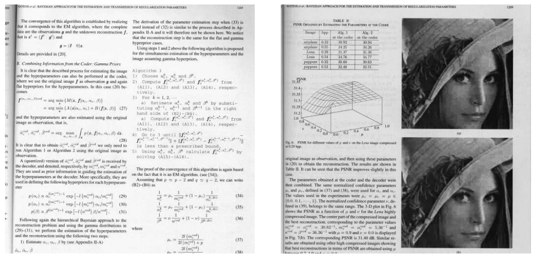

Figure 2 - Test images PP1205 (left) and PP1209 (right).

The TM prediction mode was used as an additional mode in the previously described MMP-Intra image encoder. The adaptive arithmetic encoder was adapted to cope with this new encoded value. A search region of 24 columns and 32 rows centred around the current block (figure 1) was used for all block scales used in the prediction. Besides these issues, the use of TM with MMP-Intra is quite straightforward and all optimisation techniques described in section III are compatible with this extension.

V. Experimental Results

The experimental results presented in this section compare the encoding performance of the original MMP algorithm [1], the PC-MMP method proposed initially in [2] and MMP-Intra with new TM prediction mode. Results for two state-of-the-art image encoding algorithms JPEG2000 [10] and H.264/AVC *High* profile intra encoder [3] are also presented.

R-D results are plotted for the same test images used in [2]: smooth image LENA (from [11]), text image PP1205 and compound (text and grayscale) image PP1209. Images PP1205 and PP1209 (see figure 2) 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 [12]. The presented plots are limited to rates where the visual distortion of the compressed images does not make them unusable, specially for the images with text.

Experimental observations show that the use of the template matching prediction mode increases the performance of MMP-Intra for smooth and compound images for all compression ratios and achieves an equivalent performance for the text image. The coding gains for smooth and compound images are about 0.1 dB. This gain adds to the ones achieved by the previously presented techniques and reduces the gap to JPEG2000 and H.264/AVC for image LENA. In fact, for compression ratios up to 0.25 bpp, the performance of MMP-Intra is now equivalent to that of the JPEG2000 encoder for this particular test image.

For text images, the use of the new TM prediction has no effect on the performance of the method. This is due to the highly irregular characteristics of the text images, that decrease the correlation among the neighbour areas of the image. In this case, the matching of the template region is not as accurate as for smooth images and the MMP-Intra encoder tends to use other prediction methods.

Figures 3 to 5 show the efficiency of MMP-Intra when compared with the PC-MMP method, displaying performance gains that go up to 1.5 dB for the LENA image. For the text and compound images, MMP-Intra not only

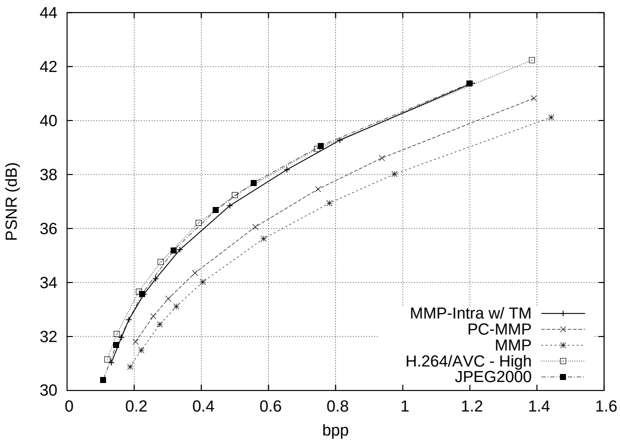

Figure 3 - Experimental results for image LENA.

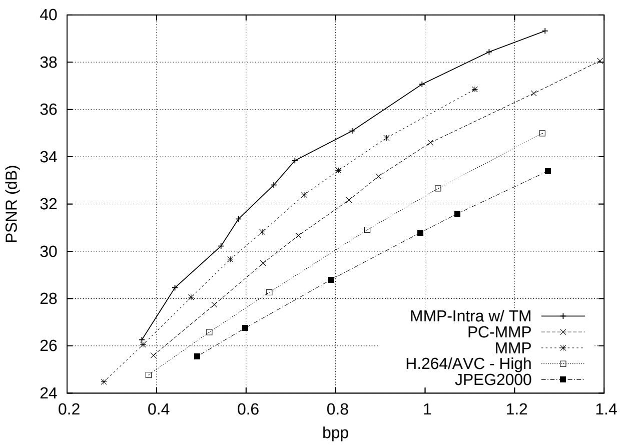

Figure 4 - Experimental results for image PP1205.

eliminates the performance loss in relation to the original MMP algorithm, but it is also capable of achieving significant gains over it. Overall, the PSNR gains of MMP-Intra in relation to PC-MMP are more then 2 dB for text image PP1205 and up to about 2 dB for compound image PP1209.

For non-smooth test images MMP-Intra presents considerable gains over the transform-based image encoders. These gains can be as much as 4 or 5 dB for text image PP1205, when compared to H.264/AVC and JPEG2000, respectively and range to about 1.5 to 2 dB for compound image PP1209, when compared with the same encoders.

VI. Conclusion

In this paper we present the use of a new prediction method for the MMP-Intra image encoder, based on template matching. The presented method allows for consistent gains in the compressed image quality, specially for smooth images.

MMP-Intra is an evolution of the PC-MMP method with significant performance gains, that result from enhanced prediction and dictionary updating techniques, that are summarised in this paper.

Experimental results that compare MMP-Intra with the JPEG2000 and H.264/AVC *High* profile intra encoder, show that this is an exciting new paradigm and a good alternative to the transform-based methods, in spite of its higher computational complexity.

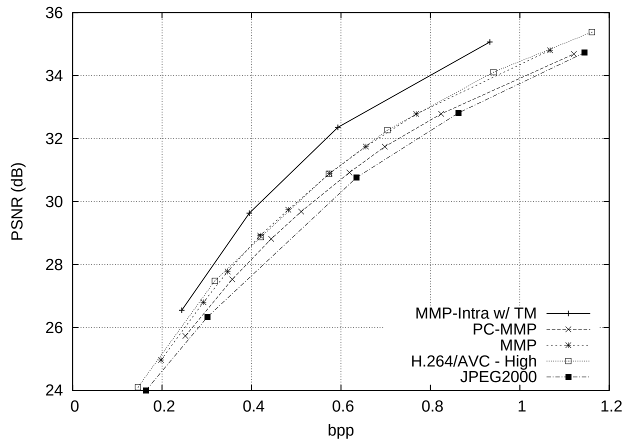

Figure 5 - Experimental results for image PP1209.

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] N. M. M. Rodrigues, E. A. B. da Silva, M. B. de Carvalho, S. M. M. de Faria, and Vitor M. M. Silva, "Image predictive coding using multiscale recurrent patterns", *International Conference on Telecommunications - ConfTele05*, April 2005.

[3] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 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)*, March 2005.

[4] N. M. M. Rodrigues, E. A. B. da Silva, M. B. de Carvalho, S. M. M. de Faria, and Vitor M. M. Silva, "Universal image coding using multiscale recurrent patterns and prediction", *IEEE International Conference on Image Processing*, September 2005.

[5] N. M. M. Rodrigues, E. A. B. da Silva, M. B. de Carvalho, S. M. M. de Faria, Vitor M. M. Silva, and Frederico Pinagé, "Effi cient dictionary design for multiscale recurrent patterns image coding", *ISCAS 2006 IEEE International Symposium on Circuits and Systems*, May 2006.

[6] N. M. M. Rodrigues, E. A. B. da Silva, M. B. de Carvalho, S. M. M. de Faria, and Vitor M. M. Silva, "Improving multiscale recurrent pattern image coding with enhanced dictionary updating strategies", *International Telecommunications Symposium - ITS 2006*, September 2006.

[7] Thiow Tan, Choong Boon, and Yoshinory Suzuki, "Intra prediction by template matching", *IEEE International Conference on Image Processing*, October 2006.

[8] Li-Yi Wei and Marc Levoy, "Fast texture synthesis using tree-structured vector quantization", *Proceeding of SIG-GRAPH 2000*, pp. 479–488.

[9] K. Sugimoto, M. Kobayashi, Y. Suzuki, S. Kato, and C. Seng Boon, "Inter frame coding with template matching spatio-temporal prediction", *IEEE International Conference on Image Processing*, October 2004.

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

[11] http://sipi.usc.edu.

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