Efficient Dictionary Design for Multiscale Recurrent Pattern Image Coding

Nuno M. M. Rodrigues*†, Eduardo A. B. da Silva‡, Murilo B. de Carvalho§,
Sérgio M. M. de Faria*†, Vítor M. M. da Silva*¶ and Frederico Pinagé||
*Instituto de Telecomunicações, Portugal; †ESTG, Instituto Politécnico Leiria, Portugal;
‡PEE/COPPE/DEL/Poli, Univ. Fed. Rio de Janeiro, Brazil; §TET/CTC, Univ. Fed. Fluminense, Brazil;
¶DEEC, Universidade de Coimbra, Portugal; ||FUCAPI, Manaus, Amazonas, Brazil.
E-mails: nuno.rodrigues@co.it.pt, eduardo@lps.ufrj.br, murilo@telecom.uff.br,
sergio.faria@co.it.pt, vitor.silva@co.it.pt, fredpinage@fucapi.br

***Abstract*— MMP-Intra was recently proposed as a recurrent patterns based image encoder that combines the Multidimensional Multiscale Parser (MMP) algorithm with intra prediction techniques. Our results show that this method is able to achieve considerable gains over state-of-the-art transform-based image encoders for a wide variety of types of images, like text, composed (text and graphics) and texture images, while having performance close to the one of traditional algorithms for smooth images. Because of this universal character, MMP-Intra can be regarded as a viable alternative to transform-based image coding.**

MMP-Intra uses a multiscale adaptive dictionary to approximate the original data blocks. It is composed of dilations, contractions and concatenations of previously encoded patterns. In this work we present a new method for controlling the dictionary adaptability, in which the dictionary is only updated if a certain distortion criterion is met in the block being encoded. Experimental results show that this scheme is able to consistently outperform the original method, while achieving relevant reductions in its computational complexity.

I. Introduction

The Multidimensional Multiscale Parser (MMP) algorithm [1] was originally proposed as a multidimensional lossy signal compression method. MMP uses a *multiscale adaptive dictionary* of vectors to approximate variable-length input vectors, that result from parsing an original input block of data. Scaling transformations allow the matching of each dictionary element to the original blocks, that may have different sizes.

MMP is an extremely versatile encoding algorithm, that has shown good results when applied to a wide variety of signal sources, ranging from voice and ECG [2] to stereoscopic images [3], and video [4]. When applied to still digital images, MMP is able to achieve excellent results for non low-pass images, like text, composed and texture images, notably outperforming the traditional state-of-the-art transform-based methods. However, for smooth images its performance is inferior to the one of state-of-the-art methods.

MMP-Intra [5] is a recently proposed encoding method that combines MMP with predictive coding techniques, like those used in H.264/AVC Intra coding. Experimental results show that this version of the MMP algorithm is able to significantly improve the performance of the original method for smooth images *and* maintain its good performance for other types of images, making it a method with universally good results.

Ongoing work in this method is focused mainly on improving it's general performance, with a special interest in the performance for smooth images, as well as reducing the reduction of its computational complexity. In this paper we study a new, improved, dictionary updating procedure, that decreases the redundancy between the dictionary vectors for the various scales. When combined with a context adaptive arithmetic coding of the MMP symbols, this scheme achieves gains of about 0.3 dB over MMP-Intra using the original update procedure, while at the same time reducing the computational complexity of the method. Experimental results show that with the proposed improvements, the rate distortion performance of MMP-Intra is now only marginally bellow JPEG2000 and H.264/AVC *high* profile for coding of smooth images, about 0.2 to 0.5 dB. For other types of images, MMP-Intra consistently maintains its excellent performance, achieving gains over standardised state-of-the-art encoders that range from 1.5 to 5 dB.

In section II we briefly introduce the main aspects of MMP-Intra for image coding. Section III presents and discusses the new dictionary design procedure and context conditioning. In section IV experimental results for this new method are compared with the results of JPEG2000 and H.264/AVC. Finally, section V summarises the conclusions of this work.

II. The MMP-Intra algorithm

This section begins with a brief description of the MMP algorithm applied to image coding and then explains the most important aspects of MMP-Intra. An exhaustive description of these methods can be found respectively in [1] and [5].

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.

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 MMP-Intra algorithm.

MMP-Intra is a successful combination of the original MMP algorithm with intra-frame prediction techniques, like the ones used in the H.264/AVC standard [6]. For each original block, X^l, MMP-Intra determines the prediction block, P_m^l, and the respective residue values, Q_m^l. The block with residual pixels is then encoded using MMP.

MMP-Intra uses the same prediction modes used by H.264/AVC intra coded blocks, with one important exception: the DC mode uses the most frequent value (MFV) among those pixels used for the prediction, instead of the average value of those pixels. Because MMP-Intra encodes the prediction error using blocks from the dictionary, the DC value of the prediction error is not a good choice for it as for the case of H.264/AVC, that uses transform coding. 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.

Intra prediction is used *hierarchically* for blocks of dimensions 16×16, 16×8, 8×8, 8×4 and 4×4 (corresponding to levels 8 to 4 of the segmentation tree). This hierarchical prediction scheme, allied to the use of Lagrangian R-D cost functions, 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.

When compared with the original MMP algorithm, MMP-Intra has to encode some additional information for the block prediction (namely the prediction mode and the block size used for the prediction). Using this information, the decoder is able to decode each image block, by finding the corresponding prediction block and adding it to the decoded residual block. Details about MMP-Intra can be found in [5].

III. Efficient Dictionary design for MMP-Intra

MMP-Intra uses the same dictionary updating procedure as the original MMP. Each time a new block B_{new} is created in a scale l' (by the concatenation of two vectors of level $l'-1$ of the dictionary) the new block becomes available for all scales of $\mathcal{D}$.

To increase the computational efficiency of the method, we keep a copy of the dictionary in every scale. This eliminates the need to apply the scale transformation before each match between the original block and a dictionary vector, hence considerably reducing the complexity of the method. Even so, each time the algorithm searches for a new match, it has to determine the distortion for all the vectors of a given level of the dictionary. This process is, by far, the one that contributes most to the computational complexity of the algorithm.

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.

Observations of the final dictionary sizes of the MMP-Intra process have shown that for high rates the final dictionary sizes are much larger than for lower target bit rates. In fact, practical tests led to the conclusion that the final dictionary size tends to grow linearly with the final bit rate. Nevertheless, further 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 in the coding. This discrepancy also grows with the target bit rate. Such observations are extensively valid across images and target bit rates.

The new dictionary design algorithm targets the redundancy reduction. The basic principle underlying this procedure is to avoid that new vectors, very close to those already available in the dictionary space, will be used to update the dictionary.

Every time a new vector is inserted in the dictionary, an additional index is created for that level of the dictionary. The average bit rate needed to transmit each index of the dictionary thus grows with the total number of indexes. Therefore, the introduction of new blocks in the dictionary that are not beneficial for the approximation of future blocks is a factor that limits the performance of the algorithm. This can be the case for blocks that are too close to each other in the dictionary space and that could be replaced by a single one, in such a way that the additional distortion could be compensated by the reduced bit rate.

In order to avoid this, 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. This introduces a "minimum distance condition" between any two vectors of each level of the dictionary. In fact, this guarantees that, for each level l, there is never more than one block inside any hypersphere of radius $d.N^l$, where N^l is the number of elements of each level l of the dictionary (in our case $N^l = 2^l$).

The value of d must be carefully chosen. If this value is too small, the aim of reducing redundancy will not be achieved, and if it is too large, the dictionary will lose its ability to accurately represent the original patterns, leading to a decrease in the algorithm performance.

In case of MMP-Intra, the optimum value for d is a function of the target bit-rate and therefore of the parameter λ. To determine an efficient relation between these two values for our case, a set of tests were performed, where we measured the performance of the method for a large range of λ and different values of d. The test results were then plotted in an R-D chart and the convex hull (that contains the points that optimised the $D(R)$ function) was determined. Observing the relation between d and λ for the points in the convex hulls of the performed tests, we were able to determine a simple model to the function $d(\lambda)$:

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

Equation 1 allows the encoder to automatically achieve a close to optimum R-D relation, for any given target bit-rate.

The new dictionary design procedure also uses a context adaptive arithmetic encoder for the dictionary indexes. This is achieved by, instead of indexing the entire dictionary, dividing the indexes into groups according to the original scale of the block that, by a scale transformation, originated the vector corresponding to the index. This means that each group l contains only those blocks that were inserted in the dictionary resulting from a new vector of scale l. Using such procedure, each index is transmitted using one symbol specifying the group, followed by another symbol that points to one index inside that group. The current tree level is used as a context for the arithmetic coding of the segment symbol, while the index uses the scale and the dictionary segment as contexts, better exploring the statistical dependencies of the MMP symbols.

IV. Experimental Results

The proposed dictionary design algorithm, using the minimum distance condition for the elements of each dictionary level was implemented. Experimental tests were performed using an MMP-Intra encoder with initial block size of 16×16 pixels. Four prediction modes are used for 16×16 blocks (level 8 of the segmentation tree): the new MFV prediction mode plus the three H.264/AVC $Intra\_16 \times 16$ *Vertical, Horizontal* and *Plane* prediction modes. For all other block sizes the nine modes defined by H.264/AVC for $Intra\_4 \times 4$ prediction, with the DC mode replaced by MFV, were used.

R-D results are presented for four different grayscale images: the smooth image LENA (from [7]), text image pp1205, compound (text and grayscale) image pp1209 and texture image D108. 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 [8]. Test image D108 is a texture image from the Brodatz Database, downloaded from [9].

Figures 1 to 4 present the experimental results of MMP-Intra and MMP when compared with JPEG2000 encoder [10] and with H.264/AVC *high* profile (JM9.2) [6], used as a still image encoder. H.264/AVC was used in this test because of its well known excellent performance in still image coding, using its intra-frame coding tools. Additionally, H.264/AVC uses essentially the same prediction of MMP-Intra, but in the context of the transform-quantisation-encoding paradigm, making it a good comparison for MMP-Intra.

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.

We can see from the plots that the new dictionary design algorithm increases the performance of MMP-Intra for all test images and for all compression ratios. For smooth image LENA, the gains are about 0.2 to 0.3 dB, reducing the gap to JPEG2000 and H.264/AVC to about 0.2 to 0.5 dB. For other test images, the gains over original MMP-Intra are even larger, ranging from about 0.4 to 0.5dB for compound and texture images PP1209 and D108, to about 1 dB for text image pp1205. It is important to note that, for non smooth images, MMP-Intra already outperforms transform-based encoder by a considerable amount.

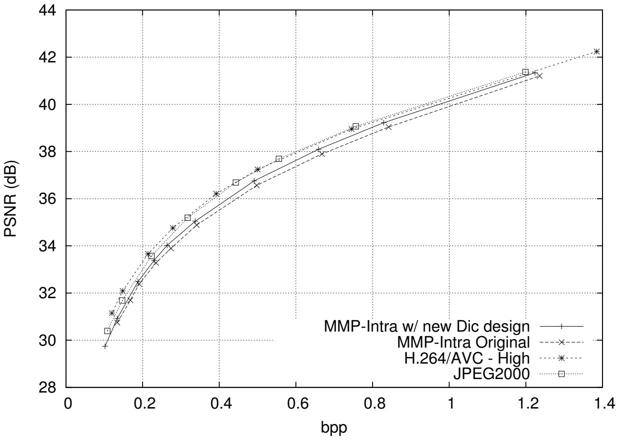

Fig. 1. Experimental results for image LENA.

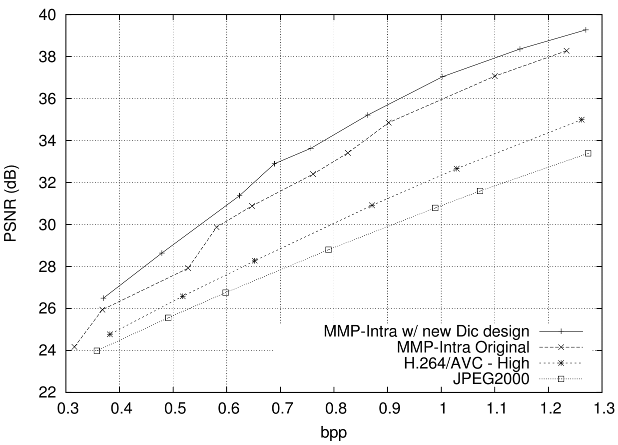

Fig. 2. Experimental results for image pp1205.

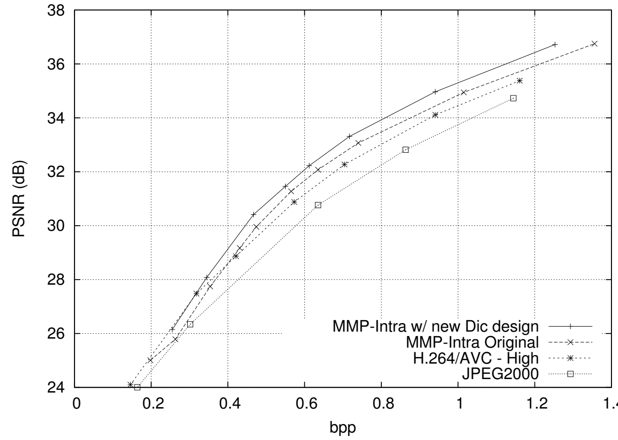

Fig. 3. Experimental results for image pp1209.

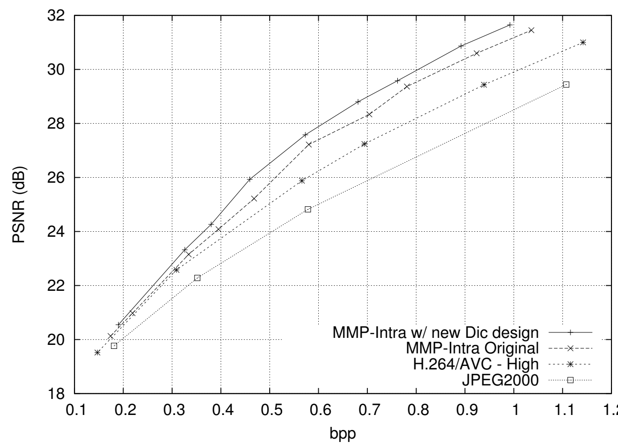

Fig. 4. Experimental results for image D108.

Besides these gains in quality, the new dictionary design procedure also allows for a reduction in computation complexity. Because this method excludes many of the vectors that would be inserted in the dictionary, specially for the larger dimensions, the search for the best block becomes much less complex. Tests were performed using equivalent versions of the new and original methods and gains of up to 50% in execution time (CPU time) were achieved.

V. Conclusion

In this paper we presented a efficient dictionary update process and improved context conditioning for the MMP-Intra image encoder. The presented methods allow for consistent gains in the compressed image quality with relevant reductions in encoding complexity.

The quality measurements confirm MMP-Intra as a universal image coding method, outperforming JPEG2000 and H.264/AVC high profile for a wide variety of image types while having competitive performance for smooth grayscale images.

These results confirm the effectiveness of the proposed procedures, introducing new research insights in ways of obtaining faster multiscale recurrent pattern image coding algorithms, that are capable of improving its performance for a wide range of image types.

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] E. B. L. Filho, E. A. B. da Silva, M. B. de Carvalho, W. S. S. Jnior, and J. Koiller, "Electrocardiographic signal compression using multiscale recurrent patterns," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 12, December 2005.

[3] M. H. V. Duarte, M. B. de Carvalho, E. A. B. da Silva, C. L. Pagliari, and G. V. Mendona, "Multiscale recurrent patterns applied to stereo image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 15, November 2005.

[4] N. M. M. Rodrigues, E. da Silva, M. B. de Carvalho, S. M. M. de Faria, and V. M. M. Silva, "H.264/AVC based video coding using multiscale recurrent patterns: First results," *International Workshop VLBV05*, September 2005.

[5] ——, "Universal image coding using multiscale recurrent patterns and prediction," *IEEE International Conference on Image Processing*, September 2005.

[6] J. V. T. J. of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and 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)*, March 2005.

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

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

[9] http://www.ux.his.no/∼tranden.

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