Fast Implementation for Multiscale Recurrent Pattern Image Coding

Danillo Graziosi[1,3], Nuno Rodrigues[1,2], Eduardo A. B. da Silva[3], Vitor Silva[1,4], Sérgio de Faria[1,2],

[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]DEEC, Univ. de Coimbra, Portugal.

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

***Abstract* – Multidimensional Multiscale Parser (MMP) algorithms have consistently achieved higher compression gains over the years, and recently outperformed state-of-the-art transform-based image encoders, like the JPEG2000 and H.264/AVC, for both smooth and non-smooth images, from low to medium compression rates.**

The use of dictionary update tools and a highly adaptable segmentation feature allowed the encoder to achieve a better rate-distortion performance, at the price of increased computational complexity.

In this paper we address the algorithm complexity problem, revising some developments towards the reduction of the computational burden and propose a new fast implementation, with just a small decrease in coding performance. Experimental results show that the modified algorithm still outperforms JPEG2000 and H.264 for smooth and compound images, with a very significant gain in coding time.

I. Introduction

MMP was initially introduced in [1], and can be regarded as an extension of traditional VQ methods. The major advantage of MMP is its powerful dictionary adaptation properties. Although it gave better results for non-smooth images than most state-of-the-art encoders, a gap still existed for smooth images. The performance of MMP was significantly improved when the concept of prediction was incorporated into the algorithm. Prediction was first used in [2], where blocks $m \times n$ were predicted from the neighboring pixels, following the prediction modes defined in H.264 standard [3]. The residue was encoded with the usual MMP algorithm, and with this method, MMP finally closed the performance gap for smooth images.

By using efficient dictionary update techniques, sophisticated context modeling (for details, see [4]) and relaxing the fixed block division (see also [5]), MMP now outperforms transform-based image encoders, and reaches the excellent rate-distortion performance for all types of images.

One of the main advantage of MMP encoders is its "universal" character. MMP can be efficiently used for a wide range of data signals, from voice and ECG [6] to stereoscopic images [7], digital images [1], [4] and video [8].

The previously mentioned improvements in rate-distortion performance of MMP arise at the cost of an increase in its computational complexity. In this paper, this critical aspect of the algorithm will be addressed, and a suggestion for reducing the coding time with little loss in coding gain will be proposed. In the following section, the fundamental MMP algorithm and its evolutionary steps will be explained in more details. The computational complexity will be addressed in section III, where the measures taken so far in reducing the computational cost will also be revisited. In section IV, a fast implementation for MMP-FP will be presented. The results and some future perspective towards reducing computational complexity will be exposed at the end of this article.

This work has been supported by Fundação Para a Ciência e Tecnologia (FCT), under project PTDC/EEA-TEL/66462/2006/

II. The MMP Algorithm

MMP divides the image into non-overlapping blocks, and uses patterns with different scales to approximate the image blocks. If no pattern from the dictionary satisfactory matches the new block, it is iteratively divided, first in the vertical, then in the horizontal direction, until an appropriate match is found for the remaining block. The new patterns determined for each block encoding are then added to the dictionary in multiple scales, therefore adapting it to the images' characteristics.

In order to find the best block segmentation and index encoding, an RD algorithm based on Lagrangian optimization is used. Before encoding the block, we use an optimization function that will find the most appropriate segmentation tree based on the coding cost of each node. We can associate to each node a distortion

$$D(X^l, S_i^l) = \sum_{x,y} (X^l(x,y) - S_i^l(x,y))^2 \qquad (1)$$

given by the sum of square differences between the original block X^l and the dictionary element S_i^l, and a rate

$$R(S_i^l) = \log_2(\Pr(i)) \qquad (2)$$

dependent on the probability of the symbol.

The cost of node n^l will be

$$J(n^l) = D(X^l, S_i^l) + \lambda R(S_i^l) \qquad (3)$$

where λ is the Lagrangian cost parameter.

The segmentation tree is optimized by deciding whether to segment a block, based on the sum of the nodes' costs, that is, the block will only be divided if

$$J(n^l) > J(n_1^{l-1}) + J(n_2^{l-1}) + \lambda R_{seg} \qquad (4)$$

where R_{seg} is the rate necessary for transmitting the segmentation flag.

In this sense, the optimization function will actually test the full segmentation of the block and prune the nodes until the minimum cost in an RD sense is found. By doing this iteratively, the optimization function will be called $\log_2(M) + 1$ times, searching for a best dictionary index every time, and adding up to the algorithm's complexity. Detailed information about the MMP algorithm and considerations on the RD optimization can be found in [1].

A. MMP with Predictive Coding: MMP-I

Prediction was first used in an MMP framework in [2]. When prediction is successfully applied, the residual blocks tend to be more homogenous with lower (and similar) energy, favoring the adaptation of the dictionary and improving the encoding efficiency. Based on the intra-prediction used in the H.264/AVC standard [3], the first algorithm used only prediction for 16×16 blocks, but revealed some inefficiency when used with non-smooth images.

MMP-Intra has taken the prediction process one step further, and used adaptive block size prediction, with blocks of dimensions 16×16 down to 4×4, and used the most frequent value (MFV) among neighboring pixels for the prediction instead of the DC mode [4].

The best prediction mode is chosen based on the Lagrangian R-D cost function, that determines the best trade-off between the prediction accuracy and the additional overhead introduced by the prediction data. For each prediction mode, the residue is encoded with the usual MMP encoder, and the cost of encoding that specific residue is evaluated for all available modes, choosing the mode with the smallest cost.

Other improvements introduced in [4] were:

- reducing the entropy of the dictionary indexes' symbols;
- limiting the insertion of new dictionary elements in every scale;
- improving the dictionary adaptation process;
- increasing the efficiency of the arithmetic encoding using adaptive contexts.

B. Flexible Partitioning: MMP-FP

In order to take further advantage of the adaptability of the MMP encoder, a new flexible partitioning scheme has been recently proposed [5]. The rigid dyadic block partitioning scheme used previously was relaxed, and now the block can be segmented into vertical or horizontal direction, according to the best RD compromise. An additional flag is sent to indicate the direction of the segmentation. By doing this, the optimization algorithm now has to consider 25 possible segmentation patterns for a 16×16 block (eg. 16×1 or 2×8).

MMP-FP can be summarized by the following steps.

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

1. find the prediction mode m_{best} that gives the minimum Lagrangian cost for coding the residue $Res^l = X^l - Pred^l(m)$, given by: $J(X^l, m) = J(Res^l) + \lambda R(m)$, where $J()$ is Lagrangian cost obtained by applying the MMP-FP algorithm to the residue of the block and $R()$ is the rate needed to encode the prediction mode m;
2. parse the original block in the horizontal direction, X^{l-1}_{left} and X^{l-1}_{right}, with half the pixels of the original block;
3. apply the algorithm recursively to X^{l-1}_{left} and X^{l-1}_{right}, until prediction for blocks of dimension 4×4 is reached;
4. calculate the cost for horizontal segmentation $J_{hor} = J(X^{l-1}_{left}, m_{best\_left}) + J(X^{l-1}_{right}, m_{best\_right}) + \lambda R_{seg\_hor}$
5. parse the original block in the vertical direction, X^{l-1}_{up} and X^{l-1}_{down}, with half the pixels of the original block;
6. apply the algorithm recursively to X^{l-1}_{up} and X^{l-1}_{down}, until prediction for blocks of dimension 4×4 is reached;
7. calculate the cost for vertical segmentation $J_{ver} = J(X^{l-1}_{up}, m_{best\_up}) + J(X^{l-1}_{down}, m_{best\_down}) + \lambda R_{seg\_ver}$
8. based on the values of the cost functions, determined in the previous steps, decide which is the best prediction mode, and whether to segment the original block in horizontal direction, in vertical direction or not segment at all;

Note that the optimization function is called to determine the cost of each prediction mode, when MMP encodes the residue of the block, with all three possible segmentation options. This adds up to a higher computational complexity, because the algorithm must test several segmentation possibilities along with different prediction schemes for each block size.

The flexible segmentation scheme improved considerably MMP's performance. MMP-FP was able to outperform state-of-the-art transform-based algorithms for medium to low compression rates, increasing even more MMPs performance for all kinds of images. Details and results for MMP-FP can be found in [5].

III. Computational Complexity for the MMP algorithm

Up to now, the focus of research on the MMP algorithm was improving the algorithm's rate-distortion performance, and little attention has been devoted to the algorithm's complexity. In what follows we will investigate methods to reduce MMPs complexity.

Since MMP-based encoders use an approximate pattern matching scheme, they have a complexity similar to that of standard Vector Quantization methods, that is traditionally higher than transform-based encoders. In [9] it is shown that in MMP there is a linear relationship between dictionary size and total rate. In additional, most of the improvements that lead to higher gains (new possible partitions, increased dictionary cardinality, etc.) [4] [5] incurred also in higher computational demands.

In [10], an algorithm for controlling the insertion of elements depending on a given threshold was presented, leading to a reduction in the size of the dictionary. Although this technique allowed a faster vector search due to a smaller dictionary, it also increased the computational cost, since each new element had to be compared with all the code-vectors present in the dictionary before insertion.

One way of avoiding extra calculations for block selection is restricting the number of scales for which new blocks will enter in the dictionary. In [4], scale restriction helps to reduce dictionary size, while not affecting (and in some cases improving) the overall coding performance. This is so because the restriction eliminates patterns that would rarely be used, decreasing the rate without affecting the PSNR.

The initial block size was also taken into consideration. An analysis made in [4] showed that for low compression ratios, a significant reduction in computational cost was achieved, without any degradation in rate-distortion performance, by changing the initial block size from 16×16 to 8×8.

The search for the best match in the dictionary is the most computationally intensive task, and is associated with the calculation of the sum of square differences (SSD), used in the search for best match for a given image block. By using a table of squared values for each possible difference, each multiplication can be substituted for one memory fetch operation, achieving gains in coding time. Other computationally intensive tasks are the calculation of logarithms, used by the encoder to estimate the rate of the encoded symbols. This operation can also be substituted for a lookup table. The use of lookup tale increases memory requirements; however these are tolerable in a standard personal computer (PC) based implementation.

IV. Fast Implementation for MMP-Intra

The use of prediction does not impose a severe additional computational cost, since it implies only few extra additions. Nevertheless, the choice of the best prediction mode is made comparing the cost of encoding the residue obtained after each prediction, which means that the encoding process has to be repeated M times, one for each prediction mode. This drastically increases the computational cost in the case of MMP-FP prediction. This is so because, besides it being done hierarchically, the number of possible segmentations is greatly increased.

In transform-based methods that use prediction, a common approach towards reducing the computational cost is by selecting only a few modes according to certain criteria, like the gradient method proposed in [11]; however in the MMP framework this would still not avoid the repeated dictionary search for determining the residue's cost of the remaining prediction modes.

One way to avoid the calculation of the cost of encoding the residue of each prediction is to use a different criterion for selecting the best prediction mode. For our fast implementation, we decided to use the residue's energy instead of the residue's coding cost, choosing the mode that results in the minimum energy. Residues with lower energy tend to be smoother, avoiding block segmentation and reducing bitrate, since less dictionary indexes are sent.

With this simple algorithm modification, we avoid several dictionary searches, resulting in a significant reduction in computational complexity, although at a cost in coding performance, as will be shown in the following section.

V. Experimental Results

The original algorithm and the fast implementation were both tested under the same conditions. Gray scale and compound images were used in the simulation, and are available for download at [12].

In Figures 1 and 2, a decrease in PSNR of about 0.2 dB can be seen in all available bit rates; however the rate-distortion performance is still above the one of H.264 for medium to low compression ratios. Table I shows the encoding time saved with the new decision criterion, reducing it in some cases more than seven times.

Table I
Percentage of time saved with the new FAST algorithm

Rate	0.4 bpp	0.7 bpp	1.1 bpp
gold	87%	88%	85%
lena	86%	85%	81%
pp1205	87%	81%	84%
pp1209	86%	86%	82%

For compound images, where smoothness criteria do not apply, the losses are larger, reaching 1 dB for the text only PP1205 image, and 0.4 dB for the compound image PP1209; however even then it outperforms the encoders used for comparison (see figures 3 and 4). Equivalently to the smooth images, the new fast implementation also speeded up the encoding time, making it seven times faster than the previous version.

VI. Conclusion and Future Work

The MMP algorithm has shown great potential for image encoding; however recent developments have focused only on enhancing algorithm's performance, and computational complexity has not been taken into consideration. With the proposed fast implementation, we showed that we can significantly reduce the computational complexity and still profit from the good performance of the MMP algorithm.

Future work will focus on reducing even further the complexity, by combining different tested techniques according to the target rate (such as smaller block initialization) and image areas (using image segmentation for encoding tools selection).

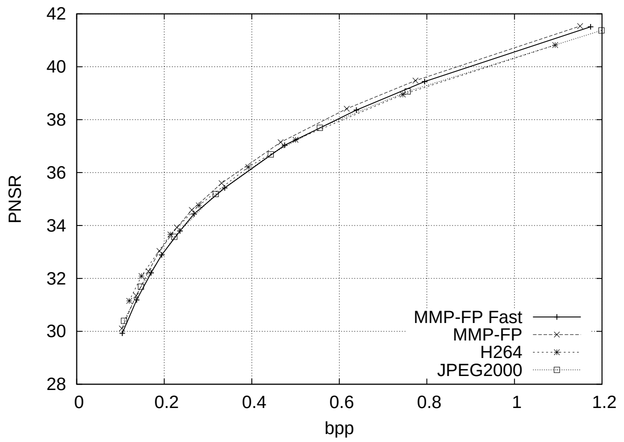

Figure 1 - Experimental results for image Lena.

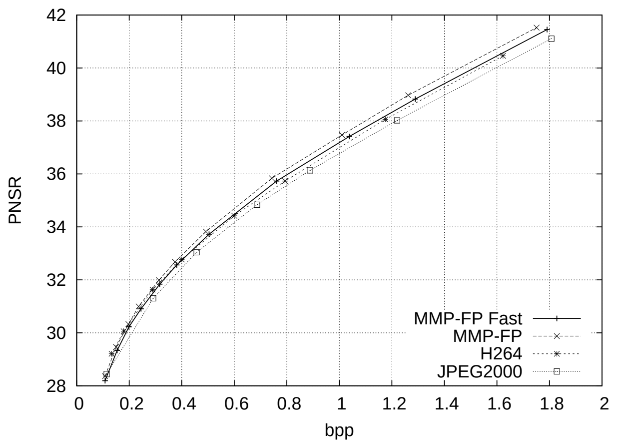

Figure 2 - Experimental results for image Gold.

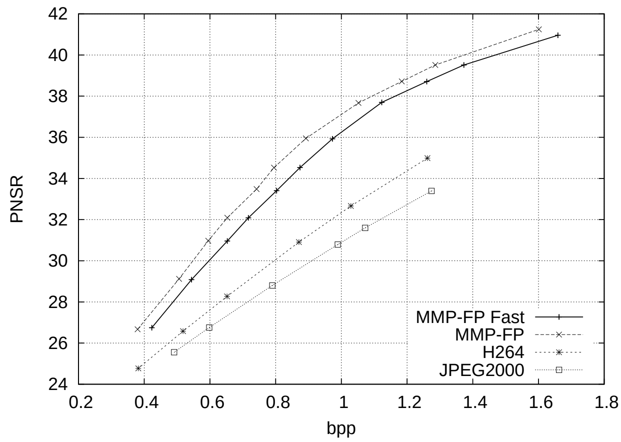

Figure 3 - Experimental results for compound image PP1205.

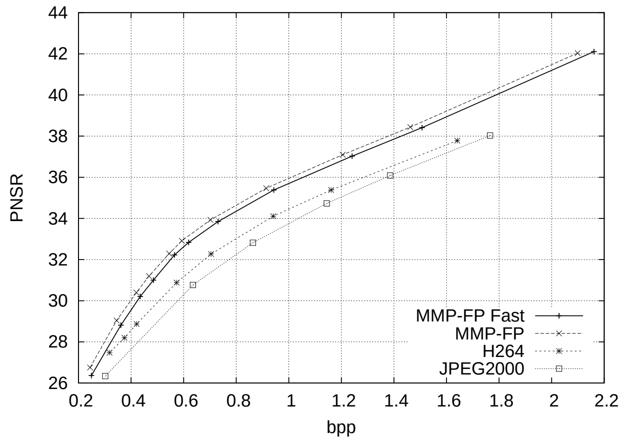

Figure 4 - Experimental results for compound image PP1209.

REFERENCES

[1] M. de Carvalho, E. da Silva, and W. Finamore, "Multidimensional signal compression using multiscale recurrent patterns", *Elsevier Signal Processing*, vol. 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", *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, "On dictionary adaptation for recurrent pattern image coding", *Image Processing, IEEE Transactions on*, vol. 17, no. 9, pp. 1640–1653, September 2008.

[5] Nelson C. Francisco, N. M. M. Rodrigues, E. A. B. da Silva, M. B. de Carvalho, S. M. M. de Faria, Vitor M. M. Silva, and Manuel J. C. S. Reis, "Multiscale recurrent pattern image coding with a flexible partition scheme", *IEEE International Conference on Image Processing*, October 2008.

[6] E. B. L. Filho, N. M.M. Rodrigues, E. A. B. da Silva, S. M. M. de Faria, V. M. M. da Silva, and M. B. de Carvalho, "ECG signal compression based on DC equalization and complexity sorting", *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 7, pp. 1923–1926, July 2008.

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

[8] N. M. M. Rodrigues, E. A. B. da Silva, M. B. de Carvalho, S. M. M. de Faria, and Vitor M. M. Silva, "On overriding H.264/AVC B-slice predicted residue coding", *PCS2007 - Picture Coding Symposium*, November 2007.

[9] E. B. L. Filho, E. A. B. da Silva, M. B. de Carvalho, W. S. S. Júnior, and J. Koiller, "Electrocardiographic signal compression using multiscale recurrent patterns", *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 12, pp. 2739–2753, December 2005.

[10] 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, "Efficient dictionary design for multiscale recurrent patterns image coding", *ISCAS 2006 IEEE International Symposium on Circuits and Systems*, May 2006.

[11] Jhing-Fa Wang Wei-Guang Lin, An-Chao Tsai and Jar-Ferr Yang, "A simple direction detection algorithm for fast h.264 intra prediction", in *Proc. 2007 IEEE Region 10 Conf. (TENCON'07)*, Taipei, Oct. 2007, pp. 1–4.

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