Computational Complexity Reduction Methods for Multiscale Recurrent Pattern Algorithms

Nelson C. Francisco[1,2,3], Nuno M. M. Rodrigues[1,2], Eduardo A. B. da Silva[3],
Murilo B. de Carvalho[4], Sérgio M. M. de Faria[1,2].

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

***Abstract*—The Multidimensional Multiscale Parser algorithm was originally proposed as a generic lossy data compression algorithm. An high degree of adaptivity and versatility allowed it to outperform state-of-the-art transform-based compression methods for a wide range of applications, from still images, compound documents, or even ECG's, just to name a few.**

However, as other pattern matching algorithms, it presents a high computational complexity. In this paper, we investigated several techniques that allowed to considerably reduce both the encoder's and the decoder's computational complexity, with marginal R-D performance losses. The most important reduction was achieved on the decoder, that reduced up to 95% the time required by the previous method. These improvements contribute to affirm MMP as an alternative to traditional transform-based encoders, approaching its computational complexity with that of transform-based algorithms.

***Index Terms*—Image Coding, Pattern Matching, Data Compression**

I. Introduction

Several decades of image and video coding have confirmed the success of transform-based algorithms. A good rate-distortion performance for natural images and video compression, dictated its adoption by many standards, from JPEG to H.264/AVC. The performance of these methods rely on the assumption that the input signal has a low-pass nature, and consequently can be subject to a coarse quantization or to be simply ignored, in order to achieve good compression ratios. Although, when this assumption is not true, their efficiency deteriorates noticeably.

The Multidimensional Multiscale Parser (MMP) algorithm [1] was presented as an alternative to the transform-based paradigm. As it does not rely on any assumption about the input signals' characteristics, it has the ability to adapt itself to their particular features. This high degree of adaptability allows MMP to outperform state-of-the-art compression methods for a wide range of applications, from lossless and lossy still images [2], video [3], compound documents [4], stereoscopic images [5] or even ECG's [6], just to name a few. However, as other pattern matching methods, MMP has a high computational complexity.

This project was funded by FCT - "Fundação para a Ciência e Tecnologia", Portugal, under the grant SFRH/BD/45460/2008, and Project COMUVI (PTDC/EEA-TEL/099387/2008).

In applications where the input data only needs to be encoded once and decoded many times, a high encoding computational complexity can be justified by a superior rate-distortion performance. Nevertheless, MMP's decoder also presents a considerable computational complexity, limiting its application for receivers with low resources. To overcome this issue and enlarge MMP's applications, several techniques that can be applied to any MMP based algorithm were studied, with significant computational complexity reduction regardless of the input signal's characteristics.

In Section II, we present a brief description of the MMP algorithm. In Section III, we discuss the critical time consuming processes in the algorithm, with the two proposed reduction techniques being described in Section III-A and III-B. Experimental results for the two methods are presented in Section IV, and Section V summarizes the conclusions of this work.

II. The MMP algorithm

MMP [1] is based on approximations of data segments of scale l, using codewords from an adaptive dictionary $\mathcal{D}^l$. For bi-dimensional input signals, these data segments are sequential blocks X^l.

For each input block, the dictionary codeword S_i^l that better represents X^l is chosen, based on an Rate-Distortion function J, given by $J = D + \lambda R$, where λ is a *Lagrangian multiplier* [7], that weights the rate R of the representation, over its resulting distortion D.

The input block is then segmented into two sub-blocks of a lower scale, X_1^{l-1} and X_2^{l-1}, each with half the pixels, and the two halves are recursively optimized. The sum of the resulting costs is then compared with that of the original block, in order to decide whether to segment or X^l, applying recursively the same procedure down to 1×1 sub-blocks.

Originally [1], segmentations were made in a pre-established direction for each scale, alternating the horizontal and vertical directions. In [2], a new segmentation scheme was proposed, where both the horizontal and vertical segmentations being tested for each scale, selecting the one with the lowest Lagrangean cost. This allow a better exploitation of the image's structure, resulting in a more adaptive algorithm, with considerable performance gains for all image types.

The segmentation pattern used for each block is represented by a binary tree, $\mathcal{T}$. Each non-segmented block, corresponds to a tree leaf, while each segmentation correspond to a child nodes n_i^l. Once $\mathcal{T}$ is obtained, it is converted into a string of symbols, using a top-down approach. Two distinct flags are used to indicate whether a node n_i^l corresponds to an horizontal or vertical segmentation, and a third flag is used to identify tree leaves. The non-segmentation flag is followed by the index of the code-vector that should be used to approximate the corresponding block. All the generated symbols are encoded using an adaptive arithmetic encoder, with separate contexts for each symbol and each scale.

Every segmentation of a block of scale l originates a new pattern, formed by the concatenation of two codewords of scale $l-1$. The use of a separable scale transformation T_l^s, allows to adjust the vector's dimensions and update the new pattern in all of the dictionary's scales, resulting in an *adaptive dictionary* that does not require any extra overhead to be transmitted. The decoder is able to keep its synchronized copy using only the segmentation flags and dictionary indices information.

In order to further improve the efficiency of the adaptive dictionary, several design techniques were proposed in [8]. Additional codewords became available for each extra pattern created, using *geometric transforms* and *displacements* of the original block. At the same time, redundancy control tools are used in order to avoid the insertion of useless blocks in the dictionary. State-of-the-art image encoders based on MMP also uses an hierarchical prediction scheme similar to the one used by H.264/AVC, that favors the statistical distribution of dictionary's codewords [8], with relevant performance improvements.

III. Computational complexity reduction methods

A. Dictionary partitioning by Euclidean norm

The most time consuming task on pattern matching algorithms, is the search for the best match among dictionary codewords. For each input block, the sum of squared errors (SSE) has to be computed for all codewords, in an exhaustive and time consuming process. For those algorithms that uses adaptive dictionaries, the updating stage also requires exhaustive searches for existing codevectors that are similar to the new pattern.

A careful organization of codewords can give an important contribute in accelerating the searching process. For example, if codewords are sorted by ascending Euclidean norms, it is possible to start the search to optimize a given block X^l in codewords with norm close to $||X^l||$. The search will then enlarge gradually, with the lowest distortion D found at each moment, being used to restrict the searching region. In this case, all the codewords with norms outside $[||X^l||-D; ||X^l||+D]$, are known to have a distortion larger than D, and consequently do not need to be tested in a strict distortion optimization.

However, sorting the dictionary every time a new codeword is inserted, is a very time consuming task, that easily counterbalances the gains achieved by the more efficient search. This issue could be overcome, for example, by using norm-based indexation of the dictionary. Codeword would remain disposed arbitrarily, and an additional field would indicate the codeword's norm, testing the ones with the closest norms first. However, this approach would impose a large amount of memory jumps, which also are known to require many time.

To overcome these two problems, we developed a method that combines the two presented techniques. The dynamic range of possible norms values, is divided in N slots, with the codewords being disposed sequentially inside their corresponding slot. This way, codewords inside each slot can be processed sequentially, minimizing the number of memory jumps, while the existence of distinct slots preserves the ability to discard codewords with distant norms. In this case, if an exact match exists for X^l, it will belong to the norm slot n, whose boundaries contain $||X^l||$, so n will be the starting point for the search. In a strict distortion optimization, the best match must belong to a norm slot contained in the interval $[||X^l||-D; ||X^l||+D]$, where D is the distortion of the best match found inside the slot n. The algorithm then proceeds to the slots $n-k$ and $n+k$, for increasing the value of k, and every time a best match is found, D is recomputed and the searching region narrows. The process will converge once all the remaining slots are outside the interval.

Nevertheless, when a R-D optimization is used, the search region will not depend on the distortion D, but on the Lagrangean cost J. In this case, the amplitude of the search region will be larger because of the λR term, with a direct dependence from λ. However, codewords located in the frontier of the searching region, would only be the optimal solution if it was possible to represent them with a null rate. In this case, the searching region radius can be reduced to $J = D+\lambda(\Delta R)$, with (ΔR) being the difference between the rate required to encode the best found match, and the minimum rate required to encode any codewords of the dictionary.

The dictionary also includes a field containing the average of each codeword, that allows to discard codewords inside the norm slot that belong to different quadrants.

Figure 1 schematizes the searching region for a two-dimensional block X^l. Each norm slot corresponds to a concentric region. S_i^l is the best match found in slot n. Slots $n-1$ and $n+1$ are tested next, with a new best match being found in $n-1$. With this approach, only the codewords represented as * are tested (norm slots $n-1$, n and $n+1$), with all the codewords represented as x (belonging to other norm slots) being discarded.

The number of norm slots is an important factor in the performance of the method. A high number of slots is more effective in reducing the search range, but imposes a large amount of memory jumps. The trade-off between the number of codewords tested in each slot, and the number of memory jumps, defines the optimal N. For the case of MMP, that uses a multiscale approach, the value of N was optimized for each dictionary level l. It revealed, by experimental test, to be properly traduced by the following expression:

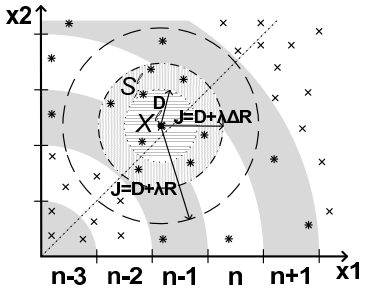

Fig. 1. Example of searching region for a two-dimensional input block X^l.

$$N(l) = \left\lceil \frac{\sqrt{Range^2 * Height(l) * Width(l)}}{4} \right\rceil, \quad (1)$$

where $Range$ represents the dynamic range of the input signal and $Width(l)$ and $Height(l)$ the dimensions of blocks of scale l.

We initially used dictionary slots with equal sizes, that added up to the maximum dictionary capacity (MDC). However, since codewords need to be discarded whenever the slot's capacity is reached, a particular issue occurred with this approach. Residue blocks have a norm distribution highly peaked near zero, and lower slots became full much earlier. This way, the algorithm was forced to discard codewords than would be available in a non-segmented dictionary. This restriction in the dictionary's growing process may have a negative impact in the R-D performance of the encoder. This was the case for our tests with MMP.

In order to minimize these losses, the capacity of each slot was adjusted according to the typical distribution of the residue norms. Experimental tests showed that when no restrictions are imposed on the dictionary's growth, the codewords' norm has a distribution than can be accurately modulated by a Rayleigh distribution, with two interesting particularities:

- The use of intra prediction makes the distribution shape highly independent of the input image's type;
- The shape depends on the target distortion, and consequently on the value of the Lagrangean operator λ. Low distortions mean better predictions, and consequently, residue blocks tend to have lower norms, concentrating the distribution around zero.

This way, it was possible to determine an expression to determine the suited capacity of each slot:

$$C(n) = a\left(\frac{2n}{b}e^{\frac{-n^2}{b}}\right) + c. \quad (2)$$

The value of b defines the concentration of the distribution around zero, and depends on λ, has was previously described. Once more we were able to define b has:

$$b = \frac{0.2log_{10}(\lambda+1)+2}{2}.N(l). \quad (3)$$

c guarantee a minimum capacity for each slot, and increases with λ, for the same reason. A logarithmic dependence proved to traduce accurately the relationship between c and λ:

$$c = \left\lceil MDC\frac{log(\lambda+1)+1}{8} \right\rceil. \quad (4)$$

a corresponds to the remaining elements over the MDC, resulting in the expression:

$$a = MDC - c.N(l). \quad (5)$$

With this approach, it was possible to practically eliminate the performance losses, at the expense slightly lower computational gains, when compared with the use of constant capacity slots.

It is important to note that this method reduced not only the complexity of the encoding process, but also of the decoder, as searches to determine block insertions will be much faster. In this case, only the norm slot corresponding to the generated pattern need to be tested in each case, instead of the entire dictionary.

B. Gradient analysis for tree expansion

In the particular case of MMP, one other time consuming task is the optimization of the block segmentation. In order to determine the optimal segmentation tree for each input block, MMP performs an hierarchical optimization for each scale, down to 1×1 blocks. However, if a given block has a very homogenous texture, the probability of finding its optimal representation with blocks from lower scales decreases.

To avoid testing segmentation patterns that are unlikely to have low Lagrangean cost, the gradient of each block is computed both in the vertical and horizontal direction, and if it is lower than a pre-established threshold τ, the block will not be further segmented in that direction. A dependence between τ and λ can be established: large λ means that the rate has an higher weight than the distortion, so τ can have larger values without compromising the optimality of the results. The expression:

$$\tau = (0.001\lambda + 1.5) * size(l), \quad (6)$$

was found to be appropriate to describe this dependence, where $size(l)$ represents de block's number of pixels in the tested direction.

IV. Experimental Results

Several experimental tests were performed in order to evaluate the results of each of the proposed methods. The capacity of norm slots and the optimal relationship between τ and λ were also obtained by performing some experimental tests, using different image types.

The computational complexity reduction of the proposed methods is summarized in Table I, for a set of 4 images with different characteristics: natural images Lena and Barbara, scanned text image PP1205, and scanned compound document PP1209. The encoder that only uses variable capacity norm slots is refereed as Enc. I, while Enc. II comprehends both of the proposed techniques. As the method proposed on Section III-B imposes no changes in the decoder, only the results for the final decoder are presented.

As it can be seen in Table I, an average gain of 69% of the encoding time and 87% of the decoding time were achieved by the proposed techniques. Figures 2 to 3 show the

TABLE I
% OF TIME SAVED OVER THE REFERENCE CODEC.

	Rate	0.25bpp	0.50bpp	0.75bpp	1.00bpp	Average
Enc. I	Lena	46%	53%	55%	57%	53%
	Barbara	51%	61%	69%	69%	63%
	PP1205	63%	72%	79%	84%	75%
	PP1209	50%	65%	66%	65%	62%
	Average	53%	63%	67%	69%	63%
Enc. II	Lena	56%	59%	63%	65%	61%
	Barbara	59%	65%	71%	72%	67%
	PP1205	73%	78%	82%	87%	80%
	PP1209	60%	68%	68%	70%	67%
	Average	62%	68%	71%	74%	69%
Decoder	Lena	73%	80%	84%	84%	80%
	Barbara	87%	81%	85%	87%	85%
	PP1205	94%	94%	92%	94%	94%
	PP1209	91%	87%	89%	90%	89%
	Average	86%	86%	88%	89%	87%

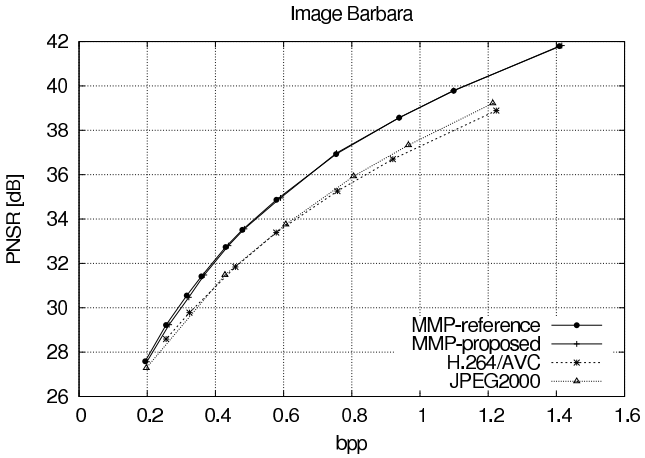

Fig. 2. Experimental results for image BARBARA 512×512.

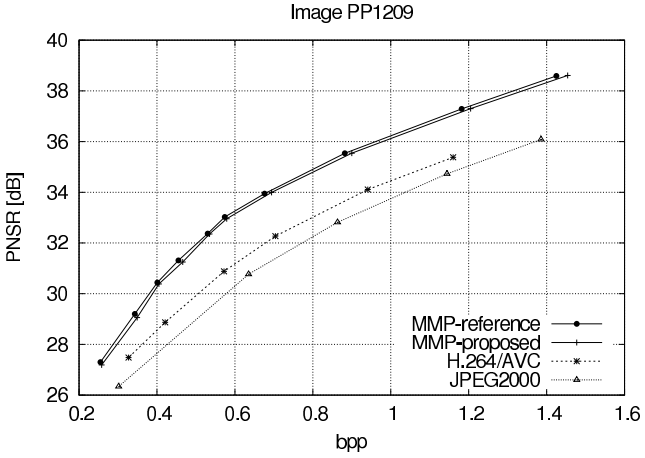

Fig. 3. Experimental results for image PP1209 512×512.

R-D performance of the reduced complexity encoder when compared with its original version, and with two state-of-the-art transform based encoders: H.264/AVC and JPEG2000. Despite the significant computational complexity reduction, the proposed method only presents residual R-D performance losses for smooth images. For text and compound images, the statistical distribution of residue norms tends to change, since the efficiency of the prediction stage decreases. For these images, the PSNR losses may go up to -0.2dB, in the worst case, but one also achieves the greater reduction in computational complexity. For both cases, the proposed encoder still considerably outperforms state-of-the-art transform based compression algorithms.

MMP's rate-distortion performance advantage, for a wide range of applications, makes the convergence between its encoding time and those from transform-based algorithms, an important factor in affirming the pattern matching paradigm as its viable successor. Despite being still considerably more complex, an important step was achieved in that direction, especially if we consider "encode once and decoded many times" application scenarios, where the encoder's high computational complexity can be easily justified by a state-of-the art rate-distortion performance.

V. Conclusions

In this paper, we present two computational complexity reduction techniques specially developed for the MMP algorithm, but that can be adapted to other pattern matching methods. These techniques considerably reduce the MMP's computational complexity, only with marginal R-D performance losses.

Several improvements can still be achieved, namely by exploiting the existence of repetitive tasks and the intensive use of integer operations. Nevertheless, these advances are implementation related. The methods described in this paper are algorithm design techniques. Another interesting area of multi-core processing, either through the use of GPUs or general purpose multi-core processors, that have enjoyed a recent increasing popularity.

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. C. Francisco, N. M. M. Rodrigues, E. A. B. da Silva, M. B. de Carvalho, S. M. M. de Faria, V. M. M. Silva, and M. J. C. S. Reis, "Multiscale recurrent pattern image coding with a flexible partion scheme," *IEEE International Conference on Image Processing*, pp. 141–144, S.Diego, California, October 2008.

[3] N. M. M. Rodrigues, E. A. B. da Silva, M. B. de Carvalho, S. M. M. de Faria, and V. M. M. Silva, "Improving H.264/AVC inter compression with multiscale recurrent patterns," *IEEE Int. Conf. on Image Processing*, Oct. 2006.

[4] N. Francisco, N. Rodrigues, E. da Silva, M. de Carvalho, S. de Faria, and V. Silva, "Scanned compound document encoding using multiscale recurrent patterns," *Image Processing, IEEE Transactions on*, vol. 19, no. 10, pp. 2712 –2724, 2010.

[5] 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 Trans. Circ. Systems for Video Technology*, vol. 11, no. 15, Nov. 2005.

[6] E. B. L. Filho, N. M. M. Rodrigues, E. A. B. da Silva, S. M. M. de Faria, V. M. M. Silva, and M. B. de Carvalho, "On ECG signal compression with one-dimensional multiscale recurrent patterns allied to pre-processing techniques," *Biomedical Engineering, IEEE Transactions on*, vol. 56, no. 3, pp. 896–900, mar. 2009.

[7] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, pp. 23–50, November 1998.

[8] N. M. M. Rodrigues, E. A. B. da Silva, M. B. de Carvalho, S. M. M. de Faria, and V. 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.