MULTISCALE RECURRENT PATTERNS AND GENERALISED SIDE-MATCH APPLIED TO IMAGE COMPRESSION

E. B. L. Filho[⋆†], *Ú. Abecassis*[‡], *W. S. S. Júnior*[⋄†], *E. A. B. da Silva*[†] *e M. B. de Carvalho*[Δ]

⋆Centro de Ciência, Tecnologia e Inovação do Pólo Industrial de Manaus,
Rua Salvador, 391, Adrianópolis, Manaus-AM, 69057-040, BRAZIL
† Universidade Federal do Rio de Janeiro,
Cx. P. 68504, Rio de Janeiro-RJ, 21941-972, BRAZIL
‡ Fundação Centro de Análise, Pesquisa e Inovação Tecnológica,
Av. Gov. Danilo de Matos Areosa, 381, Dist. Indust., Manaus - AM, 69075-351, BRAZIL
⋄Universidade Federal do Amazonas - CETELI,
Av. Gen. Rodrigo Octávio Jordão Ramos, 3000, Manaus - AM, BRAZIL
ΔTET/CTC, Universidade Federal Fluminense,
Rua Passo da pátria, 156, Niterói - RJ, 24210-240, BRAZIL
eddie@ctpim.org.br, ursula@fucapi.br, waldirjr@ufam.edu.br,
eduardo@lps.ufrj.br, murilo@telecom.uff.br

ABSTRACT

The Side-Match Multidimensional Multiscale Parser (SM-MMP) is a coding method based on the approximate multiscale pattern matching concept, where the dictionary is built considering smoothness constraints around block boundaries. This assumption can favor the coding of smooth signals, resulting in superior-quality reconstructed representations. In this work, a generalised framework for side-match is presented, in which the match attempt with neighboring blocks is performed in a hierarchical way and with a greater degree of restriction. An improved dictionary usage strategy is also presented, which employs elements from the causal neighborhood of the input block. The simulations performed on gray-scale images show that the proposed method is effective, presenting superior performance when compared to its predecessor.

Index Terms**—** Recurrent Pattern Matching, Multiscale Representation, Image Compression, Vector Quantization, Side-Match

1. INTRODUCTION

The Side-Match Multidimensional Multiscale Parser (SM-MMP) [1] is a recently developed coding algorithm built upon MMP (Multidimensional Multiscale Parser) [2], which is a lossy compression scheme based on approximate multiscale pattern matching. The SM-MMP encodes blocks of an input image in an dependent way, using state dictionaries $\mathscr{D}_s$ composed by N elements from $\mathscr{D}$ that comply with a continuity criterion regarding the causal neighborhood.

In this work, we propose a more generic approach for the problem of using multiscale pattern matching with continuity criterion. In it, unlike SM-MMP, some blocks are encoded taking into account a continuity criterion to three instead of two neighbors. This is achieved by dividing the input blocks into two groups: initial state blocks $\mathscr{X}^j$ and generalised side-match blocks $\mathscr{X}^{j(GSM)}$. The blocks of the first group are encoded with either MMP or SM-MMP; the other blocks are encoded by performing side-match with the initial states blocks, where each input block can have up to three neighbors. We also use displacements in the causal neighborhood of the current block for creating an additional dictionary, which increases the diversity for match attempts and captures periodic structures present throughout the image. The simulation results show that the proposed scheme improves the performance presented by SM-MMP, using a unified and generic approach for side-match.

The remaining of this paper is organised as follows. In section 2, the version of MMP with continuity criterion, known as SM-MMP, is presented. In section 3, we propose the generalised side-match framework, called GSM-MMP (Generalised Side-Match Multidimensional Multiscale Parser), as well as the additional neighborhood dictionary. Section 4 provides experimental results with gray-scale images and section 5 presents the conclusions of this work.

2. THE SM-MMP ALGORITHM

The SM-MMP is a lossy compression algorithm based on multiscale pattern matching, which is an extension of the ordinary pattern matching [2], where vectors of different dimensions can be matched. This is possible due to a scale transformation $T_N^M(x) : \mathscr{R}^N \mapsto \mathscr{R}^M$, which converts the dimensions of the vector before the matching attempt

In the two-dimensional SM-MMP, there is a dictionary

This work was accomplished thanks to the support provided by CT-PIM for basic research. The authors would like to thank its board of directors, represented by Mr. Wesley Alves and Mr. Admilton Pinheiro Salazar.

$\mathscr{D} = \{b_0, b_1, ..., b_{L-1}\}$ of fixed-size blocks, that it employs to encode variable-size sub-blocks of an input block $X^0_{k,l}$, where the dimensions $M \times N$ are powers of two and k and l define the location of the block on the image grid. When trying to encode $X^0_{k,l}$, MMP searches, in the state dictionary $\mathscr{D}_s$, the best block b_{i_0} for replacing $X^0_{k,l}$, without caring about the dimensions of each b_i. If b_{i_0} is chosen as an approximation for $X^0_{k,l}$, the algorithm outputs a *flag* '1', informing a successful match, and the index i_0; otherwise, the input block is split into two other blocks $X^1_{k,l}$ and $X^2_{k,l}$, with the same dimensions. When it happens, the algorithm outputs a *flag* '0', informing that a division of the current block was made, and repeats the procedure for $X^1_{k,l}$ and then $X^2_{k,l}$. This partitioning is recursively applied to all input blocks, as depicted in Fig. 1 for a block of dimensions 4×4.

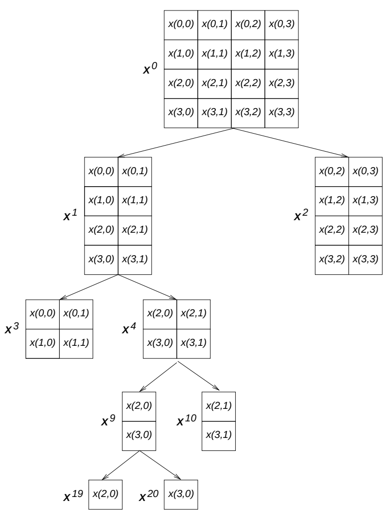


Fig. 1. Segmentation of a 4×4 block performed by SM-MMP.

The partitioning procedure defines a segmentation tree $\mathscr{S}$, where each node n_j is associated to a sub-block $X^j_{k,l}$. Regarding Fig. 1, the output generated by MMP would be $0, 0, 1, i_3, 0, 0, i_{19}, i_{20}, 1, i_{10}, 1, i_2$. The indices and flags are then encoded by an arithmetic encoder.

For a given block $\mathbf{X}^j_{k,l}$ of dimensions $M \times N$ associated to node n_j of $\mathscr{S}$, we define the *upper neighbor* $\mathbf{U}^j_{k,l}$ as the block of dimensions $N \times M$ right above $\mathbf{X}^j_{k,l}$, and the *left neighbor* $\mathbf{L}^j_{k,l}$ as the block of dimensions $N \times M$ to the left of $\mathbf{X}^j_{k,l}$, as depicted in Fig. 2. The state dictionary $\mathscr{D}_s$ used for encoding $X^j_{k,l}$ is composed by the N elements $\mathbf{b}_i$ from $\mathscr{D}$ that present the smallest *rugosities* $r_{ij} = R\left(\mathbf{U}^j, \mathbf{L}^j, \mathbf{b}_i\right)$, which is a metric defined in [1]. The size N depends on the level of *activity* $A(\mathbf{X})$ of blocks $\mathbf{U}^j$ e $\mathbf{L}^j$, also defined in [1]. N is then determined as $N_{max}\left(\left(A\left(\mathbf{U}^j\right) + A\left(\mathbf{L}^j\right)\right)/2\right) A^{-1}_{max}$, where A_{max} is the biggest value $A(\mathbf{X})$ among all input blocks and N_{max} is the maximum allowed size. The dictionary $\mathscr{D}$ is updated with expanded and contracted versions of previously encoded blocks, similar to what is done in Lempel-Ziv encoders [4].

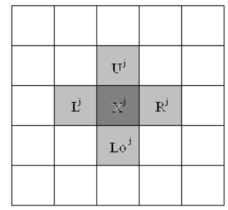


Fig. 2. Neighbors of a given block.

It is worth noticing that if $X^0_{k,l}$ presents dimensions $M \times N$, with $M = N$, the partitioning procedure will create blocks with dimensions: $\{M \times N, M \times N/2, M/2 \times N/2, M/2 \times N/4, M/4 \times N/4, ..., 1 \times 1\}$, that is, $1 + log2(M) + log2(N)$ different dimensions. To reduce the complexity of the algorithm and avoid performing the scale transformation T^M_N before each matching attempt, it is necessary to keep $1 + log2(M) + log2(N)$ copies of the dictionary, in such a way that T^M_N is needed just in the inclusion of each element.

The segmentation tree $\mathscr{S}$ can be optimised in a R-D sense. The optimization begins with $X^0_{k,l}$ and recursively splits blocks $X^{2j+1}_{k,l}$, computing the Lagrangian costs $J(n_{2j+1})$. The statistical records of elements $b_{i_{2k+1}}$ and flags, for all scales $(p, q) = \{(0, 0), (0, 1), (1, 1), (1, 2), ..., (log2(M), log2(N))\}$, are continuously updated. When the algorithm reaches the last scale (1×1), it returns analyzing the costs of subtrees with root nodes $\mathscr{S}(n_{2j+1})$ and $\mathscr{S}(n_{2j+2})$ and the cost of the parent node n_j, to decide about the removal of the subtrees, which happens if $J(n_j) \leq J(\mathscr{S}(n_{2j+1})) + J(\mathscr{S}(n_{2j+2}))$. If the algorithm decides to remove them, all updates due to $\mathscr{S}(n_{2j+1})$ and $\mathscr{S}(n_{2j+2})$ are excluded from $\mathscr{D}$ and the statistics of elements and flags are adjusted. The procedure continues through the splitting and analysis of other sub-blocks $X^{2j+2}_{k,l}$ of the current partition and then toward blocks to the left and above, until there are no more elements to analyse.

3. THE FRAMEWORK FOR GENERALISED SIDE-MATCH: GSM-MMP

In SM-MMP, the coding of each signal block depends on its causal neighbors. Given the traversing order of the tree, $X^0_{k,l}$ has just two neighbors: $\mathbf{U}^0_{k,l} = X^0_{k-1,l}$ and $\mathbf{L}^0_{k,l} = X^0_{k,l-1}$. However, it could be advantageous if one could modify this [5], such that more neighbors could be used. Then, we define the *lower neighbor* $\mathbf{Lo}^0_{k,l} = X^0_{k+1,l}$ as the block with dimensions $M \times N$ right below $\mathbf{X}^0_{k,l}$, and the *right neighbor* $\mathbf{R}^0_{k,l} = X^0_{k,l+1}$ as the block with dimensions $M \times N$ to the right of $\mathbf{X}^0_{k,l}$, as depicted in Fig. 2.

To enable the usage of the new neighbors, the new method will then classify the image blocks into two groups: initial state blocks $\mathscr{X}^0_{k,l}$ and generalised side-match blocks $\mathscr{X}^{0(GSM)}_{k,l}$. The first group is composed by the first blocks $\mathscr{X}^0_{k,l} = X^0_{2k,2l}$ to be encoded, which are organised in a way similar to a separable subsampling of 2:1 in the horizontal and vertical directions, as depicted in Fig. 3(b) (darker blocks).

These blocks can be encoded either with the original MMP or SM-MMP. However, to improve the performance at lower rates and create a side-match hierarchy, all initial state blocks are encoded with SM-MMP (see Fig. 3(a)). Note that the size of each $\mathscr{D}_s$ is quadruplicated, due to the increase in uncertainty, with the exception of $\mathscr{X}^0_{0,0}$. This way, the side-match is no longer performed with the neighbors right to the left and above, but with $\mathscr{U}^0_{k,l} = X^0_{2k-2,2l}$ and $\mathscr{L}^0_{k,l} = X^0_{2k,2l-2}$.

After encoding blocks $\mathscr{X}^0_{k,l}$, the initial grid is created, with key blocks periodically positioned throughout the image. The blocks in the next group, that is blocks $\mathscr{X}^{0(GSM)}_{k,l} = X^0_{2k/2k+1,2l+1/l}$, are encoded under the concept of generalised side-match. With the adopted organization, three configurations can happen (except for border blocks), shown in Fig. 3(b). The side-match can be performed with up to three neighbors in different ways. The rugosity and the dictionary size are computed in the same way as done in SM-MMP [1], but with one more element. For instance, for the type 3 side-match shown in Fig. 3, the rugosity and the dictionary size are

$$
\begin{aligned}
&R\left(\mathbf{U}^j,\mathbf{L}^j,\mathbf{b}_i\right) = \\
&\sum_{k=0}^{N-1}\left|\left\lfloor\frac{U^j(M-2,k) - U^j(M-1,k) + b_i(0,k) - b_i(1,k)}{2}\right\rfloor\right. \\
&\qquad \left. + b_i(0,k) - U^j(M-1,k)\right| \\
&+\sum_{p=0}^{M-1}\left|\left\lfloor\frac{L^j(p,N-2) - L^j(p,N-1) + b_i(p,0) - b_i(p,1)}{2}\right\rfloor\right. \\
&\qquad \left. + b_i(p,0) - U^j(p,N-1)\right| \\
&+\sum_{p=0}^{M-1}\left|\left\lfloor\frac{b_i(p,N-2) - b_i(p,N-1) + R^j(p,0) - R^j(p,1)}{2}\right\rfloor\right. \\
&\qquad \left. + R^j(p,0) - b_i(p,N-1)\right| \qquad (1)
\end{aligned}
$$

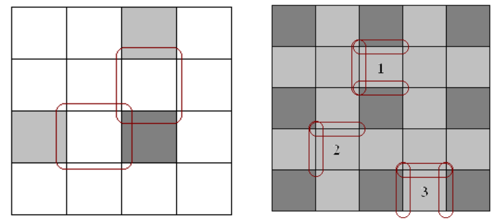
and

$$
N = N_{max}\left(\left(A\left(\mathbf{U}^j\right) + A\left(\mathbf{L}^j\right) + A\left(\mathbf{R}^j\right)\right)/3\right)A^{-1}_{max}, \qquad (2)
$$

respectively. Similar extensions can be performed for the type 1 side-match. The dictionary $\mathscr{D}$, in GSM-MMP, is built and updated with the same procedures presented in [1].

Then, the blocks used during the creation of $\mathscr{D}_s$ are chosen with a greater restriction and are better matched to their causal neighbors. SM-MMP and even the original MMP can be regarded as special cases of this new method.

For providing an enhanced dictionary and capturing periodic structures present in the signal, we also incorporate the new concept of *neighborhood dictionary* $\mathscr{D}_N$. This entity is composed as follows. For each $X^j_{k,l}$, a window with the



(a) (b)

Fig. 3. Side-match in GSM-MMP: (a) Side-match procedure for initial state blocks; (b) Side-match types found in generalised side-match blocks.

same dimensions is displaced through its causal neighborhood, where each displacement corresponds to a dictionary index b^N_i, looking for the best element to match $X^j_{k,l}$. If some blocks repeat on the image in a deterministic way, some elements b^N_i are more requested than others, which provides significant gains when using an arithmetic encoder. Note that the elements can be chosen alternately from $\mathscr{D}$ or $\mathscr{D}_N$.

4. SIMULATION RESULTS

The SM-MMP and GSM-MMP algorithms were used to compress gray-scale images, initially divided into blocks $X^0_{k,l}$ of dimensions 16×16. The initial dictionary had just elements in scale 1×1 and was initialised with $\mathscr{D}_0 = \{0, 4, 8, \ldots, 252, 255\}$. The vectors in all other scales were obtained through a separable two-dimensional scale transformation, implemented as explained in [1].

Figs. 4(a), 4(b), 4(c) e 4(d) show the R-D performance of the algorithms for images Cameraman and Einstein, of dimensions 256×256, and Lena and PP1209, of dimensions 512×512 pixels. The results for JPEG2000 [6] are also presented. The image Lena was retrieved from http://sipi.usc.edu /database, Cameraman from http ://iie.fing.edu.uy/ense/asign/ codif/material.htm, and Einstein from http://scien.stanford. edu/labsite/scien_test_images_videos.html. The PP1209 was scanned from IEEE *Transactions on Image Processing*, vol. 9, no. 7, pp. 1209, July 2000, and is composed by gray-scale images, text, formulas and graphics. The figures show that:

1. The GSM-MMP presented superior performance when compared to SM-MMP for all test images.

2. The GSM-MMP presented better performances than JPEG2000 for three images, with advantages of $\approx 2,2$ dB at $0,5$ bpp for PP1209, $\approx 1,8$ dB at $0,5$ bpp for Cameraman, and $\approx 0,35$ dB at $0,4$ bpp for Einstein.

3. The GSM-MMP slightly outperformed MMP for image Lena, although it did not achieve the same performance of JPEG2000 for this image.

The main responsible for the improvement was the GSM approach. The neighborhood dictionary, in most cases, provided from 30 to 40% of the performance gain. The image Cameraman compressed by GSM-MMP at 0.30 bpp is shown in Fig. 5. One of the possible reasons for the good

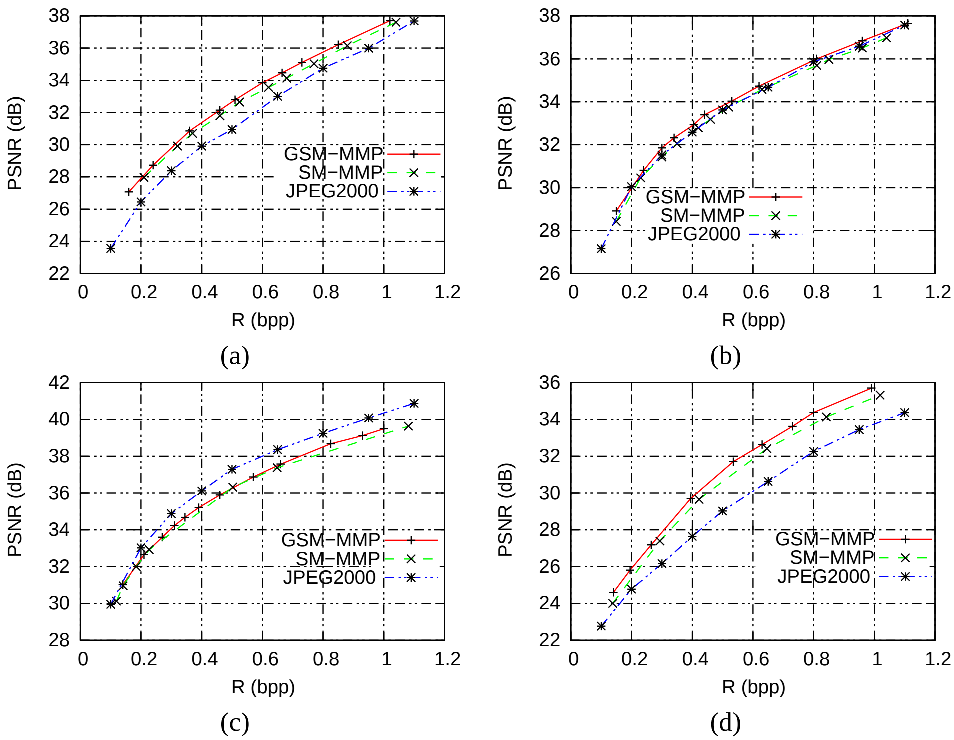


Fig. 4. R-D performance for: (a) Cameraman 256 × 256; (b) Einstein 256 × 256; (c) Lena 512 × 512; (d) PP1209 512 × 512.

performance presented by GSM-MMP for this image is that the GSM approach can predict blocks with high accuracy throughout the homogeneous regions composed by the grass, the sky and the cameraman's coat.

Fig. 5. Image Cameraman: GSM-MMP at 0.30 bpp.

5. CONCLUSIONS

In this paper, we proposed a new method for image compression which is a generalization of approximate multiscale pattern matching with side-match (SM-MMP). This approach presented superior performance when compared to SM-MMP, with significant improvements at high rates. The side-match became hierarchical and generalised, classifying the image blocks into groups that are processed in different ways. SM-MMP and even MMP can be regarded as special cases of this new method, which can be simplified or modified according to the target application.

6. REFERENCES

[1] E. B. L. Filho, M. B. Carvalho, and E. A. B. da Silva, "Multidimensional signal compression using multiscale recurrent patterns with smooth side-match criterion," *IEEE International Conference on Image Processing*, Singapore, October 2004.

[2] M. B. de Carvalho, E. A. B. da Silva and W. A. Finamore, "Multidimensional Signal Compression using Multiscale Recurrent Patterns," Elsevier's Signal Processing, Vol. 82, No. 11, pp. 1559-1580, November 2002.

[3] S. B. Yang, "General-Tree-Structured Vector Quantizer for Image Progressive Coding Using the Smooth Side-Match method", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 2, pp. 193–202, February 2003.

[4] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, Vol. it-24, No. 5, pp. 530-536, September 1978.

[5] H. C. Wei, P. C. Tsai and J. S. Wang, "Three-Sided Side Match Finite-State Vector Quantization", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, No. 1, pp. 51-58, February 2000.

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