

A COMPOUND IMAGE ENCODER BASED ON THE MULTISCALE RECURRENT PATTERN ALGORITHM

Nelson C. Francisco^{1,6}, Ricardo N. R. Sardo¹, Nuno M. M. Rodrigues^{1,2}, Eduardo A. B. da Silva³
Murilo B. de Carvalho⁴, Sérgio M. M. de Faria^{1,2}, Vitor M. M. da Silva^{1,5} and Manuel J. C. S. Reis⁶

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*

⁵*DEEC, Universidade de Coimbra, Portugal*

⁶*DE/CITAB, Univ. de Trás-os-Montes e Alto Douro, Portugal*

Keywords: Image Coding, Pattern Matching, Compound Images, Scanned Images.

Abstract: In this paper we present the current state of the project SCODE (Scanned Compound Document Encoder). The objective of this project is the development of a new image application, based on the Multidimensional Multiscale Parser algorithm (MMP), for compression of scanned documents, composed by pictures, graphs and text.

MMP is a generic compression algorithm that has been successfully applied in image coding. The use of a multiscale adaptive pattern matching coding paradigm allows it to achieve good results, consistently, for both smooth and text images. On the contrary, the traditional transform-based methods have a well known performance deficit for non-smooth image coding.

Current state-of-the-art compound image coding schemes rely on the use of segmentation techniques to split foreground and background planes of an input image. The performance of such methods, generally, degrades with the loss of efficiency of the segmentation process, namely for complex documents or low quality scans. These losses result from the use of transform-based compression for the background layer, like in DjVu or JPEG2000/Part6. The flexibility of MMP algorithm makes its efficiency independent of the segmentation process. Our experimental results show that MMP already outperforms some state-of-the-art algorithms, thus proving its usefulness as a compound image encoding algorithm.

In this paper we present the current results and the developed coding schemes, as well as an overview on the future work for this project.

1 INTRODUCTION

The increasing relevance of internet as a communication and publishing media for written documents, and the decreasing price of scanning and storage hardware, are contributing for the progressive substitution of the traditional paper by digital media support. Examples of this can easily be found in on-line digital libraries and publishing sites (namely scientific publications), that make available electronic copies of documents that were originally created in paper, or for which the original digital versions are no longer available. Other important application is the digital storage of document records, like those used in several do-

document archives. This has the advantage of avoiding the large storage and preservation requirements, associated with the original paper versions.

However, efficient compression of these digital documents is a challenging task. A simple approach to this problem would be the use of the traditional state-of-the-art image codecs to compress images resulting from the scanned documents. Nevertheless, it is a well known fact that traditional image coding algorithms are not capable of achieving a satisfactory performance for non low-pass images, like those resulting from document scanning. Because of this, several dedicated algorithms have been proposed in the literature, that were specifically optimised for cod-

ing compound images. These methods usually adopt a segmentation procedure, to separate text from the background and natural images, using different techniques to compress each of these components. Important examples of these methods are Digipaper (Huttenlocher et al., 1999) and DjVu (Bottou et al., 1998), that feature advanced technologies, such as image layer separation, progressive loading, arithmetic coding and lossy compression for bitonal images, allowing for high quality, readable images to be stored in an efficient way.

The main objective of the SCODE project is the development of an efficient compression algorithm for scanned document compression, based on a recently proposed paradigm for image coding, referred to as the Multidimensional Multiscale Parser (MMP) algorithm. This algorithm was originally proposed as a multidimensional lossy signal compression method, and its results demonstrated that it is a good alternative to the state-of-the-art transform-quantization based image encoders.

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, which may have different sizes. This makes MMP an extremely versatile encoding algorithm, that has showed good results when applied to a wide variety of signal sources, ranging from voice and ECG to stereoscopic images and video signals.

The SCODE software prototype will offer a set of tools for compress and manage scanned documents, as well as a set of features commonly found in this type of applications (the possibility of navigating documents, zooming and panning page images, producing and displaying side navigation thumbnails, saving and printing page and documents). This application implements a segmentation tool in order to split images and text planes, such the appropriate MMP algorithm can be used for each type of data. The segmentation of the digital document will be performed manually, automatically or using a combination of both, according to the application scenario.

In Section 2 we present the various versions of the MMP algorithm and the dictionary training procedure. Section 3 details the implementation of MMP for coding text images and Section 4 describes the SCODE prototype software. The Experimental results are shown in Section 5 and Section 6 devises some conclusions and further work.

2 MMP IMAGE CODING

In this section, we present the current state of MMP algorithm. A detailed presentation of this subject can be found in (de Carvalho et al., 2002).

2.1 The MMP Algorithm

The MMP algorithm was proposed has a generic lossy data compression method (de Carvalho et al., 2002). MMP coding is based on the use of an adaptive dictionary \mathcal{D}^l , to represent input data segments X^l . For each input image block, the algorithm first searches the dictionary for the element S_i^l that minimises the Lagrangian cost function $J(\mathcal{T}) = D(X^l, S_i^l) + \lambda R(S_i^l)$, where $D()$ is the sum of square differences (SSD) function and $R()$ is the rate needed to encode the approximation. The superscript l means that the block X^l belongs to *scale* l , that corresponds to a block size of $(2^{\lfloor \frac{l+1}{2} \rfloor} \times 2^{\lfloor \frac{l}{2} \rfloor})$. The algorithm then proceeds with the segmentation of the original block in two blocks, X_1^{l-1} and X_2^{l-1} , each with half the pixels of the original block, searching the dictionary of scale $(l-1)$ for the elements $S_{i_1}^{l-1}$ and $S_{i_2}^{l-1}$ that minimise the cost functions for each of the sub-blocks. The compression cost associated with each of the previous alternatives is then evaluated and the algorithm decides whether to segment or not the original block. Each non-segmented block of scale l is approximated by one word S_i^l of the dictionary \mathcal{D}^l . If a block is segmented, then the same procedure is recursively applied to each segment.

MMP uses a binary segmentation tree to represent the optimal partitioning of each block, that is encoded using two binary flags: flag '0' represents the tree nodes or block segmentations, and flag '1' represents the tree leaves (sub-blocks that are not segmented). The leaf flags are followed by the index that identifies the dictionary element selected to represent the corresponding sub-block. All these items are encoded using an adaptive arithmetic encoder, with a different context for each tree level, that corresponds to a block scale.

Figure 1 represents the segmentation of a block and its corresponding segmentation tree. In this example, i_0, \dots, i_4 are the indexes chosen to encode each sub-block. The corresponding string of symbols is as follows:

$$0 \ 1 \ i_0 \ 0 \ 1 \ i_1 \ 0 \ 0 \ i_2 \ i_3 \ 1 \ i_4.$$

The *adaptive* dictionary used by MMP is updated with new patterns originated by the blocks already processed. Each time a block of scale l is segmented into two $l-1$ sub-blocks, a new block is originated by the concatenation of two dictionary elements selected

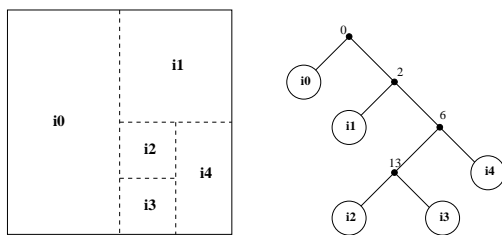


Figure 1: Segmentation of a 4×4 block (scale 4)(a) and the corresponding 5 scale binary tree (b).

for these sub-blocks. This new block is used to update the dictionaries in every scale, using a separable scale transformation T_i^s to adjust the vector's original scale l to each dictionary scale s . The decoder keeps a synchronised copy of the dictionary, using only the information of the segmentation flags and dictionary indexes.

2.2 MMP with Predictive Coding: MMP-I Algorithm

Although MMP shows results that considerably outperform those of state-of-the-art transform based algorithms for non-smooth images, this performance advantage is not verified for smooth image compression. MMP-I (Rodrigues et al., 2005) reduced this gap in performance for smooth images and brought MMP's results closer to those of transform-based algorithms, without compromising the results for non-smooth images. The MMP-I algorithm combines the MMP coding principles with intra-frame prediction techniques, like those used in the H.264/AVC standard (Joint Video Team (JVT), 2005).

For each original block X^l , MMP-intra first determines the prediction block P_m^l and the corresponding residue block Q_m^l , that is encoded by the MMP algorithm. The additional prediction overhead is evaluated by the Lagrangian R-D cost functions, allowing the encoder to determine the best trade-off between the prediction accuracy and the bit-rate needed to encode it. The prediction information is encoded together with the original MMP flags and indexes, using an adaptive arithmetic coder (Rodrigues et al., 2005). With this information, the decoder is able to reconstruct the image blocks by calculating the corresponding prediction block and adding it to the decoded residue block.

2.3 Efficient Dictionary Adaptation: MMP-II Algorithm

MMP-I uses the same dictionary updating procedure as the original MMP. However, experimental tests revealed some inefficiencies in this procedure. This observation motivated the investigation of several dictionary adaptation techniques that improved the performance of MMP-I, resulting in a new algorithm referred as MMP-II (Rodrigues et al.,)

MMP-II uses an improved context modelling for the dictionary elements, resulting in an increase of the arithmetic encoder performance. The dictionary elements are organised into partitions, and each dictionary element is identified using a partition index followed by its index inside that partition. The original block scale is used as a context, exploiting the fact that blocks generated at different levels have different matching probabilities.

An efficient redundancy control scheme for dictionary elements is also used. The insertion of a new block in the dictionary is only done if its distance, in relation to another block already available in the dictionary, is inferior to a given threshold d . This prevents the creation of a new index for blocks that bring very little distortion gains, that would also increase the overall rate.

In order to improve the dictionary approximation power, MMP-II uses extra blocks, originated by geometric transformations and translations of the original block, to update the dictionary. A norm-equalisation procedure is also used, in order to adapt the new codevector patterns to the residue signal's statistical distribution. A detailed description of the MMP-II coding can be found in (Rodrigues et al.,).

2.4 Flexible Partitioning: MMP-FP Algorithm

Experimental results have shown that the rigid dyadic block partitioning scheme used by MMP was somewhat ineffective and the compression performance of the algorithm was very dependent on the direction in which the segmentation is done in each scale. This observation clearly indicated that for some blocks the vertical segmentation performed better than the horizontal one, and (*vice versa*). This motivated the implementation of an alternative MMP segmentation scheme (Francisco et al., 2008), where each block can be segmented along either the horizontal or the vertical direction, based on a local R-D criterion.

Prior to being encoded, each image block X^l is segmented in both directions. This procedure is applied recursively for each child node, expanding the

segmentation tree of the block. The value of the Lagrangian cost function for each segmentation option is then evaluated from the bottom of the tree up, and the option with lower cost is chosen. If the decision to segment the block using one direction is taken, the child nodes generated in the other direction of the segmentation tree are pruned. If the lowest Lagrangian cost corresponds to a non-segmentation decision (*i.e.* the block corresponds to a tree leaf), all child nodes are pruned.

As a direct consequence of this new segmentation scheme, the block partition dimensions become very flexible and the method is able to adapt much more efficiently to the input signal's features. The new flexible segmentation scheme is used by MMP-II, both for the compression of the predicted residue and for the prediction step. This results in a much more accurate prediction process, creating a predicted residue with lower energy, that is more efficiently compressed by MMP. This partitioning method also uses block sizes that favour the prediction process, like very narrow blocks (*e.g.* 16×1), generating a more accurate prediction signal.

The flexible segmentation scheme improved considerably MMP's performance. For smooth images, MMP-FP is able to outperform state-of-the-art transform-based algorithms for bit-rates above 0.3bpp, increasing even more MMP's performance for non-smooth images.

2.5 A Dictionary Training Procedure

The initial dictionary used by MMP is quite simple, containing only a set of homogeneous blocks in each scale, distributed along the signals' dynamic range. The increase of its approximation power depends on the insertion of new code-vectors during image compression. Consequently, the initial blocks are coded in a less efficient way, due to the higher number of segmentations imposed before the dictionary reach a convenient variety of patterns. Therefore, a quick and appropriate growth of the dictionary is very important, in order to reduce the number of block segmentations and, consequently, enhance the compression performance of the algorithm. This motivated the development of a dictionary training procedure, such that an additional set of patterns are generated and inserted in the initial dictionary. A group of representative test images were encoded sequentially, at different bit-rates, and the dictionary blocks used for coding one image were inserted in the trained dictionary, that was used to compress the subsequent image.

Experimental results have shown that the training procedure for smooth images increases the MMP

PSNR values by up to 0.3dB for lower bit-rates. The use of an extra context for the initial blocks assures that the encoder's performance is not compromised by the entropy increase imposed by these blocks. This new method allowed MMP to outperform transform-based coding algorithm for bit-rates down to 0.15bpp.

3 MMP FOR TEXT IMAGE CODING

All previously described evolutions of MMP were developed to increase its performance for smooth images. However, the new techniques also allowed MMP to increase its performance for non-smooth images. Furthermore, experimental results have shown that the use of a predictive schemes is of little utility in text images. Low pixel correlation compromise the accuracy of the prediction stage, resulting in residue blocks with an energy level close to that of the original block. The cost for coding the prediction information will be increased more than that of non-predictive scheme compression.

This observations motivated a new implementation of the MMP algorithm, where the influence of each previously discussed technique was studied and evaluated, in order to obtain a new version of MMP, specifically optimised for text-images. The resulting encoder is not based in a predictive scheme, but uses the features of MMP-II, as well as the flexible partitioning scheme, described in section 2.4. The new method increased the MMP's performance for text images, with considerable computational complexity reduction.

Such method is adequate to be used in the SCODE application for compression of the non-smooth image layer, obtained from the segmentation process.

4 THE SCODE APPLICATION

The SCODE software application intends to be a stand-alone creator a viewer of MMP documents files. Because it is a Qt-based program (Qt is a cross-platform application framework provided by Trolltech), it can run across multiple operating systems, namely Windows, Linux/X11 and Mac OS X. This application has been developed simultaneously with the encoder algorithms, providing a GUI with the basic tools for image analysis and manipulation.

At this point, the application supports the display and processing, simultaneously, of one or more image files from various image formats. It also displays

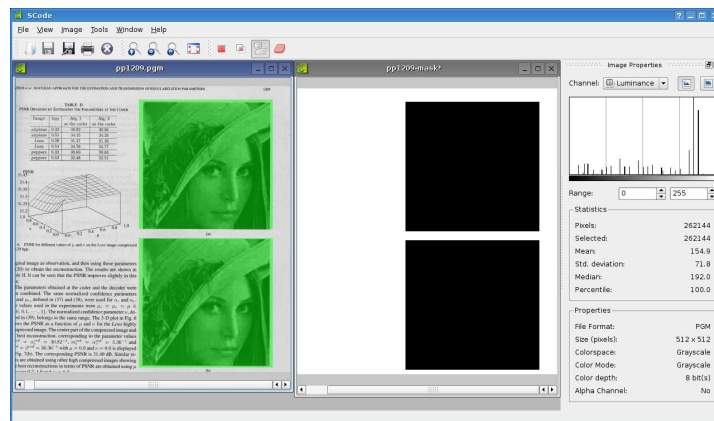


Figure 2: SCODE graphical interface.

some useful statistics about each images, or selected region, such as the histogram, the mean and standard deviation. It also offers a set of tools to navigate in documents, zoom and pan images, produce and display side navigation thumbnails, or save and print image documents.

However, the main feature is the segmentation process that splits the image into two layers, according to the digital document characteristics, namely smooth and non-smooth regions. Such segmentation is intended to be manual, automatic or assisted, where two techniques are combined. At this stage, only the manual segmentation is implemented. The application allows the user to draw a segmentation mask and save it as a new bi-level image. This process is the start point for selecting the compressing scheme (smooth/non-smooth) to compress each layer. Figure 2 shows the current graphical interface.

The original image is presented in the left window. The user is able to manually draw the segmentation mask, that in this case corresponds to the smooth image component and is highlighted in green. The window in center displays the binary mask generated, with white pixels corresponding to the text layer and black pixels to the smooth image layer. In the right, some informations about the image's statistics are displayed, including its histogram, median and standard deviation.

In the next stage of this work an automatic segmentation method will be implemented. Although assuming that an automatic segmentation process might introduce some problems, mainly due to a bad scanning, we expect that the universality of MMP will be able to overcome such problem, adapting the dictionary to the pattern of the chosen region. This characteristic cannot be showed by our counterparts encoders, that make pre-assumptions about the segmented layers and when it fails the results are catastrophic.

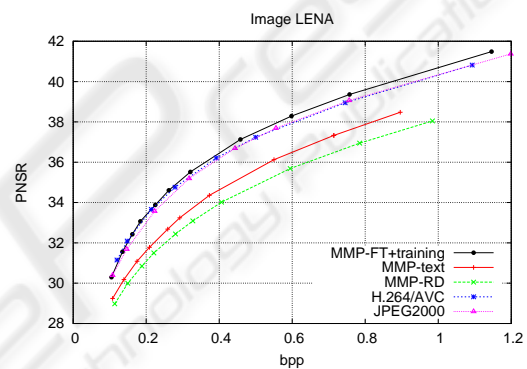


Figure 3: Experimental results for image LENA 512×512.

5 EXPERIMENTAL RESULTS

In this section, we present some experimental results obtained with the current version of MMP. Figures 3 to 5 show the experimental results for three images: smooth images (LENA and GOLDHILL) and a text image (SCAN004). These test images are available for download at (<http://www.estg.ipleiria.pt/~nuno/MMP/>).

Figures 3 and 4 show that the best results for smooth image coding are obtained with when using MMP-FP, associated with the dictionary training procedure. For non-smooth images, we notice an advantage of the MMP, version without predictive coding (MMP-text). Besides having a lower computational complexity, MMP-text consistently outperforms MMP with predictive coding.

These figures also show the improvements in performance over the original MMP, for all image types. As we can see in these figures, the current versions of MMP already outperform the transform-based algorithms (JPEG2000 and H.264/AVC), both for smooth and non-smooth images. Because the MMP-FP and

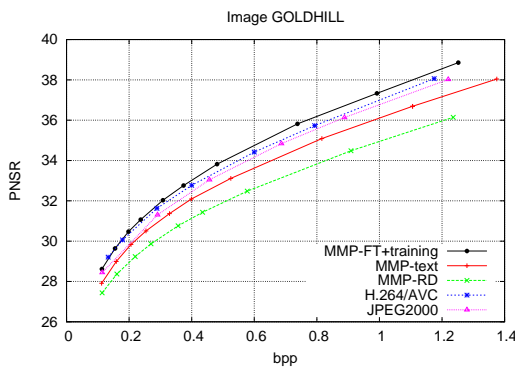


Figure 4: Experimental results for image GOLDHILL 512×512.

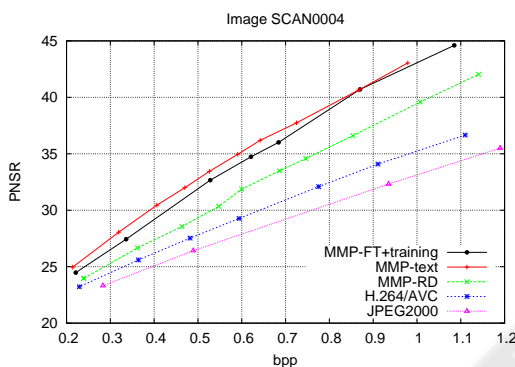


Figure 5: Experimental results for image SCAN0004 512×512.

MMP-text are able to achieve such good results for smooth and text images, we expect a light impact on performance, when region misclassification occurs during segmentation.

6 CONCLUSIONS AND FUTURE WORK

In this paper we present the objectives and the on going work on the SCODE project. The coding techniques already investigated allow our MMP-based encoders to outperform transform-based algorithms for smooth images, as well as for text images.

These results have highlighted the universality of MMP, showing its promising applicability for scanned documents. This is a good indication for the final encoding application, that will use a segmentation process to separate the smooth and text image regions, and encode them independently, using optimised versions of MMP. The method's versatility will be very useful in eliminating the performance losses observed for current state-of-the-art encoders, when

segmentation accuracy fails. Nevertheless, current MMP results demonstrate that the adaptive multiscale pattern matching paradigm is a promising option to the well established state-of-the-art transform-coding methods.

For future work we intend to perform some optimisations, in the MMP algorithm and in its implementation. The first is related to the MMP compression efficiency and the latter aims to reduce the computational complexity. Another objective is the development and implementation of the automatic and assisted segmentation for the different types of regions.

ACKNOWLEDGEMENTS

Project SCODE (PTDC/EEA-TEL/66462/2006) is financially supported by FCT - "Fundação para a Ciência e Tecnologia", Portugal.

REFERENCES

- Bottou, L., Haffner, P., Howard, P., Simard, P., Bengio, Y., and Cun., Y. (1998). High quality document image compression with djvu. *J. Electron. Imaging*, pages 410–425.
- de Carvalho, M., da Silva, E., and Finamore, W. (2002). Multidimensional signal compression using multiscale recurrent patterns. *Elsevier Signal Processing*, (82):1559–1580.
- Francisco, N. C., Rodrigues, N. M. M., da Silva, E. A. B., de Carvalho, M. B., de Faria, S. M. M., Silva, V. M. M., and Reis, M. J. C. S. (2008). Multiscale recurrent pattern image coding with a flexible partition scheme. *IEEE International Conference on Image Processing*.
- <http://www.estg.ipleiria.pt/~nuno/MMP/>.
- Huttenlocher, D., Felzenszwalb, P., and Rucklidge, W. (1999). Digipaper: A versatile color document image representation. *Proc. IEEE Int. Conf. on Image Processing*, pages 219–223.
- Joint Video Team (JVT), ISO/IEC MPEG & ITU-T VCEG, I. J. . I.-T. S. Q. (2005). *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)*.
- Rodrigues, N. M. M., da Silva, E. A. B., de Carvalho, M. B., de Faria, S. M. M., and Silva, V. M. M. On dictionary adaptation for recurrent pattern image coding. *Accepted for publication in IEEE Transactions on Image Processing*.
- Rodrigues, N. M. M., da Silva, E. A. B., de Carvalho, M. B., de Faria, S. M. M., and Silva, V. M. M. (2005). Universal image coding using multiscale recurrent patterns and prediction. *IEEE International Conference on Image Processing*.