ELSEVIER

Image and Vision Computing 17 (1999) 771–776

image AND vision COMPUTING

Linear prediction image coding using iterated function systems

M. Nappi[a,*], D. Vitulano[b]

[a]*Dipartimento di Informatica ed Applicazioni 'R.M. Capocelli', Università degli Studi di Salerno, 84081 Baronissi (SA), Italy*
[b]*Istituto per le Applicazioni del Calcolo 'M. Picone' (IAC/CNR), Rome, Italy*

Received 17 April 1998; received in revised form 14 July 1998; accepted 16 July 1998

Abstract

This paper presents a hybrid system to speed up image fractal encoding. The coding scheme, LP–IFS, consists of linear prediction (LP) and Iterated Functions Systems (IFS) applied in cascade on the image. The LP process employs a 2D auto-regressive model to estimate parameters for each block in the image partition; IFS are then used instead of adaptive quantizers to encode linear prediction errors. The stability of the resulting coding scheme is assured, since both LP and IFS are stable systems. The experiments performed have shown that LP–IFS can achieve very low bit-rates (BR) with good subjective and objective quality. Moreover, comparative studies based on extensive computer simulations have demonstrated that LP–IFS can rival standard IFS-based techniques in terms of BR and peak signal-to-noise ratio for high compression ratio and with respect to computing time. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Image coding; Linear prediction; Fractal coding

1. Introduction

The main goal of this work is that of providing a method that can both improve on the computational load imposed by IFS and effectively rival the performance results of well-known techniques such as JPEG [1], IFS [2–4] and IFS-based hybrid schemes [5,6].

To this aim, we explore the use of a cascaded combination of auto-regressive (AR) linear prediction (LP) [7,8] with iterated function systems (IFS) [2,9] to encode monochrome images. The AR model is computationally very simple and has a low number of coefficients. In addition, it has been shown to be efficient for digital image coding [10]. IFS have been developed in the last few years and are receiving major attention in research as a new technique for image coding. This interest stems from the fact that an IFS is an object simple in form and yet capable of accurately modeling complicated signals such as images [2,9] and residuals, i.e. predictive errors.

An IFS, based on iterated contractive image transformations [2,3,9], is applied to encode residuals instead of adaptive quantizers and/or decimators. In fact, IFS theory provides a convenient way to describe and to code LP residuals since it is designed to exploit the self-similarities that abound in such residuals.

An IFS usually requires a huge computing time to be carried out on an image. This is less true in the framework of LP–IFS: indeed, much less work is required on the residuals yielded by LP. In other words, the LP filter traps a big amount of visual information, and a greater tolerance is allowed for IFS encoding of the residual.

More recently, a closed related method for true color images was proposed [11]. However, it was only compared with a classical IFS coding scheme. Moreover, LP–IFS provides better performance for reducing the computational complexity, since it adopts a new lossy acceleration fractal encoding, proposed in Ref. [12].

The proposed method is attractive because of its fast encoding, that will be useful in a wide range of applications such as fractal image coding and fractal image retrieval and communication [13].

The outline of the paper is as follows. Section 2 presents a review of LP and IFS techniques, while Section 3 describes the proposed coding scheme. Experimental results and comparative studies are shown in Section 4, while Section 5 draws conclusions.

2. A review of LP and IFS techniques

In this section, we deal briefly with LP and IFS techniques. A general description is provided without too many details, but those strictly necessary to understand the remainder of the paper.

* Corresponding author.
E-mail address: micnap@udsab.dia.unisa.it (M. Nappi)

0262-8856/99/$ - see front matter © 1999 Elsevier Science B.V. All rights reserved.
PII: S0262-8856(98)00153-X

2.1. LP technique

For a given image S, any pixel $s_{m,n}$ can be predicted from a number of past samples using linear prediction [7,8]. This is the same as considering our image to be the output of some filter which has as input the past samples – which are known – and an additional input sequence. Formally, the auto-regressive moving average (ARMA) model is represented by the relation

$$s_{m,n} = -\sum_{i,j} a_{i,j} s_{m-i,n-j} + G \sum_{k,l} b_{k,l} u_{m-k,n-l} \tag{1}$$

where G is the gain factor, $a_{i,j}$ and $b_{k,l}$ are predictor coefficients and $u_{m,n}$ is the input sequence. If the latter is ignored, the second term on the right-hand side is null, and we have an auto-regressive (or all-pole or AR) model [8,14,15]. In this way, the amplitude of the predictable signal is

$$\hat{s}_{m,n} = \sum_{i,j \in \mathrm{ROS}} - a_{i,j} s_{m-i,n-j} \tag{2}$$

where ROS is the region of support. We stress that this model is not able to trap all information: there is always a difference between original and predictable signal. This quantity is

$$e_{m,n} = s_{m,n} - \hat{s}_{m,n} = s_{m,n} + \sum_{i,j \in \mathrm{ROS}} a_{i,j} s_{m-i,n-j} \tag{3}$$

and it intervenes in the estimate of the parameters $a_{i,j}$. Indeed, using the least-squares method, we can minimize the mean squared error (MSE):

$$\partial E / \partial a_{i,j} = 0 \Leftrightarrow \partial\left(\sum e_{m,n}^2\right) / \partial a_{i,j} = 0 \tag{4}$$

In this way, we obtain a set of p equations, where p is the number of parameters – well-known as *normal* or Yule–Walker equations [8].

The linear prediction operator A turns out to be linear and shift invariant. If we want filter stability, i.e. minimum phase, all poles must lie inside the unit circle [14].

Usually, LP techniques alone do not remove redundancy; in other words, they do not achieve any compression. The data can be reduced if we eliminate some of the predictive error coefficients or if we reduce the number of quantization levels [16]. Here, we make the choice of eliminating further redundancy, that is, we look for a transformation able to find self-similarity in the residual signal. With this aim, we employ IFS, described in the next section.

2.2. IFS technique

The theory of IFS [2,3,9] is based on the concept of a contractive function (CF). This can be defined as a function $f{:}X \to X$ on a complete metric space (X,h) such that there exists a constant $s < 1$ for which the following condition is satisfied:

$$h(f(x), f(y)) \leq s \cdot h(x,y) \qquad \forall x, y \in X \tag{5}$$

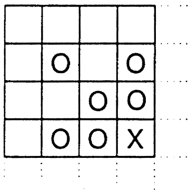

Fig. 1. The region of support (ROS) adopted for LP–IFS. The crossed pixel is predicted by a linear combination of the circled pixels.

where h is a metric function and the real number s is said to be a *contractivity factor* of f.

From Banach's fixed point theorem we know that each CF, when iterated, converges to a point $x_f \in X$, independent of the starting point of iteration. These theoretical results can be used for image coding, since we can solve the problem of finding the set of CFs that better approximates the points of a given image. Such a set is called an iterated function system (IFS). The IFS of an image can be constructed with Jacquin's algorithm [2], in which images are first partitioned into regions, and then encoded by finding CFs to transform image regions (which in this context are called *domains*) into similar image regions (here called *ranges*). These transformations are affine mappings, having the general form

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & \alpha_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ s_i \\ \beta_i \end{bmatrix} \tag{6}$$

where the coefficients represent characteristics such as the geometric contraction, the isometric transformation, grey scale factors, etc. IFS coding provides a compact but still accurate representation of an image.

3. LP–IFS coding scheme

Our goal is to find a transformation T having basically two properties. The first is that it traps information due to a strong correlation existing among contiguous pixels. The second property is that it should ideally succeed in eliminating the remaining redundancy.

In order to satisfy the first property, we chose Linear Prediction using the 2D AR model. The ROS adopted for our implementation is shown in Fig. 1. This mathematical model, developed to describe real signals, approximates them well but not perfectly: predictive residuals still contain much significant information. However, the residual – representing essentially the uncorrelated information present in the image – can be defined mathematically too.

With this aim, the possible courses of action are two:

- interpreting this error as white noise – i.e. a stochastic process – and applying classical decimation and quantization techniques [10,14,15];

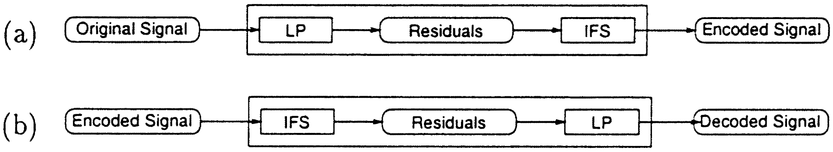

Fig. 2. LP–IFS coding (a) and decoding (b).

- finding an alternative model able to reduce further redundancy inside the residual.

Our basic idea is to follow the second course, adopting IFS to reduce the further redundancy present in the residual. In fact, the residual can be considered as a fractal object, in the sense that it is made of transformed copies of either itself or parts of itself. The main feature of the IFS-based approach is that it effectively detects and exploits piecewise self-transformability on a block-wise basis.

Formally, the transformation T can be defined by introducing two invertible operators

$A : U_1 \rightarrow U_2$ and $B : U_2 \rightarrow U_3$

where U_1 denotes the vector space of images, i.e. $gl(n,R)$, while U_2 is the vector space of residuals $gl(n,R)$ and U_3 the vector space $M_1(R)$ of polynomials of degree ≤ 1. Hence, our final operator will be invertible:

$$T \equiv B \circ A, \quad T^{-1} \equiv A^{-1} \circ B^{-1} \tag{7}$$

In the case of LP–IFS, A is the LP operator and B is the IFS operator.

The coding scheme LP–IFS consists of LP and IFS combined in cascade on the image. In other words, once LP is performed on the image, we are left with a residual on which IFS is applied. The stability of the hybrid operator is ensured by the separate stability of the two component operators.

The decoding scheme consists of IFS and LP combined in cascade on the data produced by LP–IFS. A schematic diagram of LP–IFS coding and decoding is shown in Fig. 2.

Notice that LP–IFS adoptes the algorithm proposed in Ref. [12].

4. Experimental results

The experiments performed on LP–IFS had two main goals:

- verifying the speed-up relative to traditional IFS and fractal based hybrid schemes;
- assessing the subjective and objective quality of the reconstructed images relative to compression techniques such as IFS and JPEG.

With this aim, several 512 × 512 8-bit test images have been selected, but for the sake of brevity this paper shall only show the results obtained on the luminance components of images ‘Lena’, ‘Peppers’ and ‘Baboon’ (the latter really represents a mandrill). The objective results obtained are presented in terms of bit-rate (BR), peak signal-to-noise ratio (PSNR) and execution time.

The bit-rate is the sum of two terms:

$$\mathrm{BR} = b_1 + b_2 \tag{8}$$

The first term b_1 represents information contained in our predictor, i.e. estimated filter coefficients and initial conditions. For a single 16 × 16 pixel block, we have six predictive coefficients; since reflection coefficients [11] are utilized, 48 bits are enough to guarantee stability. Furthermore, there are pixels belonging to the original image. These LP-related data can be compressed by Huffmann coding. The second term b_2 is due to IFS coding. We have utilized adaptive filters of order 6 so as not to complicate excessively our model, and also because the results demonstrate that a good quality can be achieved.

PSNR is defined as

$$\mathrm{PSNR} = 10\log_{10} \frac{M \cdot N \cdot 255^2}{\sum_{m,n} (s_{m,n} - \bar{s}_{m,n})^2} \tag{9}$$

where $s_{m,n}$ is the original image and $\bar{s}_{m,n}$ is the restored image, while M and N are the image’s side lengths.

The execution times were taken on an IBM RS/6000 and are the CPU seconds reported by `time`.

Linear prediction has been performed with (strongly causal) filters of the sixth order and fixed-size blocks of either 64 × 64, 32 × 32 or 16 × 16 pixels. On the contrary, our implementation of IFS adopts the following parameters:

1. an adaptive quadtree partitioning scheme [3], articulated on four levels, from 64 × 64 to 8 × 8;
2. the pool domain divided in 272 classes [12];
3. the eight canonical isometries of a square block [2];
4. α_i and β_i (see Eq. (6)) respectively quantized with 3 and 5 bits.

4.1. Comparison with classical IFS

The aim of this subsection is to demonstrate that LP–IFS can be an efficient coding if compared to classical [2,3] and improved [4] IFS coding. In particular, LP–IFS is faster than all IFS-based coding schemes and can achieve better quality for high compression ratios (i.e. bit-rates lower than 0.30 bpp).

Table 1 summarizes the results obtained on ‘Lena’ with

Table 1
Performance of LP–IFS on ‘Lena’ using adaptive filters with 64 × 64, 32 × 32 and 16 × 16 block sizes

Block size	BR	b_1	b_2	PSNR	PSNR (res)
64 × 64	1.00	0.030	0.97	33.89	35.33
64 × 64	0.80	0.030	0.77	32.73	34.21
64 × 64	0.65	0.030	0.62	31.77	33.18
64 × 64	0.46	0.030	0.42	30.80	32.17
64 × 64	0.18	0.030	0.15	29.07	30.49
64 × 64	0.15	0.030	0.12	28.55	30.14
64 × 64	0.11	0.030	0.08	27.71	29.12
32 × 32	1.00	0.067	0.93	34.30	37.10
32 × 32	0.75	0.067	0.68	33.50	36.07
32 × 32	0.60	0.067	0.53	32.90	34.81
32 × 32	0.48	0.067	0.41	32.05	33.95
32 × 32	0.19	0.067	0.12	30.10	31.88
32 × 32	0.15	0.067	0.08	29.93	31.63
32 × 32	0.11	0.067	0.04	28.11	31.17
16 × 16	1.00	0.200	0.80	36.09	38.56
16 × 16	0.81	0.200	0.61	35.61	37.63
16 × 16	0.65	0.200	0.45	34.90	36.44
16 × 16	0.34	0.200	0.14	32.76	34.52
16 × 16	0.21	0.200	0.01	30.84	33.23

different BR. The last column of the table, labeled ‘PSNR (res)’, reports the PSNR between the original and the modeled residual as reconstructed by IFS. The results achieved by using block sizes of 32 × 32 and 64 × 64 illustrate that larger block sizes slightly decrease the effectiveness of LP–IFS in terms of PSNR, yielding better BR.

Fig. 3 shows ‘Lena’ compressed at several ratios using 16 × 16 blocks.

The speed-up over traditional IFS yielded by LP–IFS can be appreciated by looking at Table 2. Since in the case of LP–IFS the IFS operator is applied after LP, it has to deal with a reduced dynamical range; as a result, the number of blocks ‘flat’ enough to be simply encoded by their average is much higher.

The columns labeled ‘N_T’ and ‘N_F’ in Table 2 contain the number of blocks encoded by an affine transformation and the number of ‘flat’ (average-only) blocks respectively. The columns labeled FLOPS report the number of significant floating point operations (additions and multiplications/divisions).

The reduced number of necessary affine transformations has an immediate impact on the execution time, as shown in Fig. 4. It can be seen that LP–IFS, even in its slowest version (16 × 16 pixel blocks) is faster than plain IFS by a factor of about 4.5.

It can be seen that IFS provides a slightly better PSNR for medium and low compression ratios at the cost of a heavier computational burden: the number of affine transformations required is about twice that for LP–IFS. For higher bit-rates, however, LP–IFS outperforms IFS with regard to PSNR (more than 1 dB), as shown also by Tables 3 and 4.

In order to assess the visual distortion introduced by LP–IFS at high compression ratios, Fig. 5 shows zoomed details

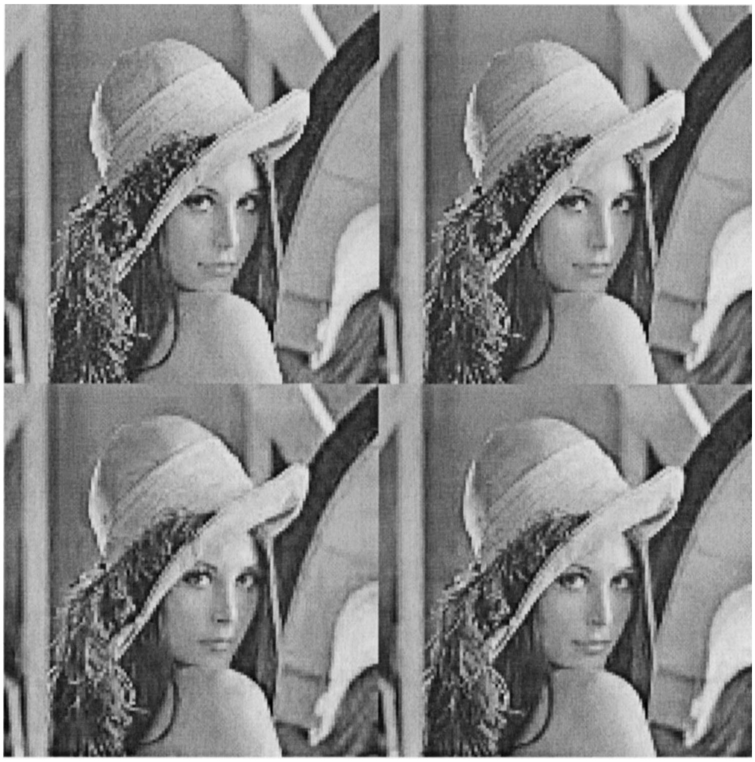

Fig. 3. 8-Bit image ‘Lena’ for several compression factors using 16 × 16 blocks. Clockwise, from upper left: original, 0.65 bpp, 0.34 bpp and 0.21 bpp.

Table 2
Performance of IFS and LP–IFS (16 × 16 pixel blocks) applied to image 'Lena' and to its residual

IFS					LP–IFS				
bpp	N_T	N_F	PSNR	FLOPS (× 10^6)	bpp	N_T	N_F	PSNR	FLOPS (× 10^6)
0.19	1737	389	29.22	24051	0.21	423	1724	30.84	6084
0.36	3094	223	32.55	45826	0.34	1420	1600	32.76	22319
0.62	5276	161	34.97	96432	0.65	2911	1573	34.90	51300
0.83	6123	87	35.67	121300	0.81	3659	1003	35.61	62077
1.02	7076	25	36.22	146277	1.00	4361	826	36.09	78865

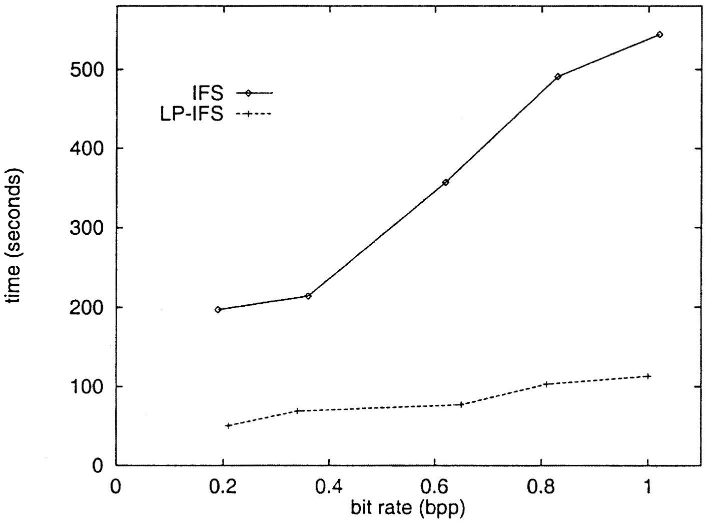

Fig. 4. Execution times for LP–IFS (16 × 16 pixel blocks) and traditional IFS on image 'Lena'.

from 'Lena'. Although the bit-rate is very low, the dreaded blocking effect is not significant: the most perceptible noise appears as a kind of diffused blurring.

Hybrid fractal methods like the ones proposed in Refs. [5,6] outperform both Fisher's method and LP–IFS with regard to PSNR. However, hybrid methods suffer from long encoding times and cannot rival sub-band encoding in terms of PSNR.

Table 3
PSNR comparison between plain IFS (Fisher) and LP–IFS (32 × 32 pixel blocks) on image 'Peppers'

bpp	PSNR–IFS	PSNR–LP–IFS
0.50	32.06	31.69
0.30	30.54	30.08
0.10	26.13	27.97

Table 4
PSNR comparison between plain IFS (Fisher) and LP–IFS (32 × 32 pixel blocks) on image 'Baboon'

bpp	PSNR–IFS	PSNR–LP–IFS
0.49	25.67	26.34

Table 5 illustrates that JPEG is able to achieve excellent results at low to moderate compression ratios, but it has not been designed for high compression in the first place. In particular, for bit-rates below 0.30, LP–IFS is definitely better.

5. Conclusion

In this paper we have described a new technique for still image compression based on a hybrid scheme that combines LP with IFS, applied in cascade on the image.

The performance of the coding schemes has been tested

Table 5
Performance of JPEG on 512 × 512, 8-bit 'Lena'

BR	PSNR
1.05	38.91
0.88	38.42
0.68	37.14
0.45	35.36
0.35	34.03
0.20	28.56
0.10	24.61

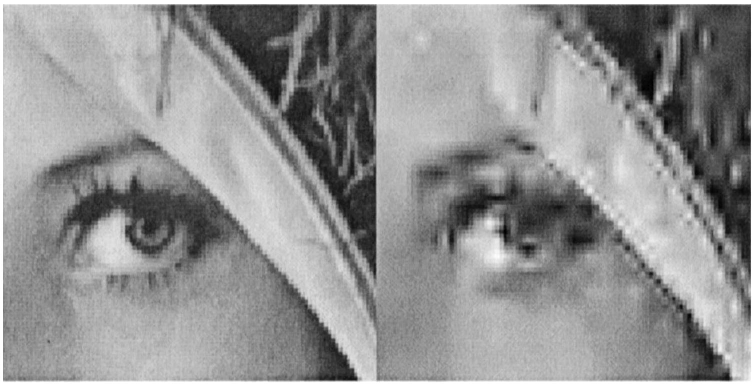

Fig. 5. Zoomed detail from 'Lena': original and compressed at 0.10 bpp.

on several test images. The main result is a significant speed-up with respect to the traditional implementation of IFS coding, with a negligible loss of quality. Indeed, for high compression ratios, LP–IFS outperforms traditional IFS as well as JPEG.

The issues that will be addressed in the improvement of the system include:

- the parallelization of the coding scheme;
- the combination between IFS with other LP schemes or sub-band based coding;
- the application on true color image and video coding.

References

[1] G.K. Wallace, The JPEG still picture compression standard, Commun. ACM 34 (1991).

[2] A.E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformation, IEEE Trans. Image Proc. 1 (1992) 18–30.

[3] Y. Fisher, Fractal Compression: Theory and Application to Digital Images, Springer, New York, 1994.

[4] D.C. Popescu, A. Dimca, H. Yan, A nonlinear model for fractal image coding, IEEE Trans. Image Proc. 6 (3) (1997).

[5] K.U. Barthel, T. Voye, Adaptive fractal image coding in the frequency domain, in: Proc. Int. Workshop on Image Processing: Theory, Methodology, Systems and Applications, Budapest, June 1994.

[6] K.U. Barthel, S. Brandau, W. Hermesmeier, G. Heising, Zerotree wavelet coding using fractal prediction, in: Proc. ICIP-97 IEEE Int. Conf. on Image Processing, Santa Barbara, CA, October 1997.

[7] A.K. Jain, Image data compression: A review, Proc. IEEE 69 (1981) 349–389.

[8] J. Makhoul, Linear prediction: A tutorial review, Proc. IEEE 63 (1975) 561–580.

[9] M. Barnsley, Fractals Everywhere, Academic Press, New York, 1988.

[10] M. Das, S.Y. Tan, N.K. Loh, Adaptive predictive coding of images based upon multiplicative time series modeling, in: Proc. IEEE Int. Conf. on Acoustic Speech Signal Processing, 1987, pp. 1386–1389.

[11] L. Moltedo, M. Nappi, D. Vitulano, S. Vitulano, Color image coding combining linear prediction and iterated function systems, Signal Processing 63 (1997) 157–162.

[12] R. Distasi, M. Nappi, S. Vitulano, Speeding up fractal encoding of images using a block indexing technique, Best Paper Award, Proc. ICIAP-97 IAPR Int. Conf. on Image Analysis and Processing, in: A. Del Bimbo (Ed.), Lecture Notes in Computer Science, Vol. 1311, Springer, 1997, pp. 101–107.

[13] M. Nappi, G. Polese, G. Tortora, FIRST: Fractal indexing and retrieval system for image databases, Image and Vision Computing 16 (14) (1998).

[14] S. Burgett, M. Das, Predictive image coding using two-dimensional multiplicative autoregressive models, Signal Processing 31 (1993) 121–132.

[15] P.A. Maragos, R.W. Schafer, R.M. Mercereau, Two dimensional linear prediction and its applications to adaptive predictive coding of images, IEEE Trans. Acoust. Speech Signal Proc. ASSP12 (6) (1984) 1213–1229.

[16] H. Kobayashi, L.R. Bahl, Image data compression by predictive coding I and II: Prediction algorithms, IBM J. R&D 18 (2) (1974) 164–179.