ELSEVIER

Signal Processing 63 (1997) 157–162

SIGNAL PROCESSING

# Color image coding combining linear prediction and iterated function systems

L. Moltedo[a], M. Nappi[b,*], D. Vitulano[a], S. Vitulano[c]

[a] *Istituto per le Applicazioni del Calcolo, M. Picone (IAC) C. N. R. Roma, Italy*
[b] *Dipartimento di Informatica ed Applicazioni, R.M. Capocelli, Università degli Studi di Salerno, 84081 Baronissi, Italy*
[c] *Policlinico di Cagliari, Università degli Studi di Cagliari, Italy*

Received 15 January 1996; received in revised form 1 May 1997

---

## Abstract

This paper describes three color image coding schemes combining linear prediction and iterated function systems. Linear prediction is based on the known autoregressive model, while iterated function systems (IFS) are implemented using quadtree partitioning. After describing the coding schemes, comparisons are carried out in terms of objective results. Simulation studies indicate that the proposed coding schemes produce images of quality ranging from fairly good to good, while performing significantly better than IFS alone with respect to computing time. © 1997 Elsevier Science B.V.

## Zusammenfassung

In diesem Artikel werden drei Farbbildkodierschemata beschrieben, die lineare Prädiktion und iterative Funktionensysteme kombinieren. Die lineare Prädiktion basiert auf dem bekannten autoregressiven Modell, während die iterativen Funktionensysteme (IFS) durch eine Baumstruktur mit vierfachen Abzweigungen implementiert werden. Zunächst werden die Kodierschemata beschrieben und nachfolgend Vergleiche mit Hilfe von objektiven Resultaten durchgeführt. Auswertungen von Simulation zeigen, daß die vorgeschlagenen Kodierschemata eine Bildqualität von recht gut bis gut liefern, während die Rechenkomplexität geringer ist als die von IFS allein. © 1997 Elsevier Science B.V.

## Résumé

Cet article décrit trois méthodes de codage d'images combinant prédiction linéaire et systèmes de fonctions itérées. La prédiction linéaire est basée sur le modèle autorégressif classique, alors que les systèmes de fonctions itérées (IFS) sont implantées à l'aide d'un partitionnement quadratique. Après la description des méthodes de codage, des comparaisons sont faites en termes de résultats objectifs. Les simulations indiquent que les méthodes de codage proposées produisent des images d'une qualité allant de raisonnablement bonne à bonne, tout en se comportant significativement mieux que les IFS seuls vis-à-vis du temps de calcul. © 1997 Elsevier Science B.V.

*Keywords:* Color image coding; Linear prediction; Iterated function systems

---

* Corresponding author. Tel.: + 39 89 965 356; fax: + 39 89 965 272; e-mail: micnap@dia.unisa.it.

*PII* S0165-1684(97)00164-3

## 1. Introduction

Coding techniques based on linear prediction (LP) have a good computing time, but the compression ratios that they achieve are suboptimal by any modern standard [2, 6, 7]. On the other hand, iterated function systems (IFS) have the opposite problem: good compression is achieved at the expense of long computing times [4, 5, 8].

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

In order to satisfy the first property, we employ linear prediction using the 2D autoregressive (AR) model [2]. This mathematical model approximates real signals well but not perfectly, since predictive errors or residuals contain still much significant information. The residual, representing essentially the uncorrelated information in the image, can be defined mathematically too. In fact, it can be considered a fractal object in the sense that, within a reasonable approximation, it is made of transformed copies of either itself or parts of itself. Because of these considerations, the residual can be effectively modeled by IFS instead of the usual decimators and/or quantizers.

This paper describes a hybrid approach combining AR-LP and IFS, called ARIFS, intended to exploit the best features of the two aforementioned techniques, with particular emphasis on the encoding of 24-bit, true-color images.

The rest of the paper is organized as follows. Section 2 introduces the ARIFS coding/decoding schemes, while the results of computer simulation are reported in Section 3. Some concluding remarks are finally made in Section 4.

## 2. The ARIFS coding scheme

For a given image $S$, any pixel $s_{m,n}$ can be predicted from a number of past samples using linear prediction [3, 6]. Formally, the autoregressive moving-average (ARMA) model is represented by the relation

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

where $G$ is 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 of Eq. (1) is null and we have an AR model [2, 6, 7]. Thus, the amplitude of the predictable signal becomes

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

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. The predictive error is $e_{m,n} = s_{m,n} - \hat{s}_{m,n} = s_{m,n} + \sum_{i,j \in \mathrm{ROS}} a_{i,j}\, s_{m-i,n-j}$ and intervenes in the estimate of the $a_{i,j}$ parameters. Indeed, using the least-squares method, we can minimize the mean-squared error (MSE) by imposing $\partial E / \partial a_{i,j} = 0 \Leftrightarrow \partial(\sum e^2_{m,n})/\partial a_{i,j} = 0$. This yields a set of $p$ equations – where $p$ is the number of parameters in our model – well known as *normal* or Yule–Walker equations [6].

The linear prediction operator just introduced is completely characterized by Eq. (2). It turns out to be linear and shift invariant; if we want filter stability, i.e., minimum phase, all poles lie inside the unit circle [6].

Techniques employing LP alone do not achieve any compression; the data can be reduced if we eliminate some of the predictive error coefficients or reduce the number of quantization levels [7]. Here, we make the choice of eliminating further redundancy, i.e., we look for another transformation able to find self-similarity in the residual signal.

The theory of IFS [1], based on the concept of contractive function (CF), can provide the transformation that we are looking for.

Banach's fixed-point theorem and Barnsley's Collage theorem [1] are the foundations for IFS-based image coding [4, 5, 8].

The IFS of an image can be constructed by an algorithm [4, 5] that first partitions the image into regions, which are then encoded by finding CFs (typically affine transformations) to transform a region (which in this context is called a domain) into a similar smaller region (called a range). The stability of IFS is implied by their continuous dependance on their coefficients [1].

The compound operator $T$ that we are looking for is obtained from the cascaded application of LP and IFS. More formally,

$$T = B \circ A. \tag{3}$$

In Eq. (3), $A: U_1 \to U_1$ is the LP operator, that maps an image $S \in U_1$ from the vector space of images, $U_1 = \mathrm{gl}(n, R)$, into a residual $R = S - \hat{S} \in U_1$. $B: U_1 \to U_2$ is the IFS operator, mapping a residual $R \in U_1$ into a point $P \in U_2$, where $U_2$ is the vector space $M_1(R)$ of all polynomials of degree $\leqslant 1$.

Invertibility and stability of the compound operator $T$ are ensured by the separate invertibility and stability of the two-component operators $A$ and $B$.

The coding scheme ARIFS consists of AR–LP and IFS combined in cascade on the image in an open loop: once LP is performed on the image, we are left with a residual on which IFS is applied.

### *2.1. Generalization to multi-channel images*

The technique just illustrated is a 1-channel coding scheme. In order to generalize such a model to $M$-channel data, we can rewrite Eq. (3) as

$$T^{(i)}: B^{(i)} \circ A^{(i)}, \quad i = 0,1,\dots,M-1. \tag{4}$$

In order not to complicate the formalism, we simply introduce a channel index $i$. For example, if we want to apply our model to a color image, three channels ($M = 3$) are enough.

If we analyze the morphological structure of the three component images, it is possible to ideate several alternative models to obtain very high compression ratios with a relatively good quality. The following were found to be particularly effective:

- ARIFS1: Applying the same ARIFS coding independently on the three channels, as shown in Fig. 1.
- ARIFS2: Sharing LP filters on the two chrominance channels, with luminance encoded with a different set of LP coefficients, as depicted in Fig. 2.

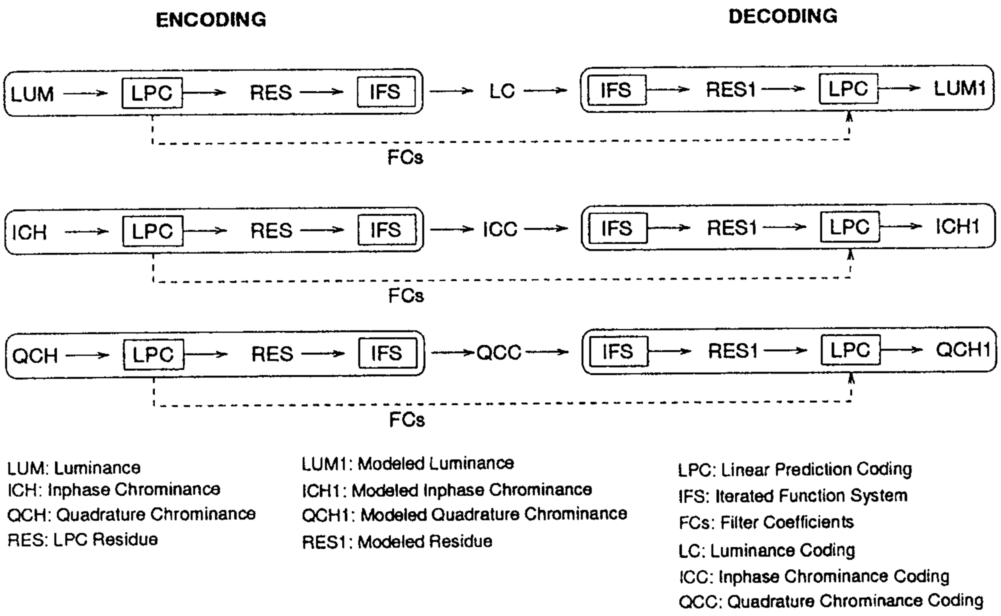

Fig. 1. ARIFS1: ARIFS performed independently on luminance and chrominance channels.

- ARIFS3: Applying ARIFS to the luminance channel but only IFS on the chrominance channels, as illustrated in Fig. 3.

The execution parameters for method ARIFS1 approximate luminance with finer detail than chrominance, since the latter is less important perceptively.

Along the same lines, the second method ARIFS2 calculates the LP filter coefficients based on the first chrominance component, and then uses them

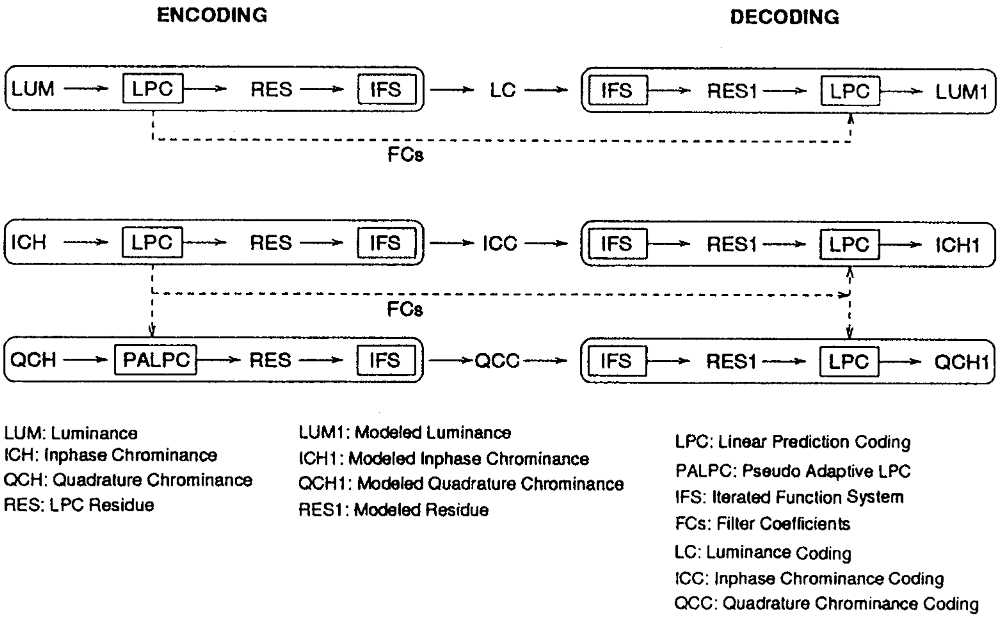

Fig. 2. ARIFS2: full ARIFS on luminance channel and shared LP filters on chrominance channels.

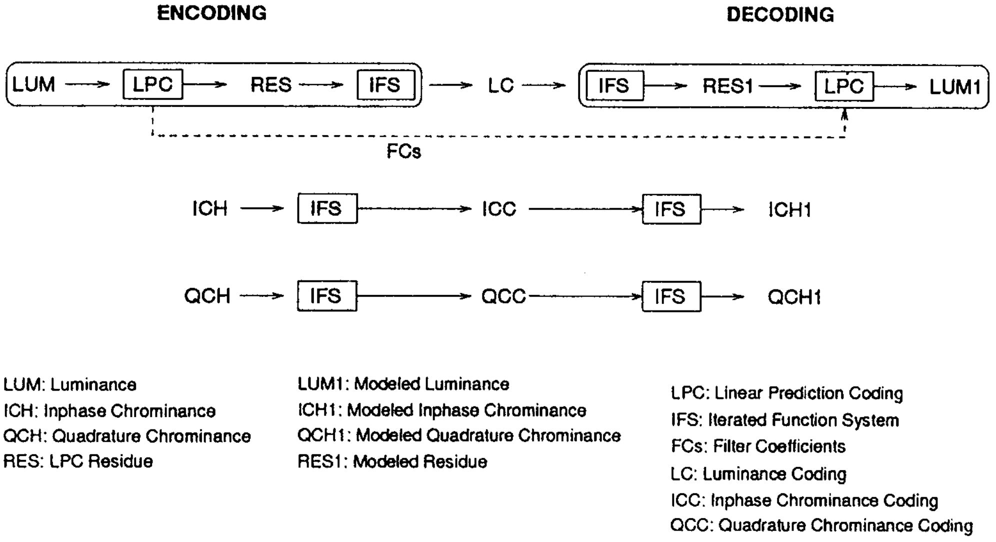

Fig. 3. ARIFS3: full ARIFS on luminance and IFS on chrominance.

to encode the second component as well. A specific filter is calculated for the encoding of the sensitive luminance information.

Finally, method ARIFS3 provides good compression because no filter coefficients have to be stored for the chrominance. In order to avoid the long computing times typical of IFS coding, the latter is performed with large tolerance values.

## 3. Experimental results

In order to evaluate the performance of ARIFS, several 24-bit 512 × 512 test images have been selected, but for the sake of brevity this paper shall only show the results obtained on Lena. The results obtained are presented in terms of two measures: peak signal-to-noise ratio (PSNR) and bit-rate (BR).

The BR is the sum of two terms: it is simply given by $\mathrm{BR} = b_1 + b_2$, where the first term $b_1$ accounts for information contained in the predictor, i.e., estimated filter coefficients and initial conditions, while the second term $b_2$ is due to IFS coding.

Both 16 × 16 and 32 × 32 blocks were used for LP. For each block there are $p = 6$ predictive coefficients; since reflection coefficients are utilized [6], 48 bits are enough to guarantee stability. Furthermore, there are pixels belonging to the original image.

IFS coding is always implemented with a 3-level quadtree (8 × 8, 16 × 16 and 32 × 32). Table 1 shows the outcome of computer simulations meant to compare the three hybrid-coding schemes. The PSNR values are averaged over the three channels of the RGB-reencoded image, while the partial bit-rates under $b_1$ can be reduced by a further 10–15% using Huffman coding.

The first two schemes (ARIFS1 and ARIFS2) provide the best results as far as PSNR is concerned. In particular, ARIFS2 performs better than ARIFS1 since the sharing of the filters allows for an improved bit-rate without significantly worsening the PSNR. In some cases, the PSNR/BR tradeoff of ARIFS2 is even better. Besides, ARIFS2 is the fastest of the three schemes.

The scheme ARIFS3 provides the best bit-rates but cannot reach the other schemes' PSNR values.

Table 1
A comparison between the three hybrid-coding schemes.

| BR | $b_1$ | $b_2$ | PSNR |
|---|---|---|---|
| ARIFS1 | | | |
| 0.45 | 0.33 | 0.12 | 28.00 |
| 0.54 | 0.33 | 0.21 | 28.51 |
| 0.65 | 0.33 | 0.32 | 28.64 |
| 0.77 | 0.33 | 0.44 | 29.05 |
| 0.89 | 0.33 | 0.56 | 29.97 |
| 0.96 | 0.33 | 0.66 | 31.30 |
| 1.20 | 0.33 | 0.87 | 31.89 |
| 1.71 | 0.33 | 1.38 | 34.86 |
| 1.97 | 0.33 | 1.64 | 36.69 |
| ARIFS2 | | | |
| 0.39 | 0.27 | 0.12 | 27.99 |
| 0.44 | 0.27 | 0.17 | 28.58 |
| 0.51 | 0.27 | 0.24 | 28.67 |
| 0.66 | 0.27 | 0.39 | 28.72 |
| 0.75 | 0.27 | 0.48 | 29.14 |
| 0.88 | 0.27 | 0.61 | 30.32 |
| 1.00 | 0.27 | 0.73 | 31.93 |
| 1.38 | 0.27 | 1.11 | 34.16 |
| 1.90 | 0.27 | 1.63 | 36.78 |
| ARIFS3 | | | |
| 0.23 | 0.20 | 0.03 | 26.53 |
| 0.26 | 0.20 | 0.06 | 27.07 |
| 0.41 | 0.20 | 0.21 | 27.51 |
| 0.47 | 0.20 | 0.27 | 27.62 |
| 0.68 | 0.20 | 0.48 | 27.83 |
| 0.86 | 0.20 | 0.66 | 27.96 |
| 1.00 | 0.20 | 0.80 | 28.04 |
| 1.52 | 0.20 | 1.32 | 30.72 |
| 2.04 | 0.20 | 1.84 | 32.34 |

An important result coming out from the experiments is that, as expected, it makes a big difference for IFS to work on the original image or on an LP residue: since the residual signal has a reduced dynamic range, it contains a larger number of 'flat' blocks, which do not require IFS computation since they can be simply represented by their average value – much more efficient than calculating a contractive transform.

Table 2 shows a comparison between a full IFS transform and the same transform performed on an LP residual: the columns labeled '$N_\mathrm{T}$' contain the number of blocks represented by actual transforms, while '$N_\mathrm{F}$' is the number of flat blocks.

Table 2
Performance of IFS applied to the original luminance component of image lena and to its residue

| Whole luminance | | | | Residue | | | |
|---|---|---|---|---|---|---|---|
| bpp | $N_T$ | $N_F$ | PSNR | bpp | $N_T$ | $N_F$ | PSNR |
| 0.19 | 1737 | 389 | 29.22 | 0.21 | 471 | 1724 | 30.54 |
| 0.36 | 3094 | 223 | 32.55 | 0.34 | 1513 | 1600 | 31.77 |
| 0.62 | 5276 | 161 | 34.97 | 0.65 | 3052 | 1573 | 33.90 |
| 0.83 | 6123 | 87 | 35.67 | 0.81 | 4127 | 1003 | 34.81 |
| 1.02 | 7076 | 25 | 36.22 | 1.00 | 5215 | 826 | 35.93 |

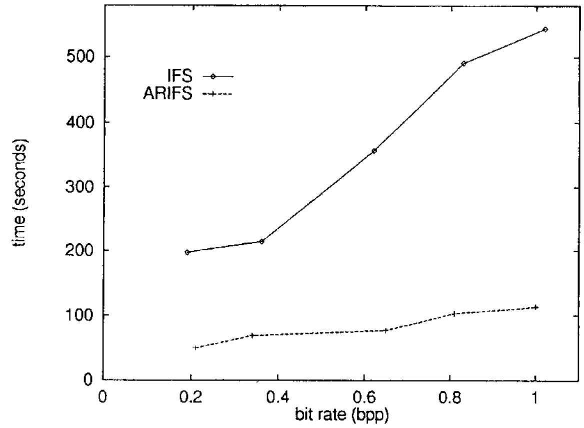

Fig. 4. Execution times for ARIFS and traditional IFS on 24-bit image Lena.

The net result of the increased number of flat blocks is that the computing time for ARIFS is significantly shorter than the time for IFS. As can be seen in Fig. 4, which plots the execution times on an IBM RS/6000, ARIFS2 is faster than IFS alone by a factor ranging from 4 to 5.

## 4. Conclusions

In this paper we have described three schemes for still-image compression based on a hybrid technique combining AR–LP with IFS on 24-bit color images.

The performance of the coding schemes has been tested on several images. Among the considered coding schemes, the sharing of LP filters yields the best bit-rate and PSNR. An interesting feature of the proposed schemes is their speed, which compares favorably to classic IFS techniques.

An interesting issue to be addressed in further study of this system might be the development of a related technique for video coding.

## References

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

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

[3] M. Das, S.Y. Tan, N.K. Loh, Adaptive predictive coding of images based upon multiplicative time series modeling, Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., 1987, pp. 1386–1389.

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

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

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

[7] 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 Process. ASSP-12 (6) (1984) 1213–1229.

[8] E.W. Jacobs, Y. Fisher, R.D. Boss, Image compression: A study of the iterated transform method, Signal Processing 29 (3) (1992) 251–263.