Hierarchical Interpretation of Fractal Image Coding and Its Application to Fast Decoding

Z. Baharav † D. Malah †

E. Karnin †§

† Technion - Israel Institute of Technology
Department of Electrical Engineering, Haifa 32000, Israel
Tel:+972-4-294745, fax:+972-4-323041, email: malah@ee.technion.ac.il
§ I.B.M. Israel Science and Technology

Abstract

The basics of a block oriented fractal image coder, are described. The output of the coder is an IFS (Iterated Function System) code, which describes the image as a fixed-point of a contractive transformation. A new hierarchical interpretation of the IFS code, which relates different scales of the fixed-point to the code, is presented and proved. The proof is based on finding a function of a continuous variable, from which different scales of the signal can be derived. Its application to a fast decoding algorithm is then described, leading typically to an order of magnitude reduction of computation time.

I Introduction

The use of fractal shapes to describe real world scenes has been shown to result in very realistic images [1]. This is due to the self-similarity property of fractal shapes, a property which is frequently encountered in real world scenes [2]. One way of creating a fractal shape is by considering it as a fixed-point of a contractive Iterated Function System (IFS) [3]. The result is that a complex image can be described by a rather small number of IFS parameters. Thus, the IFS provides an efficient representation of a fractal image. Two issues are involved in coding an image by an IFS :

encoding - Given an image, find a contractive IFS such that its fixed point is sufficiently 'close' to the image. The parameters of this IFS are then called the *IFS code* of the image. Throughout, the terms 'IFS' and 'IFS code' will be used interchangeably. This problem is known as the 'inverse problem'.

decoding - Given a contractive IFS, find its fixed point.

The problem of encoding has been widely addressed [3] - [6]. Since general images are not necessarily self similar, a way to find the IFS code is to partition the image into blocks, and to restrict the functions forming the IFS to a set of transformations on these blocks. Given the set of allowed transformations and the image partition, the task is to find an IFS such that its fixed point is as 'close' to the original image as possible. The problem of decoding can be solved by applying the contractive-mapping theorem [7]. Namely, iterate repeatedly the IFS on any starting image, and the resulting image will converge to the desired fixed-point.

The main drawback with this decoding scheme is its computational cost. This cost is high because the iterations are done on a full size image. In this paper we suggest a new hierarchical interpretation of the IFS code, and propose a *fast* decoding algorithm which is based on this interpretation.

II Mathematical background

Let X be a metric space with metric d. The transformation $T: X \rightarrow X$ is *contractive* if

$$\exists\, s \in [0,1) \mid \forall\, x,y \in X \quad d(T(x),T(y)) \leq s\, d(x,y) \tag{1}$$

If X is a complete metric space, the contraction mapping theorem [7] ensures the following :

- There exists a unique fixed-point $x_f \in X$, such that $T(x_f) = x_f$.
- For any point $x \in X$ the sequence $\{T^n(x) : n = 0, 1, \ldots\}$ converges to x_f. That is, $\lim_{n\to\infty} T^n(x) = x_f \ \ \forall x \in X$.

Thus, the point x_f is uniquely specified by T, and can be approached to by iterations. In a typical application, the space X is the space of images, x_f is an image which should be as close as possible to the image being encoded, and T is actually a system of functions, namely the IFS itself.

III IFS coding

The method to be described is mainly attributed to Barnsley and Jacquin [3], and is performed block-wise. In order to make the development more lucid, it will be described in terms of coding a 1-dimensional signal (vector), instead of an image (matrix). The results are fully applicable and easily extendible to 2-dimensional signals (images), as exemplified in section VI.

Let $\mathbf{U}_N$ denote the vector to be encoded. For convenience, the length of the vector, N, is assumed to be an integer power of 2. Partition $\mathbf{U}_N$ into $M_R \stackrel{\triangle}{=} \frac{N}{B}$ sub-vectors $\mathbf{R}_i$, of length B each, such that

$$\mathbf{R}_i(j) = \mathbf{U}_N((i-1)B + j) \quad , \quad i = 1, 2, \cdots, M_R \quad ; \quad j = 1, 2, \cdots, B \tag{2}$$

$\{\mathbf{R}_i\}_{i=1}^{M_R}$ are called the *range-blocks* of $\mathbf{U}_N$, with i denoting the range-block index. Note that the range-blocks are tiling the vector $\mathbf{U}_N$. From $\mathbf{U}_N$ create $M_D \stackrel{\triangle}{=} (\frac{N-D}{D_h} + 1)$ sub-vectors of length D such that

$$\mathbf{D}_i(j) = \mathbf{U}_N((i-1)D_h + j) \quad , \quad i = 1, 2, \cdots, M_D \quad ; \quad j = 1, 2, \cdots, D \tag{3}$$

$\{\mathbf{D}_i\}_{i=1}^{M_D}$ are called the *domain-blocks* of $\mathbf{U}_N$, and their ensemble is named the *domain-pool*. D_h is the shift between two adjacent domain-blocks. Note that if $D_h < D$ the domain-blocks are overlapping.

Now, let the set of allowed transformations be of the form :

$$\mathbf{T}(\mathbf{D}) = a\varphi(\mathbf{D}) + b\mathbf{1}_B \tag{4}$$

- a - Scalar scaling factor, $|a| < 1$. • b - Scalar offset value, $b \in \Re$; $\mathbf{1}_B$ - A vector of size B of all 1's.
- φ - Spatial contraction function, defined here to be:

$$\varphi(\mathbf{D})(j) \stackrel{\triangle}{=} \frac{1}{2}(\mathbf{D}(2j-1) + \mathbf{D}(2j)) \quad , \quad j = 1, 2, \cdots, B \tag{5}$$

i.e., φ contracts blocks of size $D = 2B$ into blocks of size B, by averaging pairs of adjacent elements in $\mathbf{D}$.

To simplify the development, we assume here that B is also an integer power of 2, $B = 2^l$, hence $D = 2^{l+1}$. We also assume that the shift is $D_h = B$.

The encoding/decoding process is as follows:

encoding - For each range-block $\mathbf{R}_i$, find the domain-block and the parameters (a_i, b_i), for which $d(\mathbf{R}_i, \hat{\mathbf{R}}_i)$ is minimized, where $\hat{\mathbf{R}}_i$ is the best approximation (in terms of the metric $d(\cdot,\cdot)$) to $\mathbf{R}_i$ achievable by the allowed transformations. In equation form

$$\hat{\mathbf{R}}_i = a_i\varphi(\mathbf{D}_{m_i}) + b_i\mathbf{1}_B \quad , \quad i = 1, 2, \cdots, M_R \tag{6}$$

where m_i is the index of the domain-block which best matches the i'th range-block. The parameters (a_i, b_i, m_i) are called the *transformation parameters* . The IFS code of the image is the ensemble of these M_R transformations, which relate domain-blocks to range-blocks, and thus consists of $3M_R$ transformations parameters.

decoding - Start from *any* vector of length N, $\mathbf{V}_N^0$. Set $n = 0$, choose $\epsilon > 0$, and do the following:

1. Create from the vector $\mathbf{V}_N^n$ a pool of domain-blocks of length D each, using a shift of D_h.
2. Compute from the IFS and the set of domain-blocks a set of range-blocks, according to (6).
3. Form a new vector of length N, $\mathbf{V}_N^{n+1}$, by concatenating the last computed range-blocks.
4. Check the termination-condition: $d(\mathbf{V}_N^{n+1}, \mathbf{V}_N^n) \leq \epsilon$. If not satisfied, set $n \leftarrow n+1$ and goto step 1.
5. $\mathbf{V}_N^{n+1}$ is the decoded vector.

For demonstration, we show in Fig. 1(a) an IFS representation, and in Fig. 1(b) its fixed point. In this case $N = 16$, $B = 4$, $D_h = 4$, and there are therefore $M_R = 4$ range-blocks and $M_D = 3$ domain-blocks.

Range-block index i	Scale a_i	Domain-block index m_i	Offset b_i
1	0.5	1	12
2	0.5	3	8
3	0.5	2	0
4	0.5	1	4

$D/B = 2$, $D_h = B = 4$, $M_R = 4$

(a)

20	12	4	12

(d)

22	18	10	14	6	2	14	10

(c)

23	21	17	19	11	9	15	13	5	7	3	1	15	13	9	11

(b)

Fig. 1: (a) IFS code , (b) decoding with B=4 , (c) B=2 (d) B=1

IV IFS and Resolution

As previously explained, the IFS describes relations between different parts of the IFS fixed-point vector which we denote as $\mathbf{V}_N$. These parts were referred to as domain-blocks and range-blocks. The IFS determines directly the number of range-blocks, M_R, but it *does not*, however, imply any specific size of these blocks (only the ratio $\frac{D}{B}$ must be set according to the definition of the spatial-contraction function φ; in (5) this ratio is 2). Hence, decoding an IFS code results in a vector of length N if the value of B is chosen to be $B = N/M_R$, or it can result in a vector of a different length, e.g., $N/2$, by using $B = N/(2M_R)$. An example is shown in Fig. 1(b)-(d), where the decoding is done with $B = 4$, $B = 2$, and $B = 1$, respectively. The fact that for each value of B a different fixed-point is obtained is by no means contradictive to the contraction mapping theorem! Since although the IFS has a unique fixed point in a complete metric space, each of the decodings in Fig. 1(b)-(d) is actually done in a different space, namely in $\Re^{16}, \Re^8$ and $\Re^4$, respectively. A relation among all the achievable vectors, related to a given IFS, will now be established.

Definition 1 *Let* $f \in L^\infty[0,1]$. *Define* $f_r(i)$, $i = 1, \cdots, r$, *by*

$$f_r(i) \triangleq r \int_{(i-1)\frac{1}{r}}^{i\frac{1}{r}} f(x)\, dx \tag{7}$$

$f_r(i)$ *denotes the* function $f(x)$ at resolution r . *We say that* $f_{r_1}(i)$ *is* finer (*i.e., with higher resolution*) *than* $f_{r_2}(i)$ (*which is* coarser) *if* $r_1 > r_2$.

Theorem 1 *Given an IFS code, there exists a unique function* $f(x) \in L^\infty[0,1]$ *such that a vector* $\mathbf{V}_N$ *is a fixed point of the IFS iff it is equal to the function* $f(x)$ *at resolution* $r = N$, *i.e.,*

$$\mathbf{V}_N(j) = f_N(j) \quad , \quad j = 1, 2, \cdots, N \tag{8}$$

The function $f(x)$ *will be called the* IFS embedded function .

The proof is provided in [8].
From Theorem 1 the following corollary is obtained:

Corollary 1 *Let $\mathbf{V}_N$ be a fixed-point in $\Re^N$ of a given IFS, then $\mathbf{V}_{\frac{N}{2}}$ is a fixed-point (in $\Re^{\frac{N}{2}}$) of the same IFS iff :*

$$\mathbf{V}_{\frac{N}{2}}(j) = \frac{1}{2}(\mathbf{V}_N(2j-1) + \mathbf{V}_N(2j)) \quad , \quad j = 1, 2, \cdots, \frac{N}{2} \tag{9}$$

The proof follows immediately from the Theorem.

V Hierarchical representation

After establishing a relation between $\mathbf{V}_{\frac{N}{2}}$ and $\mathbf{V}_N$, as described by Corollary 1, we further investigate this relation, and extend it.

Given an IFS code and N, it can be decoded using $B = B^{(\frac{N}{2})} = N/(2M_R)$ to get the vector $\mathbf{V}_{\frac{N}{2}}$. The domain-blocks of $\mathbf{V}_{\frac{N}{2}}$ are created on the basis of (3), with size $D^{(\frac{N}{2})} = 2B^{(\frac{N}{2})}$ and ${D_h}^{(\frac{N}{2})} = B^{(\frac{N}{2})}$, by

$$\mathbf{D}_i^{(\frac{N}{2})}(j) = \mathbf{V}_{\frac{N}{2}}((i-1){D_h}^{(\frac{N}{2})} + j) \quad , \quad i = 1, 2, \cdots, M_D \quad ; \quad j = 1, 2, \cdots, D^{(\frac{N}{2})} \tag{10}$$

where we use the superscript in $\mathbf{D}_i^{(\frac{N}{2})}$ to indicate that the domain-blocks are retrieved from a vector of length $\frac{N}{2}$. Comparing equations (3),(5),(9), and (10) we get :

$$\varphi(\mathbf{D}_i^{(N)}) = \mathbf{D}_i^{(\frac{N}{2})} \tag{11}$$

Hence, (6) can be written as

$$\hat{\mathbf{R}}_i^{(N)} = a_i \mathbf{D}_{m_i}^{(\frac{N}{2})} + b_i \mathbf{1}_B \quad , \quad i = 1, 2, \cdots, M_R \tag{12}$$

Now, if $\mathbf{V}_{\frac{N}{2}}$ is a fixed point of the IFS , then the vectors $\varphi(\mathbf{D}_i^{(N)})$ obtained from (11) and (10) are equal to those computed directly from $\mathbf{V}_N$ via equations (3) and (5) (replacing $\mathbf{U}_N$ by $\mathbf{V}_N$), *if* $\mathbf{V}_N$ is a fixed point. Thus, the $\hat{\mathbf{R}}_i^{(N)}$ computed in (12) are the range-blocks of the fixed-point vector $\mathbf{V}_N$.

In summary, if $\mathbf{V}_{\frac{N}{2}}$ is a fixed-point of the IFS, we have shown a direct way for computing $\mathbf{V}_N$ from it, by using (12), without the need to perform iterations!

Using the same reasoning once more, we start by assuming that $\mathbf{V}_{\frac{N}{4}}$ is given as a fixed-point of the IFS, then $\mathbf{V}_N$ can be computed by transforming twice: once from $\mathbf{V}_{\frac{N}{4}}$ to $\mathbf{V}_{\frac{N}{2}}$, and then from $\mathbf{V}_{\frac{N}{2}}$ to $\mathbf{V}_N$. The algorithm for transforming $\mathbf{V}_{\frac{N}{4}}$ to $\mathbf{V}_{\frac{N}{2}}$, and then $\mathbf{V}_{\frac{N}{2}}$ to $\mathbf{V}_N$, is the *same* , and is derived from the IFS code, as described by (12). The reason the algorithm is the same is that by Corollary 1, $\mathbf{V}_{\frac{N}{4}}$, $\mathbf{V}_{\frac{N}{2}}$, and $\mathbf{V}_N$, are all fixed-points of the *same* IFS in the appropriate spaces. An example of a hierarchical structure is given in Fig. 1(b)-(d). This structure is a pyramid of the IFS fixed-points, with $\mathbf{V}_{\frac{N}{2^p}}$ comprising the p'th level in the pyramid. The level with the coarsest resolution (Fig. 1(d)) is called the *top-level*. Such an hierarchical structure is natural to fractals, which have the property of self-similarity at different resolutions.

The question arises, what is the smallest size of the top-level that will still enable us to use the same technique for finding $\mathbf{V}_N$? This is answered by the following theorem.

Theorem 2 *Let $\mathbf{V}_N$ be a fixed-point in $\Re^N$ of a given IFS, and let $B = B^{(N)} = 2^l$, $D = D^{(N)} = 2B$ and $D_h = B$, then the number of levels in the pyramid of IFS fixed-points is*

$$\log_2(B) + 1 = l + 1 \tag{13}$$

leading to a top-level size of $N/2^l$ $(= M_R)$.

PROOF : Ascending one level in the hierarchy, means halving the size of the range-block, B. Since this size must be at least 1 in order that the IFS could be applied, the theorem follows. Q.E.D ■

VI Fast decoding

A fast decoding method, which we call *hierarchical decoding*, follows directly from the above interpretation of the IFS code. In this method, one begins by computing the *top-level*. This can be done by iterations (using $B = 1$). Then, one follows with the deterministic algorithm (12) to advance to a higher resolution. The process of advancing to a higher resolution is repeated, until the desired vector-size is achieved. This method is compared below with the conventional *iterative decoding*, in which the iterations are done on a full scale image. The computational savings obtained in using the hierarchical decoding method stems from the fact that the iterations are done *only* in order to find the top-level, which is a small size vector.

Computational cost

Throughout, the following notations and definitions will be used:

- S - computation time of one summation.
- M - computation time of one multiplication.
- t_c - total computation time.
- I - number of iterations.

Furthermore, multiplications by $\frac{1}{2}$ or $\frac{1}{4}$ will not be counted (counting them will show even greater savings using the hierarchical decoding, because they occur mainly in the iterative method, when performing the computation φ).

1. <u>One dimensional case</u>

   With N the vector length and B the range-block size in the original vector, we have:

   - *Iterative decoding* - Referring to (4) and (5). For a single iteration the computation time is: $N \cdot [(1+1)S + 1M] = N \cdot [2S + 1M]$. Thus, the the total computation time is:

   $$t_c^I = I \cdot N \cdot [2S + 1M] \tag{14}$$

   - *Hierarchical decoding* - Recall that at the top-level there are $\frac{N}{B} = M_R$ elements (by Theorem 2). Hence, according to the result in (14), we know the computation time needed to compute the top-level. According to Theorem 2 there are $\log_2(B)$ levels in the pyramid, excluding the top-level, with $2^p \frac{N}{B}$ elements in the p'th level ($p = 0$ at the top-level). Referring to (12), we can compute the cost of transforming from the top-level to $\mathbf{V}_N$:

   $$\sum_{p=1}^{\log_2(B)} (2^p \frac{N}{B}) \cdot [1S + 1M] = (B-1)\frac{N}{B} 2 \cdot [S + M] \doteq N \cdot [2S + 2M] \tag{15}$$

   The total computation time is therefore

   $$t_c^H \doteq N \cdot \left\{ \frac{I}{B} [2S + 2M] + [2S + 2M] \right\} \tag{16}$$

2. <u>Two dimensional case</u>

   Note that in the 2D case, performing the contraction φ requires the computation of the mean of 4-elements. With N^2 the number of elements in the original image (matrix) and $B \times B$ the range-block size, we have:

   - *Iterative decoding* - For a single iteration, the computation time is: $N^2 \cdot [(3+1)S + 1M] = N^2 \cdot [4S + 1M]$ Thus, the total computation time is:

   $$t_c^I = I \cdot N^2 \cdot [4S + 1M] \tag{17}$$

- *Hierarchical decoding* - Here, there are $(\frac{N}{B})^2$ elements in the top-level. Hence, according to the previous result, we know the computation time needed to compute the top-level. Similar to the 1-dimensional case, the cost of transforming from the top-level to the $N \times N$ image is:

$$\sum_{p=1}^{\log_2(B)} (4^p(\frac{N}{B})^2) \cdot [1S + 1M] = \frac{4}{3}(B^2 - 1)(\frac{N}{B})^2 \cdot [1S + 1M] \doteq N^2\frac{4}{3} \cdot [1S + 1M] \tag{18}$$

Thus, the total computation time is:

$$t_c^H \doteq N^2 \cdot \left\{ \frac{I}{B^2} \cdot [4S + 1M] + \frac{4}{3} \cdot [1S + 1M] \right\} \tag{19}$$

Usually B^2 is much larger than the number of iterations, I, (typically $B^2 = 64$ and $I < 8$) so that the first term can be neglected . Hence:

$$t_c^H \doteq N^2 \cdot \frac{4}{3} \cdot [1S + 1M] \tag{20}$$

For example : Assuming $M = k \cdot S$, we find that the ratio of computation times is :

$$Q \triangleq \frac{t_c^H}{t_c^I} = \frac{N^2 \cdot \frac{4}{3} \cdot S \cdot [1 + k]}{N^2 \cdot I \cdot S \cdot [4 + k]} = \frac{4 \cdot [1 + k]}{3 \cdot I\,[4 + k]} \tag{21}$$

(a) For a floating point processor, we can assume $k = 1$, leading to $Q = \frac{8}{15 \cdot I}$ which shows an order of magnitude savings if $I \geq 6$.

(b) For a fixed point processor, we can assume $k = 8$, leading to $Q = \frac{1}{I}$.

It is seen that the larger the value of I, the more advantageous is the hierarchical decoding method.

VII Conclusion

A new hierarchical interpretation of the IFS code was presented and proved. Its application to fast decoding was shown to reduce the computation time by more then an order of magnitude. The hierarchical interpretation provides an insight into the encoding/decoding procedure. This insight is natural to fractals, which have a self-similarity characteristic under different scales. Other applications of the hierarchical interpretation, such as applying it to obtain super-resolution of a given image, are currently being investigated.

References

[1] Heinz-Otto Peitgen and Dietmar Saupe. *The Science of Fractal Images.* Springer-Verlag, Berlin, 1988.

[2] Benoit B. Mandelbrot. *The Fractal Geometry of Nature.* W. H. Freeman and Company, San Francisco, 1982.

[3] Arnaud E. Jacquin. *A Fractal Theory of Iterated Markov Operators with applications to Digital Image Coding.* PhD thesis, Georgia Institute of Technology, August 1989.

[4] M. F. Barnsley and A. E. Jacquin. Application of recurrent iterated function systems to images. SPIE-Visual Communications and Image Processing, Vol. 1001, pp. 122–131, 1988.

[5] J. M. Beaumont. Image data compression using fractal technique. *BT Technol. J.*, Vol.9,No.4, pp.93–109, October 1991.

[6] Steve Kocsis. Digital compression and iterated function systems. SPIE-Application of Digital Image Processing, Vol. 1153, pp. 19–27, 1989.

[7] M. F. Barnsley. *Fractals Everywhere.* Academic Press, San Diego, 1988.

[8] Baharav Zachi. Hierarchical interpretation of fractal image coding. Technion, M.Sc thesis. In preparation.