A Multi-Resolution Framework for Fractal Image Representation and its applications

Z. Baharav *, H. Krupnik *, 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 starting point of this paper is the basic fractal coder suggested by Jacquin. The coder finds and encodes the parameters of a partitioned iterated function system (PIFS), which approximates the signal as a fixed-point of a contractive transformation.

The work presented here can be divided into two parts. The first part begins with a presentation of the hierarchical structure of the PIFS code. This structure relates the code and its fixed-point in different resolutions. It is shown that there exists a function of a continuous variable which is directly related to the PIFS. It is shown that by properly manipulating this function, called the PIFS embedded-function*, one can compute the fixed-points related to the code in any desired resolution.*

We end the first part with a brief description of several applications, such as a fast non-iterative decoder, a method for fractal interpolation of the signal via its PIFS code, and an improved collage-bound.

This research was supported by the fund for the promotion of research at the Technion.

The hierarchical representation of the IFS code establishs a link to multiresolution analysis of the code. In the second part of the work we present a method for mapping similar regions within an image in the wavelet domain. We first show a new formulation by which mappings, which are identical to conventional IFS mappings, are found in the Haar wavelet transform domain. We then propose to use different *mother-wavelet than Haar. The use of other wavelets than Haar results in the overlap of mapped regions, thus avoiding the typical blockiness associated with conventional blockwise IFS.*

By Using mappings between the wavelet coefficients we can enrich the family of transformations. This leads to a new fractal-wavelet image coder that reduces blockiness, and improves the PSNR of the reconstructed image.

1 Introduction

The subject of fractals began as a pure mathematical subject related to chaos (e.g. von-Koch snowflake, Sierpanski-gasket [6]). In 1992, Jacquin [9] used the idea of fractals in a compression algorithm for images. Many interesting features associated with the 'original' fractals seem to fade away as the IFS coding of pictures evolved. For example, the property of *self similarity* at different resolutions, inherent to fractals, does not show up in a simple form in IFS codes.

In this work we will show that the properties of self similarity at different resolutions, and the related notion of fractal-dimension, do exist in IFS coding. Moreover, we will use the hierarchical interpretation to arrive at a new blockless wavelet-fractal coder. The original coder, suggested by Jacquin [9], can then be viewed as a special case of this coder, when one uses Haar wavelets.

The work is organized as follows:

Section 2 briefly reviews the formulation of IFS coding, introduces notation and presents basic examples.

Section 3, contains a theorem, relating the different resolutions of a signal to its IFS code. It then gives an hierarchical interpretation of the theorem.

Section 4 describes various applications, like a fast decoding method, and super resolution via the IFS code. It also introduces the imporatnt notion of the IFS embedded function.

Section 5 briefly reviews the wavelet-series and their relation to octave-band subband decomposition.

Section 6 describes a new formulation of the common IFS code, by which similar regions within an image are mapped using the Haar-discrete-wavelet-transform.

Section 7 describes a new improved coding algorithm based on the IFS mappings in the DWT domain (DWT-IFS).

Section 8 investigates super-resolution via the DWT-IFS code.

Section 9, which is the last one, contains the summary and conclusions.

2 Formulation of IFS Coding/Decoding

In this section a formulation of the IFS coding/decoding is discussed, accompanied by an example. To simplify notation we usually refer to 1-dimensional signals, and we will call them either vectors or blocks. Extensions to 2-dimensions are ususally immediate, and are partly considered in the applications sections, as well as in the discussion of wavelets. The following notation will be used: vectors are in bold-face letters (like $\mathbf{a}$), matrices are in bold upper-case ones (like $\mathbf{A}$), and scalars are in regular letters (like a, A).

2.1 Encoding

The task of finding the IFS code of a vector $\mathbf{u_0} \in \Re^N$ is the task of finding a contractive transformation W, such that its fixed point is as close as possible to $\mathbf{u_0}$.

In the encoding process one restricts the set of allowed transformations W, to be systems of M_R functions w_i. Thus, by loosely using the union notation, we can write the operation of W on a vector $\mathbf{v} \in \Re^N$ as:

$$\begin{aligned} \mathbf{u} &= W(\mathbf{v}), \qquad \mathbf{u} \in \Re^N \\ W(\mathbf{v}) &= \bigcup_{i=1}^{M_R} w_i(\mathbf{v}) \end{aligned} \tag{1}$$

Each w_i is further restricted to be of the form:

$$\begin{aligned} w_i \;:\; \Re^D &\mapsto \Re^B \\ \mathbf{d}_{m_i} &\mapsto \mathbf{r}_i = w_i(\mathbf{d}_{m_i}) = a_i \varphi(\mathbf{d}_{m_i}) + b_i \mathbf{1}_B \end{aligned} \tag{2}$$

where:

$\mathbf{d}$ - is called the *domain-block*, and is of size D. $\mathbf{d}_{m_i}$ is thus the m_i'th domain-block, which is simply a block of D consecutive elements extracted from $\mathbf{v}$. The use of the subscript m_i comes to stress the fact that the domain-block $\mathbf{d}_{m_i}$ is mapped to $\mathbf{r}_i$. For now, no specific mechanism for extracting the blocks will be discussed.

$\mathbf{r}$ - is called the *range-block*, and is of size $B < D$. $\mathbf{r}_i$ is thus the i'th range-block, and $\mathbf{r}_i \in \mathbf{u}$.

φ - *Spatial contraction* function, which transforms blocks of size D to blocks of size B .

a_i - Scalar *scaling* factor,

$$a_i \in \Re, \qquad |a_i| < 1 \tag{3}$$

b_i - Scalar *offset* value, $b_i \in \Re$.

$\mathbf{1}_B$ - A vector of size B of all 1's.

The group of all triplets (a_i, b_i, m_i) is called the *transformation parameters.*

The length of $\mathbf{u}$, which is the result of concatenating M_R range-blocks of size B each (1), is

$$N = M_R \cdot B \tag{4}$$

Moreover, the concatenation of range-blocks can also be written as:

$$\mathbf{u}((i-1) \cdot B + j) = \mathbf{r}_i(j) \quad ; \quad i = (1, \cdots, M_R), \qquad j = (1, \cdots, B) \tag{5}$$

Now the mechanism of computing $\mathbf{u} = W(\mathbf{v})$, where all the parameters describing W are known, can be described as follows:

I : Algorithm for Computing the Transformation $\mathbf{u} = W(\mathbf{v})$

1. `For` $i = 1$ `to` M_R

   (a) `Extract the` $\mathbf{d}_{m_i}$ `block from the vector` $\mathbf{v}$.

   (b) `Compute`

$$\mathbf{r}_i = w_i(\mathbf{d}_{m_i}) = a_i \varphi(\mathbf{d}_{m_i}) + b_i \mathbf{1}_B \tag{6}$$

2. *Concatenate* the range-blocks thus obtained, $\mathbf{r}_i, i = (1, \cdots, M_R)$, in the natural order, to get the new vector $\mathbf{u}$. The length of the vector $\mathbf{u}$ is $N = M_R \cdot B$.

W described above is called a *block-wise* transformation, the reason being evident from the computational algorithm.

So far the discussion of the w_i's was quite general. At this stage we will make further restrictions and assumptions about the parameters, in order to make the discussion both more practical and lucid.

II : Parameters Restrictions and Assumptions

1. N - The size of the original vector to be encoded is an integer power of 2.

2. $B = 2^r$ - The size of a range-block. B is therefore also some integer power of 2.

3. $D = 2B$ - The size of a domain-block.

4. $D_h = B$ - D_h is defined to be the shift between two adjacent domain-blocks. Thus, the number of domain-blocks is $M_D \stackrel{\triangle}{=} (\frac{N-D}{D_h} + 1)$, and each domain-block is given by

$$\mathbf{d}_{m_i}(j) = \mathbf{v}((m_i - 1)D_h + j) \tag{7}$$

$$m_i = 1, 2, \cdots, M_D \quad ; \quad j = 1, 2, \cdots, D \tag{8}$$

Note that the domain-blocks are overlapping, since $D_h < D$.

5. $\varphi(\cdot)$ - The spatial contraction function is defined to be:

$$\varphi(\mathbf{d}_{m_i})(j) \stackrel{\triangle}{=} \frac{1}{2}(\mathbf{d}_{m_i}(2j) + \mathbf{d}_{m_i}(2j-1)), \qquad j = 1, 2, \cdots, B \tag{9}$$

i.e., $\varphi(\cdot)$ contracts blocks of size $D = 2B$ into blocks of size B, by averaging pairs of adjacent elements in $\mathbf{d}_{m_i}$.

The contents of the IFS code of a vector, namely the parameters which define W, can now be summarized:

III : IFS-code

1. B - The size of the range-blocks.
2. M_R - The number of range-blocks.
3. M_R triplets of the transformation-parameters (a_i, b_i, m_i).

All other relevant parameters needed for decoding, such as $D = 2B, D_h = B, N = M_R B$, and others, are derived from the IFS code using the previous assumptions.

The encoding process to be described consists of finding a W which satisfies $\mathbf{u}_0 \cong W(\mathbf{u}_0)$. Thus, $\mathbf{u}_0$ is approximately the fixed-point of W. Since W defines uniquely its fixed-point, storing W (namely the parameters that defines it) defines a lossy code for $\mathbf{u}_0$.

IV : IFS-Coding of $\mathbf{u}_0$

1. Store B in the code-file.
2. Store M_R in the code-file, where $M_R = N/B$, and N is the length of $\mathbf{u}_0$.
3. Partition $\mathbf{u}_0$ into M_R range-blocks, as described in (5),

$$\mathbf{r}_i(j) = \mathbf{u}_0((i-1) \cdot B + j) \tag{10}$$

$$i = (1, \cdots, M_R), \qquad j = (1, \cdots, B)$$

4. Extract from $\mathbf{u}_0$ the $M_D = (\frac{N-D}{D_h} + 1)$ domain-blocks, according to (7)

$$\mathbf{d}_q(j) = \mathbf{u}_0((q-1)D_h + j) \tag{11}$$

$$q = 1, 2, \cdots, M_D, \qquad j = 1, 2, \cdots, D$$

5. For $i = 1$ to M_R

(a) Find the best parameters (a_i, b_i, m_i), such that

$$d(\mathbf{r}_i, a_i\varphi(\mathbf{d}_{m_i}) + b_i\mathbf{1}_B) \tag{12}$$

is minimized, where $d(\cdot)$ is a distance function.

(b) Store the parameters (a_i, b_i, m_i) in the code file.

This formulation is now demonstrated by the use of a numerical example. Fig. 1(a)-(b) presents a vector $\mathbf{u}_0$ and its IFS code. The IFS is given in a table form. By performing the transformation described by the IFS on $\mathbf{u}_0$, one can verify that the vector $\mathbf{u}_0$ in this example is a fixed-point of the transformation (namely $\mathbf{u}_0 = W(\mathbf{u}_0)$), and thus the coding in this case is lossless.

2.2 Decoding

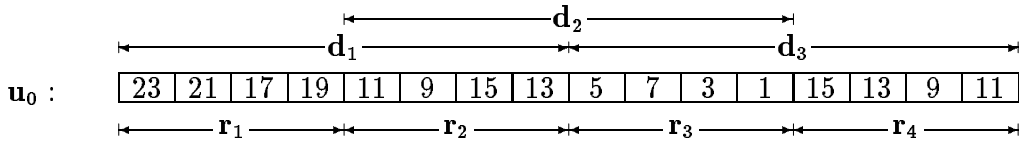
The process of decoding is straight-forward, since it involves the finding of a fixed-point of a contractive transformation W. This can be done by repeatedly iterating W on *any* initial vector, until a desired closeness to the fixed-point is reached [4].

In Table 1 the decoding of the IFS code in Fig. 1b is demonstrated, starting from an initial vector of all 0's.

iteration	vector															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	12	12	12	12	0	0	0	0	4	4	4	4	8	8	8	8
2	18	18	12	12	10	10	12	12	0	0	2	2	10	10	4	4
3	21	18	17	18	8	9	13	10	5	6	0	1	13	10	9	10

Table 1: Decoding by iterations

Following the example, we will now demonstrate an imporatant fact about the decoding (and the IFS defintion). In the above example, the IFS code, with the prescribed $B = B_1 = 4$, resulted a transformation $W^1 : \Re^{16} \mapsto \Re^{16}$, with a fixed point $\mathbf{f}^1 \in \Re^{16}$. Suppose, however, that the value of B is changed to some other value than the one prescribed in the IFS code, e.g.



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

$B = 4$, $M_R = 4$

(b)

$$0.5 \cdot \varphi(\mathbf{d}_1) + 12 =$$
$$0.5 \cdot \varphi([23, 21, 17, 19, 11, 9, 15, 13]) + 12 =$$
$$0.5 \cdot [22, 18, 10, 14] + 12 =$$
$$[23, 21, 17, 19]$$

(c)

Figure 1: (a) An original vector $\mathbf{u}_0$ (b) IFS code of $\mathbf{u}_0$ (c) Example of computing the first code-line on $\mathbf{u}_0$

$B = \frac{1}{2}B_1 = 2$. We have thus created a new transformation, denoted by $W^{\frac{1}{2}}$, which is clearly a contractive transformation in $\Re^8$. The decoding process of $W^{\frac{1}{2}}$ will therefore yield a fixed-point vector $\mathbf{f}^{\frac{1}{2}}$ of length $N = N_{\frac{1}{2}} = 8$, which is half the length of $\mathbf{f}^1$. Thus, we conclude that the IFS can be decoded in different spaces, yielding a different fixed-point in each space. Throughout the remainder of this work, we will use the notion that $B = B_1$ results in $\mathbf{f}^1$, and $B = jB_1$ results in $\mathbf{f}^j$. In Fig. 2(a)-(c) 3 fixed points of the same IFS code (using different B's) are described, each one being in a different space, $\Re^{16}$, $\Re^8$, and $\Re^4$, respectively.

The exact relation between those different fixed-points, and its interpre-

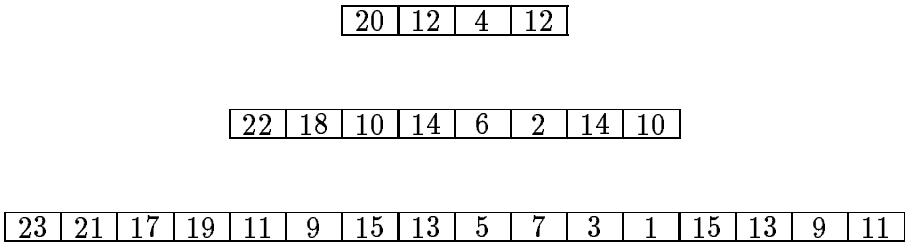


Figure 2: Decoding with (a) B=4 , (b) B=2 , and (c) B=1

tation, are examined in detail in the sequel.

3 Hierarchical interpretation

As described in the previous section, each value of B leads to a different transformation, with a different fixed-point. The following theorem describes relations between two different fixed-points, given that B is halved.

Theorem 1 (Zoom) *Given an IFS code, it leads to W^1 with $B = B_1$, and to $W^{\frac{1}{2}}$ with $B = B_1/2$. Let the fixed-points of these transformations be $\mathbf{f}^1$ and $\mathbf{f}^{\frac{1}{2}}$, respectively, then:*

Zoom-out:

$$\mathbf{f}^{\frac{1}{2}}(j) = \frac{1}{2}\left\{\mathbf{f}^1(2j) + \mathbf{f}^1(2j-1)\right\}, \qquad j = (1, \cdots, \frac{N_1}{2}) \tag{13}$$

where $N_1 \stackrel{\triangle}{=} M_R \cdot B_1$.

Zoom-in:

$$\mathbf{f}^1((i-1)B_1 + j) = a_i \mathbf{f}^{\frac{1}{2}}((m_i - 1){D_h}^{\frac{1}{2}} + j) + b_i \tag{14}$$
$$i = (1, \cdots, M_R), \qquad j = (1, \cdots, B_1)$$

where ${D_h}^{\frac{1}{2}} \stackrel{\triangle}{=} \frac{{D_h}^1}{2} = \frac{B_1}{2}$.

For proof see [19].

An interpretation of the theorem is as follows: The theorem establishes a relation between the pair $\mathbf{f}^1$ and $\mathbf{f}^{\frac{1}{2}}$. The same relation is carried over to the pair $\mathbf{f}^{\frac{1}{2}}$ and $\mathbf{f}^{\frac{1}{4}}$,

$$\mathbf{f}^{\frac{1}{4}}(j) = \frac{1}{2}\left\{\mathbf{f}^{\frac{1}{2}}(2j) + \mathbf{f}^{\frac{1}{2}}(2j-1)\right\} \tag{15}$$

$$\mathbf{f}^{\frac{1}{2}}((i-1)B_{\frac{1}{2}} + j) = a_i \mathbf{f}^{\frac{1}{4}}((m_i - 1)D_h{}^{\frac{1}{4}} + j) + b_i \tag{16}$$

The relation also holds for the pair $\mathbf{f}^{\frac{1}{4}}$ and $\mathbf{f}^{\frac{1}{8}}$, and so on. The collection of fixed-points can thus be described in terms of an hierarchical structure, as was shown in Fig. 2. This structure is a *pyramid of the IFS fixed-points*, with $\mathbf{f}^{\frac{1}{2^p}}$ comprising the p'th level of the pyramid. Thus $\mathbf{f}^{\frac{1}{2^p}}$ has $N/2^p$ elements. The level with the coarsest resolution (Fig. 2c) is called the *top-level*.

An intuitive understanding of the process can be gained by noting that the domain-blocks of $\mathbf{f}^1$, after their contraction, are actually blocks which are contained in $\mathbf{f}^{\frac{1}{2}}$. In a formalistic form, this can be shown by writing down the expression for the j'th element in the contraction of the m_i'th domain-block of $\mathbf{f}^1$ (eqns. (7)-(9)):

$$\varphi(\mathbf{d}_{m_i}{}^1)(j) = \frac{1}{2} \cdot \left\{\mathbf{f}^1((m_i-1)D_{h1} + 2j - 1) + \mathbf{f}^1((m_i-1)D_{h1} + 2j)\right\} \tag{17}$$

$$j \in (1, \cdots, B_1)$$

According to (13), the right hand side is *exactly* $\mathbf{f}^{\frac{1}{2}}((m_i-1)D_{h1}/2 + j)$, and if we further denote as before, $D_h{}^{\frac{1}{2}} \stackrel{\triangle}{=} D_{h1}/2$ and $\mathbf{d}_{m_i}{}^{\frac{1}{2}}$ as the m_i'th domain-block of $\mathbf{f}^{\frac{1}{2}}$, we conclude that:

$$\varphi(\mathbf{d}_{m_i}{}^1)(j) = \mathbf{d}_{m_i}{}^{\frac{1}{2}}(j), \qquad j \in (1, \cdots, B_1) \tag{18}$$

The question arises, what is the smallest size of the top-level such that the above relations between two adjacent levels still hold ? This is answered by the following corollary.

Corollary 1 *Let $\mathbf{f}^1$ be a fixed-point in $\Re^N$ of a given IFS, and let $B = B_1 = 2^l$, $D = D_1 = 2B$ and $D_h = B_1$, then the number of levels in the pyramid of IFS fixed-points is*

$$\log_2(B) + 1 = l + 1 \tag{19}$$

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

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 corollary follows. Q.E.D ■

3.1 DC Orthogonalization

A modification of the conventional mapping described in (6) is a mapping with orthogonalization to the block averages (DC) [13]. The reconstructed block and the transformation can be described as:

$$\mathbf{r}_i = w_i(\mathbf{d}_{m_i}) = a\varphi(\mathbf{d}_{m_i} - \overline{\mathbf{d}}_{m_i}\mathbf{1}_B) + \overline{\mathbf{r}}_i\mathbf{1}_B \tag{20}$$

Where $\overline{\mathbf{d}}_{m_i}$ and $\overline{\mathbf{r}}_i$ are the mean values of the corresponding blocks. notice that in this case $\overline{\mathbf{r}}_i$ replaces the conventional offset factor b_i in (6). Therefore the transformation parameters in this case are :

$$W = \bigcup w_i, \qquad w_i = \{a_i, \overline{\mathbf{r}}_i, m_i\} \tag{21}$$

The two main advantages of defining W as in (20), over the one in (6), are :

1. Its fixed point can be found in no more than $\log_2(B)+1$ iterations [13].
2. The scaling factors values a_i do not have to be restricted.

The DC block orthogonalization does not affect the pyramidal relations between the different scales of the fixed point as described in Sec.(3). To go up the fixed point pyramid (decrease resolution) one should average samples, as in (13), while to go down the pyramid one should replace (14) with :

$$\mathbf{f}^1((i-1)B_1 + j) = a_i\left\{\mathbf{f}^{1/2}((m_i-1){D_h}^{1/2} + j) - \overline{\mathbf{d}}_{m_i}\right\} + \overline{\mathbf{r}}_i \tag{22}$$
$$\mathrm{i} = (1, \cdots, M_R), \qquad j = (1, \cdots, B_1)$$

4 Applications

4.1 Fast Decoding

A fast decoding method, which we call *hierarchical decoding*, follows directly from the hierarchical interpretation of the IFS code. In this method, one begins by computing the *top-level*. This can be done by iterating the IFS with $B = 1$, i.e. applying W to an initial vector of size M_R, until a fixed-point is reached (or closely approached). Then, one follows the deterministic algorithm (14) to advance to a higher resolution. The process of advancing to a higher resolution is repeated, until the desired vector-size is achieved.

In [2] a detailed account of the computations is given, and it is shown that for the 2-dimensional case, the savings are in an order of magntitude.

An even faster non-iterative decoding method is echived by hierarachical decoding of a fixed point of a transformation that consist mappings with DC orthogonalization. In this case the top-level can be computed in a non-iterative way. The decoding algorithm will be as follows [13] :

V : Hierarchical decoding with DC-orthogonalization

1. Construct the pyramid level $\mathbf{f}^{\frac{1}{B}}$ in which $\mathbf{f}^{\frac{1}{B}}(i) = \overline{\mathbf{r}}_i$, i.e. the range blocks mean values that are included in the IFS code.
2. 'Zoom-in', $\log_2(B)$ times, starting from $\mathbf{f}^{\frac{1}{B}}$, to get $\mathbf{f}^1$ as described in Sec.(3) and Sec.(4.1) This time use (22) to copy domain blocks from a given pyramid level to tile the next lower level (higher resolution).

In the following sections we refer to this procedure as the *'reference algorithm'* and show that an equivalent algorithm can be performed in the *Haar Discrete Wavelet Transform.*

4.2 Super-resolution

The subject of resolution is inherently related to the discretization of a function of a continuous variable. The process of discretization is called sampling. In the following definition we define a specific method of sampling.

Definition 1 *Given a function* $G(x) \in L^{\infty}[0,1]$, *define* $G_r(i)$ *by*

$$G_r(i) \stackrel{\triangle}{=} r \int_{(i-1)\frac{1}{r}}^{i\frac{1}{r}} G(x)\, dx, \qquad i = (1, \cdots, r) \tag{23}$$

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

Theorem 2 (IFS Embedded Function) *Given an IFS code, there exists a unique function* $G(x) \in L^{\infty}[0,1]$ *such that a vector* $\mathbf{v}_N \in \Re^N$ *is a fixed point of the IFS iff it is equal to the function* $G(x)$ *at resolution* $r = N$, *i.e.,*

$$\mathbf{v}_N(j) = G_N(j), \qquad j = 1, 2, \cdots, N \tag{24}$$

The function $G(x)$ *is called the* IFS embedded function.

For proof see [19].

In Fig. 3 a somewhat intuitive demonstration of the IFS embedded function theorem is given. The IFS is the one described previously, in Fig. 1. Its fixed-points, for $B = 1$, $B = 2$, and $B = 4$ are shown in the figure as functions of a continuous variable $x \in [0,1]$. For example, the fixed-point using $B = 1$ is the vector $[20, 12, 4, 12]$, which has $N = 4$ elements. Thus, the fixed-point is drawn as a piece-wise constant function:

$$f(x) = \begin{cases} 20 & x \in [0, \frac{1}{4}) \\ 12 & x \in [\frac{1}{4}, \frac{1}{2}) \\ \vdots & \vdots \end{cases} \tag{25}$$

The embedded function $G(x)$ is also shown. One easily sees that the fixed-points 'approach' the IFS embedded function. Moreover, it is seen that the value of each function, in each of its intervals, equals the mean of the IFS Embedded function over the appropiate interval, as described by (23).

Super-resolution deals with finding a higher resolution of a given discrete signal. For example, suppose a vector $\mathbf{v}^1$ with elements $\mathbf{v}^1(i) = g_N(i), i = 1, 2, \cdots, N$ is given, which is the signal $g(x) \in L^{\infty}[0,1]$ at resolution N, where $g(x)$ is unknown to us. The goal is to find a vector $\mathbf{v}^2$ of length $2N$, which approximates the signal $g(x)$ at resolution $2N$, namely $g_{2N}(i), i = 1, 2, \cdots, 2N$. The process of transforming from a given resolution to a higher one is also called zoom-in.

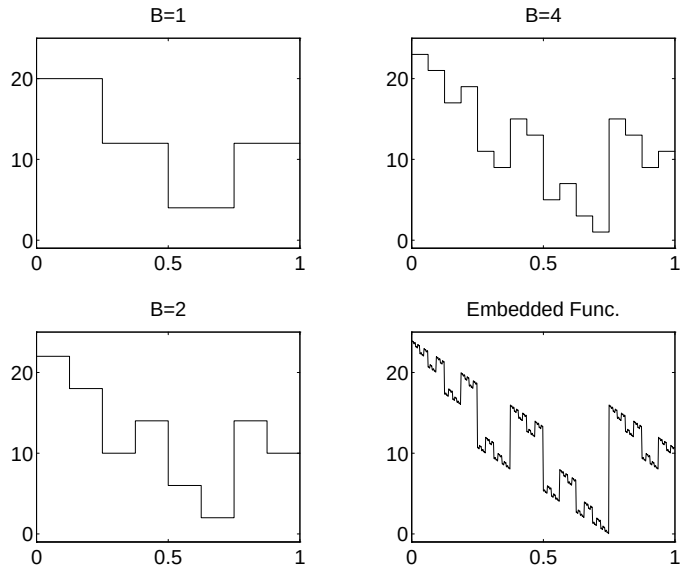


Figure 3: Fixed-points with B=1 ,B=2 ,B=4 ,and the corresponding IFS Embedded Function

The hierarchical representation suggests a simple method for finding higher resolution representations from a given resolution. Given a fixed-point vector $\mathbf{f}^1$ of length N (which is the signal $g(x) \in L^\infty[0,1]$ at resolution N, where $g(x)$ is not known to us), its IFS code enables us to build a hierarchical structure, called the pyramid of fixed-points. The IFS code also gives us an algorithm for relating between two adjacent levels in the pyramid (Zoom theorem, eqns. (14)-(13)). Thus, after finding the IFS code of $\mathbf{f}^1$, all that is needed in order to get a vector of length $2N$ is to apply the zoom-in algorithm in (14) to $\mathbf{f}^1$, and get a new vector $\mathbf{f}^2$. The vector $\mathbf{f}^2$ is an integral part of the pyramid of fixed-points, and is the fixed-point of the IFS code when using $B = 2B_1$. This vector can be used as an approximation of the higher resolution representation, namely an approximation of g_{2N}. The subject of super resolution will be pursued further in Sec.(8) in conjuction with a wavelet based encoder.

We have already introduced the IFS embedded function. Moreover, indicative bounds on its fractal dimension can be found directly from the matrix representation of the IFS code [19]. This dimension tells us about the na-

ture of the super-resolution vectors (see [4] for detailed discussion concerning fractal interpolation functions). By introducing certain constraints in the coding procedure (such as on $\frac{D}{B}$, or $|a_i|$), one can affect the dimension of the resultant IFS embedded function.

We will not elaborate here upon the possibility to arrive at any desired rational zoom factor, nor about the possibility of incorprating different sampling methods in the above disccussion. These subjects appear in [19].

4.3 Improved collage theorem

A detailed description of an improved collage theorem can be found in [14], where simulations and a discussion are also given. Here, we will merely describe it in general terms, because its full description involves additional notation, which we have no room for in this paper.

We have previously described the fact that the IFS fixed points can be arranged in a pyramid structure of different resolutions. Similarily, one can create a pyramid structure composed of different resolutions of the original signal (by simply averaging elements), where the highest resolution is that of the original vector. The idea in the new collage theorem is that if we require that the original vector and the fixed point should be 'close' as possible in the highest resolution, it implies a requirement on the 'closeness' of the other levels of the two pyramids.

This idea can be utilized to arrive at a new bound on the decoding error. This new bound considers the signal in different resolutions. This idea has many implications, and perhaps the most important one is that it suggests a different method for performing the encoding in a manner which takes into account the error in each level of the pyramid, and not only at the original resolution level.

5 The Discrete Wavelet Transform

5.1 Wavelet Series

An *Orthonormal Wavelet Series* [16] [11] is an orthonormal basis of $\mathrm{L}_2(\Re)$ whose members $\psi_{k,l}$ are self-similar functions. That is, each of the functions $\psi_{k,l}$ is a scaled and shifted version of a *mother wavelet* Ψ,

$$\psi_{k,l}(t) = \sqrt{2^l}\Psi(2^l t - k), \qquad (k,l) \in \mathcal{Z}^2 \tag{26}$$

where k is the translation index and l is the scaling index.

As a result, every function $f(t) \in \mathrm{L}_2(\Re)$ can be described as a linear combination of the wavelet functions :

$$f(t) = \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_{k,l}\psi_{k,l}(t) \tag{27}$$

and the coefficients are the inner product between $f(t)$ and $\psi_{k,l}(t)$

$$d_{k,l} = \int_{-\infty}^{\infty} f(t)\psi_{k,l}(t)dt \tag{28}$$

The approximation of f at resolution L is defined as

$$A_L\{f(t)\} = \sum_{l=-\infty}^{L} \sum_{k=-\infty}^{\infty} d_{k,l}\psi_{k,l}(t) \tag{29}$$

This approximation can also be calculated as a linear combination of translations of scaled versions of a *Scaling Function* Φ

$$\phi_{k,L}(t) = \sqrt{2^L}\Phi(2^L t - k)k \in \mathcal{Z} \tag{30}$$

$$A_L\{f(t)\} = \sum_{k=-\infty}^{\infty} a_{k,L}\phi_{k,L}(t) \tag{31}$$

The set of coefficients $a_{k,L}$ is defined as *the discrete approximation* of f at resolution L, and is denoted $A_L = \{a_{k,L}|k \in Z\}$.

The *detail signal* is a signal that is spanned by the set of wavelet functions $\psi_{k,L}$. I.e,

$$D_L\{f(t)\} = \sum_{k=-\infty}^{\infty} d_{k,L}\psi_{k,L}(t) \tag{32}$$

The coefficients that represent the detail signal are denoted :

$$D_L = \{d_{k,L}|k \in \mathcal{Z}\} \tag{33}$$

It is easy to see that the detail signal is the difference between two approximations of the signal in two sequential resolutions L and $L-1$:

$$D_{L-1}\{f(t)\} = A_L\{f(t)\} - A_{L-1}\{f(t)\} \tag{34}$$

With the mother wavelet (and the scaling function) one can associate a pair of digital filters h_h and h_l such that all discrete approximations A_l and detail signal coefficients D_l $(l < L)$ can be calculated from A_L by means of digital filtering and decimation :

$$A_{L-1} = (A_L * h_l) \downarrow 2, \qquad D_{L-1} = (A_L * h_h) \downarrow 2 \tag{35}$$

A_L can be reconstructed from A_{L-1} and D_{L-1} using the folowing:

$$A_L = (A_{L-1} \uparrow 2) * h_l + (D_{L-1} \uparrow 2) * h_h \tag{36}$$

where $\uparrow 2$ represents upsampling by adding a zero valued sample between every two consecutive samples. These relations between the aproximations and the detail signal in different resolutions are also known as subband decomposition of a discrete signal as described in the folowing subsection.

5.2 Wavelet Transforms of Discrete Signals

Let us describe the *Discrete Wavelet Transform* (*DWT*) of a 1D discrete signal $\mathbf{v}(n)$ as an octave-band subband decomposition (Fig. 4, top), using a pair of *Para-Unitary Quadrature Mirror Filters* (*QMF*)[16]. The octave-band filtering is implemented by splitting over and over the lowest subband with the two-band pair of filters. Such a decomposition with three splits is shown in Fig. 4(bottom). At each split the higher branch represents convolution with high pass filter h_h and 2:1 decimation, and the lower branch represents convolution with lowpass filter h_l and decimation. Let us denote the signal as $\mathbf{v}(n) \equiv \mathbf{v}_{\mathcal{L}}^1(n)$ and the subbands as :

$$\mathbf{v}_{\mathcal{L}}^{1/2^l} = (\mathbf{v}_{\mathcal{L}}^{1/2^{l-1}} * h_l) \downarrow 2, \qquad \mathbf{v}_{\mathcal{H}}^{1/2^l} = (\mathbf{v}_{\mathcal{L}}^{1/2^{l-1}} * h_h) \downarrow 2 \tag{37}$$

It is easy to see that the relations between the discrete aproximations $(a_{k,l})$ and the detail signal coefficients of a countinuous variable function $(d_{k,l})$ are, in fact, *the same* as the relations between a discrete signal and its

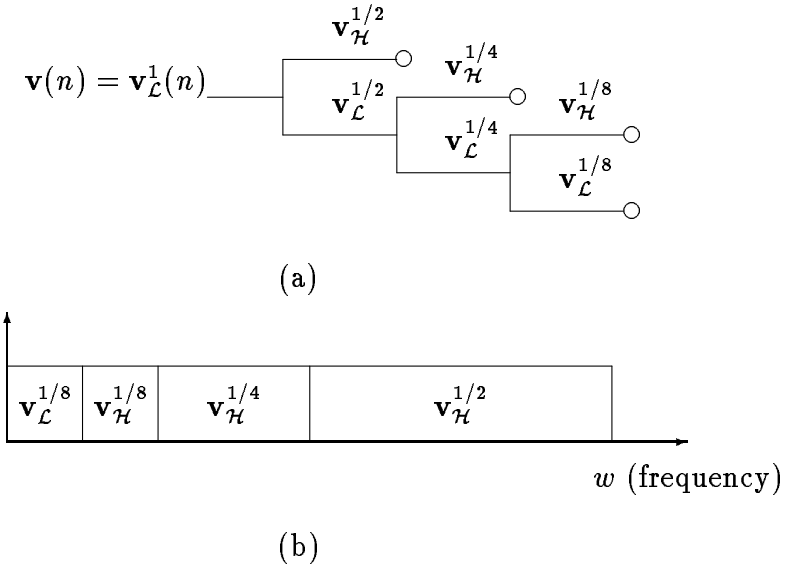


Figure 4: Octave band subband decomposition. (a) Splitting tree. (b)The subbands on the frequency axis.

subband coefficients that result from splitting it into octave bands. I.e., if the discret signal $\mathbf{v}(n)$ is a discrete approximation A_0 of a continuous function $f(t)$ ($\mathbf{v}(n) \equiv a_{n,0}$) then :

$$\mathbf{v}_{\mathcal{L}}^{1/2^l} = A_l, \qquad \mathbf{v}_{\mathcal{H}}^{1/2^l} = D_l, \qquad \forall L < 0 \tag{38}$$

5.2.1 Finite Length DWT Representation

Since the DWT is an orthonormal transformation, the subband decomposition of a *finite length* signal $\mathbf{v}(n)$ can be represented by a unitary matrix. We will denote this matrix by $U_{L,H}^{N}$. Here L stands for the Number of subband splittings, H for a specific choice of the QMF filters and N is the length of the signal. We will denote the DWT of a signal $\mathbf{v}$ by $\mathbf{v}_{dwt}$ i.e :

$$[\mathbf{v}_{\mathcal{L}}^{1/2^L}, \mathbf{v}_{\mathcal{H}}^{1/2^L}, \ldots, \mathbf{v}_{\mathcal{H}}^{1/2}] = \mathbf{v}_{dwt} \equiv U_{L,H}^{N}\mathbf{v}, \qquad \mathbf{v} = (U_{L,H}^{N})^T\mathbf{v}_{dwt} \tag{39}$$

In the following sections we use a pyramidal description of the DWT, in which the subband splitting is described by two pyramids. The first is a lowpass pyramid that consists of $l = 0, \ldots, L$ levels, where only $\mathbf{v}_{\mathcal{L}}^{1/2^L}$ is part of the DWT. The second is a highpass pyramid that consist of $\mathbf{v}_{\mathcal{H}}^{1/2^l}$, $l = 1, \ldots, L$

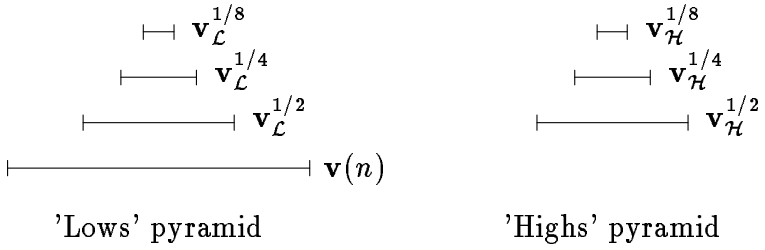


Figure 5: pyramidal description of the DWT

and is fully contained in the DWT. The two pyramids are illustrated in Fig. 5 (in this figure $L = 3$).

6 IFS mappings via the Haar DWT

The use of wavelets in fractal image coding should not be surprising. IFS fractal images have self similarities among different scales of the images and Wavelet basis functions are self similar. Therefore, it is natural to believe that Wavelet transforms of an IFS fixed point should have a unique structure. Such a structure has been explored in [8] on synthetic IFS fractals. In this work we wish to devlope the connection between a PIFS fixed point and the discrete wavelet transform. A conection that has been intorduced recently in [10] and [5]

In section 4.1 we summarized the *'reference algorithm'*. In this section we wish to construct an equivalent algorithm in the Haar-DWT. The aim of this section is to show the following :

1. Given an IFS, the Haar-DWT high-bands coefficients of its fixed point can be calculated *directly* from the lowest band - which is the fixed point at the lowest resolution ($\mathbf{f}_{\mathcal{L}}^{1/B}$).

2. The IFS mapping parameters, obtained by the *reference algorithm* in section Sec.(3.1) can be found directly from the Haar-DWT coefficients of the input signal $\mathbf{v}$.

The Haar-DWT is an orthonormal transform that is associated with the lowpass filter $h_l \equiv [\sqrt{1/2}, \sqrt{1/2}]$, and the highpass filter $h_h \equiv [\sqrt{1/2}, -\sqrt{1/2}]$, as a QMF pair. Note that 2:1 decimation of the output of the convolution

of a signal $\mathbf{v}$ with h_l is equivalent (up to a $\sqrt{1/2}$ factor) to applying the operator φ that simply averages pairs of samples (9) of the signal $\mathbf{v}$. I.e., $\mathbf{v}_{\mathcal{L}}^{1/2} = \sqrt{2}\varphi(\mathbf{v})$. As a result, if D_h is even, one can extract the scaled-down domain blocks $\varphi(\mathbf{d}_{m_i})$ *directly* from $\mathbf{v}_{\mathcal{L}}^{1/2} = \sqrt{2}\mathbf{v}^{\frac{1}{2}}$.

It is observed that, since the length of end of each of the Haar filters is 2, the Haar-DWT is a blockwise transform which maps 2^L input samples (where L is the number of splittings) to 2^L DWT coefficients. As a result, every range block $\mathbf{r}_i$ of size $B = 2^L$ can be represented by a set of 2^L DWT coefficients. We call the set of coefficients that represent a range block a *range-tree* and denote the range-tree obtained from a block $\mathbf{r}_i$ by $\mathbf{S}_{\mathbf{r}_i}$:

$$\mathbf{S}_{\mathbf{r}_i} = [r_i^{(1)}, \ldots, r_i^{(B)}]^t = U_{L,Haar}^B \mathbf{r}_i \tag{40}$$

The coefficients $\mathbf{S}_{\mathbf{r}_i}$ are part of the DWT of the whole signal $\mathbf{v}$

$$\begin{aligned} \mathbf{v} &= \bigcup \mathbf{r}_i \\ r_i^{(1)}, \ldots, r_i^{(B)} &\in U_{L,Haar}^N \mathbf{v} \end{aligned} \tag{41}$$

the coefficients in $\mathbf{S}_{\mathbf{r}_i}$ can be subdivided into groups based on the subband that they have been taken from :

$$\begin{aligned} r_i^{(1)} &= \mathbf{v}_{\mathcal{L}}^{1/B}(i) \\ r_i^{(2)} &= \mathbf{v}_{\mathcal{H}}^{1/B}(i) \\ \left[r_i^{(3)}, r_i^{(4)}\right] &= [\mathbf{v}_{\mathcal{H}}^{2/B}(2i-1), \mathbf{v}_{\mathcal{H}}^{2/B}(2i)] \\ &\cdots \\ \left[r_i^{(B/2+1)}, \ldots, r_i^{(B)}\right] &= [\mathbf{v}_{\mathcal{H}}^{1/2}((B/4)i+1), \ldots, \mathbf{v}_{\mathcal{H}}^{1/2}((B/2)i)] \end{aligned} \tag{42}$$

We have seen that the scaled-down domain-block is part of the subband $\mathbf{v}_{\mathcal{L}}^{1/2}$. Let us assume that $D_h = 2B$, i.e., the scaled-down domain-blocks tile $\mathbf{v}_{\mathcal{L}}^{1/2}$. This is somewhat more restrictive assumption than the one in Sec.(3) where we selected D_h to be B. As a result, one can extract a sub-tree of coefficients $\mathbf{S}_{\mathbf{d}_{m_i}}$ that is the Haar-DWT of the *scaled-down* domain block $\varphi(\mathbf{d}_{m_i})$ from the DWT of $\mathbf{v}$ with $L+1 = \log_2(B)+1$ splittings :

$$
\begin{aligned}
\mathbf{S}_{\mathbf{d}_{m_i}} &= [d_{m_i}^{(1)}, \ldots, d_{m_i}^{(B)}] = \sqrt{2} U_{L,Haar}^{B} \varphi(\mathbf{d}_{m_i}) \\
d_{m_i}^{(1)}, \ldots, d_{m_i}^{(B)} &\in U_{L,Haar}^{N/2} \mathbf{v}_{\mathcal{L}}^{1/2} \subset U_{L+1,Haar}^{N} \mathbf{v} \\
d_{m_i}^{(1)} &= \mathbf{v}_{\mathcal{L}}^{1/2B}(m_i) \\
d_{m_i}^{(2)} &= \mathbf{v}_{\mathcal{H}}^{1/2B}(m_i) \\
\left[d_{m_i}^{(3)}, d_{m_i}^{(4)}\right] &= [\mathbf{v}_{\mathcal{H}}^{1/B}(2i-1), \mathbf{v}_{\mathcal{H}}^{1/B}(2m_i)] \\
&\ldots \\
\left[d_{m_i}^{(B/2+1)}, \ldots, d_{m_i}^{(B)}\right] &= [\mathbf{v}_{\mathcal{H}}^{1/4}((B/4)m_i + 1), \ldots, \mathbf{v}_{\mathcal{H}}^{1/4}((B/4)m_i)] \quad (43)
\end{aligned}
$$

Note that $r_i^{(1)}$ and $d_{m_i}^{(1)}$ are the only lowpass coefficients in $\mathbf{S}_{\mathbf{r}_i}$ and $\mathbf{S}_{\mathbf{d}_{m_i}}$, respectively. These coefficients are equal, up to a gain factor of $\sqrt{2}^L$, to their corresponding blocks mean values.

Let us now consider a signal $\mathbf{f}^1$ that is a fixed point of an IFS transformation W. $\mathbf{f}^1$ is tiled with range blocks such that for any block, $\mathbf{r}_i$, there is a corresponding scaled-down domain block $\varphi(\mathbf{d}_{m_i})$ in $\mathbf{f}^{\frac{1}{2}}$ such that

$$
\mathbf{r}_i - \overline{\mathbf{r}}_i = a_i(\varphi(\mathbf{d}_{m_i}) - \overline{\mathbf{d}}_{m_i}\mathbf{1}_B) \quad (44)
$$

Applying the unitary matrix $U_{L,Haar}^{B}$ on both sides of (44) gives :

$$
[0, r_i^{(2)}, \ldots, r_i^{(B)}] = \sqrt{2} a_i [0, d_{m_i}^{(2)}, \ldots, d_{m_i}^{(B)}] \quad (45)
$$

Suppose now that we decompose the whole fixed point $\mathbf{f}$ with a $L+1 = \log_2(2B)$-split, Haar-DWT, as in Fig. 6. Assuming that $D_h = 2B$, and based on the previous observations, each of the range and domain blocks can be reconstructed from the coefficients tree that is extracted from the DWT of the *whole* fixed point. In Fig. 6, the DWT coefficients of a four samples range block in $\mathbf{f}$ (shaded) and scaled-down domain-block (white) in $\mathbf{f}^{1/2}$, are marked.

Since all AC coefficients $r_i^{(p)}$ and $d_{m_i}^{(p)}$ $(2 \leq p \leq B)$ are contained in the DWT of $\mathbf{f}$ (with $L+1$-splits), and since for each p, $d_{m_i}^{(p)}$ is from a subband of lower frequency than $r_i^{(p)}$, (45) shows that the coefficients of the higher bands can be calculated from those of the lower bands. That is, one can start with $\mathbf{f}_{\mathcal{H}}^{1/2B}$, calculate all the coefficients which are in $\mathbf{f}_{\mathcal{H}}^{1/B}$, and so on. This

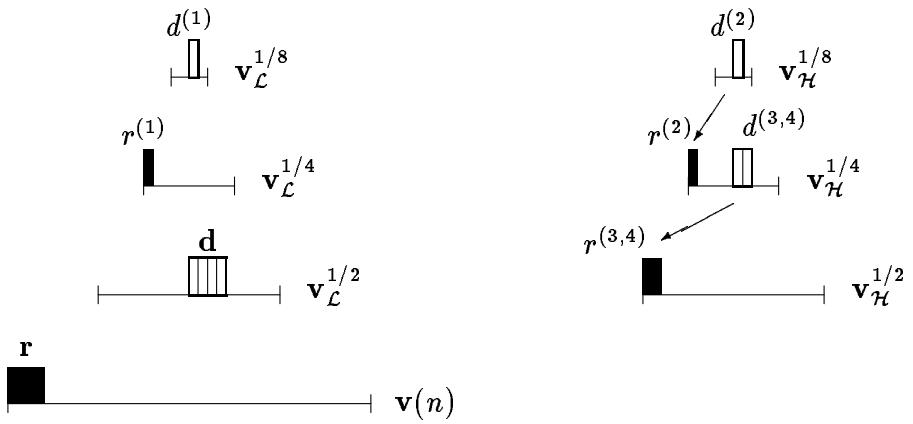


Figure 6: Representing coefficients trees of a range and a domain block.

calculation can be interpreted as a recursive extrapolation, or prediction of the higher bands of the signal from the lowest one. At each stage another subband is extrapolated and so a higher resolution of the fixed point can be obtained, quite similar to the hierarchical decoder which is described in Sec.(3).

Can we find the IFS code in the wavelet domain as well? The answer depends on the metric. If one uses the standard inner product metric l^2 (a very common choice), then, due to the norm preserving characteristics of the DWT transform, the approximation error minimization can be performed in the DWT domain:

$$\|(\mathbf{r}_i - \overline{\mathbf{r}}_i) - a_i\varphi(\mathbf{d}_{m_i} - \overline{\mathbf{d}}_{m_i})\|_2^2 = \sum_{p=2}^{B}(r_i^{(p)} - \sqrt{2}a_i d_{m_i}^{(p)})^2 \qquad (46)$$

Thus, the minimization of the distance can be performed by finding the best matches between AC range and domain-trees. In each tree there are $2^L - 1$ coefficients as in the right pyramid of Fig. 6. Each range and domain-tree consists of coefficients from different subbands, representing the same location of the signal.

We can now gather the observetaions above and describe an encoding/decoding algorithm in the Haar-DWT. This algorithm yields exactly the same results as the *reference algorithm* and will be refered to as the *DWT-IFS* algorithm.

A summary of that algorithm is as follows :

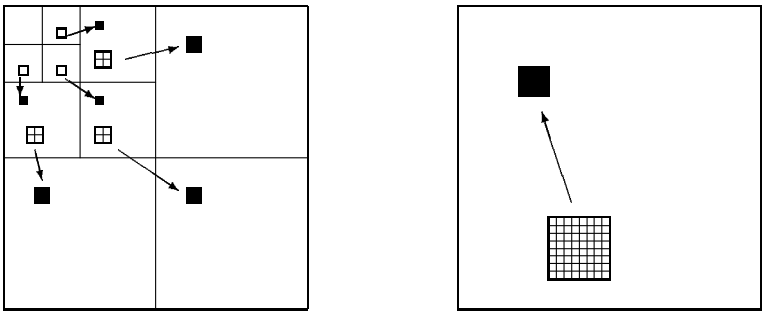
VI : Finding a DWT-IFS (Encoding)

1. Compute the DWT with $L+1(=log_2(B)+1)$ splits of the signal.
2. Extract range and domain-trees (with AC coefficients only).
3. For each range-tree $\mathbf{S}_{\mathbf{r}_i}$, find the best domain-tree $\mathbf{S}_{\mathbf{d}_{m_i}}$ and scaling factor a_i which minimizes (46).
4. The IFS code consists of the upper levels of the lowpass and highpass pyramids ($\mathbf{f}_{\mathcal{L}}^{1/2B}$ and $\mathbf{f}_{\mathcal{H}}^{1/2B}$), the scale factors a_i, and the indices m_i for each tree.

VII : Finding the Fixed-Point of the DWT-IFS (Decoding)

1. Copy the lowpass and highpass levels of the corresponding pyramids from the code.
2. Extrapolate downwards in the highpass pyramid using the scale factors and the indices from the IFS code.
3. Compute the Inverse DWT.

The term "DWT-IFS" has been choosen for this algorithm inspite of the fact that, for finite resolution images, the process is ended in deterministic number of steps, since the decoded image is identical to the one that is echieved using the conventional IFS decoding process. Moreover, in Sec.(8), we show that the algorithm can iterate "infinite" number of times and achieve an infinite-resolution continuous image.



(a) (b)

Figure 7: Representing coefficients trees of a range block and a corresponding domain block. (a) The coefficients in the DWT domain. (b) The blocks in the spatial domain.

6.1 2D-IFS in the Haar-Wavelet Domain

In order to apply the above algorithm to images there is a need to extend it to two dimensions. This is most easily done with *separable* 2D QMF's. The subband decomposition of a separable octave-band QMF is a quadtree partition of the 2D frequency domain as illustrated in the left side of Fig. 7. In such a decomposition every split of the low band $v_{\mathcal{LL}}^{1/2^l}$ consists of four bands, labeled $v_{\mathcal{LL}}^{1/2^{l+1}}$, $v_{\mathcal{LH}}^{1/2^{l+1}}$, $v_{\mathcal{HL}}^{1/2^{l+1}}$, $v_{\mathcal{HH}}^{1/2^{l+1}}$

Each DC-less BxB range or scaled-down domain block can be represented by a tree of $2^L x 2^L - 1$ ($L = \log_2(B)$) highpass coefficients that are extracted from the 2D separable Haar transform (2D Haar-DWT) of the whole image. The subdivision of the 2D frequency plane, the coefficient-tree that represent a range block and the coefficient-tree that represends a scaled-down domain block are shown in Fig. 7. In the left side of the figure, there are 15 shaded 2D Haar-DWT coefficients. These coefficients are the DWT of the 4x4 (shaded) range block of an image shown in the right. The 15 white coefficients represent a single domain-tree corresponding to a scaled-down version of the 8x8 square that is also shown on the right of Fig. 7. Therefore, it easy to see that the 2D encoding and decoding process is as described for the 1D case.

The extrapolation of the subbands is also marked (by the arrows) in Fig. 7(a). We would like to emphasize that the 2D extrapolation to calculate the higher bands is done here with the same parameters in all three directions

(HL, LH and HH)

6.2 block flipping and rotation in the DWT domain

The conventional fractal image coder uses rotated and flipped versions of the domain blocks in order to enrich the domain pool [9]. It can be shown that all the conventional isometries of a scaled-down domain block can also be implemented at a very low computational cost in the 2D-DWT transform domain. This is due to the fact that linear phase, even length, separable 2D-Filters have the following properties :

1. The lowpass 2D filter (h_{ll}) is invariant to all the 8 rotations and flips that are commonly used in fractal coders.

2. The h_{hh} filter is rotation and flip invariant (up to a sign reversal)

3. The 'cross' filters h_{lh} and h_{hl} may switch (due to rotation and flipping) from one to the other. That is, a rotated h_{lh} turns out to be $\pm h_{hl}$.

To summarize : The coefficients-tree which represents an isometry of a scaled-down domain block can be found from the representing tree of the scaled-down block by:

1. applying the same isometries in every sub-block of coefficients that belong to the domain-tree and come from a specific subband.

2. switching between HL and LH subbands (for diagonal flips and 90^0 rotations only)

3. inverting the sign of the coefficients of a subband if the result of flipping/rotating its coresponding filter (h_{lh}, h_{hl}, h_{hh}) are the negative of one of those filters.

7 An Adaptive Blockless DWT-IFS Coder

7.1 A Blockless IFS

The investigation of the IFS in the Haar-DWT domain gives a better understanding of the frequency characteristics of the code. Although many

practical benefits can be achieved from the subband interpretation, the main benefit comes, probably, from changing the filters used in the QMF pair.

In such case the domain trees come from the decomposition of $v_{\mathcal{L}}^{1/2}$. A scaled-down version of the image. However the scaling function φ in now not a four pixel averaging operator as defined in (9). Instead it is equivalent to filtering the image with 2D LP QMF separable filter followed by a decimator. Such a filter has a much smoother characteristics as a scaling operator. Finding the best domain-tree $\mathbf{S}_{\mathbf{d}_{m_i}}$ and the best scaling factor a_i, that describe each range-tree, is equivalent to finding similar regions in an image and in its scaled-down version. However those regions overlap, and their local energy fade smoothly towards their borders. Thus a code that is not in blocks have been obtained. To summarize, if the algorithm for finding an IFS and its fixed point, as in Sec.(6) is applied, using a QMF pair other than Haar, one effectively obtains a blockless reconstructed image.

The chosen QMF should have the following properties:

1. Since the IFS matches coefficients from different frequency bands (in different locations), it is very important that the filters should have zero (linear) phase. Such filters have an additional advantage in coding finite length signals (images) since there is a simple way for symmetrically expanding them to prevent discontinuities along the boundries of the image (instead of cyclic expansions) [3].

2. The filters should be relatively short due to the finite size of the image, and because it is desireable to prevent a local error (which usually comes from an edge) from being diffused to smooth areas and create 'ringings' or 'ripples'.

3. Since the minimization is done in the transform domain, it is of advantage to use an *orthonormal* DWT. If a non-orthonormal transform is used, the distance between the the iamge v and $W(v)$ is not equal to the distance between their DWT transforms.

4. It is sufficient, however, that the QMF will provide near-perfect reconstruction (almoust orthogonal), as long as the error created by the subband splitting is small relative to the expected error (of the IFS prediction).

7.1.1 Preliminary Results

In order to evaluate the quality of the IFS that can be obtained in the DWT domain (DWT-IFS), its ability to compress images has been examined. Several separable QMF's were examined on a few images, and all of the filters performed better than the Haar-QMF (which is, as explained earlier, equivalent to the reference algorithm). Out of the examined filters, the best results, in terms of PSNR, as well as subjectively, were obtained with Adelson's et. al. 9 taps QMFs [1] Changing the QMF filter improves the PSNR by about 1dB, and reduces the blockiness of the reconstructed image

It is known that in order to achieve high compression results one has to adapt the amount of bits that represent a block (tree) according to each block activity. One of the common ways to do so is by Quadtree partitioning of the (large) blocks with high prediction error [7]. An equivalent procedure can be applied to range/domain trees in the DWT domain (This is currently under investigation). However, in the following subsections we would like to describe other ways to adapt the complexity of a transformation w_i to the activity of its corresponding regions.

7.2 Zerotrees and DWT-IFS

The DWT-IFS coder that has been described in the last section can be interpreted as a Wavelet-coder in which the coefficients of the base-band are quantized while and higher bands coeficients are predicted from the base band by the DWT-IFS. In recent years several high performance wavelet image coders based on *Zerotrees* were reported [18]. A Zerotree is a subtree of wavelet coefficients that is encoded with a special flag that declares that a whole subtree has negligible energy.

Zerotrees can improve the DWT-IFS coder as well. If a range-tree of AC coefficients has negligible energy, then the bits that are used for decoding its coresponding scaling factor and domain-tree index can be saved. Simulation results show that for standard images (such as 'Lena' 512x512) more than 30% of the range-trees can be zeroed with negligible degradation in the PSNR (about 0.1 dB). We would also like to mention that zerotrees of variouse *depths* are associated with the DWT-domain equivalent of a Quad-tree based IFS, in which range blocks may have different sizes, corresponding to trees with different depths.

Note that using Zerotrees in the DWT-IFS coder, with Haar filters, is equivalent to representing a range block by its mean. A procedure that has been previously used for low variance blocks (denoted by Jacquin as *shaded blocks* [9]), However, such a procedure does not work well because of the discontinuities between the blocks. Using filters other than Haar, allows zeroing low energy trees without noticeable visual artifacts.

7.3 Matching pursuit

A simple way to enrich a transformation is to represent a DC-less range-tree (block) by a combination of several domain-trees (blocks). I.e.,

$$\mathbf{r}_i - \overline{\mathbf{r}}_i = a_i^{(1)}\varphi(\mathbf{d}_{m_i}{}^{(1)} - \overline{\mathbf{d}}_{m_i}^{(1)}) + a_i^{(2)}\varphi(\mathbf{d}_{m_i}{}^{(2)} - \overline{\mathbf{d}}_{m_i}^{(2)}) + \dots \qquad (47)$$

Finding the optimal scaling factors, $a_i^{(p)}$, and domain blocks $\mathbf{d}_{m_i}{}^{(p)}$, through exhaustive search is not practical, therefore it is suggested to apply a *matching-pursuit* approach, which is similar to the one in [12], to find a sub-optimal approximation of every single range block (tree) as follows :

VIII : Range-block approximation using matching pursuit

1. **Initialize** $p = 1$ **and the estimation error :** $\mathbf{e}_i^{(1)} = \mathbf{r}_i - \overline{\mathbf{r}}_i$.
2. **If** $\|\mathbf{e}_i^p\|_2^2$ **is smaller than a given threshold or** $p = P+1$ **(where** P **is the maximum number of domain blocks to be combined) goto step 7.**
3. **Find the best** *single* **domain index** $m_i^{(p)}$ **and scaling factor** $a_i^{(p)}$ **that minimize :**

$$\|\mathbf{e}_i^{(p)} - a_i^{(p)}\varphi(\mathbf{d}_{m_i}{}^{(p)} - \overline{\mathbf{d}}_{m_i}^{(p)})\|_2 \qquad (48)$$

4. **If** $p > 1$ **re-optimize the scaling factors by minimizing the norm of**

$$(\mathbf{r}_i - \overline{\mathbf{r}}_i) - a_i^{(1)}\varphi(\mathbf{d}_{m_i}{}^{(1)} - \overline{\mathbf{d}}_{m_i}^{(1)}) - a_i^{(2)}\varphi(\mathbf{d}_{m_i}{}^{(2)} - \overline{\mathbf{d}}_{m_i}^{(2)}) - \dots - a_i^{(p)}\varphi(\mathbf{d}_{m_i}{}^{(p)} - \overline{\mathbf{d}}_{m_i}^{(p)}) \qquad (49)$$

given **the blocks indices (requires a** pxp **matrix inversion)**

5. Calculate the new estimation error

$$\mathbf{e}_i^{(p+1)} = \mathbf{e}_i^{(1)} - \sum_{k=1}^{p} a_i^{(k)} \varphi(\mathbf{d}_{m_i}{}^{(k)} - \overline{\mathbf{d}}_{m_i}^{(k)}) \qquad (50)$$

6. increase p by 1 and go back to 2

7. The parameters of w_i are the number of domain-blocks p, and the p pairs of scaling factors and domain indices.

7.4 Directional trees and directional mappings

An alternative way to improve the representation of a range-tree, when it is not well estimated by a single domain-tree, is to subdivide the range-tree into three groups of coefficients based on the orientation of the subband from which they come from. (LH, HL and HH). In this manner one gets three range subtrees of size $(2^L \mathbf{x} 2^L - 1)/3$ from a single range-tree of $2^L \mathbf{x} 2^L - 1$ coefficients. Each of these directional range subtrees is approximated separately from directional domain subtrees (extracted in the same manner). As a result, the IFS that is used to reconstruct the region (the block in the Haar-DWT case) consists of three domain indices and three scaling factors, one for each directional sub-tree. A similar idea for coefficients subblocks has been recently proposed by Rinaldo and Calvango [15]. However, they use a different set of parameters for each resolution level (pyramid level).

7.5 Proposed coding algorithm

Based on the above suggestions the following Adaptive DWT-IFS coding scheme is proposed :

IX : Encoding

1. Compute the DWT of the image $\mathbf{v}(m,n)$ with $L+1$ splits of the signal.

2. Quantize and store the low bands $\mathbf{v}_{\mathcal{LL}}^{1/2B}$, $\mathbf{v}_{\mathcal{LH}}^{1/2B}$, $\mathbf{v}_{\mathcal{HL}}^{1/2B}$, $\mathbf{v}_{\mathcal{HH}}^{1/2B}$.

3. Extract domain pool of trees $\mathbf{S}_{\mathbf{d}_{m_i}}$ (with AC coefficients only).

4. For each range-tree $\mathbf{S}_{\mathbf{r}_i}$:
   ◇ If $\|\mathbf{S}_{\mathbf{r}_i}\|_2^2 < Thershold \Longrightarrow$ encode as a Zerotree.
   ◇ else approximate the range-tree with a single domain-tree (Sec.(7.2)).
   ◇ Is the error is too large, approximate the range-tree with two domain-trees using the matching pursuit formulation (Sec.(7.3)).
   ◇ If the error is still too large, split the range-tree into three *directional* subtrees and approximate each of them using three directional domain subtrees. (Sec.(7.4)).

X : Decoding

1. Copy the quantized bands from the code.

2. Extrapolate recursively from the low bands (included in the code) to the higher bands. Compute each of the groups of coefficients according to the information about the type of transformation, using the scaling factors and the indices.

3. Compute the Inverse DWT.

7.6 coding results

The Adelson et al. 9 taps QMF wavelet coefficients of the lower bands were quantized with 7 bits/coeff for the LP band and 6 bit/coeff for the three (lower) HP bands. The scaling factors were also quantized with 6 bits each. All the quantizers used were uniform. The quantization was followed by adaptive arithmetic encoding of each of the quantized variables. In Fig. 8 we compare the results obtained in coding the image 'Lena' (512x512) using the reference IFS algorithm with a fixed block size $B = 8$ - (bottom line) to DWT-IFS coding using Adelson et. al. 9 taps QMF pair [1] with $L = 3$ (middle) and Adaptive coding scheme with the same 9 Taps filter (upper).

The lines of the fixed IFS/DWT-IFS pass through only two points. The lower point come from IFS without use of rotations and the upper from IFS using Rotations. The use of isometries in the Adaptive DWT-IFS was not found to improve the results, and therefore isometries were not applied.

From Fig. 8 it is evident that the improvement from using the Adaptive-DWT-IFS over the reference algorithm is more than 2dB. In the range 0.2-0.35 bit/pixel, the proposed coder also outperforms the conventional coder using the Quadtree approach described in [7](Chapt. 3).

However the decrease of the slope of the graph teaches us that in order to achieve higher PSNR results one should combine the Quadtree aproach with the addaptive DWT-IFS. I.e., use range-trees of different depth as well. Such a scheme is currentlly under investigation.

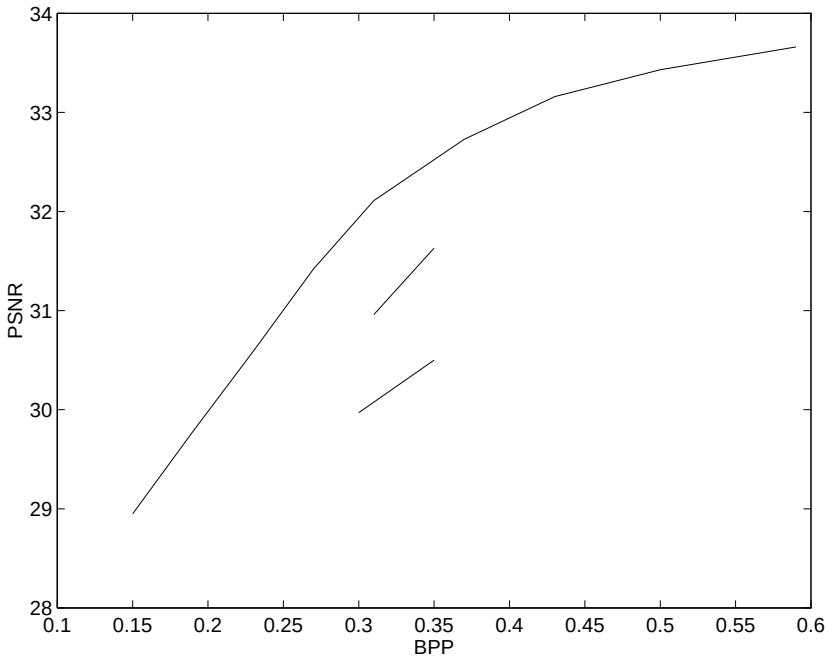


Figure 8: Results of Coding the image 'Lena' (512x512) with Adaptive DWT-IFS, $L = 3$. see text for details

.

8 Super Resolution

In Sec.(4.2) we have shown that the pyramidal interpretation of an IFS code may also be used to increase the resolution of signals. We showed that instead of using an IFS code to decode signals in arbitrary resolutions, one can 'zoom-in' from a fixed point at a given resolution and obtain a representation at twice the resolution. In this section we would like to generalize the results of Sec.(4.2) to the DWT-IFS.

Let us assume that one uses a given DWT-IFS set of parameters (base-bands coefficients, scaling factors and domain indices) but instead of extrapolating from the base band to the upper bands L times (as one should do according to the DWT-IFS decoding procedure and is shown in Fig. 6) one extrapolates $L+1$ times. Such a procedure is equivalent to zooming-in $L+1$ time in the low pass pyramid as described in Sec.(4.2), when one uses the Haar QMF. However, using other QMFs reduces the visual artifacts typical to super-resolution techniques because of the reasons explained before, concerning IFS coding/decoding.

The result of this extra band extrapolation is a fixed point with the same subbands content as in the *fixed point* at the original resolution, but with an additional subband that does not exist in the original resolution. Note that if one wishes to use the IFS code for "super-resolution" *only*, then one can extrapolate the higher (nonexisiting) band(s) from the DWT of the *original* signal. As a result the lower subbands of the signal in super resolution will be the same as the lower subbands in the *original* signal.

8.0.1 DWT-IFS Embedded Function

In Sec.(4.2) we have also pointed out that the limit of a zooming-in process applied to a fixed point is a continuous-time fixed point in "infinite" resolution denoted *IFS Embedded Function*

Using the Wavelet framework (and under some constrains on the scaling factors), the DWT-IFS defines a uniqe continuous time fixed point in $\mathcal{L}^2$. This function can be constructed in the following way :

1. extrapolate a given DWT-IFS fixed point to higher bands again and again, "infinite" number of times.

2. Calculate the continuous time function $f(t)$ - assuming that the DWT coefficients are actually its "Wavelet Series" coefficients.

Due to the relation between DWT and wavelet series, it is observed that *by definition* the fixed points at various resolutions $\mathbf{f}^{1/2^l}$ $(-\infty < l \leq L)$ (Here L is the number of splits in the DWT of the analyzed signal) are discrete approximations of this function.

It should be noted that this is a generalization of the IFS Embedded function, and that the functions are the same in the case of Haar-DWT-IFS.

9 Summary and Conclusions

In this paper a link between farctal image coding and multiresolution analysis has been established. We have shown that the hierarchical interpretation of the IFS code can be used to achieve a very fast decoding scheme.

The investigation of the hierarchical structure of the fixed point naturally leads to a wavelet framework. Such a framework enables us to create a blockless IFS code.

We belive that the combination of fractal image coding and wavelet coding, where self-similarities among wavelet-subtrees are used for representing similar regions at different resolutions in the image, will provide a common useful ground for future activity in this area

References

[1] E. D. Adelson, E. Simoncelli, and R. Hingorani. Orthogonal pyramid transforms for image coding. volume 845, pages 50–58. SPIE-Visual Communications and Image Processing II, 1987.

[2] Z. Baharav, D. Malah, and E. D. Karnin. Hierarchical interpretation of fractal image coding and its application to fast decoding. Intl. Conf. on Digital Signal Processing. pages 190–195. Cyprus, July 1993.

[3] R. H. Bamberger, S. L. Eddins, and V. Nuri. Generalized symmetric extensions for size limited multirate filter banks. *IEEE Transactions on Image Processing*, 3(1):82–87, January 1994.

[4] M. F. Barnsley. *Fractals Everywhere.* Wiley and Sons, 1988.

[5] G. Davis. Self quantized wavelet subtrees: A wavelet based theory for fractal image compression. DCC-Data Compression Conference, pages 232–241, Snowbird-Utah, May 1995.

[6] K. Falconer. *Fractal Geometry, mathematical foundations and applications.* Wiley and Sons, 1990.

[7] Y. Fisher. *Fractal Compression: Theory and applications to Digital Images.* Springer Verlag, New-York, 1995.

[8] G .C. Freeland and T. S. Durrani. Ifs fractlas and the wavelet transform. pages 2345–2348. ICASSP, 1990.

[9] A. E. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, 1(1):18–30, January 1992.

[10] H. Krupnik, D. Malah, and E. Karnin. Fractal representation of images via the discrete wavelet transform. IEEE 18th conv. of EE in Israel, pp. 2.2.2 : 1–5, May 1995.

[11] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on PAMI*, 11(7):674–693, July 1989.

[12] S. G. Mallat and Z. Zhang. Matching pursuits with time frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, December 1993.

[13] G. E. Øien and S. Lepsøy. Fractal image coding with fast decoder convergence. *Signal Processing*, 40:105–117, 1994.

[14] G. E. Øien, Zachi Baharav, et al. A new improved collage theorem with applications to multiresolution fractal image coding. pages V:565–568, ICASSP, 1994.

[15] R. Rinaldo and G. Calvango. Image coding by block prediction of multiresolution subimages. *IEEE Transactions on Image Processing*, 4(7):909–920, July 1995.

[16] O. Rioul. A discrete time multiresolution theory. *IEEE Transactions on Signal Processing*, 41(8):2591–2606, August 1993.

[17] H. L. Royden. *Real Analysis*. Macmillan Pub., 1988.

[18] J. M. Shapiro. Embedded image codec using zerotrees of wavelets coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, December 1993.

[19] Z. Baharav, D. Malah, and E. Karnin. Hierarchical interpretation of fractal image coding and its applications. In Y. Fisher (ed) *Fractal Compression: Theory and applications to Digital Images*, chapter 5, pp 91 – 117. Springer Verlag, New-York, 1995.