A Hitchhiker's Guide to "Fractal-Based" Function Approximation and Image Compression

Edward R. Vrscay
Department of Applied Mathematics
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
e-mail: `ervrscay@links.uwaterloo.ca`
Waterloo Fractal Compression Project Website: http://links.uwaterloo.ca

This is a slightly expanded version of two articles which appeared in the February and August 1995 issues of the UW Faculty of Mathematics Alumni newpaper, *Math Ties.*

Part I

In 1981, a paper by a mathematician named John Hutchinson showed how systems of contraction maps on suitable spaces (e.g. mappings on [0,1] and $[0,1]^2$ which map points closer to each other) could be used to generate self-similar "fractal" sets. In 1984, Michael Barnsley and coworkers at Georgia Tech independently came upon this idea and developed it from a more probabilistic viewpoint, calling such systems "Iterated Function Systems" (IFS). (I should mention that I was fortunate enough to be visiting Georgia Tech as an NSERC Postdoctoral Fellow during the "formative years of IFS", 1984-86, which accounts for my own interest in the subject.) In a series of papers, the Georgia Tech workers explored some methods of using IFS to approximate sets, measures and, eventually, images. Indeed, the use of IFS as a possible method of *image compression* has received much publicity. I shall try to outline below why people are interested in this application as well as some promising results - both theoretical as well as computational - obtained recently at Waterloo. Only the major points will be sketched, but I hope to show that with a little effort, you, too, can generate fractal sets and perhaps even compress images with the comfort and privacy of your own PC/workstation. Part I is devoted to the basic ideas of IFS. It is necessary to know these basics in order to understand how IFS-type methods for function approximation and image compression could be developed. These latter developments, the subject of Part II, were the result of an ongoing collaboration with Prof. B. Forte, formerly a member of the Applied Mathematics Department, and now a Professor of Mathematics (Analysis) at the University of Verona in Italy. (Many Math grads may remember Bruno for his service over many years as coach of the Waterloo Putnam teams, as well as being a former Chair of the Applied Math Department. He is currently an Adjunct Professor in our Department.) A Natural Sciences and Engineering Research Council of Canada (NSERC) Collaborative Project Grant for 1994-97 has made possible an even richer collaboration with Prof. C. Tricot, Ecole Polytechnique, U. de Montréal and Dr. J. Lévy-Véhel of INRIA, France.

One aspect of *dynamical systems theory* involves the iteration of mappings on a suitable space X. Given a mapping $f : X \to X$, pick a "seed" point $x_0 \in X$ and construct the sequence of points $x_{n+1} = f(x_n), n = 0, 1, ...$ Then examine the behaviour of the points x_n as $n \to \infty$. Do they approach a limiting point, a limiting set, a cycle of points, etc.? Moreover, does the behaviour depend on the seed point x_0? When f is nonlinear, the sequence $\{x_n\}$ can exhibit *chaotic* behaviour, a phenomenon which has received much attention and publicity over the past ten years. (**Exercise:** Let $f(x) = 4x(1-x)$ and investigate the behaviour of the x_n for various x_0.)

In contrast, IFS generally employ contractive maps over a complete metric space (X, d). (X is typically $[0, 1]$ or $[0, 1]^2$, with d being the Euclidean metric.) A map $f : X \to X$ is contractive if there exists a constant $c \in [0, 1)$ such that $d(f(x), f(y)) \leq cd(x, y)$ for all $x, y \in X$. The constant c is known as the *contraction factor* of f. Banach's celebrated *Contraction Mapping Theorem* guarantees the existence of a unique "fixed point" $\overline{x} \in X$ such that $f(\overline{x}) = \overline{x}$. Moreover, this fixed point is "attractive": For any $x_0 \in X$, and the iteration procedure defined above, $d(x_n, \overline{x}) \to 0$ as $n \to \infty$. From a dynamical systems viewpoint, the dynamics associated with iterated contraction mappings is rather trivial since all points x in our space X are attracted to $\bar{x}$. As well, the contraction maps used in practical applications are typically affine (e.g. shears and rotations followed by translations). The iteration dynamics associated with affine maps - including those transformations we saw in high school - is not as interesting as that for nonlinear maps. However, when we consider a system of contractive and possibly affine maps acting in a "parallel" fashion - the essence of IFS - the resulting behaviour is quite remarkable, as we show below.

Suppose we let $X = [0, 1]$ with d the usual Euclidean metric. Consider the following two contraction maps on (X, d):

1. $f_1(x) = \frac{1}{3}x$. Its fixed point is $\overline{x} = 0$. It is easy to see that if $x_0 \in [0, 1]$, then $x_1 = x_0/3$, $x_2 = x_1/3 = x_0/9$. In general, $x_n = x_0/3^n$. Clearly, $x_n \to 0$ as $n \to \infty$. Rather trivial dynamics, n'est-ce pas?

2. $f_2(x) = \frac{1}{3}x + \frac{2}{3}$. Then $\overline{x} = 1$. Trivial dynamics as well, with $x_n \to 1$ as $n \to \infty$ for any $x_0 \in [0, 1]$.

It is instructive to examine the action of each of the above maps on the interval [0,1]. To do this, let us define the following associated *set-valued* mappings: For $i \in \{1, 2\}$ and any subset $S \subseteq [0, 1]$, we denote $\hat{f}_i(S) = \{f_i(x) | x \in S\}$. For example, $\hat{f}_1([0, 1]) = [0, \frac{1}{3}]$. In other words, $\hat{f}_1$ "shrinks" the interval [0,1] to $[0, \frac{1}{3}]$. Likewise, $\hat{f}_2([0, 1]) = [\frac{2}{3}, 1]$. Furthermore $\hat{f}_1^{\circ 2}([0, 1]) = \hat{f}_1(\hat{f}_1([0, 1]) = [0, \frac{1}{9}]$, etc..

Now define the following set-valued IFS mapping $\hat{\mathbf{f}}$ whose action is defined as follows: For any subset $S \subseteq [0, 1]$, let

$$\hat{\mathbf{f}}(S) \equiv f_1(S) \cup f_2(S). \tag{1}$$

In other words, the mapping $\hat{\mathbf{f}}$ represents a kind of parallel action of the maps $\hat{f}_1$ and $\hat{f}_2$. If we denote $I_0 = [0, 1]$, then

$$I_1 = \hat{f}(I_0) = [0, \frac{1}{3}] \cup [\frac{2}{3}, 1],$$

$$I_2 = \hat{f}(I_1) = [0, \frac{1}{9}] \cup [\frac{2}{9}, \frac{1}{3}] \cup [\frac{2}{3}, \frac{7}{9}] \cup [\frac{8}{9}, 1].$$

The repeated application of our IFS mapping $\hat{\mathbf{f}}$, illustrated in Figure 1, mimics the "middle-thirds dissection procedure" on [0,1] which is used to construct the classical ternary Cantor set C. One might then conjecture that the I_n "tend to" C as $n \to \infty$. In fact, Hutchinson showed that the IFS mapping $\hat{\mathbf{f}}$ is a contraction mapping on an appropriate complete metric space. (The space is $\mathcal{H}(X)$, the set of all nonempty compact subsets of X. The metric h is called the *Hausdorff metric.* The sets I_n are elements of $\mathcal{H}(X)$.) Banach's theorem is then applicable: The fixed point or attractor of the IFS mapping $\hat{\mathbf{f}}$ is the Cantor set C:

$$C = \hat{\mathbf{f}}(C) = \hat{f}_1(C) \cup \hat{f}_2(C). \tag{2}$$

Eq. (2) expresses the fact that C is a union of two smaller copies of itself, as shown in Figure 1. This is an example of "self-tiling" or "self-similarity", a property which is often exhibited by "fractal" sets. (Note that $C \in \mathcal{H}(X)$.)

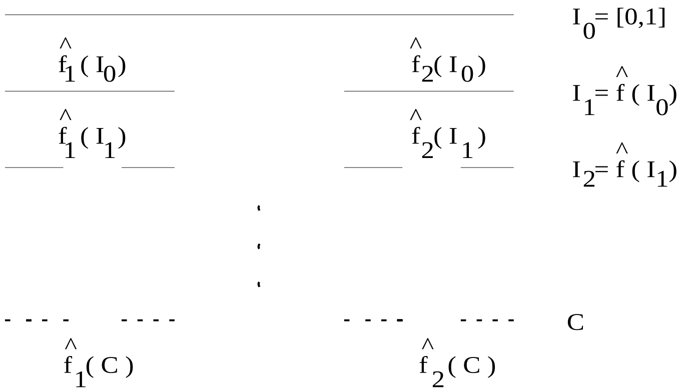

Figure 1: Repeated action of the 2-map IFS on [0,1] which performs "middle-thirds dissection". C is the ternary Cantor set on [0,1].

In general, for a system of N contraction maps $\mathbf{f} = \{f_1, f_2, ..., f_N\}$ (with contraction factors $c_i < 1$) on a complete metric space (X, d), the associated IFS operator will have the form

$$\hat{\mathbf{f}}(S) = \bigcup_{i=1}^{N} \hat{f}_i(S). \tag{3}$$

The IFS mapping $\hat{\mathbf{f}}$ is contractive (with contraction factor $c = \max_{1 \le i \le N}\{c_i\}$) and possesses a fixed point "attractor" A which satisfies the property $A = \hat{\mathbf{f}}(A)$ or

$$A = \bigcup_{i=1}^{N} \hat{f}_i(A). \tag{4}$$

(**Exercise:** Now add a third map to the 2-map IFS given above, namely $f_3(x) = \frac{1}{3}x + \frac{1}{3}$. What is the attractor of this 3-map IFS?)

In two dimensions, e.g. $X = [0,1]^2$, consider the IFS formed by the following three affine maps

$$f_1(x,y) = (\frac{1}{2}x, \frac{1}{2}y), \quad f_2(x,y) = (\frac{1}{2}x + \frac{1}{2}, \frac{1}{2}y), \quad f_3(x,y) = (\frac{1}{2}x + \frac{1}{4}, \frac{1}{2}y + \frac{1}{4}\sqrt{3}).$$

(All maps have contraction factor $\frac{1}{2}$ and their fixed points $\overline{x}_i$ lie on the vertices of an equilateral triangle - an exercise for the reader.) The resulting attractor, the so-called "Sierpinski gasket" is shown in Figure 2(a). The subset of the attractor enclosed in the lower left third triangle is $f_1(A)$, etc.. There is a lot of room for variation in two dimensions and the results can be fascinating. For example, Figure 2(b), Barnsley's famous "spleenwort fern", is the attractor of a four-map affine IFS in the plane. (**Exercise:** See if you can determine the four maps or at least a set of maps which produce a fernlike object as an attractor.)

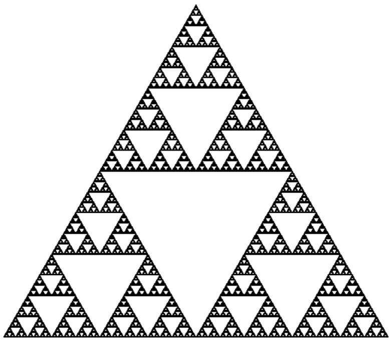

Figure 2(a): Sierpinski gasket.

Figure 2(b): Spleenwort fern.

Along with the creation of these fern-type attractors in 1984 came the idea of using IFS to approximate other shapes and figures occuring in nature and, ultimately, images in general. The IFS was seen to be a possible method of *data compression.* A high-resolution picture of a shaded fern normally requires on the order of one megabyte of computer memory for storage. Current compression methods might be able to cut this number by a factor of ten or so. However, as an attractor of a four map IFS with probabilities, this fern may be described totally in terms of only 28 IFS parameters! This is a staggering amount of data compression. Not only are the storage requirements reduced but you can also send this small amount of data quickly over communications lines to others who could then "decompress" it and reconstruct the fern by simply iterating the IFS mapping $\hat{\mathbf{f}}$. In fact, this led to some rather sensationalistic early claims of compression factors of the order of tens of thousands that could be achieved using "fractal-based" image representation. (The compression factors achieved by IFS-methods, such as those shown in Part II, are currently on the order of 10 or 20 to 1.)

However, not all objects in nature - in fact, very few - exhibit the special self-similarity of the spleenwort fern. Nevertheless, as a starting point there remains the interesting general problem to determine to determine how well sets and images can be approximated by the attractors of IFS. We pose the so-called *inverse problem* for geometric approximation with IFS as follows:

> Given a "target" set S, can one find an IFS $\mathbf{f}$ whose attractor A approximates to some desired degree of accuracy in an appropriate metric "D" (for example, the Hausdorff metric h)?

From an important corollary of Banach's Contraction Mapping Theorem, often referred to as the "Collage Theorem", the inverse problem may be restated as follows:

> Given a target S and an $\epsilon > 0$, can one find an IFS $\mathbf{f}$ such that $D(S, \hat{\mathbf{f}}(S)) < \epsilon$?

The term $\hat{\mathbf{f}}(S)$, which from Eq. (3) is the union of shrunken copies of S, is known as the "collage" of S. The term $D(S, \hat{\mathbf{f}}(S))$ is referred to as the "collage distance". The smaller the collage distance, i.e. the closer $\hat{\mathbf{f}}(S)$ is to S, the better we can "tile" S with contracted copies of itself. (In the case that S is the attractor of an N-map IFS $\mathbf{f}$, then, by Eq. (4), the collage distance is zero.) The basic idea is illustrated in Figure 3(a) . A leaf is viewed as an approximate union of shrunken copies of itself. Each smaller copy is obtained by an appropriate contractive IFS map f_i. If we restrict ourselves to affine IFS maps in the plane, i.e. $f_i(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$, then the coefficients of the matrix $\mathbf{A}$ and the column vector $\mathbf{b}$ can be determined by just determining where any three points of the original leaf are mapped to in each shrunken copy. The Collage Theorem then states that the attractor A of the resulting IFS lies close to the target leaf S. The attractor A of the IFS constructed from Figure 3(a) is shown in Figure 3(b). (The interested reader might wish to return to the spleenwort fern and determine the four IFS maps required to obtain a successful collage. Hint: One of the maps is responsible for the stem.)

In general, the determination of optimal IFS maps by looking for approximate geometric self-similarities in a set is a very difficult problem with no simple solutions, *especially if one wishes to automate the process.* Fortunately, we can proceed by another route by realizing that there is much more to a picture than just geometric shapes. There is also shading. For example, a real fern has veins which may be darker than the outer extremeties of the fronds.

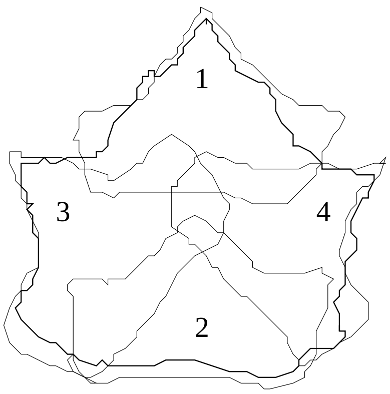

Figure 3(a): Approximating a leaf as a "collage", i.e. a union of shrunken copies of itself.

Figure 3(b): The attractor A of the four-map IFS obtained from a "collage" of the type in Figure 3(a).

Thus it is more natural to think of a picture as defining a function: At each point or pixel (x, y) in a photograph or a computer display (represented, for convenience, by the region $X = [0,1]^2$) there is an associated grey level $u(x, y)$, which may assume a finite nonnegative value. (In practical applications, i.e. digitized images, each pixel can assume one of only a finite number of discrete values.) In the next (and last!) instalment of this article, I shall show how an IFS-type approach can be set up for functions. It will involve a "collaging" of the graphs of functions, leading to an effective method of approximating and compressing images.

Before closing, many of you may be thinking that the IFS iteration procedure outlined above represents a rather tedious way of generating pictures of attractors, since you must keep track of many points during each iteration. It is possible to generate IFS attractors by following only one point in an iteration procedure called the "chaos game". Briefly, suppose you have an N-map IFS $\mathbf{f} = \{f_1, f_2, ..., f_N\}$ on $X = [0,1]$ or $[0,1]^2$. Now take any point $x_0 \in X$. Pick a number i_1 from the set $\{1, 2, ..., N\}$ randomly and let $x_1 = f_{i_1}(x_0)$. Now continue this procedure: For $n = 1, 2, ...$ let $x_{n+1} = f_{i_n}(x_n)$, where each index i_n is chosen randomly from $\{1, 2, ..., N\}$. Plot the points x_n for $n \geq 50$. These points will appear to perform a random walk on the attractor A of $\mathbf{f}$. As more and more points are plotted, the resulting picture will provide a better representation of A. (There is an intimate relation between this *Markov process* and what is called an *invariant measure* which is supported on the attractor A.)

References: A very readable account of these "early IFS" methods is: M.F. Barnsley and A. Sloan, "A better way to compress images," *BYTE* Magazine, January issue, pp. 215-223 (1988). For a more detailed treatment, see M.F. Barnsley, "Fractals Everywhere", Academic Press (1988).

Part II

In the previous exciting episode, I introduced the idea of Iterated Function Systems (IFS) and what they can do geometrically. (Now go and get your previous edition of *Math Ties* because I'm not going to review everything here!) Instead of looking at the action of a *single* mapping $f : X \to X$, where X is some suitable (complete metric) space, e.g. [0,1], we consider a set of maps $\mathbf{f} = \{f_1, f_2, ..., f_N\}$ These maps are then considered to act in union, like a "parallel machine". Mathematically, associated with the IFS $\mathbf{f}$ is an operator $\hat{\mathbf{f}}$ whose action on a subset $S \subset X$ is given by

$$\hat{\mathbf{f}}(S) = \bigcup_{i=1}^{N} \hat{f}_i(S). \tag{5}$$

If each of the maps $f_i : X \to X$ is contractive, then the IFS operator $\hat{\mathbf{f}}$ is contractive (in *Hausdorff metric*, but let's not worry about that here). From Banach, there must exist a unique *attractor* set A which satisfies the fixed point equation

$$A = \bigcup_{i=1}^{N} \hat{f}_i(A). \tag{6}$$

(Recall the Cantor set C on [0,1] as the attractor of a 2-map IFS.) The next, and most challenging, step is the *inverse problem*:

> Given a "target" set S (e.g. a black-and-white shape of a leaf) can one find a set of contraction maps $f_i, 1 \leq i \leq N$, hence an N-map IFS, whose attractor A is close to S in some metric D?

Figures 3(a) and 3(b) in Part I showed the idea of finding an IFS by tiling, or making a collage of, a leaf with smaller copies of itself. In general, however, such a determination of optimal IFS maps is quite difficult, especially if one would like to accomplish this on a computer without direct human intervention. Part I was concluded with the comment that we can regard a picture as being more than merely geometric shapes. There is also shading. As such, it is more natural to think of a picture as defining a function: At each point or pixel (x, y) in a photograph, there is an associated "grey level" $u(x, y)$ which assumes a finite and nonnegative value. (Here, $(x, y) \in X = [0, 1]^2$, for convenience.) For example, consider

Figure 4, a standard test case in image processing studies named "Lena". The image is a 512×512 pixel array. Each pixel assumes one of 256 shades of grey (0 = white, 255 = black). From the point of view of continuous real variables (x, y), the image is represented as a piecewise constant function $u(x, y)$. If the grey level value of each pixel is interpreted as a value in the z direction, then the graph of the *image function* $z = u(x, y)$ is a surface in $\mathbf{R}^3$.

Figure 4: The target image "Lena", a 512×512 pixel array, 8 bits (256 grey-level values) per pixel.

Our goal is to set up an IFS-type approach to work with functions $u : X \to \mathbf{R}^+$ instead of sets. Before writing any mathematics, let us illustrate schematically what can be done. For ease of presentation, we consider for the moment only one-dimensional images, i.e. positive real-valued functions $u(x)$ where $x \in [0, 1]$. An example is sketched in Figure 5(a). Suppose our IFS is composed of only two contractive maps f_1, f_2. Each of these functions f_i will map the "base space" $X = [0, 1]$ to a subinterval $\hat{f}_i(X)$ contained in X. Let's choose

$$f_1(x) = 0.6x, \qquad f_2(x) = 0.6x + 0.4. \tag{7}$$

For reasons which will become clear below, it is important that $\hat{f}_1(X)$ and $\hat{f}_2(X)$ are *not* disjoint - they will have to overlap with each other, even if the overlap occurs only at one point.

The first step in our IFS procedure is to make two copies of the graph of $u(x)$ which are distorted to fit on the subsets $\hat{f}_1(X) = [0, 0.6]$ and $\hat{f}_2(X) = [0.4, 1]$ by "shrinking" and translating the graph in the x-direction. This is illustrated in Figure 5(b). Mathematically, the two "component" curves $a_1(x)$ and $a_2(x)$ in Figure 5(b) are given by

$$a_1(x) = u(f_1^{-1}(x)) \;\; x \in \hat{f}_1(X), \qquad a_2(x) = u(f_2^{-1}(x)) \;\; x \in \hat{f}_2(X), \tag{8}$$

(**Exercise:** Check this and understand it since it is important for the ideas that follow!) We're not finished, however, since some additional flexibility in modifying these curves would be desirable. Suppose that are allowed to modify the y (or grey level) values of each component function $a_i(x)$. For example, let us

1. multiply all values $a_1(x)$ by 0.5 and add 0.5,

2. multiply all values $a_2(x)$ by 0.75.

The modified component functions, denoted as $b_1(x)$ and $b_2(x)$, respectively, are shown in Figure 5(c). What we have just done can be written as

$$\begin{aligned} b_1(x) &= \phi_1(a_1(x)) = \phi_1(u(f_1^{-1}(x))) \quad x \in \hat{f}_1(X), \\ b_2(x) &= \phi_2(a_2(x)) = \phi_2(u(f_2^{-1}(x))) \quad x \in \hat{f}_2(X), \end{aligned} \tag{9}$$

where

$$\phi_1(y) = 0.5y + 0.5, \qquad \phi_2(y) = 0.75y, \quad y \in \mathbf{R}^+. \tag{10}$$

The ϕ_i are known as *grey-level maps*: They map (nonnegative) grey-level values to grey-level values.

We now use the component functions b_i in Figure 5(c) to construct a new function $v(x)$. How do we do this? Well, there is no problem to define $v(x)$ at values of $x \in [0,1]$ which lie in only one of the two subsets $\hat{f}_i(X)$. For example, $x_1 = 0.25$ lies only in $\hat{f}_1(X)$. As such, we define $v(x_1) = b_1(x) = \phi_1(u(f_1^{-1}(x)))$. The same is true for $x_2 = 0.75$, which lies only in $\hat{f}_2(X)$. We define $v(x_2) = b_2(x) = \phi_2(u(f_2^{-1}(x)))$.

Now what about points that lie in *both* $\hat{f}_1(X)$ and $\hat{f}_2(X)$, for example $x_3 = 0.5$? There are two possible components that we may use to define our resulting function $v(x_3)$, namely $b_1(x_3)$ and $b_2(x_3)$. How do we suitably choose or combine these values to produce a resulting function $v(x)$ for x in this region of overlap?

To make a long story short, this is a rather complicated mathematical issue and has been a subject of ongoing research, in particular at Waterloo. There are many possibilities of combining these values, including (1) adding them, (2) taking the maximum or (3) taking some weighted sum, for example, the average. In what follows, we consider the first case, i.e. *we simply add the values.* The resulting function $v(x)$ is sketched in Figure 6(a). The observant reader may now be able to guess why we demanded that the subsets $\hat{f}_1([0,1])$ and $\hat{f}_2([0,1])$ overlap, touching at least at one point. If they didn't, then the union $\hat{f}_1(X) \cup \hat{f}_2(X)$ would have "holes", i.e. points $x \in [0,1]$ at which no component functions $a_i(x)$, hence $b_i(x)$, would be defined. (Remember the Cantor set?) Since want our IFS procedure to map functions on X to functions on X, the resulting function $v(x)$ *must* be defined for all $x \in X$.

The 2-map IFS $\mathbf{f} = \{f_1, f_2\}$, $f_i : X \to X$, along with associated grey-level maps $\Phi = \{\phi_1, \phi_2\}$, $\phi_i : \mathbf{R}^+ \to \mathbf{R}^+$, is referred to as an *Iterated Function System with Grey-Level Maps* (IFSM), $(\mathbf{f}, \Phi)$. What we did above was to associate with this IFSM an operator T which acts on a function u (Figure 5(a)) to produce a new function $v = Tu$ (Figure 6(a)). Mathematically, the action of this operator may be written as follows: For any $x \in X$,

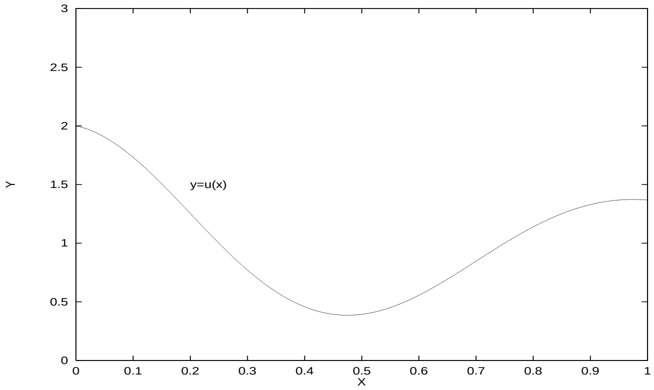

Figure 5(a): A sample "one-dimensional image" $u(x)$ on [0,1].

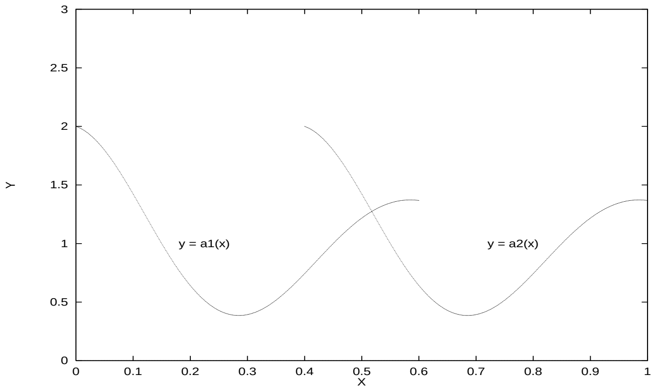

Figure 5(b): The component functions given in Eq. (8).

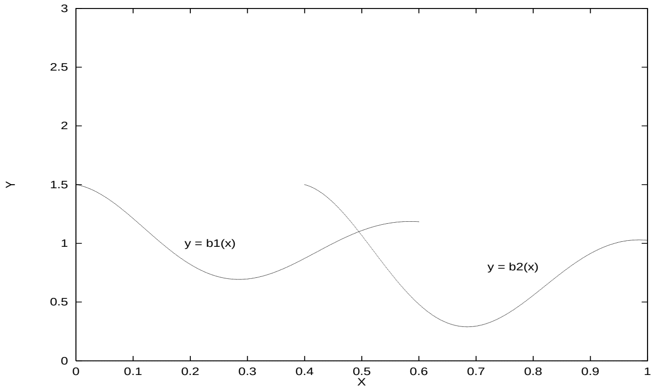

Figure 5(c): The modified component functions given in Eq. (9).

$$v(x) = (Tu)(x) = \sum_{i=1}^{N}{}' \phi_i(u(f_i^{-1}(x))). \tag{11}$$

The prime on the summation signifies that for each $x \in X$ we sum over only those $i \in \{1,2\}$ for which a "preimage" $f_i^{-1}(x)$ exists. (Because of the "no holes" condition, it guaranteed that for each $x \in X$, there exists at least one such i value.) For $x \in [0, 0.4)$, i can be only 1. Likewise, for $x \in (0.6, 1]$, $i = 2$. For $x \in [0.4, 0.6]$, i can assume both values 1 and 2. The extension to a general N-map IFSM is straightforward.

There is nothing preventing us from applying the T operator to the function v, so let $w = Tv = T(Tu)$. Again, we take the graph of v and "shrink" it to form two copies, etc.. The result is shown in Figure 6(b). As T is applied repeatedly, we produce a sequence of functions which converges to a function $\bar{u}$ in an appropriate space, called $\mathcal{L}^1(X)$). (This is the space of functions $f : X \to \mathbf{R}$ for which $\| f \|_1 \equiv \int_X |f(x)|dx < \infty$. In this space, the distance between two functions $u, v \in \mathcal{L}^1(X)$ is given by the norm $\| u - v \|_1 \equiv \int_X |u(x) - v(x)|dx$.) The function $\bar{u}$ is sketched in Figure 6(c). (Because it has so many jumps, it is better viewed as a histogram plot.)

In general, under suitable conditions on the IFS maps f_i and the grey-level maps ϕ_i, the operator T associated with an IFSM $(\mathbf{w}, \Phi)$ is contractive in $\mathcal{L}^1(X)$. Therefore, from the Banach Contraction Mapping Theorem, it possesses a unique "fixed point" function $\bar{u} \in \mathcal{L}^1(X)$. This is precisely the case with the 2-map IFSM given above. Its attractor is sketched in Figure 6(c). Note that from the fixed point property $\bar{u} = T\bar{u}$ and Eq. (11), the attractor $\bar{u}$ of an N-map IFSM satisfies the equation

$$\bar{u}(x) = \sum_{i=1}^{N}{}' \phi_i(\bar{u}(f_i^{-1}(x))), \qquad . \tag{12}$$

In other words, the graph of $\bar{u}$ satisfies a kind of "self-tiling" property: it may be written as a sum of distorted copies of itself.

Before going on, let's consider the three-map IFSM composed of the following IFS maps and associated grey-level maps:

$$\begin{aligned} f_1(x) &= \frac{1}{3}x, & \phi_1(y) = \frac{1}{2}y, \\ f_2(x) &= \frac{1}{3}x + \frac{1}{3}, & \phi_2(y) = \frac{1}{2}, \\ f_3(x) &= \frac{1}{3}x + \frac{2}{3}, & \phi_3(y) = \frac{1}{2}y + \frac{1}{2}, \end{aligned} \tag{13}$$

Notice that $\hat{f}_1(X) = [0, \frac{1}{3}]$ and $\hat{f}_2(X) = [\frac{1}{3}, 1]$ overlap only at one point, $x = \frac{1}{3}$. Likewise, $\hat{f}_2(X)$ and $\hat{f}_3(X)$ overlap only at $x = \frac{2}{3}$. The fixed point attractor function $\bar{u}$ of this IFSM is sketched in Figure 7. It is known as the "Devil's Staircase" function. You can see that the attractor satisfies a self-tiling property: If you shrink the graph in the x-direction

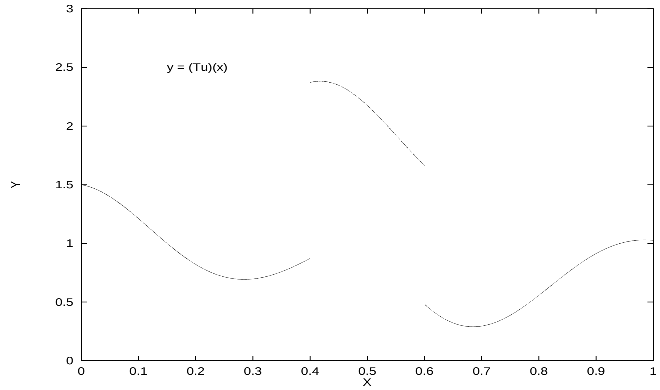

Figure 6(a): The resulting "fractal transform" function $v(x) = (Tu)(x)$ obtained from the component functions of Figure 5(c).

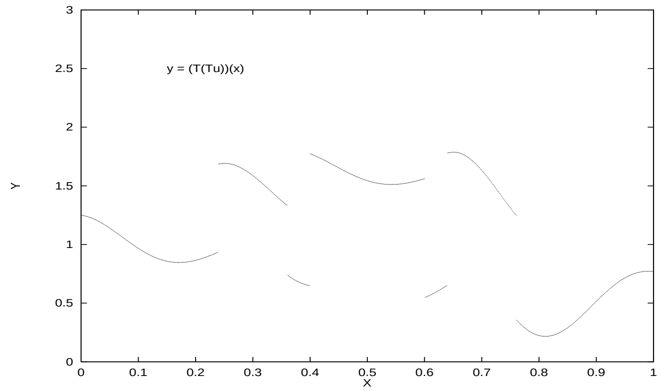

Figure 6(b): The function $w(x) = T(Tu)(x) = (T^{\circ 2}u)(x)$: the result of two applications of the fractal transform operator T.

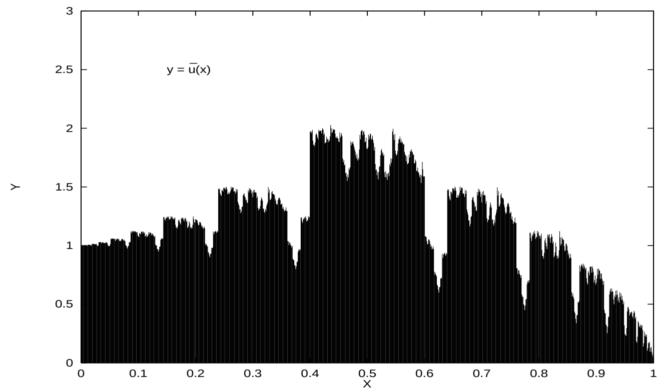

Figure 6(c): The "attractor" function $\bar{u} = T\bar{u}$ of the two-map IFSM given in the text.

onto the interval $[0, \frac{1}{3}]$ and shrink the in y-direction by $\frac{1}{3}$, you obtain one piece of it. The second copy, on $[\frac{1}{3}, \frac{2}{3}]$, is obtained by squashing the graph to produce a constant. The third copy, on $[\frac{2}{3}, 1]$, is just a translation of the first copy by $\frac{2}{3}$ in the x-direction and $\frac{1}{2}$ in the y-direction. (Note: The observant reader can complain that the function graphed in Figure 6 is **not** the fixed point of the IFSM operator T as defined in Eq. (11): The value $v(\frac{1}{3})$ should be $\frac{3}{2}$ and not $\frac{1}{2}$, since $x = \frac{1}{3}$ is a point of overlap. In fact, this will also happen at $x = \frac{2}{3}$ as well as an infinity of points obtained by the action of the f_i maps on $x = \frac{1}{3}$ and $\frac{2}{3}$. What a mess! Well, not quite, since the function in Figure 7 and the true attractor differ on a countable infinity of points. Therefore, the the $\mathcal{L}^1$ distance between them is zero! The two functions belong to the same *equivalence class* in $\mathcal{L}^1([0, 1])$.)

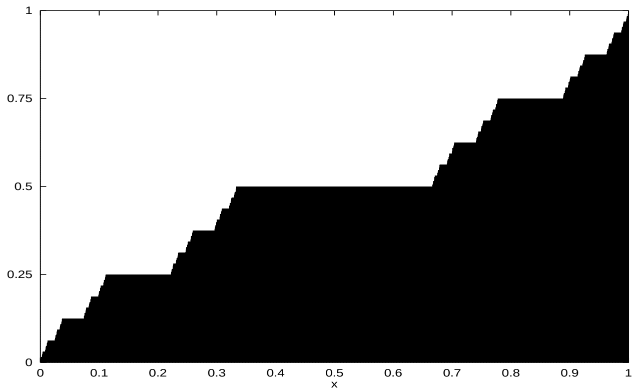

Figure 7: The "Devil's staircase" function, the attractor of the three-map IFSM given in Eq. (13).

Now we have an IFS-method of acting on functions. Along with a set of IFS maps f_i there is a corresponding set of grey-level maps ϕ_i. Together, Under suitable conditions, the determine a unique attracting fixed point function $\bar{u}$ which can be generated by iterating operator,T, defined in Eq. (11). As was the case with the "geometrical IFS" earlier, we are naturally led to the following *inverse problem for function (or image) approximation*:

> Given a "target" function (or image) v and an $\epsilon > 0$, can we find an IFSM $(\mathbf{f}, \Phi)$ whose attractor $\bar{u}$ approximates v to within ϵ, i.e. satisfies the inequality $\| v - \bar{u} \|_1 < \epsilon$?

For the same reason as in Part I, namely the "Collage Theorem" (an important corollary of Banach's Contraction Mapping Theorem), the above inverse problem may be conveniently posed as follows:

> Given a target function v and an $\delta > 0$, can we find an IFSM $(\mathbf{f}, \Phi)$ with associated operator T, such that the "collage distance" satisfies $\| v - Tv \|_1 < \delta$? This basically asks the question, "How well can we 'tile' the graph of v with distorted copies of itself (subject to the operations given above)?"

Now, you might comment, it looks like we're right back where we started. We have to examine a graph for some kind of "self-tiling" symmetries, involving both geometry (the f_i)

as well as grey-levels (the ϕ_i), which sounds quite difficult. The response is "Yes, in general it is." However, it turns out that an enormous simplification is achieved if we give up the idea of trying to find the best IFS maps f_i. Instead, we choose to work with a *fixed* set of IFS maps f_i, $1 \leq i \leq N$, and then find the "best" grey-level maps ϕ_i associated with the f_i.

> **Question:** What are these "best" grey-level maps?
>
> **Answer:** They are the ϕ_i maps which will give the best "collage" or tiling of the function v with shrunken copies of itself using the fixed IFS maps, w_i.

To illustrate, consider the target function $v = \sqrt{x}$. Suppose that we work with the following two IFS maps on [0,1]: $f_1(x) = \frac{1}{2}x$ and $f_2(x) = \frac{1}{2}x + \frac{1}{2}$. Note that $\hat{f}_1(X) = [0, \frac{1}{2}]$ and $\hat{f}_1(X) = [\frac{1}{2}, 1]$. The two sets $\hat{f}(X)$ overlap only at $x = \frac{1}{2}$.

(**Note:** It is very convenient to work with IFS maps for which the overlapping between subsets $\hat{f}_i(X)$ is minimal, referred to as the "nonoverlapping" case. In fact, this is the usual practice in applications. The remainder of this discussion will be restricted to the nonoverlapping case, so you can forget all of the earlier headaches involving "overlapping" and combining of fractal components.)

We wish to find the best ϕ_i maps, i.e. those that make $\| v - Tv \|_1$ small. Roughly speaking, we would like that

$$v(x) \approx (Tv)(x), \qquad x \in X, \tag{14}$$

or at least for as many $x \in X$ as possible. Recall from our earlier discussion that the first step in the action of the T operator is to produce copies of v which are "shrunk" in the x-direction onto the subsets $\hat{f}_i(X)$. These copies, $a_i(x) = v(f_i^{-1}(x))$, $i = 1, 2$, are shown in Figure 8(a) along with the target $v(x)$ for reference. The final action is to modify these functions $a_i(x)$ to produce functions $b_i(x)$ which are to be as close as possible to the pieces of the original target function v which sit on the subsets $\hat{f}_i(X)$. Recall that this is the role of the grey-level maps ϕ_i since $b_i(x) = \phi_i(a_i(x))$ for all $x \in \hat{f}_i(X)$. Ideally, we would like grey-level maps that give the result

$$v(x) \approx b_i(x) = \phi_i(v(f_i^{-1}(x))), \qquad x \in \hat{f}_i(X). \tag{15}$$

Thus if, for all $x \in \hat{f}_i(X)$, we plot $v(x)$ vs. $v(f_i^{-1}(x))$, then we have an idea of what the map ϕ_i should look like. Figure 8(b) shows these plots for the two subsets $\hat{f}_i(X)$, $i = 1, 2$. In this particular example, the exact form of the grey level maps can be derived: $\phi_1(t) = \frac{1}{\sqrt{2}}t$ and $\phi_2(t) = \frac{1}{\sqrt{2}}\sqrt{t^2 + 1}$. I leave this as an exercise for the interested reader.

In general, however, the functional form of the ϕ_i grey level maps will not be known. In fact, such plots will generally produce quite scattered sets of points, often with several $\phi(t)$ values for a single t value. The goal is then to find the "best" grey level curves which pass through these data points. But that sounds like *least squares*, doesn't it? In most such "fractal transform" applications, only a straight line fit of the form $\phi_i(t) = \alpha_i t + \beta_i$ is assumed. For the functions in Figure 8(b), the "best" affine grey level maps associated with the two IFS maps given above are:

$$\begin{aligned} \phi_1(t) &= \frac{1}{\sqrt{2}}t, \\ \phi_2(t) &\approx 0.35216t + 0.62717. \end{aligned} \tag{16}$$

The attractor of this 2-map IFSM, shown in Figure 8(c), is a very good approximation to the target function $v(x) = \sqrt{x}$.

In principle, if more IFS maps w_i and associated grey level maps ϕ_i are added, albeit in a careful manner, then a better accuracy can be achieved. (Prof. Forte and I have shown this theoretically, subject to conditions on the IFS maps as well as the grey-level maps.) The primary goal of IFS-based methods of image compression, however, is not necessarily to provide approximations of arbitary accuracy, but rather to provide approximations of acceptable accuracy "to the discerning eye" with as few parameters as possible. As well, it is desirable to be able to compute the IFS parameters in a reasonable amount of time. A significant improvement, which follows a method introduced in 1989 by A. Jacquin, a Ph.D. student of Barnsley (now at Bell Labs), lies in the method of "local" IFS. Rather than attempting to express a set or an image as a union of copies of itself, the local IFSM method seeks to express it as a union of copies of *subsets* of itself. It seems reasonable to look for more local self-similarities in a function or image as opposed to global properties assumed by the traditional IFS approach. (For anyone interested, try to approximate the function $v(x) = \sin(\pi(x))$ on $[0, 1]$ using the two-map IFS in the $\sqrt{x}$ example above. You will see the problems in fitting the grey-level maps. It is much better to split the graph into two halves, i.e. one on $[0, \frac{1}{2}]$ and the other on $[\frac{1}{2}, 1]$.)

Let us now return to "real" two-dimensional images, in particular, "Lena" in Figure 4(a). Suppose that we perform a Jacquin-type method on this 512×512 pixel target image as follows. Divide the image into nonoverlapping square blocks 16 pixels wide and 16 pixels high. There are $32^2 = 1024$ of these square blocks which we shall call the "parent blocks", I_j. Now divide each of these blocks into 4 smaller blocks, .i.e. blocks which are 8 pixels by 8 pixels square. There are $64^2 = 4096$ of these "child blocks", J_k. There are eight possible contractive IFS-type maps, $f^{(l)}_{j,k}, 1 \leq l \leq 8$, from a given parent block I_j to a given child block J_k (4 rotations, 4 mirror inversions).

It now remains to choose, for each child block J_k a parent block $I_{j(k)}$ (note the dependence on k, and an IFS map $f^{l(k)}_{k,j(k)}$. Once this choice is made (a number of approaches are shown below), optimal affine grey level map $\phi_k(t) = \alpha_k(t) + \beta_k$ as was done for the one-dimensional case above, i.e. the map which prodces the best least squares fit of the subimage on the parent block onto the subimage on the child block. When this is done for all child blocks, we have a "code" for the target image: in this case, a set of numbers

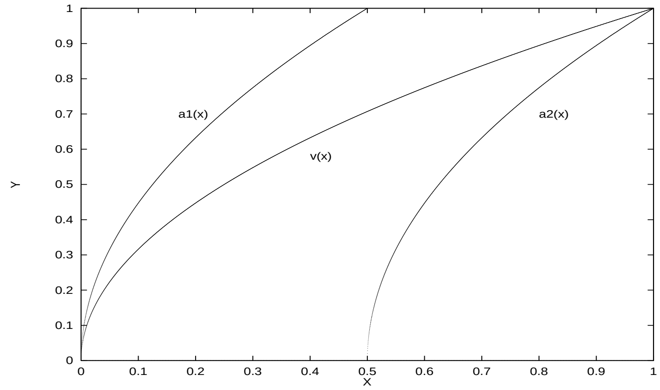

Figure 8(a): The target function $v(x) = \sqrt(x)$ on [0,1] along with its contractions $a_i(x) = v(w_i^{-1}(x))$, $i = 1, 2$, where the two IFS maps are $w_1(x) = \frac{1}{2}x$, $w_2(x) = \frac{1}{2}x + \frac{1}{2}$.

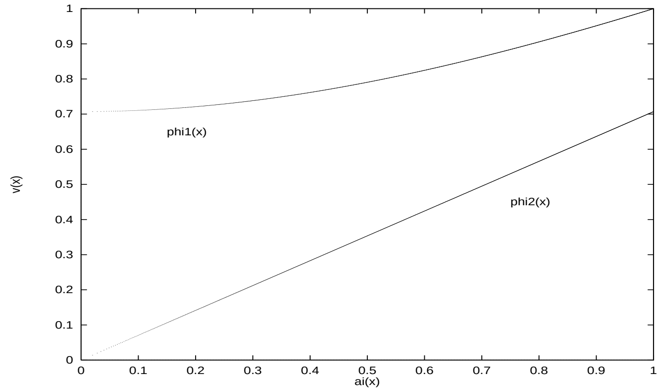

Figure 8(b): Plots of $v(x)$ vs $a_i(x) = v(w_i^{-1}(x))$ for $x \in w_i(X)$, $i = 1, 2$. These graphs reveal the grey level maps ϕ_i associated with the two-map IFSM.

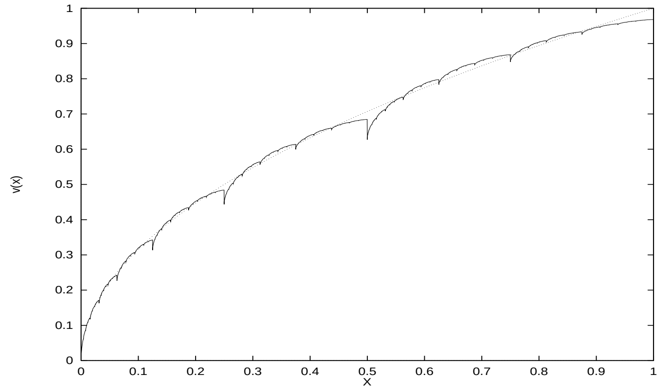

Figure 8(c): The attractor of the two-map IFSM with grey level maps given in Eq. (16).

$(j(k), \alpha_k, \beta_k), 1 \leq k \leq 4096$ which define a unique contractive operator T associated with the local IFSM. Starting with any "seed image", even a totally black or white screen, we may iterate this operator T to generate the attractor function $\bar{u}$ of the IFSM which, hopefully, will be a reasonable approximation to the target image. In the results presented below, the "$\mathcal{L}^1$ error" of the approximation $\bar{u}$ to the target image v is simply the distance $\| v - \bar{u} \|_1$.

(An embarrassing disclaimer: I actually performed these calculations a couple of years ago using completely unoptimized FORTRAN programs. Since then, my students - and others - have written much better routines in C which have reduce the computational time by factors of up to 100. The relative computational costs of the various methods are still preserved, however.)

Method 1: For each child block, J_k, find the "best" parent block $I_{j(k)}$. This means that for each of the eight IFS transformations of the parent to the child, we find the optimal grey-level map and then select the case which gives the best fit. For each child, we then perform a search over *all* possible parents. This method requires over three hours of computer time to determine the optimal IFS approximation, shown in Figure 9(a). The $\mathcal{L}^1$ error of this approximation is 0.018.

Method 2: If we eliminate the search of optimal parent blocks and choose, for example, the parent block "closest" to the child (in fact, the parent which contains the given child), the approximation in Figure 9(b) is obtained. The $\mathcal{L}^1$ error of this approximation is 0.029, somewhat higher than that of Fig. 4. The compression time was only 34 sec.. However, from a visual perspective, the approximation is not acceptable.

Method 3: One possibility to reduce computer time without sacrificing much error in approximation is to use "place-dependent" grey-level maps. These maps necessarily involve an extra couple of parameters in the "collaging" of parent-child pairs, so the compression factor is lowered slightly. In Figure 9(c) we show the IFS approximation obtained with this method, using the same number of child and parent blocks as above but with no searching for optimal parent blocks. The $\mathcal{L}^1$ error of this approximation to "Lena" is 0.022, lying slightly closer to the approximation of Method 1 than that of Method 2. This approximation is more acceptable visually since the "blockiness" introduced by the IFS partitioning is less apparent. As well, the "compression time" is a mere 27 seconds. The compression associated with this IFSM is about 10 to 1. (Actually, this ratio can be improved to something like 15 or 20 to 1 by means of efficient coding of the IFS parameters, e.g. "entropy coding", but that's another story.) Not bad, but we're looking for even greater accuracy in possibly less time (the goal being a fraction of a second) and with the smallest possible number of IFS parameters, i.e. an even higher compression ratio.

Recently (in fact, since the time Part I appeared in print) there have been several significant developments which show much promise in fractal image compression. Some of these include what can be called (if I may borrow terminology from Electrical Engineering)

Figure 9(a): Approximation to "Lena" using Method 1.

Figure 9(b): Approximation to "Lena" using Method 2.

Figure 9(c): Approximation to "Lena" using Method 3.

"IFS methods in the frequency domain". Briefly, if we assume a more traditional expansion of our target image/function in terms of a set of orthonormal basis functions $\{q_n\}_{n=0}^{\infty}$, i.e.

$$v(x) = \sum_{n=0}^{\infty} c_n q_n(x), \tag{17}$$

then one can perform a kind of IFS compression on the expansion coefficients c_n. This turns out to be quite natural if the $q_n(x)$ functions are *wavelets.*

Finally, the compression of images is not the only practical application of this approximation method that we intend to explore. Our goal is to use this technique in to study some of the more difficult "boundary value" problems in applied mathematics, for example, (i) the vibration of fractured solids and (ii) the behaviour of water waves near an irregular coastline.

References

Original research papers:

J. Hutchinson, Fractals and self-similarity, Indiana Univ. J. Math. **30**, 713-747 (1981).

M.F. Barnsley and S. Demko, Iterated function systems and the global construction of fractals, Proc. Roy. Soc. London **A399**, 243-275 (1985).

A. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations, IEEE Trans. Image Proc. **1** 18-30 (1992).

Books:

M.F. Barnsley, *Fractals Everywhere*, Academic Press, New York (1988).

M.F. Barnsley and L.P. Hurd, *Fractal Image Compression*, A.K. Peters, Wellesley, Mass. (1993).

Y. Fisher, *Fractal Image Compression, Theory and Application*, Springer-Verlag (1995).

Expository papers:

M.F. Barnsley and A. Sloan, A better way to compress images, *BYTE* Magazine, January issue, pp. 215-223 (1988).

Y. Fisher, A discussion of fractal image compression, in *Chaos and Fractals, New Frontiers of Science*, H.-O. Peitgen, H. Jürgens and D. Saupe, Springer-Verlag (1994).

Current research papers:

B. Forte and E.R. Vrscay, "Theory of generalized fractal transforms" and "Inverse problem methods for generalized fractal transforms", to appear in *Fractal Image Encoding and Analysis*, edited by Y. Fisher (Springer Verlag, NY, 1996). Proceedings of the NATO Advanced Study Institute held in Trondheim, Norway, July 8-17, 1995. (available from the Waterloo Website or by anonymous ftp)

(Revised 5 September 1996)