Fractal Wavelet Compression of Audio Signals

Robert A. Wannamaker and Edward R. Vrscay

Department of Applied Mathematics, Faculty of Mathematics
University of Waterloo, Waterloo, ON, CANADA N2L 3G1
rob@audiolab.uwaterloo.ca, ervrscay@links.uwaterloo.ca

Abstract

A novel approach to audio signal compression is described which marries wavelet transform coding with fractal-based compression techniques. The signal is divided into frames which are subjected to a Fast Wavelet Transform, with subsequent fractal coding of high frequency wavelet coefficients using a collage approach. Results from initial implementations are examined. Compression ratios above 6:1 should ultimately be attainable with good fidelity signal reconstruction.

1 Introduction

Over the past decade, audio compression has become almost ubiquitous due to the enormous appetite of multimedia systems for data. Current state-of-the-art audio compression schemes transform the incoming signal (using a discrete cosine transform or a set of subband filter banks) and quantize each transformed datum according to its psychoacoustic salience as determined by some psychoacoustic model [1]. The signal reconstructed from the coded data will not be identical to the original signal, but should (ideally) be *perceived* as identical by a human listener.

In this paper we describe an audio compression scheme which adopts a somewhat different approach. It also begins by transforming the data—in this case by means of a discrete wavelet transform—but then proceeds to compress the resulting transformed data using fractal-based compression. In a sense we are applying fractal compression *indirectly* to our signals. This hybrid *fractal wavelet compression (FWC)* scheme was discovered and developed independently by a number of researchers [2, 3, 4, 5, 6, 8].

The fractal wavelet hybrid transform attempts to combine the best of the two worlds of fractal-based compression and wavelet representation of signals. Purely fractal-based methods for the approximation of images have shown much potential in data compression [8, 9]. Their effectiveness in image compression is due in part to their ability to approximate discontinuous functions. As such, they are not as suitable for the approximation of audio signals which usually exhibit greater smoothness.

Earlier experiments [10] have shown that a direct application of fractal compression to audio signals does not lead to any significant compression when good fidelity reconstruction is required. It is here that the wavelet representation is advantageous. Although the crude audio realization described herein does not attempt to use psychoacoustic information in the coding process, the compression ratios obtained are nonetheless competitive with conventional schemes. The potential for considerable further improvement exists as well.

The next section presents a elementary outline of wavelet theory and multiresolution analysis. Section 3 introduces the theory of fractal-based approximation with Section 3.2 containing the main ideas behind the hybrid fractal-wavelet transform. Section 4 discusses the implementation of these ideas for audio compression purposes and some empirical results from our current implementations are presented in Section 5.

2 Wavelet Representations

This section presents a brief introduction to wavelets and multiresolution analysis. For a comprehensive treatment, the reader is referred to [11] and [12].

2.1 Wavelet Bases

The term *wavelets* refers to certain basis functions for subspaces of $\mathcal{L}^2(\mathbf{R})$, which is the set of functions $f : \mathbf{R} \to \mathbf{R}$ such that $\| f \|_2^2 = \int_{-\infty}^{\infty} |f(x)|^2 dx < \infty$. These subspaces are associated with scale-independent signal representations known as *multiresolution analyses (MRA)*. The most succinct definition of an MRA has been provided by Mallat [11]: it is a sequence $\{V_i\}_{i=-\infty}^{\infty}$ of linear subspaces of $\mathcal{L}^2$ which satisfies the following requirements:

1. there exists a linear operator A_i mapping functions $f \in \mathcal{L}^2$ to V_i,
2. A_i is an orthogonal projection,
3. $V_i \subset V_{i+1}$,
4. $f(t) \in V_i \Leftrightarrow f(2t) \in V_{i+1}$,
5. $\forall k \in Z, \qquad A_0 f_k(x) = A_0 f(x-k)$, where $f_k(x) = f(x-k)$,
6. there exists an isomorphism I between V_i and l^2 such that[1]

$$I(A_0 f(x)) = (a_j)_{j \in Z} \Leftrightarrow I(A_0 f_k(x)) = (a_{j-k})_{j \in Z},$$

[1] A sequence $\{a_n\}$ is in the Hilbert space l^2 if $\sum_{(n \in \mathbf{Z})} a_n^2 < \infty$.

7. $\lim_{i\to\infty} V_i$ is dense in $\mathcal{L}^2$,

8. $\lim_{i\to-\infty} = \{0\}$.

Roughly speaking, V_i represents the space of all functions in $\mathcal{L}^2$ whose details of resolution less than 2^{-i} have been removed. We can thus construct a sequence of sets W_i which contains the difference between functions projected on V_i and V_{i+1}. This difference is referred to as a *detail signal*:

$$(A_{i+1} - A_i)f \in W_i.$$

Here W_i is the orthogonal complement of V_i in V_{i+1} so that

$$V_{i+1} = V_i \oplus W_i$$

where $\oplus$ denotes a direct sum of function spaces.

For any given MRA, it can be shown [11] that there exists a unique *scaling function*, $\phi(t)$, such that $\{\phi_{ij}\}_{j=-\infty}^{\infty}$ form an orthonormal basis of V_i, where

$$\phi_{ij}(t) = 2^{i/2}\phi(2^i t - j)$$

with[2]

$$\langle \phi_{ij}, \phi_{il} \rangle = \delta_{jl}.$$

Thus we may write

$$A_i f = \sum_{j=-\infty}^{\infty} \langle \phi_{ij}(t), f \rangle \phi_{ij}(t). \tag{1}$$

(Note that if $i = 0$, then the inner products supply the required isomorphism of Requirement 6 above.)

Furthermore, there exists a unique *mother wavelet*, $\psi(t)$, such that $\{\psi_{ij}\}_{j=-\infty}^{\infty}$ form an orthonormal basis for W_i, where

$$\psi_{ij}(t) = 2^{i/2}\psi(2^i t - j)$$

with

$$\langle \psi_{ij}, \psi_{il} \rangle = \delta_{jl}.$$

Thus we may alternatively write

$$A_k f = \sum_{i=-\infty}^{k-1} \sum_{j=-\infty}^{\infty} \langle \psi_{ij}(t), f \rangle \psi_{ij}(t). \tag{2}$$

[2]The Kronecker delta δ_{ij} equals one for $i = j$ and is otherwise equal to zero.

We know from Requirement 7 above that for $f \in \mathcal{L}^2$

$$f = \lim_{i \to \infty} A_i f,$$

with $A_i f$ given by either Eq. (1) or (2). We can also combine these equations to represent f as a *partial wavelet decomposition,* where all details below some level k are represented by a linear combination of dilated translates of the scaling function, ϕ and finer resolution details are represented with the wavelet basis functions ψ_{ij}:

$$f(x) = \sum_{j=-\infty}^{\infty} \langle f, \phi_{kj} \rangle \phi_{kj}(t) + \sum_{i=k}^{\infty} \sum_{j=-\infty}^{\infty} \langle f, \psi_{i,j} \rangle \psi_{i,j}(t). \tag{3}$$

A simple example of a wavelet basis is the Haar basis associated with scaled and translated copies of the following scaling function and mother wavelet:

$$\phi(t) = \begin{cases} 1 & \text{if } 0 \leq t < 1, \\ 0 & \text{otherwise,} \end{cases}$$

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < \frac{1}{2}, \\ -1 & \text{if } \frac{1}{2} \leq t < 1, \\ 0 & \text{otherwise,} \end{cases}$$

Representative members of the Haar basis are illustrated in Fig. 1. Many other bases can be constructed, including smooth ones, but we will not explore the techniques for doing so here (see, for instance, [12]). Instead, Fig. 2 simply presents some characteristic examples of smooth scaling functions and mother wavelets.

2.2 Advantages of the Wavelet Representation

Wavelets can be chosen so as to tend to zero as rapidly as desired, and some are compactly supported [12]. Thus, an error in a wavelet coefficient value introduces an error which is localized in time. This is in contrast with an error in a Fourier transform coefficient, which introduces an error distributed *throughout* the associated time signal.

On the other hand, the Fourier transform of a wavelet is a bandpass function [12], so that errors are also localized in frequency. This joint localization in time and frequency is one advantage of a wavelet representation for audio applications, since small errors associated with coefficient approximation reside nearby in both time and frequency to the larger (primary) signal and thus tend to be *masked* (i.e., rendered imperceptible) by it [1].

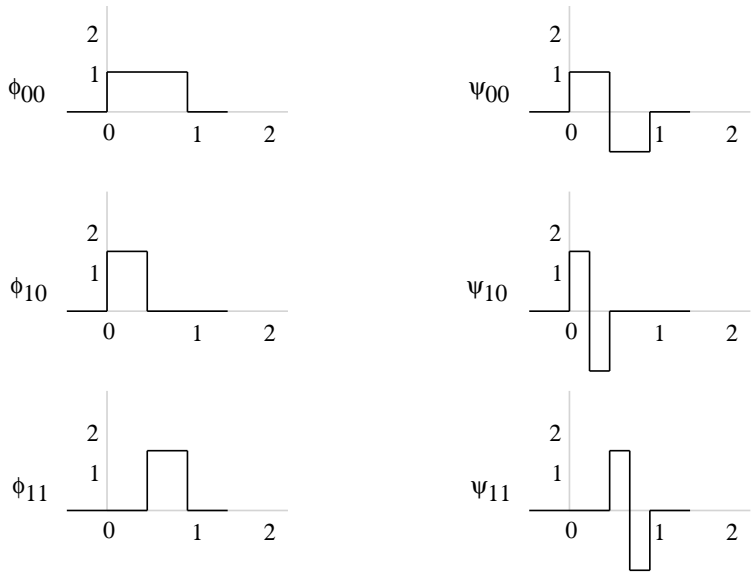


Figure 1: Some representative functions in the Haar basis.

The time/frequency localization of wavelets is often illustrated as a tiling of the time/frequency plane (see Fig. 3). Each region in the figure corresponds to one basis function[3].

2.3 The Fast Wavelet Transform

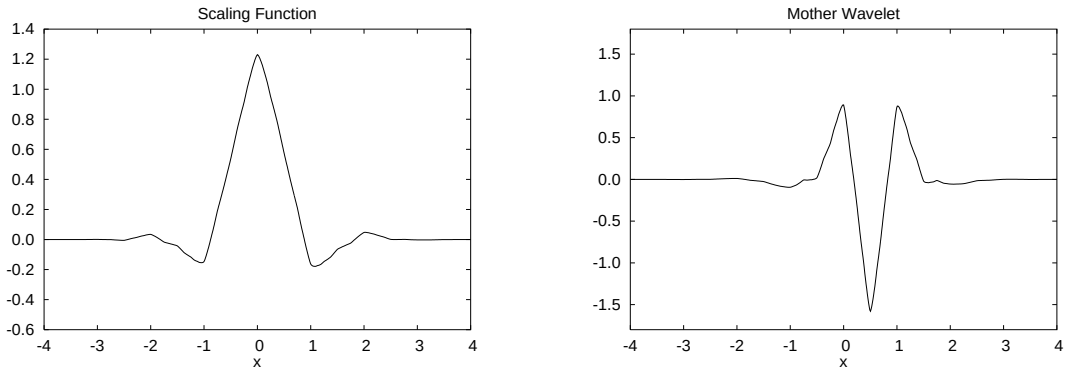
Frequently it is of interest to find the wavelet representation of a discrete-time (i.e., sampled) signal. Customarily, it is assumed that the signal is sampled frequently enough that the inner products $\langle \phi_{ij}, f \rangle$ in Eq. (1) are close to the sample values. The rationale for this assumption is that, in general,

$$\lim_{i \to \infty} 2^{i/2} \phi_{ij}(t) = \delta(t - 2^{-i} j).$$

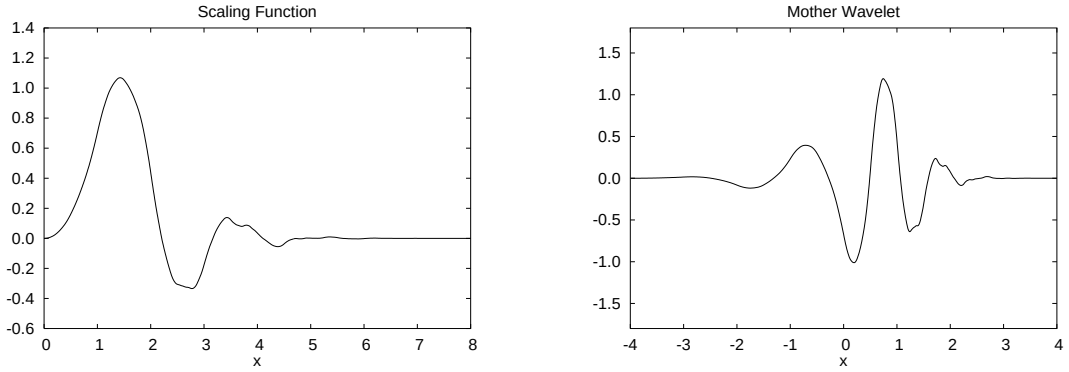
To begin, suppose we wish merely to compute $A_{i-1} f$ from $A_i f$. Following [5] we note that $\phi_{i-1,j} \in V_{i-1} \subset V_i$. Thus we may write

$$\phi_{i-1,j} \quad = \quad \sum_{k=-\infty}^{\infty} \phi_{i,k} \langle \phi_{i,k}, \phi_{i-1,j} \rangle$$

[3] In general, the regions overlap, although they are represented as distinct in the figure for reasons of clarity.



(a)



(b)

Figure 2: Some representative smooth scaling functions, $\phi(x)$, (left) and mother wavelets, $\psi(x)$, (right); (a) Coifman-12, (b) Daubechies-10.

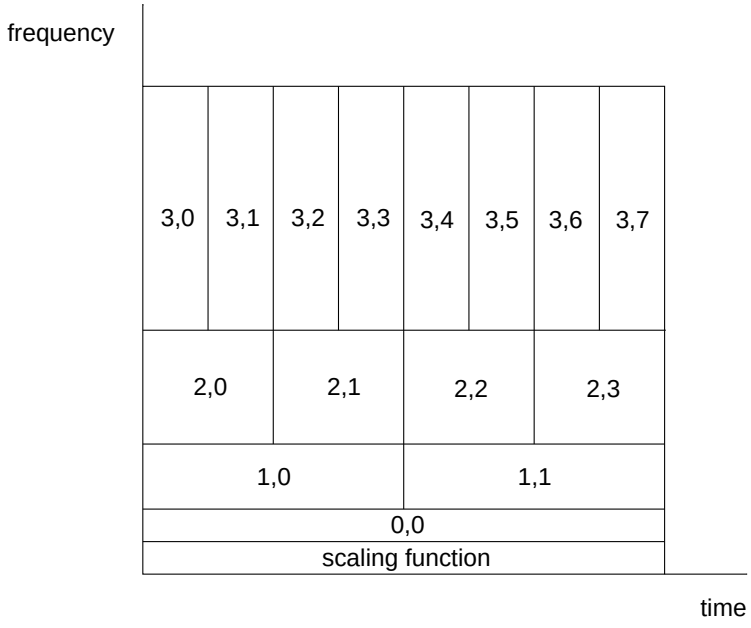


Figure 3: Tiling of time-frequency plane by wavelet basis functions. Coefficient indices i,j are indicated.

$$
\begin{aligned}
&= \sum_{k=-\infty}^{\infty} \phi_{i,k} \int_{-\infty}^{\infty} 2^{i/2}\phi(2^i t-k)2^{(i-1)/2}\phi(2^{(i-1)/2})\phi(2^{i-1}t-j)dt \\
&= \sum_{k=-\infty}^{\infty} \phi_{i,k}\sqrt{2}\int_{-\infty}^{\infty} \phi(2s+2j-k)\phi(s)ds \qquad (\text{where } s=2^{i-1}t-j) \\
&= \sum_{k=-\infty}^{\infty} \phi_{i,k}\langle \phi_{1,k-2j},\phi\rangle \\
&= \sum_{k=-\infty}^{\infty} h_{k-2j}\phi_{i,k},
\end{aligned}
$$

where

$$h_n = \langle \phi_{1,n}, \phi\rangle. \tag{4}$$

Now we define the *scaling function coefficients*

$$b_{i-1,j} = \langle \phi_{i-1,j}, f\rangle.$$

Then

$$
\begin{aligned}
b_{i-1,j} &= \left\langle \sum_{k=-\infty}^{\infty} h_{k-2j}\phi_{i,k}, f \right\rangle \\
&= \sum_{k=-\infty}^{\infty} h_{k-2j}\langle \phi_{i,k}, f\rangle
\end{aligned}
$$

so that

$$b_{i-1,j} = \sum_{k=-\infty}^{\infty} h_{k-2j} b_{ik}. \tag{5}$$

Note that this corresponds to a filtering of the b_{ik}'s followed by a downsampling by a factor of 2.

We can also find the *wavelet coefficients*

$$a_{i-1,j} = \langle \psi_{i-1,j}, f\rangle$$

in terms of the b_{ij}'s by exploiting the fact that $\psi_{i-1,j} \in W_{i-1} \subset V_i$ and writing

$$\psi_{i-1,j} = \sum_{k=-\infty}^{\infty} \langle \phi_{i,k}, \psi_{i-1,j}\rangle \phi_{ik}.$$

Proceeding as above for $b_{i-1,j} = \langle \phi_{i-1,j}, f \rangle$ we find that

$$a_{i-1,j} = \sum_{k=-\infty}^{\infty} g_{k-2j} b_{ik} \tag{6}$$

where

$$g_n = \langle \phi_{1,n}, \psi \rangle. \tag{7}$$

We can also express the $b_{i,j}$'s in terms of the $b_{i-1,j}$'s and $a_{i-1,j}$'s by noting that $\phi_{ij} \in V_i$ and $V_i = V_{i-1} \oplus W_{i-1}$. That is, we may write that

$$\phi_{ij} = \sum_{k=-\infty}^{\infty} \langle \phi_{i-1,k}, \phi_{ij} \rangle \phi_{i-1,k} + \sum_{k=-\infty}^{\infty} \langle \psi_{i-1,k}, \phi_{ij} \rangle \psi_{i-1,k}.$$

Proceeding as above yields

$$b_{ij} = \sum_{k=-\infty}^{\infty} h_{j-2k} b_{i-1,k} + \sum_{k=-\infty}^{\infty} g_{j-2k} a_{i-1,k}. \tag{8}$$

The sets of coefficients $\{h_n\}$ and $\{g_n\}$ are referred to as a pair of *quadrature mirror filters (QMF's)* and uniquely characterize a given scaling function/wavelet pair (ϕ, ψ). The application of h removes details below a given resolution so that h is essentially a lowpass filter, while the application of g extracts the same information which h removes, and thus g is essentially highpass. It can be shown [12] that the the coefficients of these filters are given in terms of one another by the simple relation

$$g_n = (-1)^n h_{1-n}. \tag{9}$$

For instance, for the Haar basis the corresponding QMF's are straightforwardly computed to be

$$h_0 = \frac{1}{\sqrt{2}} \qquad h_1 = \frac{1}{\sqrt{2}}$$

$$g_0 = \frac{1}{\sqrt{2}} \qquad g_1 = -\frac{1}{\sqrt{2}}$$

with all other coefficients being zero.

Sometimes it is of interest to find a wavelet representation for a finite data set or *frame*. By way of example, suppose we start with an input signal consisting of $M = 2^m$ samples in $X = [0, 1]$ located at $x = 2^{-m} j$, $0 \leq j < 2^m$. We begin by treating these as scaling function coefficients ($b_{m,j}$'s) and periodizing the data frame.

This operation consists of constructing a periodic signal with period equal to one frame length and consisting of contiguous copies of the data frame of interest. In practice, this is accomplished by placing the data frame in a circular buffer. For a data frame with length $M = 2^m$, the first application of the QMF's (Eqs. (5) and (6)) yields two new circular buffers, containing $M/2$ wavelet coefficients $\{a_{m-1,j}\}$ and $M/2$ scaling function coefficients $\{b_{m-1,j}\}$, respectively. The QMF's are applied to successive generations of scaling function coefficients until only one scaling function coefficient, b_{00}, remains, which, together with the $M-1$ wavelet coefficients, provides a unique wavelet representation of the data frame. The set of numbers consisting of all the wavelet coefficients and b_{00} is referred to as the *Fast Wavelet Transform* of the input signal[4]. The input samples $\{b_{i,j}\}$ can be reconstructed from the transformed signal by repeated application of Eq. (8), generating a sequence of circular buffers increasing in size.

3 Fractal-Based Data Compression

In this section we outline the main ideas behind fractal wavelet coding of signals. It is first necessary to *briefly* introduce the basic ideas of Iterated Function Systems (IFS). For a more complete introduction to the topics of IFS and compression, the reader is referred to [9]. A more detailed mathematical treatment is given in [7, 8]. The development in the following sections is a hybrid of that found in [7, 8] and [5].

3.1 Iterated Function Systems with Grey Level Maps (IFSM)

"Iterated Function Systems" (henceforth IFS) is the name introduced by Barnsley and Demko [13] to denote a system of contraction mappings in a complete metric space, e.g. [0,1]. The idea was independently developed earlier by Hutchinson [14] who showed how typically self-similar fractal sets could be generated by the "parallel" action of such systems of contraction mappings. The IFS maps plus a set of associated probabilities (IFSP) define operators which act on probability measures. As a result, early IFS research work focussed on the representation of images by measures and the approximation of these measures by IFSP measures. However, it is more convenient to represent images and signals by functions. The goal of IFS-type methods is then to approximate image/signal functions by functions which are generated by iterating an IFS-type operator. The IFSM is an example of an IFS or *fractal transform* method over an appropriate space of functions $\mathcal{F}(X)$ which represent our images or signals.

[4] It is, indeed, very fast having computational complexity of order N, as compared with the Fast Fourier Transform which has complexity of order $N \log(N)$.

Here, (X, d) denotes a complete metric space, referred to as the "base space". In what follows, $X = [0, 1]$ and d is the Euclidean metric on X. (In the case of images, X is typically $[0, 1]^2$ with Euclidean metric.) The space of functions $\mathcal{F}(X)$ will be $\mathcal{L}^2(X)$, i.e. the set of functions $f : X \to \mathbf{R}$ such that $\| f \|_2^2 = \int_X |f(x)|^2 dx < \infty$. The metric d_2 will be defined in terms of the norm, i.e. $d_2(f, g) = \| f - g \|_2$.

Now let $\mathbf{w} = \{w_1, w_2, \ldots, w_N\}$ denote a set of N contraction maps on X, i.e. $w_i : X \to X$ such that for each $i \in \{1, 2, \ldots, N\}$, there exists a $c_i \in [0, 1)$ such that $d(w_i(x), w_i(y)) \leq c_i d(x, y)$ for all $x, y \in X$. The set of maps $\mathbf{w}$ is a contractive IFS on X. For convenience, we work with affine IFS maps; i.e.

$$w_i(x) = s_i x + a_i, \quad c_i = |s_i| < 1, \quad 1 \leq i \leq N. \tag{10}$$

Associated with the IFS maps $\mathbf{w}$ is a set of grey level maps $\Phi = \{\phi_1, \phi_2, \ldots, \phi_N\}$, where $\phi_i : \mathbf{R} \to \mathbf{R}$. Again for convenience, we work with affine grey-level maps; i.e.

$$\phi_i(t) = \alpha_i t + \beta_i, \quad 1 \leq i \leq N. \tag{11}$$

The collection of maps $(\mathbf{w}, \Phi)$ is referred to as an N-map (affine) Iterated Function System with Grey Level Maps (IFSM). Associated with such an IFSM is a *fractal transform* operator $T : \mathcal{F}(X) \to \mathcal{F}(X)$ whose action will now be defined. For each $x \in X$ define its N *fractal components* f_i as follows:

$$f_i(x) = \begin{cases} \phi_i(u(w_i^{-1}(x))), & x \in w_i(X), \\ 0, & x \notin w_i(X). \end{cases} \tag{12}$$

We shall impose the additional condition on the w_i that for each $x \in X$ there exists at least one "preimage" $w_i^{-1}(x)$. Now, given a $u \in \mathcal{F}(X)$, its image Tu will be defined as follows:

$$(Tu)(x) = \sum_{i=1}^{N} f_i(x). \tag{13}$$

Geometrically, the action of the operator T may be viewed in terms of its action on the graph of u. Each term $u(w_i^{-1}(x))$ represents a shrunken copy of the graph of u which is supported on the set $w_i(X)$. The grey level map ϕ_i then distorts this shrunken copy in the grey level direction to produce the fractal component f_i. The fractal transform then adds up these fractal components to produce $(Tu)(x)$. This is illustrated below.

Example: Consider the following two-map IFSM on $X = [0, 1]$:

$$\begin{aligned} w_1(x) &= 0.6x, & \phi_1(t) &= 0.5t + 0.2, \\ w_2(x) &= 0.6x + 0.4, & \phi_2(t) &= 0.5t + 0.5. \end{aligned} \tag{14}$$

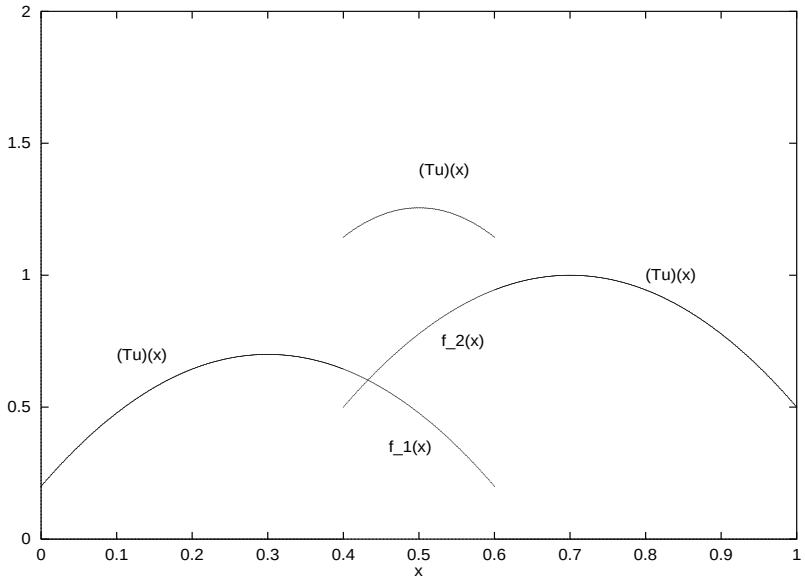


Figure 4: Graph of $(Tu)(x)$ where $u(x) = 4x(1-x)$ and T is the contractive operator for the two-map IFSM given in Eq. (14).

The action of the associated fractal transform operator T on the function $u(x) = 4x(1-x)$ is shown in Fig. 4. There are two fractal component functions, $f_1(x)$ and $f_2(x)$, which are supported on the subsets $w_1(X) = [0, 0.6]$ and $w_2(X) = [0.4, 1]$, respectively. All points $x \in w_1(X) \cap w_2(X) = [0.4, 0.6]$, have both fractal components which must be then added to produce $(Tu)(x)$, as shown in the Figure. For points $x \in [0, 0.4)$ and $x \in (0.6, 1]$, the function value $(Tu)(x)$ is determined by only one fractal component, namely, $f_1(x)$ for the former and $f_2(x)$ for the latter.

Let $u, v \in \mathcal{L}^2(X)$ with fractal components f_k and g_k, $1 \le k \le N$, respectively. Then from the relation

$$\begin{aligned} \| Tu - Tv \|_2 &= \| \sum_{k=1}^{N} [f_k(x) - g_k(x)] \|_2 \\ &\le \sum_{k=1}^{N} \| f_k(x) - g_k(x) \|_2, \end{aligned} \tag{15}$$

we obtain the result

$$d_2(Tu, Tv) \le C_2 d_2(u, v), \qquad C_2 = \sum_{k=1}^{N} c_k^{1/2} |\alpha_k|. \tag{16}$$

The standard assumption in the literature is that the sets $w_i(X)$ overlap at most on sets of zero Lebesgue measure, e.g. $N = 2$ with $w_1(x) = 0.5x$ and $w_2(x) = 0.5x + 0.5$. Then almost all points $x \in X$ have only one fractal component, which simplifies all procedures. In this special "nonoverlapping case", the above bound may be strengthened:

$$d_2(Tu, Tv) \leq \bar{C}_2 d_2(u, v), \qquad \bar{C}_2 = \left[\sum_{k=1}^{N} c_k \alpha_k^2\right]^{1/2}. \tag{17}$$

If $C_2 < 1$, then from Eq. (14), the operator T is *contractive* in $\mathcal{L}^2(X)$. From Banach's Fixed Point Theorem (Contraction Mapping Principle), there exists a unique function $\bar{u} \in \mathcal{F}(X)$ such that $T\bar{u} = \bar{u}$, i.e. $\bar{u}$ is the *fixed point* of T. Moreover, for any function $u \in \mathcal{F}(X)$, $d_2(T^n u, \bar{u}) \to 0$ as $n \to \infty$. As such, $\bar{u}$ is an *attractive* fixed point. The IFSM operator of the above example whose action is depicted in Fig. 4 is contractive in $\mathcal{L}^2(X)$: From Eq. (14), $C_2 = (2/3)^{1/2} < 1$. A histogram approximation of its fixed point function $\bar{u}$ is shown in Fig. 5. (This approximation was obtained by repeated application of T on a function, as begun in Fig. 4, until satisfactory convergence was achieved.)

We now pose the so-called *inverse problem* of function approximation using IFSM, the basis for fractal-based compression:

> Given a "target" function/image/signal $v \in \mathcal{F}(X)$ find an N-map IFSM $(\mathbf{w}, \Phi)$, $N > 0$, whose associated fractal transform operator T is contractive and whose fixed point $\bar{u}$ is "close" to v in d_2 metric.

From an important corollary of Banach's Fixed Point Theorem, referred to as the "Collage Theorem" [16], the inverse problem may be reformulated as follows:

> Given a target v and a $\delta > 0$ find an IFSM $(\mathbf{w}, \Phi)$ with contractive operator T such that $\Delta = d_2(v, Tv) < \delta$.

The term Δ is referred to as the "collage distance". In other words, we look for an operator T which sends the target v close to itself. This is basically looking for a way in which to "tile" the graph of the target function v with shrunken and distorted copies of itself. The Collage Theorem states that the distance between the attractor $\bar{u}$ of the above IFSM and the target v is within some multiple of δ; i.e., $d_2(\bar{u}, v) \leq K\delta$ for some $K \geq 0$. (If we were lucky enough to find the exact IFSM operator T whose fixed point is $Tv = v$, then $K = 0$.) If, for a suitable δ, such an IFSM is found, then the operator T is stored in computer memory. The approximation $\bar{u}$ to the target v may then be regenerated by taking any initial image/function u_0 (a blank screen, for

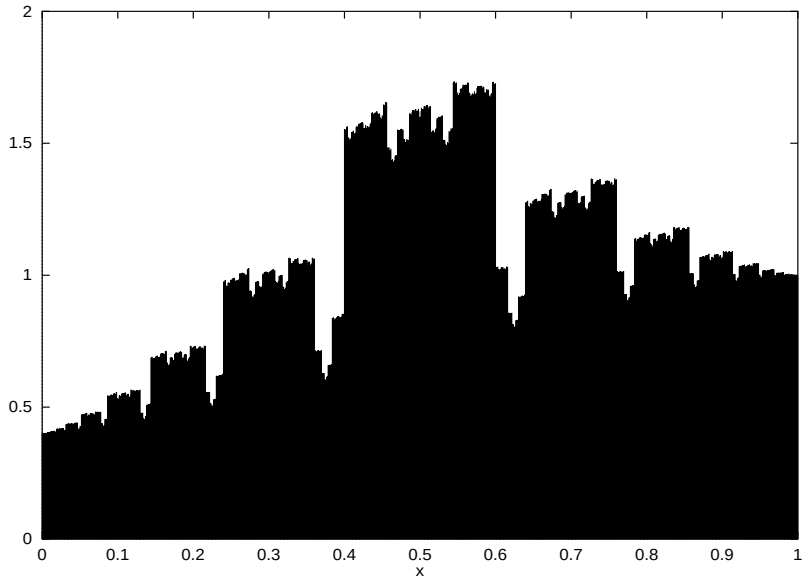


Figure 5: The fixed point attractor function $\bar{u}(x)$ of the IFSM operator T of Fig. 4. Since $\bar{u} = T\bar{u}$, the graph of this function is a union of two shrunken and modified copies of itself, as determined by the IFS and grey level maps in Eq. (14).

example, if v is an image) and forming the sequence $u_{n+1} = Tu_n$. The sequence of images u_n is guaranteed to converge to $\bar{u}$ from Banach's Fixed Point Theorem. From a compression point of view, of course, it is hoped that the storage of the operator T requires much less memory than the original target image/signal v.

A significant improvement which follows a method introduced in 1989 by A. Jacquin lies in the method of "local IFS" [17, 15]. Rather than attempting to express a function as a union of copies of subsets of *itself*, the local IFSM, or LIFSM, method seeks to express it as a union of copies of *subsets* of itself. In other words, the contractive IFS maps $\mathbf{w}$ map subsets of X into smaller subsets of X. We write these maps as $w_{jk} : I_j \to J_k$, where I_j is the *parent* or *domain block* which is mapped into the (smaller) *child* or *range block* J_k.

Example: Let $X = [0, 1]$. Suppose that there are two domain blocks $I_1 = [0, 0.5]$ and $J_2 = [0.5, 1]$ and four range blocks $J_i = [(i-1)/4, i/4]$, $1 \leq i \leq 4$. Thus, the IFS maps w_{jk} will have contraction factors $1/2$. Each range block J_k must have a domain block $I_{j(k)}$, $1 \leq j(k) \leq 2$, which is mapped onto it. Associated with each such IFS map $w_{j(k),k}$ is also a grey level map $\phi_k(t)$ which will modify the function values on $I_{j(k)}$ to produce the function values on J_k, $1 \leq k \leq 4$. Suppose that the indices $j(k)$ and grey level maps ϕ_k for the range blocks J_k are given by

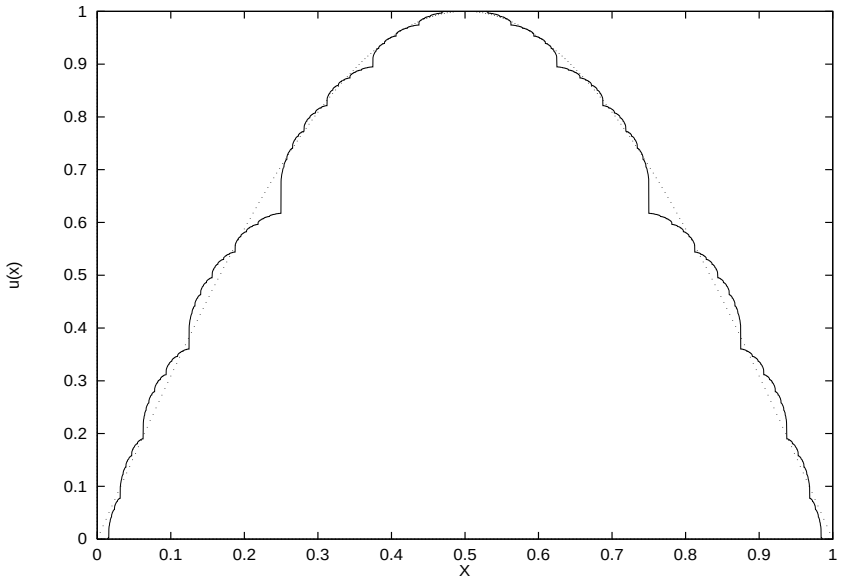


Figure 6: The attractor function $\bar{u}(x)$ of a four-map local IFSM. It is an approximation to the target function $v(x) = \sin(\pi x)$ which is also shown.

k	$j(k)$	$\phi_k(t)$
1	1	$0.66195t - 0.04843$
2	1	$0.28564t + 0.71847$
3	2	$0.28564t + 0.71847$
4	2	$0.66195t - 0.04843$

(In other words, block I_1 is mapped onto range blocks J_1 and J_2, but with different grey level maps.) An approximation of the attractor of this four-map local IFSM is shown in Fig. 6. In fact, the domain-range pairs and grey level maps ϕ_k minimize the collage distance associated with approximating the function $\sin(\pi x)$ on [0,1] with a two domain block/four range block LIFSM. A much superior approximation to the graph of this function is obtained by this method as compared to the "normal IFSM" method introduced in this Section. Roughly speaking, it is easier to "tile" the graph of $\sin(\pi x)$ with parts of itself. For example, the portion of the graph supported on $[0, 1/4]$ is better approximated as a shrunken copy of the graph supported on $[0, 1/2]$ as opposed to a shrunken copy of the entire graph supported on $[0, 1]$.

The hybrid fractal wavelet approach described in the next section is quite close in spirit to LIFSM, except that the space in which the iteration is carried out is not $\mathcal{L}^2(X)$, but a space of wavelet transform coefficients.

3.2 Fractal Wavelet Compression (FWC)

The object of fractal data compression is to approximate an arbitrary data set as the fixed point of an IFS. This is done in the hope that a satisfactory approximation may be achieved with a small enough value of N that storing the IFS mapping parameters requires less memory than storing the original data set.

Using Eq. (3), we can represent a function $u \in \mathcal{L}^2(\mathbf{R})$ as a *partial wavelet decomposition* where all details below level $k = 0$ are represented by a linear combination of translates of the scaling function, ϕ (not to be confused with the IFSM grey level maps) and finer resolution details are represented with the wavelet basis functions $\psi_{i,j}$:

$$u(x) = \sum_{j=-\infty}^{\infty} \langle u, \phi_{0,j} \rangle \phi_{0,j}(t) + \sum_{i=0}^{\infty} \sum_{j=-\infty}^{\infty} \langle u, \psi_{i,j} \rangle \psi_{i,j}(t). \tag{18}$$

We wish now to consider functions on $X = [0, 1]$. For arbitrary wavelet bases we could proceed using the periodization procedure introduced in Section 2.3, but for the sake of clarity we will restrict ourselves to use of the Haar basis in what follows. Then the following terms in Eq. (18) represent the wavelet expansion of a function on [0,1]:

$$u(x) = \langle u, \phi_{0,0} \rangle \phi_{0,0}(t) + \sum_{i=0}^{\infty} \sum_{j=0}^{2^i - 1} \langle u, \psi_{i,j} \rangle \psi_{i,j}(t).$$

Suppose we extract the following terms from this summation:

$$\sum_{i=k}^{\infty} \sum_{j=2^{i-k}l}^{2^{i-k}(l+1)-1} \langle u, \psi_{i,j} \rangle \psi_{i,j}(t), \qquad k \geq 0. \tag{19}$$

We will refer to these terms as comprising a single *block*. Indeed, if the wavelet basis in use is the Haar basis, then these terms correspond to the decomposed function within the interval $[2^{-k}l, 2^{-k}(l + 1)]$, less the mean value of the function on this interval, $\langle u, \phi_{k,l} \rangle \phi_{k,l}(t)$, which is included in the $0 \leq i < k$ terms not present in Eq. (19).

Now consider a local IFSM whose w_{jl}'s map domain blocks involving wavelets of lower resolution to range blocks involving wavelets of higher resolution. This would involve blocks corresponding to intervals

$$I_{i,j} = \left[\frac{j}{2^i}, \frac{j+1}{2^i} \right], \qquad 0 \leq j < 2^i, \quad i > 0,$$

with the domain blocks

$$I_{i^*,j}, \qquad 0 \le j < 2^{i^*}$$

and range blocks

$$I_{k^*,l}, \qquad 0 \le l < 2^{k^*},$$

where $k^* > i^*$. For each range block $I_{k^*,l}, 0 \le l < 2^{k^*}$ there will be an associated domain block $I_{i^*,j(l)}$ and the mapping between them will have a contraction factor $2^{i^*-k^*}$. Together these mappings constitute an IFS operator T with grey-level scaling parameters not yet specified.

It is of interest to express the coefficients of the wavelet expansion of Tu in terms of those for u. Using the orthogonality properties of the wavelet basis functions, it can be shown [5, 8] that if $c_{i,j} = \langle u, \psi_{i,j} \rangle$ and $d_{i,j} = \langle Tu, \psi_{i,j} \rangle$, then

$$d_{k^*+k',2^{k'}l+l'} = \alpha_l 2^{(i^*-k^*)/2} c_{i^*+k',2^{k'}j(l)+l'}, \quad k' \ge 0, 0 \le l' < 2^{k'}. \tag{20}$$

The coefficients $b_{0,0} = \langle u, \phi_{0,0} \rangle$ and $c_{i,j}, 0 \le i < k^*, 0 \le j < 2^i$ are unaffected by the action of T.

The net result is the following: Associated with the IFSM operator T is an operator M which maps blocks of wavelet coefficients to other blocks (see Fig. 7). In the above, the wavelet coefficients of each subtree with root entry $c_{i^*,j(l)}$ are multiplied by the factor $\alpha_l 2^{(i^*-k^*)/2}$. The resulting subtree then replaces the subtree with root $c_{k^*,l}$. The operator M is contractive in an appropriate metric involving the entire tree of wavelet coefficients [8]. Hence there exists a unique fixed point wavelet tree which is invariant under the action of M.

In terms of signal compression, given a signal with a known wavelet expansion, it remains to find a contractive mapping M as above which maps the expansion close to itself (i.e. its "collage distance" is small). From Eq. (20), this can be accomplished using a least squares procedure to find the α_l's which minimize the distance between the wavelet expansion coefficients of u and Tu. The distance function[5] associated with each range block is given by

$$\Delta_l^2 = \sum_{k'=0}^{\infty} \sum_{l'=0}^{2^{k'}-1} \left(c_{k^*+k',2^{k'}l+l'} - \alpha_l 2^{(i^*-k^*)/2} c_{i^*+k',2^{k'}j(l)+l'} \right)^2 .$$

The optimal scaling parameter $\bar{\alpha}_l$ can be found by straightforward differentiation [8]:

$$\bar{\alpha}_l \;=\; 2^{(k^*-i^*)/2} \frac{S_{k^*,l,i^*,j}}{S_{i^*,j,i^*,j}}, \tag{21}$$

[5] The wavelet transform is a unitary transform, so that the mean square distance between the signal and its approximation is equal to the mean square distance between their wavelet transforms.

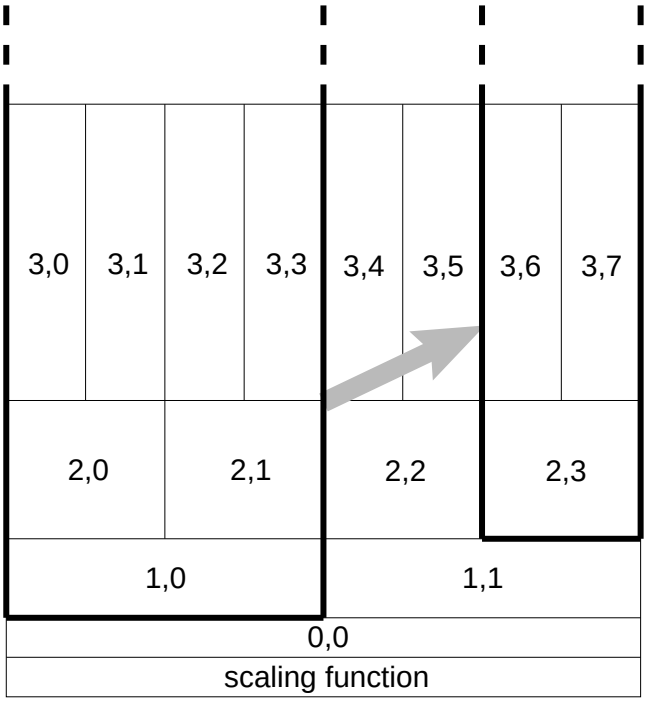


Figure 7: A typical blockwise mapping of wavelet coefficients with $k^* = 2$ and $i^* = 1$. In this case domain block 0 is mapped to range block 3; i.e., $j(3) = 0$.

where

$$S_{\alpha,\beta,\gamma,\delta} = \sum_{k'=0}^{\infty} \sum_{l'=0}^{2^{k'}-1} c_{\alpha+k',2^{k'}\beta+l'} c_{\gamma+k',2^{k'}\delta+l'}. \tag{22}$$

The minimized collage distance associated with this optimal parameter $\bar{\alpha}_l$ is given by

$$\Delta_{l,min} = [S_{k^*,l,k^*,l} - (S_{k^*,l,i^*,j(l)})^2 / S_{i^*,j(l),i^*,j(l)}]^{1/2}. \tag{23}$$

Recall that this is the minimum collage distance for the approximation of the range block $I_{k^*,l}$ using a *given* domain block $I_{k^*,j(l)}$. One may now proceed to search for the *best* domain block; i.e., the block $I_{k^*,\bar{j}(l)}$ which yields the lowest minimum collage distance $\Delta_{l,min}$. When this has been done for all range blocks, our optimal IFSM operator T has been defined. The fixed point of this operator (i.e., the desired approximation to the given signal) can be obtained by iteration. The function $\bar{u}$ corresponding to this wavelet tree can then be constructed by summing the wavelet series or inverting the Fast Wavelet Transform. (Note that this procedure involves a *full search* of all possible domain blocks. There are also algorithms which restrict the search, for example, to domain blocks which lie within a specified distance from a range block.)

The compressed signal consists of the optimal α_l's and $j(l)$'s along with the coefficients of the wavelet expansion which are unaffected by the IFSM operator, namely $b_{0,0}$ and

$$c_{i,j}, \qquad 0 \le i < k^*, 0 \le j < 2^i.$$

The value k^* is chosen at the outset. The wavelet coefficients $c_{i,j}$ below level k^* are stored directly while coefficients in or above that level are subjected to fractal compression.

It is important to note that if the given signal has finite resolution (i.e., if it corresponds to a sequence of samples) then all of the wavelet coefficients for the approximation signal can be reconstructed in a finite number of iterations. In order to see this, suppose that all coefficients are known for level i. Each of these may be expressed as a rescaling of some coefficients below level i, so that after one iteration all coefficients below level $i+1$ are known. Complete signal reconstruction entails iteration of the IFS to the finite resolution desired, followed by an inversion of the wavelet transform.

Technically speaking, the fractal transform operator M on wavelet coefficients associated with an IFSM/LIFSM operator on a function was derived from a look at the Haar basis functions. In fact, the fractal transform operator in Eq. 20 may also be defined for other wavelet bases in general. In essence, we may now forget about the "IFS roots" of the fractal wavelet transform and work exclusively in wavelet coefficient space.

4 Implementation for Audio

Since a signal which has been reconstructed following fractal wavelet compression is only an approximation to the original, the characteristics of the associated error signal bear careful consideration with regard to the application at hand (i.e., audio compression).

The choice of wavelet basis has a strong bearing on the character of the error signal. Non-smooth bases such as the Haar basis tend to engender "blockiness" in the reconstructed signal in the form of jump discontinuities at block boundaries. These are produced by the addition of errors in multiple wavelet coefficients which occurs when reconstructing the signal in these regions. Smooth wavelet bases, such as the Coifman bases [12], are preferable since they have longer QMF's which naturally interpolate between blocks to reduce the edge discontinuities. Eq. (19) shows that the blocks actually overlap in this case, but different blocks remain orthogonal to one another (so that no redundancy is introduced into the signal representation). The Coifman bases are also compactly supported so that "ringing" before and after approximately reconstructed signal spikes can be reduced by choosing wavelets of relatively small support.

Usually it is computationally infeasible to encode the entirety of an audio signal at once. Conventional transform-based compression algorithms such as MPEG Audio partition the signal into frames with a length on the order of 2^{10} samples, and then code each frame separately [1]. This can also be done in fractal wavelet coding schemes by periodizing the data frame before doing a Fast Wavelet Transform as discussed in Section 2.3. The resultant set of coefficients may be compressed using the fractal compression techniques described in the previous section.

This approach is fine if only one data frame is to be encoded, but edge effects reappear at the frame boundaries of multiframe signals (see Fig. 8). There is obviously no interpolation between frames even if smooth basis functions are chosen. This problem can be alleviated by allowing the QMF's to access data in frames adjacent to the current frame at the expense of having to retain these extra data frames in memory. The amount of extra data required depends on the length of the QMF's and the desired compression ratio. If all wavelet coefficients a_{ij} with $i \geq k^*$ are to be approximated, then only these coefficients need to be computed using data from adjacent frames. Coefficients with $i < k^*$ will be stored directly, and thus may be computed by periodizing the coefficients $\{a_{i+1,j}\}$ without introducing spurious edge effects.

Fig. 9 indicates the number of additional data frames required in order to compute

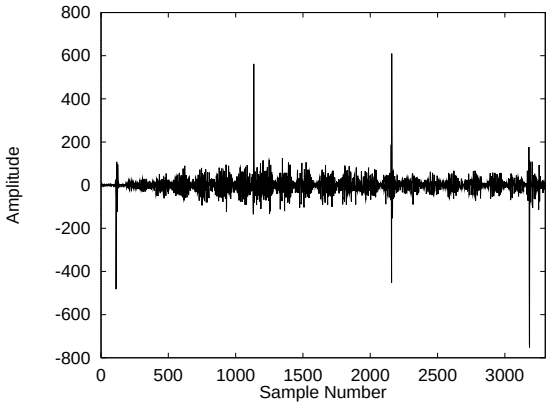


Figure 8: Error signal waveform (difference between original and reconstructed signals) showing error spikes near frame edges due to lack of interpolation between frames.

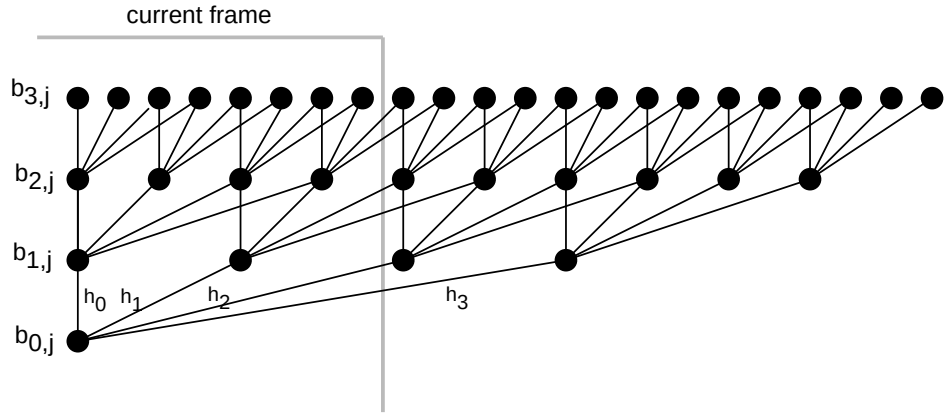


Figure 9: Extra points required from outside the current frame in order to calculate $D = 3$ levels of wavelet coefficients when $n_c = 4$ coefficient QMF's are in use. Scaling function coefficients are indicated by nodes, QMF coefficients by branches.

three levels of wavelet coefficients using a set of four-coefficient QMF's [6]. In general, if n_c-coefficient QMF's are in use, then the total number of data points required from outside of the current frame in order to calculate D levels of scaling function coefficients will be

$$\begin{aligned} n_{\text{ext}} &= \sum_{k=0}^{D-1} (n_c - 2)2^k \\ &= (n_c - 2)(2^D - 1). \end{aligned}$$

The same number of extra data points will be required at the leading frame edge in order to calculate the scaling function coefficients. If an integral number of data frames of length $M = 2^m$ data points is to be stored, then the required number of frames is

$$\kappa = \left\lceil \frac{n_{\text{ext}}}{M} \right\rceil$$

on either side of the current frame.

To code the first and last κ frames of the signal, it may be assumed that κ frames of zeros precede and follow the data of interest. When coded, these *zero frames* will, in general, yield non-zero wavelet coefficients since the QMF's will draw data from the signal proper. Proper reconstruction of the beginning and end of the signal requires transmission or storage of these zero frames for use as the extra frames of wavelet coefficients preceding/following the first/last frames of the signal. The overhead required to transmit or store these extra frames is negligible for signals of any appreciable length.

The error waveform in Fig. 10 illustrates the elimination of error spikes at frame boundaries achieved using this technique. Table 1 shows a typical frame of compressed output using $M = 16$ point frames and $k^* = 2$. For these computations, $i^* = k^* - 1$ so that the largest possible number of domain blocks were searched for matches.

5 Empirical Results

The compression and reconstruction algorithm was applied to a 20 second excerpt of piano music with (relatively) low background noise sampled at 22.05 kHz. Some noteworthy results are presented in Table 2. The mean square error (MSE) in the reconstructed signal was averaged over the entire record and the time required to

[6] The form of the filters shown here is of the sort where $h_i = 0 \quad \forall i < 0$, or, equivalently, $g_i = 0 \quad \forall i > 1$. Any QMF not in this form can be put in this form by introducing a pure time shift in the signal, formally equivalent to reindexing the coefficients so that $h_i = 0 \quad \forall i < 0$.

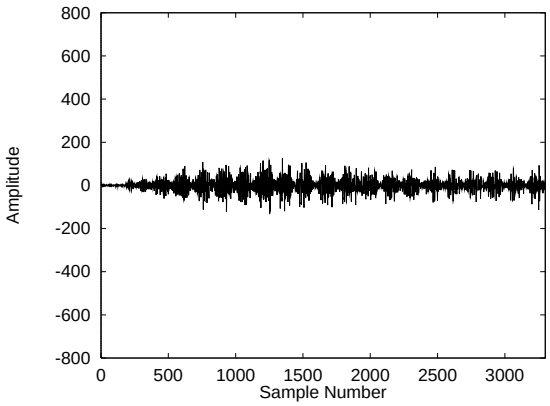


Figure 10: Error signal waveform (difference between original and reconstructed signals) without spikes at the frame edges, generated by using data from adjacent frames near the frame boundaries.

```
Stored wavelet coefficients:
b[0][0]:   1.19980
a[0][j]:  -0.86028
a[1][j]:  -0.41285 1.13481

IFS mapping parameters:  iStar=1  kStar=2
l=0 j(l)=1 alpha(l)=-0.01858
l=1 j(l)=0 alpha(l)=-0.24022
l=2 j(l)=1 alpha(l)= 0.01764
l=3 j(l)=1 alpha(l)=-0.10778

Mean Square Error: 0.01107
```

Table 1: A typical frame of compressed output using $M = 16$ point frames and $k^* = 2$.

Basis Functions	D	MSE per sample [bits^2]	Execution time per second of signal [sec]	
			Encode	Decode
Haar	5	97589	0.6	1.0
	4	7175	0.9	1.0
	3	110	1.7	1.1
	2	1.17	3.7	1.2
	1	0.000	8.3	1.5
Daubechies 10	5	77156	1.3	1.7
(extremal phase)	4	883	1.7	1.8
	3	11.3	2.5	1.9
	2	0.117	4.5	2.0
	1	0.000	9.2	2.4
Coifman 12	5	79002	1.5	1.9
(symmetric)	4	869	1.8	2.0
	3	10.4	2.6	2.0
	2	0.210	4.7	2.2
	1	0.000	9.4	2.6

Table 2: Mean Square Error, MSE, and computation time for coding and reconstruction of a 22.05 kHz piano signal with various choices of wavelet basis and degrees of compression, D. The data frames were 1024 points long. All experiments were performed using a 166 MHz Pentium processor.

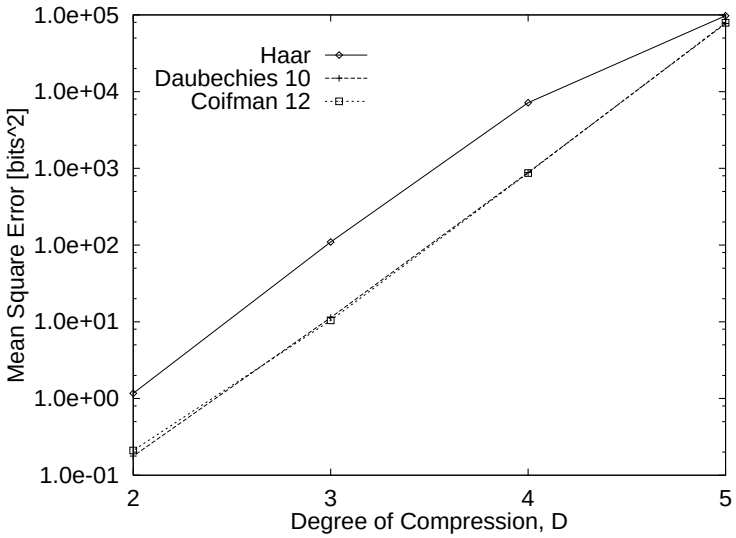


Figure 11: Mean Square Error in reconstructed signal as a function of compression factor, D, for Haar, Daubechies 10 and Coifman 12 bases.

complete each calculation on a 166 MHz Pentium processor was recorded. A variety of degrees of compression, D, were used with $i^* = k^* - 1$. D represents the number of levels of wavelet coefficients which are approximated; i.e.,

$$D = m - k^*$$

where the frame length is $M = 2^m$ and where all wavelet coefficients in or above level k^* are to be fractal coded. A frame length of $M = 2^{10} = 1024$ was chosen for the computations. The wavelet bases employed were the Haar basis, Daubechies 10-tap minimum-phase basis [12, p. 195], and Coifman's 12-tap maximally-symmetric basis [12, p. 261].

From the standpoint of MSE and computation time, there is little to choose between the Coifman and Daubechies bases, both of which yield much lower error levels than does the Haar basis (see Fig. 11). This is due to the smoothness of these bases, which matches the generally smooth nature of audio signals[7].

The MSE falls to zero when $D = 1$ since a scaling factor α is stored for each of the $M/2$ wavelet coefficients $\{a_{m-1,j}\}$, allowing exact computation of their values from the unapproximated $\{a_{m-2,j}\}$'s so that all wavelet coefficients are known precisely. This

[7] Audio signals are typically sampled at a frequency much higher than the highest frequency components in the signal. In these tests, the original signal was sampled a 22.05 kHz. Most audio signal energy resides below 5 kHz.

level of compression requires storage of $M/2$ multiresolution analysis coefficients, $M/2$ IFS scaling factors, and the locations of the domain block associated with each range block. If all data, compressed or uncompressed, is represented as 16-bit binary words (as are audio samples on a compact disc), then this represents *negative compression* in the sense that the compressed signal contains more data than the original signal. True compression using these wordlengths is not achieved unless $D \geq 2$. For $D > 3$, the MSE rises rapidly, but listening tests have shown that for $D = 3$ the smooth basis functions yield good results. Simply comparing the number of bits of information per frame of compressed and uncompressed signal, this corresponds to a compression ratio of about 3.4 if 16-bit words are used for all data storage. (The storage of the domain block locations requires 6 bits of memory since there are 2^6 possible domain blocks if $M = 2^{10}$ and $D = 3$).

The algorithm used was written in C++ and moderately optimized for speed. Encoding takes longer than decoding due to the domain block searching required, but approaches have recently been proposed which should allow this process to be made more efficient [5]. That some decoding times are greater than the corresponding encoding times reflects the fact that more time is spent in output routines when writing the reconstructed data than when writing the compressed data. (These decode times fall well below the encode times if the final data output stage is omitted.) Decoding should be possible in real time at standard audio rates if the algorithm is rewritten in lower-level (assembly) language, since both the inverse wavelet transform and iteration of the IFS are very fast.

It should be noted that the MSE figures in no way reflect the perceptual benefits obtained from the time-frequency localization of wavelet coefficient errors. As with any suitable audio compression scheme, the perceived fidelity of the reconstructed signals is superior to what the gross MSE figures might suggest.

Table 3 presents the rather disappointing results obtained in the coding and reconstruction of a 6.5 second speech signal. The algorithm has less luck finding good matches between domain and range blocks amidst the irregular waveforms and explosive noise bursts characteristic of speech, even at low degrees of compression. Frequently, the best match found for a random noise segment of the original is a domain block with a IFS scaling factor close to zero, thus effectively removing noise bursts from the signal and giving the speaker a definite lisp. Such problems indicate the necessity of an adaptive coding algorithm which can handle signals of varying character (see below).

If the size of the data frame (i.e., M) is increased, more domain blocks are searched for a fixed degree of compression, D. It is thus to be expected that better matches between domain and range blocks, and correspondingly lower error values, should be obtained. Table 4 indicates the mean square errors and computation times associated

Basis Functions	D	MSE per sample [bits^2]
Haar	4	47135
	3	11035
	2	637
Daubechies 10	4	36291
(extremal phase)	3	10571
	2	997
Coifman 12	4	33456
(symmetric)	3	9959
	2	895

Table 3: Mean Square Error (MSE) and for coding and reconstruction of speech signal with various choices of wavelet basis and degrees of compression (D).

with various values of M using Coifman 12 wavelets. Although the prospect of lower error values is enticing, the improvement with doubling the number of domain blocks is not as striking as that associated with increasing the value of k^*. Increasing the frame size and accepting the resulting increase in computation time cannot yield the sort of dramatic improvement in perceptual quality which is associated with making k^* adaptive.

One simple approach to making k^* adaptive involves coding the current frame of data with successively higher values of k^* until the total mean square coding error associated with that frame falls below some predetermined value. Results of experiments using this approach are summarized in Table 5. The results for MSE limits of 32 bits^2/sample and less are very good with few audible errors for the piano signal. (Recall that the errors are localized in time and frequency, thus promoting error masking.) In this case, encoding took about 6.8 seconds per second of 22.05 kHz input. The speech signal tolerates much higher mean square error levels due partially to its having a higher overall signal power and the relative robustness of speech intelligibility in the presence of noise. The quality of the speech remains very good for MSE limits of 4096 bits^2/sample and less. Thus both the speech and piano signals were found to be tolerant of compression by a factor of roughly 4.3, which took about 3 sec per second of 22.05 kHz input to encode and 2.0 sec per second to decode. It should be possible to render remaining artifacts in either signal inaudible and to handle signals of either sort using a single algorithm by using more sophisticated adaptive schemes which

Number of Points Per Frame, M	MSE per sample [bits^2]		Execution time per seconds of signal [sec]	
	Piano Signal	Speech Signal	Encode	Decode
256	30.2	13763	1.8	2.0
512	17.5	11819	2.1	2.0
1024	10.4	9958	2.6	2.0
2048	6.0	6087	3.7	2.1
4096	4.0	3964	5.8	2.1

Table 4: Mean Square Error (MSE) and computation time for coding and reconstruction of piano and speech signals with compression corresponding to $D = 3$ for various sizes of data frame. Coifman 12 wavelets were used.

PIANO SIGNAL		
Maximum MSE per sample [bits^2]	Average D	Average Compression Ratio
2	2.6	2.7
4	2.8	3.0
8	2.9	3.3
16	3.1	3.7
32	3.3	4.3
64	3.4	4.7
SPEECH SIGNAL		
512	2.7	2.8
1024	2.9	3.2
2048	3.1	3.7
4096	3.3	4.4
8192	3.6	5.3
16384	3.9	6.5

Table 5: Maximum allowed Mean Square Error (MSE) and degrees of compression, D, attained using an adaptive FWC algorithm with 1024-point data frames and Coifman 12 wavelets. MSE is calculated frame-by-frame, but indicated per sample. Input and output data are stored with 16 bits of precision.

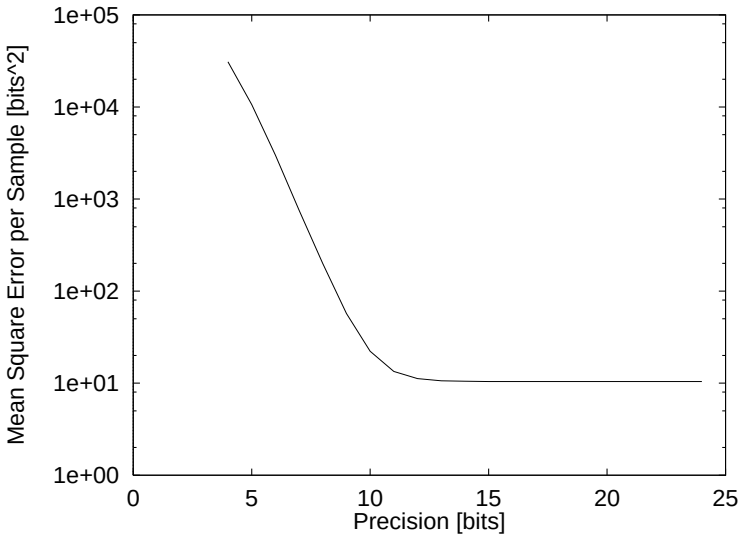


Figure 12: Mean Square Error (MSE) versus precision of compressed data for coding and reconstruction of piano signal using 1024-point frames, Coifman 12 wavelets, and degree of compression $D = 3$.

would restrict the allowable error power associated with individual wavelet coefficients relative to their magnitude.

As has been pointed out elsewhere [5, 18], the wavelet transform (like the DCT) is an approximation to a Karhunen-Loève transform and as a result the wavelet coefficients have a lower variance than the original signal and can be coded with fewer bits. This will further increase the compression ratio. Although it is difficult to predict the variance of the IFS scaling factors, some preliminary experiments with reducing the number of bits used to code them suggests that it is quite low. Fig. 12 indicates the degradation in signal quality as a decreasing number of bits are used to code both the unapproximated wavelet coefficients and IFS scaling factors. It is clear that in this case no appreciable increase in MSE due to coefficient quantization occurs unless fewer than 12 bits are used. If only 12 bits are used to store the data, this represents an additional compression factor of 1.27 relative to 16-bit data (keeping in mind that the domain block locations must be stored as before).

Finally, no attempt was made at entropy encoding the coefficients obtained. Fractal wavelet image coding typically experiences additional compression by a factor of roughly 1.2 due to the inclusion of an arithmetic coding scheme. A similar increase is expected for fractal wavelet audio coding. If the piano signal were coded using the adaptive FWC algorithm with an upper limit to the MSE of 32 bits-squared per

sample, 12 bit data, and arithmetic coding, a compression ratio of about 6.5 would be expected.

6 Conclusions

It seems that fractal wavelet compression (FWC) can be made competitive with conventional modes of audio compression. Compression by a factor greater than 3 can be achieved with generally satisfactory reconstruction using 16 bit data in the compressed signal, so a factor greater than 6 should be achievable using proper arithmetic coding and coarser quantization of the data.

The audio quality of the signals reconstructed following FWC exhibits error levels occasionally exceeding those produced by conventional schemes. It should be possible to eliminate these by using more sophisticated adaptive coding in which k^* is adjusted so that the error in any given wavelet coefficient never exceeds some predetermined value after reconstruction. An even better adaptive scheme would ensure that the error remains small relative to the coefficient itself in order to promote error masking. Ultimately the success of such schemes must be assessed using psychoacoustic models which might be incorporated into the coding scheme to ensure the perceptual transparency of compression/decompression. Such modelling should also allow higher compression ratios. Clearly, there is a great deal of work yet to be done in this area, although the inherently promising nature of fractal wavelet compression now seems evident.

7 Acknowledgements

This research was supported in part by funds from the Ontario Graduate Scholarship Program (RAW) and by the Natural Sciences and Engineering Council of Canada (NSERC) in the form of an individual Operating Grant (ERV) as well as a Collaborative Projects Grant (with C. Tricot, École Polytechnique, Université de Montréal).

References

[1] Brandenburg, Karlheinz, and Marina Bosi, "Overview of MPEG-Audio: Current and Future Standards for Low Bit-Rate Audio Coding," presented at the 99th Convention of the Audio Eng. Soc., 1995 Oct. 6–9, preprint 4130.

[2] Davis, G., "Self-Quantized Wavelet Subtrees: A Wavelet-Based Theory for Fractal Image Compression," 1995 Data Compression Conference Proceedings (preprint).

[3] Krupnik, H., D. Malah and E. Karnin, "Fractal Representation of Images via the Discrete Wavelet Transform," Proc. IEEE Conv. EE, Tel Aviv, Israel, 1995 Mar.

[4] Simon, B., "Explicit Link Between Local Fractal Transform and Multiresolution Transform," Proc. IEEE Conf. Image Processing, 1995 Oct.

[5] van de Walle, Axel, "Relating Fractal Image Compression to Transform Methods," Master's Thesis, University of Waterloo, Waterloo, Canada, 1995.

[6] van de Walle, Axel, "Merging Fractal Image Compression and Wavelet Transform Methods," Proceedings of the NATO Advanced Study Institute, Trondheim, Norway, 1995, Jul. 8-17.

[7] Forte, Bruno, and Edward R. Vrscay, "Theory of Generalized Fractal Transforms," Proceedings of the NATO Advanced Study Institute, Trondheim, Norway, 1995, Jul. 8-17; to appear in *Fractal Image Encoding and Analysis,* Fisher, Yuval, ed.

[8] Forte, Bruno, and Edward R. Vrscay, "Inverse Problem Methods for Generalized Fractal Transforms," Proceedings of the NATO Advanced Study Institute, Trondheim, Norway, 1995, Jul. 8-17; to appear in *Fractal Image Encoding and Analysis,* Fisher, Yuval, ed.

[9] Fisher, Yuval, *Fractal Image Compression, Theory and Application,* Springer-Verlag, 1995.

[10] Ullrich, Chris, "Iterated Function Systems,", unpublished report, University of Waterloo, 1995 Aug.

[11] Mallat, Stephane G., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 11, No. 7, pp. 674–693, 1989 Jul.

[12] Daubechies, Ingrid, *Ten Lectures on Wavelets,* Society of Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[13] Barnsley, M.F., and S. Demko, "Iterated Function Systems and the Global Construction of Fractals," Proc. Roy. Soc., London, vol. A399, pp. 243–275, 1985.

[14] Hutchinson, J., "Fractals and Self-Similarity," Indiana Univ. J. Math., vol. 30, pp. 713–747, 1981.

[15] Barnsley, M.F., and L.P. Hurd, *Fractal Image Compression,* A.K. Peters, Wellesley, MA, 1993.

[16] Barnsley, M.F., *Fractals Everywhere,* Academic Press, New York, NY, 1988.

[17] Jacquin, A., "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations," *IEEE Trans. Image Proc.*, Vol. 1, pp. 18–30, 1992.

[18] Jain, Anil K., *Fundamentals of Digital Image Processing,* Prentice Hall, Englewood Cliffs, NJ, 1989.