Implicit Image Models in Fractal Image Compression

Geoffrey Davis

Email: *geoff.davis@dartmouth.edu*
6211 Sudikoff Laboratory, Dartmouth College, Hanover, NH 03755

August 5, 1996

ABSTRACT

Why does fractal image compression work? What properties must an image have for fractal block coders to work well? What is the implicit image model underlying fractal image compression? The behavior of fractal block coders is clear for deterministically self-similar structures. In this paper we examine the behavior of these coders on *statistically* self-similar structures. Specifically, we examine their behavior for fractional Brownian motion, a simple texture model. Our analysis suggests that the properties necessary for fractal block coders to work well are not so dissimilar from those required by DCT and wavelet transform based coders. Fractal block coders work well for images consisting of ensembles of locally self-similar regions together with locally stationary regions with decaying power spectra, local statistical similarity, and local isotropy. Our analysis motivates a generalization of fractal block coders that leads to substantial improvements in coding performance and also illuminates some of the fundamental limitations of current fractal compression schemes.

Keywords: fractal image compression, wavelets, self-quantization of subtrees, self-similarity, fractional Brownian motion

1 INTRODUCTION

JPEG and other common image compression are based on a simple transform coder paradigm. Images are modeled as vectors drawn from a wide-sense stationary random process. Transform coders perform an approximate Karhunen-Loève (K-L) transform on an image, quantize the resulting coefficients, and entropy code them. Fractal image compression, introduced by Barnsley and Jacquin [1][10], is based on very different principles. Fractal block coders, as described by Jacquin, assume that "image redundancy can be efficiently exploited through *self-transformability* on a blockwise basis" [11]. They store images as contraction maps of which the images are approximate fixed points. Images are decoded via iterative application of these maps.

In this paper we examine the behavior of fractal block coders for textured regions, as modeled by fractal Brownian motion processes. We compare their performance to that of transform coders and find many similarities. We show that the ability of fractal block coders to efficiently exploit deterministic self-similarities in images does not carry over to the representation of statistically self-similar regions such as fractional Brownian motion texture. Our analysis does suggest ways in which the performance of fractal block coders can be greatly improved with little additional complexity.

Our analysis relies on a new wavelet-based analytical framework for block-based fractal compression schemes first introduced in [2]. Within this framework we are able to draw upon insights from the well-established transform coder paradigm in order to address the issue of why fractal block coders work.

The balance of the paper is organized as follows. Section 2 gives an overview of a basic fractal block coding scheme. In Section 3 we introduce a wavelet-based framework for analyzing fractal block coding. Using this framework and a simple texture model we make Jacquin's assumption of "self-transformability" more concrete and we discuss *why* fractal block coding works for complex image features.

2 Overview of Fractal Block Coding

In this section we describe a generic fractal block coding scheme based on those in [11][6], and we provide some heuristic motivation for the scheme. A more complete overview of fractal coding techniques can be found in [5][12].

Transform coders are designed to take advantage of very simple structure in images, namely that values of pixels that are close together are correlated. Fractal compression is motivated by the observation that important image features, including straight edges and constant regions, are invariant under rescaling. Constant gradients are covariant under rescaling, i.e. rescaling changes the gradient by a constant factor. Scale invariance (and covariance) presents an additional type of structure for an image coder to exploit.

Fractal compression takes advantage of this local scale invariance by using coarse-scale image features to quantize fine-scale features. Fractal block coders perform a vector quantization (VQ) of image blocks. The vector codebook is constructed from locally averaged and subsampled isometries of larger blocks from the image. This codebook is effective for coding constant regions and straight edges due to the scale invariance of these features. The vector quantization is done in such a way that it determines a contraction map from the plane to itself of which the image to be coded is an approximate fixed point. Images are stored by saving the parameters of this map and decoded by iteratively applying the map to find its fixed point. An advantage of fractal block coding over VQ that it does not require separate storage of a fixed vector codebook.

The ability of fractal block coders to represent straight edges, constant regions, and constant gradients efficiently is important, as transform coders fail to take advantage of these types of spatial structures. Indeed, recent wavelet transform based techniques that have achieved particularly good compression results have done so by augmenting scalar quantization of transform coefficients with a zerotree vector that is used to efficiently encode locally constant regions [16].

For fractal block coders to be effective, images must be composed of features at fine scales that are also present at coarser scales up to a rigid motion and an affine transform of intensities. This is the "self-transformability" assumption described by [11]. It is clear that this assumption holds for images composed of isolated straight lines and constant regions, since these features are self-similar. That it should hold when more complex features are present is much less obvious. In section 4 we use a rudimentary texture model and our wavelet framework to provide a more detailed characterization of "self-transformable" images.

2.1 Image Encoding

We now describe a simple fractal block coding scheme based on those in [11][6]. Let $\mathcal{I}$ be a $2^N \times 2^N$ pixel grayscale image. Let $\mathbf{B}^J_{K,L}$ be the linear "get-block" operator which when applied to $\mathcal{I}$ extracts the $2^J \times 2^J$ subblock with lower left corner at (K, L). The adjoint of this operator, $(\mathbf{B}^J_{K,L})^*$, is a "put-block" operator that

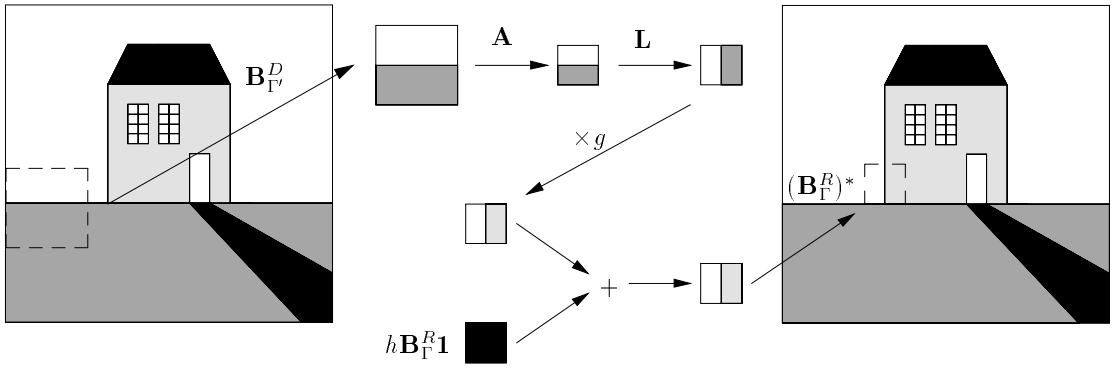

Figure 1: We quantize the small range block $\mathbf{B}^R_\Gamma\mathcal{I}$ on the right using the codebook vector $g\mathbf{L}\mathbf{A}\mathbf{B}^D_{\Gamma'}\mathcal{I} + h\mathbf{B}^R_\Gamma\mathbf{1}$ obtained from the larger domain block on the left. $\mathbf{A}$ averages and subsamples the block, $\mathbf{L}$ rotates it, multiplication by the gain g modifies the contrast, and the addition of the offset $h\mathbf{B}^R_\Gamma\mathbf{1}$ adjusts the block DC component.

inserts a $2^J \times 2^J$ image block into a $2^N \times 2^N$ all-zero image such that the lower left corner of the inserted block is at (K, L). We will use capital letters to denote block coordinates and lower case to denote individual pixel coordinates. We use a capital Greek multi-index, usually Γ, to abbreviate the block coordinates K, L and a lower-case Greek multi-index to abbreviate pixel coordinates within blocks.

We partition $\mathcal{I}$ into a set of non-overlapping $2^R \times 2^R$ *range blocks*. The goal of the compression scheme is to approximate each range block with a block from a codebook constructed from a set of $2^D \times 2^D$ *domain blocks*, where $0 < R < D \leq N$. Forming this approximation entails the construction of a contraction map from the image to itself (from the domain blocks to the range blocks) of which the image is an approximate fixed point. We store the image by storing the parameters of this map, and recover the image by iterating the map to its fixed point. Iterated function system theory motivates this general approach to storing images, but gives little guidance on questions of implementation. The basic form of the block coder described below is the result of considerable empirical work. In Section 3 we see that this block-based coder arises naturally in a wavelet framework.

The range block partition is a disjoint partition of the image consisting of the blocks $\{\mathbf{B}^R_{K,L}\mathcal{I} | (K, L) \in \mathcal{R}\}$. Here $\mathcal{R} = \{(2^R m, 2^R n) | 0 \leq m, n < 2^{N-R}\}$. The domain blocks from which the codebook is constructed are drawn from the *domain pool*, the set $\{\mathbf{B}^D_{K,L}\mathcal{I} : (K, L) \in \mathcal{D}\}$. A variety of domain pools are used in the literature. Here we will focus on a disjoint tiling of $\mathcal{I}$, the *disjoint domain pool*, $\mathcal{D} = \{(2^D m, 2^D n) | 0 \leq m, n < 2^{N-D}\}$. For discussion of more general domain pools, see [3].

Two basic operators are used for codebook construction. The "average-and-subsample" operator $\mathbf{A}$ maps a $2^J \times 2^J$ image block to a $2^{J-1} \times 2^{J-1}$ block by averaging each pixel in $\mathbf{B}^J_\Gamma$ with its neighbors and then subsampling. We define $(\mathbf{A}\mathbf{B}^J_\Gamma\mathcal{I})(k, l) = \frac{1}{4}[(\mathbf{B}^J_\Gamma\mathcal{I})(2k, 2l) + (\mathbf{B}^J_\Gamma\mathcal{I})(2k+1, 2l) + (\mathbf{B}^J_\Gamma\mathcal{I})(2k, 2l+1) + (\mathbf{B}^J_\Gamma\mathcal{I})(2k+1, 2l+1)]$ where $\mathbf{B}^J_\Gamma\mathcal{I}(k, l)$ is the pixel at coordinates (k, l) within the subblock $\mathbf{B}^J_\Gamma\mathcal{I}$. A second operator is the symmetry operator $\mathbf{L}_k$, $1 \leq k \leq 8$, which maps a square block to one of the 8 isometries obtained from compositions of reflections and 90 degree rotations.

Range block approximation is similar to shape-gain vector quantization[9]. Range blocks are quantized to a linear combination of an element from the codebook and a constant block. The codebook used for quantizing range blocks consists of averaged and subsampled isometries of domain blocks, the set $\mathcal{C} = \{\mathbf{L}_k\mathbf{A}^{D-R}\mathbf{B}^D_\Gamma\mathcal{I} : \Gamma \in \mathcal{D}, 0 \leq k \leq 8\}$. Here $\mathbf{A}^{D-R}$ denotes the operator $\mathbf{A}$ applied $D - R$ times. The contrast of the codewords in $\mathcal{C}$ is adjusted by a gain factor g, and the DC component is adjusted by adding a subblock of the $2^N \times 2^N$ matrix of ones, $\mathbf{1}$, multiplied by an offset factor h. For each range block $\mathbf{B}^R_\Gamma\mathcal{I}$ we have

$$\mathbf{B}^R_\Gamma\mathcal{I} \approx g_\Gamma\mathbf{L}_{P(\Gamma)}\mathbf{A}^{D-R}\mathbf{B}^D_{\Pi(\Gamma)}\mathcal{I} + h_\Gamma\mathbf{B}^R_\Gamma\mathbf{1}. \tag{1}$$

Here $\Pi : \mathcal{R} \to \mathcal{D}$ assigns an element from the domain pool to each range element and $P : \mathcal{R} \to \{1 \ldots 8\}$ assigns each range element a symmetry operator index. Ideally the parameters g, h, Π, and P should be chosen so that they minimize the error in the decoded image. The quantization process is complicated by the fact that the codebook used by the decoder is different from that used by the encoder, since the decoder doesn't have access to the original domain blocks. Hence errors made in quantizing range blocks are compounded because they affect the decoder codebook. These additional effects of quantization errors have proven difficult to estimate, so in practice g, h, Π, and P are chosen to minimize the l^2 quantization error.

2.2 Decoding Fractal Coded Images

The approximations for the range blocks (1) determine a constraint on the image $\mathcal{I}$ of the form $\mathcal{I} \approx \mathbf{G}\mathcal{I} + \mathcal{H}$. Expanding $\mathcal{I}$ as a sum of range blocks we obtain

$$\begin{aligned} \mathcal{I} &= \sum_{\Gamma \in \mathcal{R}} (\mathbf{B}_\Gamma^R)^* \mathbf{B}_\Gamma^R \mathcal{I} \\ &\approx \sum_{\Gamma \in \mathcal{R}} g_\Gamma (\mathbf{B}_\Gamma^R)^* \mathbf{L}_{P(\Gamma)} \mathbf{A}^{D-R} \mathbf{B}_{\Pi(\Gamma)}^D \mathcal{I} + \sum_{\Gamma \in \mathcal{R}} h_\Gamma (\mathbf{B}_\Gamma^R)^* \mathbf{B}_\Gamma^R \mathbf{1} \\ &= \mathbf{G}\mathcal{I} + \mathcal{H}. \end{aligned}$$

Provided the matrix $\mathbf{I} - \mathbf{G}$ is nonsingular, there is a unique fixed point solution $\mathcal{I}_{fp}$ satisfying

$$\mathcal{I}_{fp} = \mathbf{G}\mathcal{I}_{fp} + \mathcal{H} \tag{2}$$

given by $\mathcal{I}_{fp} = (\mathbf{I} - \mathbf{G})^{-1}\mathcal{H}$. Because $\mathbf{G}$ is a $2^{2N} \times 2^{2N}$ matrix, inverting $\mathbf{I} - \mathbf{G}$ directly is an inordinately difficult task. If (and only if) the eigenvalues of $\mathbf{G}$ are all less than 1 in magnitude, we can find the fixed point solution $\mathcal{I}_{fp}$ by iteratively applying (2) to an arbitrary image $\mathcal{I}_0$. Decoding of fractal coded images proceeds by forming the sequence

$$\mathcal{I}_n = \mathbf{G}\mathcal{I}_{n-1} + \mathcal{H} = \mathbf{G}^n \mathcal{I}_0 + \sum_{k=0}^{n-1} \mathbf{G}^k \mathcal{H}. \tag{3}$$

In general the image to be coded $\mathcal{I}$ is not an exact fixed point of (2), i.e. $\mathcal{I} = \mathbf{G}\mathcal{I} + \mathcal{H} + \mathcal{E}$ where $\mathcal{E}$ is an error image. We can bound the error in the reconstructed image $\|\mathcal{I}_{fp} - \mathcal{I}\|$ in terms of $\|\mathcal{E}\|$; this bound is known as the *collage theorem* bound. Although the bound is a weak one, small errors $\|\mathcal{E}\|$ are found in practice to yield small reconstruction errors $\|\mathcal{I}_{fp} - \mathcal{I}\|$. Extensions of the collage theorem bound and a more detailed discussion of reconstruction errors may be found in [3].

3 A Wavelet Framework

3.1 Notation

The wavelet transform is a natural tool for analyzing fractal block coders since wavelet bases possess the same type of dyadic self-similarity that fractal coders seek to exploit. In particular, the Haar wavelet basis possesses a regular block structure that is aligned with the range block partition of the image. We show below that the maps generated by fractal block coders reduce to a simple set of equations in the wavelet transform domain.

Separable 2-D biorthogonal wavelet bases consist of translates and dyadic scalings of a set of oriented wavelets $\psi_{LH}(x, y)$, $\psi_{HL}(x, y)$, and $\psi_{HH}(x, y)$ together with translates of a scaling function $\phi(x, y)$. We will use the

subscript ω to represent one of the three orientations in $\Omega = \{LH, HL, HH\}$. We will limit our attention to symmetrical (or antisymmetrical) bases. The discrete wavelet transform of a $2^N \times 2^N$ image $\mathcal{I}$ expands the image into a linear combination of the basis functions in the set $\mathcal{W}_J = \{\phi^J_{\omega,k,l} | \omega \in \Omega, 0 \leq k,l < 2^J\} \cup \{\psi^j_{k,l} | J \leq j < N, 0 \leq k,l < 2^j\}$. We will use a single lower-case Greek multi-index, usually γ, to abbreviate the orientation and translation subscripts of ϕ and ψ. The coefficients for the basis functions $\phi^j_{k,l}$ and $\psi^j_{\omega,k,l}$ are given by $\langle \tilde{\phi}^j_{k,l}, \mathcal{I} \rangle$ and $\langle \tilde{\psi}^j_{\omega,k,l}, \mathcal{I} \rangle$, respectively, where $\tilde{\phi}^j_{k,l}$ and $\tilde{\psi}^j_{\omega,k,l}$ are dual scaling functions and wavelets.

An important property of wavelet basis expansions, especially Haar expansions, is that they preserve the spatial localization of image features. For example, the coefficient of the Haar scaling function $\phi^J_{k,l}$ is proportional to the average value of an image in the $2^J \times 2^J$ block of pixels with lower left corner at $2^J k, 2^J l$. The wavelet coefficients associated with this region are organized into three quadtrees. We call this set of trees a *wavelet subtree*; coefficients forming such a subtree are shaded in each of the transforms in Figure 2. At the root of a wavelet subtree are the coefficients of the wavelets $\psi^J_{\omega,k,l}$, where $\omega \in \Omega$. These coefficients correspond to the block's coarse-scale information. Each wavelet coefficient $\langle \tilde{\psi}^j_{\omega,k,l}, \mathcal{I} \rangle$ in the tree has four children that correspond to the same spatial location and the same orientation. The children consist of the coefficients of the wavelets of the next finer scale, $\psi^{j+1}_{\omega,2k,2l}$, $\psi^{j+1}_{\omega,2k+1,2l}$, $\psi^{j+1}_{\omega,2k,2l+1}$, and $\psi^{j+1}_{\omega,2k+1,2l+1}$. A wavelet subtree consists of the coefficients of the roots, together with all of their descendents in all three orientations. The scaling function $\phi^J_{k,l}$ is localized in the same region as the subtree with roots given by $\psi^J_{\omega,k,l}$, and we refer to this $\phi^J_{k,l}$ as the scaling function associated with the subtree.

3.2 A Wavelet Transform Domain Analog of Fractal Block Coding

We now describe a wavelet transform domain analog of fractal block coding first introduced in [2]. Related schemes have been described independently in [13] and [19]. The wavelet-based fractal coder of [15] is similar in spirit but differs fundamentally in its goals and implementation.

Fractal block coders approximate a set of $2^R \times 2^R$ range blocks using a set of $2^D \times 2^D$ domain blocks. The wavelet analog of an image block, a set of pixels associated with a small region in space, is a wavelet subtree together with its associated scaling function coefficient. We define a linear "get-subtree" operator $\mathbf{S}^J_{K,L} : \mathbb{R}^{2^{2N}} \rightarrow \mathbb{R}^{2^{2(N-J)}-1}$ which extracts from an image the subtree whose root level consists of the coefficients of $\psi^J_{\omega,K,L}$. The adjoint of $\mathbf{S}^J_{K,L}$ is a "put-subtree" operator which maps a subtree to the subtree with root coefficients corresponding to $\psi^J_{\omega,K,L}$ in an all-zero image. For the Haar basis, subblocks and their corresponding subtrees and associated scaling function coefficients contain identical information, i.e. the transform of a range block $\mathbf{B}^R_\Gamma \mathcal{I}$ yields the coefficients of subtree $\mathbf{S}^{N-R}_\Gamma \mathcal{I}$ and the scaling function coefficient $\langle \tilde{\phi}^{N-R}_\Gamma, \mathcal{I} \rangle$. For the remainder of this section we will take our wavelet basis to be the Haar basis. The actions of the get-subtree and put-subtree operators are illustrated in Figure 2.

The linear operators used in fractal block coding have simple behavior in the transform domain. We first consider the wavelet analog $\widehat{\mathbf{A}}$ of the average-and-subsample operator $\mathbf{A}$. Averaging and subsampling the finest-scale Haar wavelets sets them to 0. The local averaging has no effect on coarser scale Haar wavelets, and subsampling ψ^j_γ yields the Haar wavelet at the next finer scale, ψ^{j+1}_γ, multiplied by $\frac{1}{2}$. Similarly, averaging and subsampling the scaling function ϕ^j_γ yields $\frac{1}{2}\phi^{j+1}_\gamma$ for $j < N-1$ and 0 for $j = N-1$. The action of the averaging and subsampling operator thus consists of a shifting of coefficients from coarse-scale to fine, a multiplication by $\frac{1}{2}$, and a truncation of the finest-scale coefficients. The operator $\widehat{\mathbf{A}}$ prunes the leaves of a subtree and shifts all remaining coefficients to the next finer scale. The action of $\widehat{\mathbf{A}}$ is illustrated in Figure 2.

For symmetrical wavelets, horizontal/vertical block reflections correspond to a horizontal/vertical reflection of wavelet coefficients within each scale of a subtree. Similarly, 90 degree block rotations correspond to 90 degree rotations of wavelet coefficients within each scale and a switching of the ψ_{LH} coefficients with ψ_{HL} coefficients.

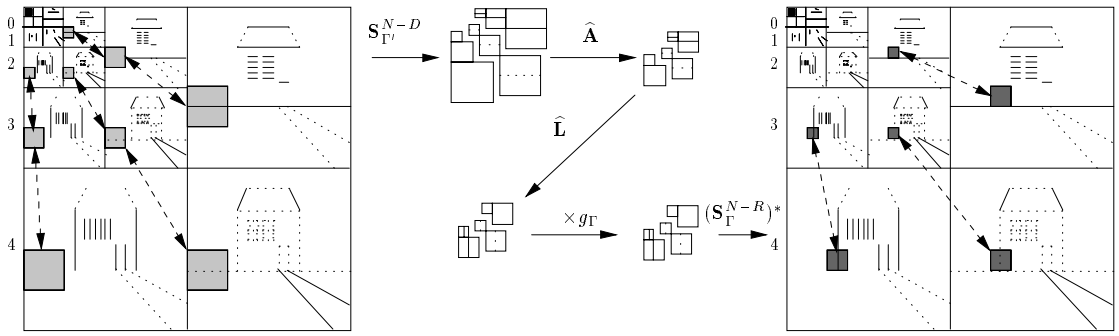

Figure 2: We approximate the darkly shaded range subtree $\mathbf{S}_\Gamma^{N-R}\mathcal{I}$ using the codebook element $g_\Gamma\widehat{\mathbf{L}}\widehat{\mathbf{A}}\mathbf{S}_{\Gamma'}^{N-D}\mathcal{I}$ derived from the lightly shaded domain subtree $\mathbf{S}_{\Gamma'}^{N-D}\mathcal{I}$. $\widehat{\mathbf{A}}$ truncates the finest scale coefficients of the domain subtree and multiplies the coefficients by $\frac{1}{2}$, and $\widehat{\mathbf{L}}$ rotates it. When storing this image we save the coarse-scale wavelet coefficients in subbands of scale 2 and lower, and we save the encodings of all subtrees with roots in scale subband 3.

Hence the wavelet analogs $\widehat{\mathbf{L}}_k$ of the block symmetry operators $\mathbf{L}_k$ permute wavelet coefficients within each scale. Figure 2 illustrates the action of a symmetry operator on a subtree. Note that the Haar basis is the only orthogonal basis we consider here, since it is the only compactly supported symmetrical wavelet basis[20]. When we generalize to non-Haar bases, we must use biorthogonal bases to obtain symmetry and compact support.

The approximation (1) leads to a similar relation for subtrees in the Haar wavelet transform domain,

$$\mathbf{S}_\Gamma^{N-R}\mathcal{I} \approx g_\Gamma\widehat{\mathbf{L}}_{P(\Gamma)}\widehat{\mathbf{A}}^{D-R}\mathbf{S}_{\Pi(\Gamma)}^{N-D}\mathcal{I}. \tag{4}$$

We refer to this form quantization of subtrees using other subtrees as the *self-quantization* of $\mathbf{S}_\Gamma^{N-R}$. The offset terms h_Γ from (1) affect only the scaling function coefficients because the left hand side of (4) is orthogonal to the subblocks of $\mathbf{1}$. Breaking up the subtrees into their constituent wavelet coefficients, we obtain a system of equations for the coefficients of the ψ_γ^j in $\mathbf{S}_\Gamma^{N-R}\mathcal{I}$,

$$\langle\tilde{\psi}_\gamma^j,\mathcal{I}\rangle \approx \frac{g_\Gamma}{2^{D-R}}\langle\tilde{\psi}_{\gamma'}^{j-(D-R)},\mathcal{I}\rangle = \frac{g_\Gamma}{2^{D-R}}\langle\mathbf{T}(\tilde{\psi}_\gamma^j),\mathcal{I}\rangle. \tag{5}$$

Here $\mathbf{T}$ is the map induced by the domain block selection followed by averaging, subsampling, and rotating. We obtain a similar relation for the scaling function coefficients,

$$\begin{aligned}\langle\tilde{\phi}_\Gamma^{N-R},\mathcal{I}\rangle &\approx \frac{g_\Gamma}{2^{D-R}}\langle\tilde{\phi}_{\Pi(\Gamma)}^{N-D},\mathcal{I}\rangle + h_\Gamma \\ &= \frac{g_\Gamma}{2^{D-R}}\langle\mathbf{T}(\tilde{\phi}_\Gamma^{N-R}),\mathcal{I}\rangle + h_\Gamma\end{aligned} \tag{6}$$

From the system (5) and (6) we see that, roughly speaking, the fractal block quantization process constructs a map from coarse-scale wavelet coefficients to fine. It is important to note that the function $\mathbf{T}$ in (5) and (6) does *not* necessarily map $\mathcal{W}_{N-D}$ to $\mathcal{W}_{N-D}$ for domain pools other than the disjoint domain pool. The reason is that translation of domain blocks by distances smaller than 2^D leads to non-integral translates of the wavelets in their corresponding subtrees. This issue is addressed in more detail in [3].

4 Fractal Block Coding of Textures

In section 2 we motivated the codebook used by fractal block coders by emphasizing the scale-invariance (covariance) of isolated straight edges, constant regions, and constant gradients. More complex image structures

lack this deterministic self-similarity, however. How can we explain fractal block coders' ability to compress images containing complex structures? Why should the codebook constructed from the domain blocks be an effective one for regions that are not self-similar?

4.1 Fractional Brownian Motion

Although complex image features such as textures are not in general self-similar, they do tend to possess characteristics that can be exploited by fractal coders. Because of the physical structure underlying natural images, pixel values are far from independent, especially for pixels that are close together. The continuity of the physical objects that give rise to natural images produces a local pixel covariance structure with correlations that decrease with pixel separation. Pentland [14] proposes a fractional Brownian motion (fBm) texture model that mimics this type of covariance structure. Fractional Brownian motion models have proved quite successful for generating synthetic landscapes and clouds in computer graphics applications [21] which lends additional support to using fBm as a model for complex image structures. An alternative but closely related image model that is used to motivate the use of the discrete cosine transform in JPEG is a first order Gauss-Markov model with high correlation. The form of this model is $x[n] = ax[n-1] + u[n]$ where $x[n]$ and $x[n-1]$ are adjacent pixels and the $u[n]$ are i.i.d. Gaussian random variables. When $a = 1$ this process is in fact ordinary Brownian motion.

A random process is $x(t)$ is said to be *wide-sense statistically self-similar* with parameter H if its second order statistics scale according to the relations

$$E[x(t)] = a^{-H} E[x(at)] \tag{7}$$

$$E[x(s)x(t)] = a^{-2H} E[x(as)x(at)]. \tag{8}$$

Fractional Brownian motion is a self-similar, zero-mean Gaussian random process where $0 < H < 1$. When the Hurst exponent $H = \frac{1}{2}$ we have ordinary Brownian motion. A key property of fBm processes is that their measured power spectra decay as $|\omega|^{-\gamma}$, where $\gamma = 2H + 1$.

Natural images observe a similar power-law spectral decay. Measurements of spectral decay in natural images show decay rates between ω^{-2} and ω^{-3}. Field[4] hypothesizes that image contrast is roughly invariant across scale, which implies that image luminance power spectra decay like ω^{-2}. His measurements of the spectra of 85 photographs of natural scenes show an overall decay rate of roughly $\omega^{-2.2}$. Voss[21] claims that shapes with fractal dimension of 0.2 to 0.3 greater than their Euclidean dimension are particularly common in nature; fractional Brownian motions exhibiting such Hausdorff-Besicovitch dimensions possess spectral decay rates of $\omega^{-2.4}$ to $\omega^{-2.35}$, respectively.

Flandrin [7] has shown that the wavelet transform coefficients of a fractional Brownian motion process are stationary sequences with a self-similar covariance structure. This means that the codebook constructed from domain subtrees will possess the same statistics as the set of range subtrees. Hence for fBm textured regions, the quantization in (4) involves matching two random vectors drawn from sources with the same statistics.

Obtaining a close match between pairs of high dimensional random vectors is an extremely difficult task unless the distribution of these vectors is such that the vectors are highly clustered. Fractal coders can avoid this difficult high-dimensional problem to some extent by adapting the size of the range blocks. Quantization becomes much easier as the range block size decreases since the dimension of the problem is reduced. In the extreme case, single pixel range blocks are trivial to quantize using (1) since setting $g_\Gamma = 0$ turns (1) into a scalar quantization. Adaptation alone does not explain the performance of fractal block coders in complex regions, however. In numerical experiments we find that although the quantized range blocks/subtrees tend to be smaller in textured regions, they are still considerably larger than the trivial case, often containing 15 coefficients or more Finding a close match between a random 15-dimensional range subtree and a random vector from our relatively small codebook is extremely unlikely unless both range and domain subtrees are tightly clustered.

Why should such clustering occur in natural images? Understanding this is the key to explaining the performance of fractal block coders in textured regions. The answer lies in the fact that the Haar transform acts as an approximate Karhunen-Loève (K-L) transform for ordinary Brownian motion, concentrating the energy in the coarse-scale coefficients. One can show that the variances of the Haar wavelet coefficients decay in scale as 2^{-2j} [22]. The result is that for Brownian motion processes, the Haar subtrees are clustered around the low-dimensional subspace consisting of subtrees with all-zero fine-scale coefficients. Moreover, because of the statistical self-similarity of Brownian motion, the covariance structures of these clusters are the same (up to a constant factor) for range and domain subtrees. Matching random subtrees that lie near this low-dimensional subspace is a *much* easier problem than matching arbitrary random subtrees. The problem is made easier still by the fact that the human visual system is less sensitive to high frequency errors than to low.

We emphasize that statistical self-similarity alone is not enough to enable fractal coders to perform effectively. For example, we can construct a process that is statistically self-similar over the range of frequencies accessible by our sampling for which wavelet coefficient variances *increase* for fine scales. Although this process is self-similar, the clustering effects described above that enable fractal block coding to function effectively have a negligible impact here. Thus, our texture model suggests that fractal block coders owe much of their performance in complex regions to the decaying power spectra of these regions. Additional characteristics that contribute to subtree clustering and therefore to fractal coder performance include statistical self-similarity, local stationarity (since our codebook contains translates of domain subtrees), and local isotropy (since the codebook contains isometries of domain subtrees).

4.2 Improving Fractal Block Coders

The Haar transform is not very effective at decorrelating fBm processes with rates of spectral decay corresponding more closely to observed values. When the decay is $O(\omega^{-\beta})$ for $2 < \beta < 3$, the autocorrelation function for a coefficient lag of n decays as $|n|^{\beta-3}$ for n large [7]. Tewfik and Kim[18] have shown that for such fBm's, wavelet transforms for bases with 2 or more vanishing moments yield much better approximations to the K-L transform than does the Haar basis. Our texture model therefore motivates the use of bases with additional vanishing moments. Although our development in 3 focused on the Haar basis, there is no reason we cannot replace this basis with a different symmetric, biorthogonal wavelet basis. Indeed, numerical experiments in [3] show dramatic improvements in coder performance when using wavelets with higher numbers of vanishing moments than the Haar basis.

An important observation is that quantization in our texture model entails matching pairs of random vectors. Although the clustering of subtrees makes reasonably accurate quantization of high dimensional subtrees possible, the codebook vectors are unlikely to be distributed in an optimal fashion. Analysis by Gersho [8] of asymptotic high resolution vector quantization with an entropy constraint reveals that for high dimensional VQ's the optimal quantizer is very close to a uniform quantizer.

The codebook generated by fractal block coders for fractional Brownian motion textured regions is far from uniform. On the contrary, code vectors are tightly clustered around the origin. Although having a free gain parameter leaves open the possibility of spreading the codebook out, the restrictions on the gain parameter imposed by convergence requirements for general domain pools limit its effect. A comparison with the results of [8] suggests that codewords for our fractal block scheme will be unnecessarily densely distributed in high probability regions of the space of subtrees and too sparsely distributed in low probability regions. This conjecture is borne out in our numerical experiments, described in [3] in which a tight clustering of codewords is found around the all-zero subtree, leading to an inefficient codebook.

A second way, then, in which fractal block coders can be improved is by modifying the codebook to eliminate codeword duplication arising from the overly dense clustering of codewords around all-zero subtrees. Signes [17] has suggested optimizing fractal codebooks by pruning them before quantizing. Such a strategy can easily be

applied to our self-quantization scheme

5 CONCLUSION

Our analysis of the fBm texture model suggests that the central "self-transformability" assumption used to motivate fractal block coders may be restated as the assumption that images consist of ensembles of regions that are locally scale-invariant together with locally stationary regions with decaying power spectra, local statistical similarity, and local isotropy. Our texture model motivates the use of wavelets with more vanishing moments than the Haar basis.

6 ACKNOWLEDGMENTS

This research was supported in part by a National Science Foundation Mathematical Sciences Postdoctoral Research Fellowship, DMS-9627391, and ARPA as administered by the AFOSR under contracts AFOSR-90-0292 and DOD F4960-93-1-0567.

7 REFERENCES

[1] M. F. Barnsley and A. Jacquin. Application of recurrent iterated function systems to images. *Proc. SPIE*, 1001:122–131, 1988.

[2] G. M. Davis. Self-quantization of wavelet subtrees: a wavelet-based theory of fractal image compression. In J. A. Storer and M. Cohn, editors, *Proc. Data Compression Conference, Snowbird, Utah*, pages 232–241. IEEE Computer Society, Mar. 1995.

[3] G. M. Davis. A wavelet-based analysis of fractal image compression. *Preprint*, See http://www.cs.dartmouth.edu/∼gdavis, 1996.

[4] D. J. Field. Scale-invariance and self-similar 'wavelet' transforms: an analysis of natural scenes and mammalian visual systems. In M. Farge, J. C. R. Hunt, and J. C. Vassilicos, editors, *Wavelets, Fractals, and Fourier Transforms*. Oxford University Press, Oxford, 1993.

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

[6] Y. Fisher. Fractal image compression with quadtrees. In Y. Fisher, editor, *Fractal Compression: Theory and Application to Digital Images*. Springer Verlag, New York, 1994.

[7] P. Flandrin. Wavelet analysis and synthesis of fractional Brownian motion. *IEEE Transactions on Information Theory*, 38(2):910–917, Mar. 1992.

[8] A. Gersho. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory*, IT-25(4):373–380, July 1979.

[9] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic, Boston, 1992.

[10] A. Jacquin. Fractal image coding based on a theory of iterated contractive image transformations. In *Proc. SPIE Visual Comm. and Image Proc.*, pages 227–239, 1990.

[11] A. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Trans. Image Proc.*, 1(1):18–30, Jan. 1992.

[12] A. Jacquin. Fractal image coding: a review. *Proc. IEEE*, 81(10):1451–1465, Oct. 1993.

[13] H. Krupnik, D. Malah, and E. Karnin. Fractal representation of images via the discrete wavelet transform. In *IEEE 18th Conv. of EE in Israel*, Tel-Aviv, Mar. 1995.

[14] A. P. Pentland. Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:661–673, 1984.

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

[16] J. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, Dec. 1993.

[17] J. Signes. Geometrical interpretation of IFS based image coding. In Y. Fisher, editor, *Fractal Image Coding and Analysis: a NATO ASI Series Book*. Springer Verlag, New York, 1996.

[18] A. H. Tewfik and M. Kim. Correlation structure of the discrete wavelet coefficients of fractional Brownian motion. *IEEE Transactions on Information Theory*, 38(2):904–909, Mar. 1992.

[19] A. van de Walle. Merging fractal image compression and wavelet transform methods. In Y. Fisher, editor, *Fractal Image Coding and Analysis: a NATO ASI Series Book*. Springer Verlag, New York, 1996.

[20] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[21] R. F. Voss. Fractals in nature: from characterization to simulation. In H.-O. Peitgen and D. Saupe, editors, *The Science of Fractal Images*. Springer-Verlag, New York, 1988.

[22] G. W. Wornell. *Signal Processing with Fractals: A Wavelet-Based Approach*. Prentice Hall, Englewood Cliffs, NJ, 1996.