Adaptive self-quantization of wavelet subtrees: A wavelet-based theory of fractal image compression *

Geoffrey Davis

Email: *geoff.davis@dartmouth.edu*
Department of Mathematics, Bradley Hall
Dartmouth College, Hanover, NH 03755

ABSTRACT

Fractal image compression was one of the earliest compression schemes to take advantage of image redundancy in scale. The theory of iterated function systems motivates a broad class of fractal schemes but does not give much guidance for implementation. Fractal compression schemes do not fit into the standard transform coder paradigm and have proven difficult to analyze. We introduce a wavelet-based framework for analyzing fractal block coders which simplifies these schemes considerably. Using this framework we find that fractal block coders are Haar wavelet subtree quantization schemes, and we thereby place fractal schemes in the context of conventional transform coders. We show that the central mechanism of fractal schemes is an extrapolation of fine-scale Haar wavelet coefficients from coarse-scale coefficients. We use this insight to derive a wavelet-based analog of fractal compression, the self-quantization of subtrees (SQS) scheme. We obtain a simple SQS decoder convergence proof and a fast SQS decoding algorithm which simplify and generalize existing fractal compression results. We describe an adaptive SQS compression scheme which outperforms the best fractal schemes in the literature by roughly 1 dB in PSNR across a broad range of compression ratios and which has performance comparable to some of the best conventional wavelet subtree quantization schemes.

Keywords: image compression, fractal compression, wavelets, vector quantization, collage theorem

1 INTRODUCTION

Lossy image compression schemes such as JPEG achieve most of their data reduction by eliminating spatial redundancy within images. Recent wavelet-based techniques which have achieved particularly high quality rate-distortion results have done so by taking advantage of additional redundancy in *scale* [15, 17]. Encoding in these wavelet schemes, as well as in JPEG, follows a standard, well-understood paradigm. First an invertible transform is performed on an image; then the transform coefficients are quantized, entropy-coded, and stored.

Fractal image compression [9, 6] also takes advantage of redundancy in scale, but its operating principles are very different from those of transform coders. Fractal compression is related to vector quantization, but it uses a self-referential vector codebook, drawn from the image itself, rather than a fixed codebook. Images are not stored as a set of quantized transform coefficients, but instead as fixed points of maps on the plane. The theory

* To appear at *SPIE Conference on Mathematical Imaging: Wavelet Applications in Signal and Image Processing*, San Diego, July 1995. Updated versions may be obtained from *http://www.cs.dartmouth.edu/~gdavis*

of iterated function systems (IFS) motivates a broad class fractal compression schemes but does not give much insight as to why fractal schemes work well. IFS theory also gives little guidance as to how efficient schemes should be implemented. Basic components of fractal schemes, such as decoder convergence properties, methods of estimating quantization error, and bit allocation methods are poorly understood.

In this paper we introduce a wavelet-based framework for analyzing fractal block coders. The wavelet framework simplifies the analysis of fractal compression considerably and, more importantly, gives a clear picture of *why* such schemes work. We show that existing fractal block coding schemes function essentially by extrapolating Haar wavelet coefficients across scales. Our analysis gives insight into a number of important implementation issues which are not well-understood, including bit allocation methods, error estimation, quantization vector search strategies, and super-resolution of images. Using the insights from our analysis we derive a wavelet-based analog of fractal compression, the self-quantization of subtrees (SQS) scheme. In our experiments SQS-compressed images have PSNR's of roughly 1 dB higher than the best existing fractal compression schemes over a broad range of compression ratios. The overall performance of our SQS coder is comparable to that of Shapiro's embedded zerotree wavelet coder [15]. SQS decoding is fast, requiring $O(N)$ operations for an N-pixel images, unlike the asymptotically converging standard fractal decoders.

The balance of the paper is organized as follows. In the next section we give an overview of fractal block coding schemes. In section 3 we show that fractal block coders are Haar wavelet subtree quantization schemes. We discuss the convergence properties of fractal decoders and show that these properties are closely related to wavelet reproducing kernels. In section 4 we introduce a modification of fractal coding which enables us to obtain a simple decoder convergence proof and a fast decoding algorithm. We generalize this modified coder to the self-quantization of subtrees scheme. Our analysis of the decoder shows that the central component of fractal coding schemes is an extrapolation of fine-scale information from coarse-scale. Here scale is in the sense of the detail spaces of a multiresolution analyses [14]. Fractal decoding is thus a cascading of information from coarse scales to fine. We see that the convergence problems which affect conventional fractal compression schemes are due to dependencies of fine-scale coefficients on finer-scale ones. In section 5 we describe an adaptive self-quantized subtree compression scheme which is a generalization of a wavelet compression scheme of [17], and in section 6 we present the results of this compression scheme.

2 FRACTAL IMAGE COMPRESSION

Fractal image compression techniques, introduced by Barnsley and Jacquin [2], have proved very successful for compressing images at low bitrates. An overview of these techniques can be found in [8]. The motivation for fractal image compression is that many basic features in images are invariant under rescaling. Constant regions in images are invariant under local averaging and subsampling, as are straight edges. Fractal compression takes advantage of this redundancy by using coarse-scale image features to quantize fine-scale features. In this work we will focus on fractal block coders.

Fractal block coders perform a vector quantization of image blocks. The codebook consists of larger blocks from the image which are locally averaged and subsampled. This codebook is very effective for coding constant regions and straight edges due to the scale invariance of these features. An important advantage over standard vector quantization coders is that fractal coders do not require separate storage of a fixed vector codebook. Fractal encoding algorithms entail the construction of a map from the plane to itself of which the unique fixed point is an approximation to the image to be coded. Compressed images are stored by storing this map and recovered by iteratively applying the map to find its fixed point. We now describe a simple fractal image compression scheme based on those in [9] and [6].

Let $\mathcal{I}$ be a $2^N \times 2^N$ pixel image, and let $\mathbf{B}^J_{K,L}\mathcal{I}$ be the $2^J \times 2^J$ subblock of $\mathcal{I}$ with lower left hand corner at $(2^J K, 2^J L)$. $\mathbf{B}^J_{K,L}\mathcal{I}$ is the result of applying the linear "get-block" operator $\mathbf{B}^J_{K,L} : \mathbb{R}^{2^{2N}} \to \mathbb{R}^{2^{2J}}$ to the image $\mathcal{I}$.

The adjoint of the get-block operator, $(\mathbf{B}^J_{K,L})^*$, is a "put-block" operator which maps a $2^J \times 2^J$ subblock to a $2^N \times 2^N$ image containing the block with its lower left hand corner at the point (K, L). We will use capital letters to denote block coordinates and lower case to denote individual pixel coordinates. To simplify our notation we will 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.

We first partition $\mathcal{I}$ into a set of non-overlapping $2^R \times 2^R$ *range blocks*, the blocks $\mathbf{B}^R_{2^R K, 2^R L}\mathcal{I}$, with $(K, L) \in \mathcal{R}$ where $\mathcal{R} = \{(K, L) : K, L \in \mathbf{Z} \text{ and } 0 \leq K, L < 2^{N-R}\}$ (by construction, integer-valued subscripts correspond to a disjoint partition of the image). The goal of the compression scheme is to approximate each range block with a block from a codebook constructed from *domain blocks* $\mathbf{B}^D_{2^D K, 2^D L}\mathcal{I}$, with $(K, L) \in \mathcal{D}$ where $D > R$ and $\mathcal{D}$ is a set called the *domain pool*. In [9] the codebook is constructed from all unit translates of the domain blocks, so the domain pool $\mathcal{D} = \{(K, L) : 2^D K, 2^D L \in \mathbf{Z} \text{ and } 0 \leq K, L < 2^{N-D}\}$ is an collection of overlapping subblocks. We discuss other possible domain pools below. In our implementation we will usually take $D = R + 1$, i.e. the domain blocks are blocks with twice the width and height of the range blocks.

We now define several operators which will be used to construct the codebook from the domain blocks. Let $\mathbf{A}$ be the "average-and-subsample" operator which maps $2^J \times 2^J$ image blocks to $2^{J-1} \times 2^{J-1}$ blocks by averaging each pixel in $\mathbf{B}^J_\Gamma$ with its neighbors and then subsampling. We have $(\mathbf{A}\mathbf{B}^J_\Gamma\mathcal{I})(k, l) = \frac{1}{4}[(\mathbf{B}^J_\Gamma\mathcal{I})(2k, 2l) + (\mathbf{B}^J_\Gamma\mathcal{I})(2k + 1, 2l) + (\mathbf{B}^J_\Gamma\mathcal{I})(2k, 2l + 1) + (\mathbf{B}^J_\Gamma\mathcal{I})(2k + 1, 2l + 1)]$ where $\mathbf{B}^J_\Gamma\mathcal{I}(k, l)$ is the pixel at coordinates (k, l) within the subblock $\mathbf{B}^J_\Gamma\mathcal{I}$. Let $\{\mathbf{L}_i\}_{1 \leq i \leq 8}$ be the 8 isometries of the square obtained from compositions of reflections and 90 degree rotations.

We approximate each range block in the image with a linear combination of a codebook element and a subblock of the matrix $\mathbf{1}$, the $2^N \times 2^N$ matrix of 1's. This subblock of the matrix of 1's allows us to adjust the DC component of our approximation. We have

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

where $\Pi(\Gamma)$ assigns an element from the domain pool to each range element, $P(\Gamma)$ assigns each range element a symmetry operator index, and $\mathbf{A}^{D-R}$ denotes the operator $\mathbf{A}$ applied $D - R$ times. The scalars g_Γ and h_Γ, the domain block $\Pi(\Gamma)$, and the symmetry operator index $P(\Gamma)$ are chosen to minimize the l^2 approximation error $\|\mathbf{B}^R_\Gamma\mathcal{I} - (g_\Gamma \mathbf{L}_{P(\Gamma)}\mathbf{A}^{D-R}\mathbf{B}^D_{\Pi(\Gamma)}\mathcal{I} + h_\Gamma \mathbf{B}^R_\Gamma\mathbf{1})\|$. We note that while it is possible to bound the l^2 quantization error in the decoded image in terms of this l^2 error, minimizing this l^2 error is not an optimal selection criterion. An improved criterion is described in an earlier work [5].

The image $\mathcal{I}$ can be written as a sum of its range blocks, $\mathcal{I} = \sum_{\Gamma \in \mathcal{R}} (\mathbf{B}^R_\Gamma)^* \mathbf{B}^R_\Gamma\mathcal{I}$, so we have

$$\mathcal{I} \approx \sum_{\Gamma \in \mathcal{R}} g_\Gamma (\mathbf{B}^R_\Gamma)^* \mathbf{L}_{P(\Gamma)}\mathbf{A}^{D-R}\mathbf{B}^D_{\Pi(\Gamma)}\mathcal{I} + \sum_{\Gamma \in \mathcal{R}} h_\Gamma (\mathbf{B}^R_\Gamma)^* \mathbf{B}^R_\Gamma\mathbf{1} = \mathbf{G}\mathcal{I} + \mathcal{H}. \tag{2}$$

To store the image we store for each $\Gamma \in \mathcal{R}$ the scalars g_Γ and h_Γ, the symmetry operator index $P(\Gamma)$, and the domain block $\Pi(\Gamma)$ used for quantization.

We recover the image iteratively from this stored information. We start with an arbitrary image $\mathcal{I}_0$, and we compute $\mathcal{I}_n = \mathbf{G}\mathcal{I}_{n-1} + \mathcal{H}$. It can be shown that this process converges pointwise when the scaling factors $|g_\Gamma| < 1$ [8]. Numerical experiments show that this upper bound is not a necessary condition for convergence, and that the use of larger bounds on the scaling factors can yield improved compression results [6].

3 FRACTAL COMPRESSION IN THE WAVELET DOMAIN

3.1 The Discrete Wavelet Transform

Wavelets are a natural tool for analyzing fractal compression since wavelet bases possess the same type of dyadic similarity which fractal compression exploits. The Haar wavelet basis, in particular, possesses the same regular block structure that the partition into range blocks imposes on the image to be compressed. We first establish some notation. For simplicity we will consider orthogonal wavelets only; it is straightforward to generalize the results to the biorthogonal case.

Let $\psi(x)$ and $\phi(x)$ be an orthogonal wavelet and its associated scaling function (see [10] and [14] for an overview of wavelets). The scaling function together with a set of dyadic rescalings and integral translations of $\psi(x)$ form a basis of $L^2(\mathbb{R})$. We can form a similar basis of $L^2(\mathbb{R}^2)$ using the function $\phi(x,y) = \phi(x)\phi(y)$, a separable 2-D scaling function, together with dyadic rescalings and integer translations of the separable 2-D wavelets $\psi_H(x,y) = \phi(x)\psi(y)$, $\psi_V(x,y) = \psi(x)\phi(y)$, and $\psi_D(x,y) = \psi(x)\psi(y)$. The wavelets ψ_H, ψ_V, and ψ_D act as horizontal, vertical, and diagonal edge detectors, respectively. We use the subscript ω to represent one of the three orientations in $\Omega = \{H, V, D\}$.

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 $\mathcal{W}_{J_0}$, the functions $\phi^{J_0}_{k,l} = 2^{J_0}\phi(2^{J_0}x - k, 2^{J_0}y - l)$ and $\psi^j_{\omega,k,l} = 2^j\psi_\omega(2^j x - k, 2^j y - l)$, for $J_0 \le j < N$ (the scale J_0 of the scaling functions is taken to be 0 in the standard version of the discrete wavelet transform). We will use a single lower-case Greek multi-index, usually γ, to abbreviate the orientation and translation subscripts of ϕ and ψ.

Each wavelet $\psi^j_{\omega,k,l}$ has four children, 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 $\psi^{j+1}_{\omega,2k+1,2l+1}$, that correspond to the same spatial location and the same orientation as $\psi^j_{\omega,k,l}$. A *wavelet subtree* is a set of wavelet coefficients that correspond to the same spatial location but different scales and orientations. The subtree $\mathbf{S}^J_{K,L}\mathcal{I}$ consists of the coefficients of the three oriented wavelets $\psi^J_{\omega,K,L}$, for $\omega \in \{H, V, D\}$, together with the coefficients of their children, their children's children, and so on. $\mathbf{S}^J_{K,L} : \mathbb{R}^{2^{2N}} \to \mathbb{R}^{2^{2(N-J)}-1}$ is a linear "get-subtree" operator analogous to our get-block operator which extracts a subtree with root wavelets $\psi^J_{\omega,K,L}$ from the wavelet transform of an image. The adjoint of $\mathbf{S}^J_{K,L}$ is a "put-subtree" operator which inserts a subtree into an all-zero wavelet transform at scale J, offset (K, L). The subtree $\mathbf{S}^J_{K,L}\mathcal{I}$ essentially provides local high-pass information about the image; the associated scaling function $\phi^J_{K,L}$ provides the local low-pass information. Two wavelet subtrees are shown as shaded regions in Figure 3.2.

3.2 A Wavelet Analog of Fractal Compression

We now obtain a wavelet-based analog of fractal compression. In fractal compression we 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. 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 $<\mathcal{I}, \phi^{N-R}_\Gamma>$. 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 3.2.

We now examine the wavelet-domain behavior of the linear operators used in fractal compression. We first consider the wavelet analog $\hat{\mathbf{A}}$ of the average-and-subsample operator $\mathbf{A}$. Averaging and subsampling of the finest-scale Haar wavelets sets them to 0. For scales other than the finest, the local averaging has no effect, and

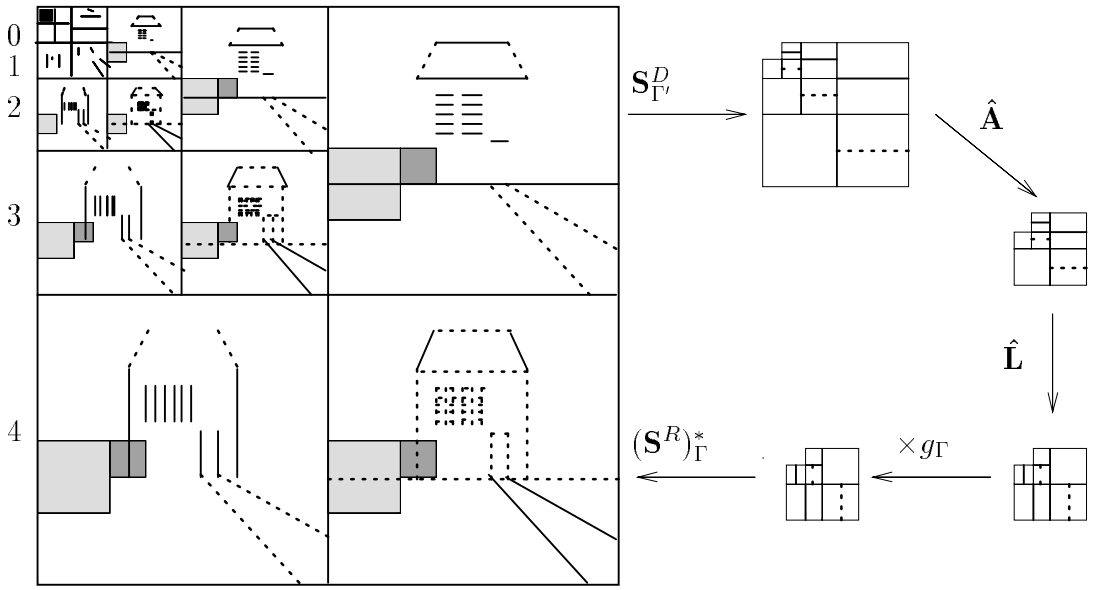

Figure 1: We approximate the darkly shaded range subtree $\mathbf{S}^R_\Gamma \mathcal{I}$ using the codebook element $g_\Gamma \hat{\mathbf{L}} \hat{\mathbf{A}} \mathbf{S}^D_{\Gamma'} \mathcal{I}$ which is derived from the lightly shaded domain subtree $\mathbf{S}^D_{\Gamma'} \mathcal{I}$. $\hat{\mathbf{A}}$ truncates the finest scale coefficients of the domain subtree and multiplies the coefficients by $\frac{1}{2}$, and $\hat{\mathbf{L}}$ rotates it. When storing this image we save the coarse-scale wavelet coefficients in bands 2 and below, and we save the encodings of all subtrees with roots in band 3.

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 the scaling function at the next finer resolution, ϕ^{j+1}_γ except for the finest-scale scaling functions, which are set to 0. The action of the averaging and subsampling operator is thus seen to be a shift of coefficients from coarse-scale to fine, a multiplication by $\frac{1}{2}$, and a truncation of the finest-scale coefficients. We see, then, that the effect of $\hat{\mathbf{A}}$ is to prune the leaves of a subtree and shift all coefficients to the next finer scale. The action of $\hat{\mathbf{A}}$ is illustrated in Figure 3.2.

For symmetrical wavelets, the only wavelets we will consider here, the horizontal reflection of a block corresponds to a horizontal reflection of wavelet coefficients within each scale of a subtree, and 90 degree block rotations correspond to 90 degree rotations of wavelet coefficients within each scale and a switching of the ψ_H coefficients with ψ_V coefficients. Hence the wavelet analogs $\hat{\mathbf{L}}_i$ of the block symmetry operators $\mathbf{L}_i$ permute wavelet coefficients within each scale. Figure 3.2 illustrates the action of a symmetry operator on a subtree.

The important thing to note is that the operators we use preserve subtrees, and the basic steps in fractal coding have simple wavelet analogs. The extraction of a domain block by the operator $\mathbf{B}^D_{\Pi(\Gamma)}$ corresponds to the extraction of the subtree by the operator $\mathbf{S}^{N-D}_{\Pi(\Gamma)}$ plus the extraction of the scaling function coefficient $< \mathcal{I}, \phi^{N-D}_{\Pi(\Gamma)} >$. The averaging and subsampling performed by $\mathbf{A}$ followed by the rotation and reflection performed by $\mathbf{L}$ corresponds to a shifting of the wavelet coefficients from coarse scales to fine and a permutation of the subtree coefficients within each scale. The result is that we obtain a relation similar to (1) for the Haar wavelet subtrees,

$$\mathbf{S}^{N-R}_\Gamma \mathcal{I} \approx g_\Gamma \hat{\mathbf{L}}_{P(\Gamma)} \hat{\mathbf{A}}^{D-R} \mathbf{S}^{N-D}_{\Pi(\Gamma)} \mathcal{I}. \tag{3}$$

The offset terms h_Γ from (1) affect only the scaling function coefficients because all translates of Haar wavelets are 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 $\mathbf{S}_\Gamma^{N-R}\mathcal{I}$,

$$< \mathcal{I}, \psi_\gamma^j > \approx \frac{g_\Gamma}{2^{D-R}} < \mathcal{I}, \psi_{\gamma'}^{j-(D-R)} > = \frac{g_\Gamma}{2^{D-R}} < \mathcal{I}, \mathbf{T}(\psi_\gamma^j) > . \tag{4}$$

Here $\mathbf{T}$ is the linear map induced by the domain block selection followed by averaging, subsampling, and rotating which, as we noted above, maps wavelet coefficents to wavelet coefficients. We obtain a similar relation for the ϕ's,

$$< \mathcal{I}, \phi_\Gamma^{N-R} > \approx \frac{g_\Gamma}{2^{D-R}} < \mathcal{I}, \phi_{\Pi(\Gamma)}^{N-D} > + h_\Gamma = \frac{g_\Gamma}{2^{D-R}} < \mathcal{I}, \mathbf{T}(\phi_\Gamma^{N-R}) > + h_\Gamma \tag{5}$$

The system of equations (4) and (5) reveal that fractal image compression is essentially a map from coarse-scales to fine. The relationships are complicated by the fact that the wavelets and scaling functions on the right hand sides of (4) and (5) are *not* always members of our basis $\mathcal{W}$ since general domain pools may contain domain blocks which require non-integral translates of wavelets in the subtrees. We discuss this idea of a mapping from coarse to fine scales in greater detail in the next section.

We obtain a wavelet-based analog of fractal compression by replacing the Haar basis used in (4) and (5) with a symmetric orthogonal or biorthogonal wavelet basis (when using a biorthogonal basis the above relations have ψ and ϕ replaced by their dual functions $\tilde{\psi}$ and $\tilde{\phi}$). There are two effects of this change of basis. First, when the new wavelets are smooth, we switch from quantizing image subblocks, which possess sharp, discontinuous edges, to quantizing subtrees, which correspond to a partition with smooth boundaries. Second, we replace the local averaging and subsampling procedure with more general filtering. Our numerical experiments below show that changing from the Haar basis to a smooth basis results in considerable improvement in compressed image quality due in part to the elimination of the block boundary artifacts.

3.3 Conditions for Convergence

Our generalized quantization scheme yields a relation for an image $\mathcal{I}$ of the form

$$(\mathbf{W}\mathcal{I}) = \mathbf{G}(\mathbf{W}\mathcal{I}) + \mathcal{H} \tag{6}$$

where $\mathbf{W}\mathcal{I}$ is the discrete wavelet transform of $\mathcal{I}$. Given $\mathbf{G}$ and $\mathcal{H}$ we decode $\mathcal{I}$ by the same iterative process used in fractal schemes. Our first goal is to show that the above encoding and iterative decoding process yields a well-defined image for non-Haar wavelets. While showing that $(\mathbf{I} - \mathbf{G})$ is non-singular is sufficient to ensure decodability, inverting $(\mathbf{I} - \mathbf{G})$ is prohibitively computationally expensive, requiring $O(2^{6N})$ work for a $2^N \times 2^N$ pixel image. The iterative decoding scheme converges if $\mathbf{G}^n \to 0$ as $n \to \infty$, which will happen if and only if all eigenvalues of $\mathbf{G}$ are strictly less than 1 in magnitude. We note that $\mathbf{G}$ is not in general diagonalizable, so we are interested in its Jordan blocks.

By expanding the image $\mathcal{I}$ into a linear combination of basis functions in (4) and (5) we find that the entries of the matrix $\mathbf{G}$ in the row corresponding to the basis function $w \in \mathcal{W}$ are the projections of $\mathbf{T}(w)$ onto the basis multiplied by $2^{R-D} g_\Gamma$, i.e. $\mathbf{G}_{k,l} = 2^{R-D} g_{\Gamma_k} < w_l, \mathbf{T}(w_k) >$ for $w_k, w_l \in \mathcal{W}$. When $\mathbf{T}(w_k) \in \mathcal{W}$, the matrix $\mathbf{G}$ has only one nonzero entry per row, but for the general case we are currently considering, the offsets of $T(w_k)$ in general correspond to non-integral translates. By the Geršgorin circle limit theorem, all eigenvalues will have magnitudes strictly less than 1 if for all g_Γ we have

$$g_\Gamma < 2^{D-R} \min_{w'} \frac{1}{\sum_{w \in \mathcal{W}_{N-R}} | < w, w' > |}. \tag{7}$$

where the where the minimum is taken over all wavelets contained in the subtrees in the domain pool and all their associated scaling functions. We thus obtain sufficient conditions for convergence based on the l^1 norm of sampled reproducing kernels for ϕ and ψ. This bound on the g_Γ's is not a necessary condition for convergence. Indeed, the sum in the denominator in general considerably larger than 2^{D-R}, so for the Haar basis this is a weaker

result than can be obtained for subblock methods via pointwise methods [8]. The bound does show that we can decode our wavelet encoding via an iterative algorithm for some range of scale factors. In the next section we see that the contractivity of the map we obtain by restricting the constants g_Γ is not as important as showing that information is transferred from coarse scales to fine. We introduce a refinement of our wavelet analog to fractal compression for which the decoding process converges unconditionally and which has a fast decoding algorithm.

4 SELF-QUANTIZATION OF SUBTREES

4.1 Extrapolation in Scale

The approximation in (5) gives an implicit scheme for obtaining scaling function coefficients, and it requires that we store a constant h_Γ for each scaling function coefficient. We can greatly simplify our scheme by storing the coefficients $< \mathcal{I}, \phi_\Gamma^{N-R} >$ directly. The scaling function coefficients contain removable spatial redundancy, so as we discuss in section 5, we can store them very efficiently.

We can further simplify our scheme by restricting the domain pool to the disjoint partition $\mathcal{D} = \{(K, L) : K, L \in \mathbf{Z} \text{ and } 0 \leq K, L < 2^{N-D}\}$. With this restriction, all wavelets corresponding to coefficients in the domain pool subtrees are members of the basis $\mathcal{W}_{N-D}$. Hence in the system (4) each wavelet coefficient of scale $\geq R$ depends only on a coefficient of a coarser-scale wavelet.

We say that a map from one set of wavelet coefficients to another is *R-scale-extending* if each wavelet coefficient of scale $j \geq R$ in the range is dependent only on wavelet coefficients of scale $< j$. The system given in (4) is a particularly simple form of scale-extending map. One can show that (4) also yields an R-scale-extending map for the larger domain pool $\mathcal{D} = \{(K, L) : 2^{D-R}K, 2^{D-R}L \in \mathbf{Z} \text{ and } 0 \leq K, L < 2^{N-D}\}$. In the case of the the Haar basis this set corresponds to the set of domain blocks which share boundaries with range blocks. This particular restricted domain pool has been studied for fractal block coders in [13] and [3]. The theorem below generalizes their results, extends them into the general wavelet framework, and gives new insight into why their results hold.

THEOREM 4.1 (RECONSTRUCTION THEOREM). *Let $\mathcal{I}$ be a $2^N \times 2^N$ image for which the scaling function coefficients $< \mathcal{I}, \phi_\gamma^{N-R} >$ are known, and suppose that we know that $\mathcal{I}$ is the fixed point of a linear R-scale-extending map $\mathbf{M}$. Then we can find $\mathcal{I}$ using R applications of the map $\mathbf{M}$.*

Proof: By applying the wavelet transform to the image $\mathcal{I}_R = \sum_\gamma < \mathcal{I}, \phi_\gamma^{N-R} > \phi_\gamma^{N-R}$, we obtain all the coarse-scale wavelet coefficients $< \mathcal{I}, \psi_\gamma^j >$ for $j < N - R$ for $\mathcal{I}$. We can now obtain the wavelet coefficients $< \mathcal{I}, \psi_\gamma^{N-R} >$ by applying the map $\mathbf{M}$ since these coefficients depend only on the coefficients we already know. Each time we apply the map $\mathbf{M}$ we obtain the wavelet coefficients at the next finer scale, so by induction the result is proved. A more detailed version of this proof may be found in [5].

□

When we use the disjoint domain pool described above, the matrix $\mathbf{G}$ from (6) has a very simple form. The rows of $\mathbf{G}$ corresponding to the scaling functions are all zero since we have transferred all the scaling function information to the coefficients in $\mathcal{H}$. Because the wavelet permutation map $\mathbf{T}$ maps each basis element to a multiple of a basis element, the rows corresponding to wavelets in subtree $S_\Gamma^{N-R}\mathcal{I}$ contain a single nonzero entry with value $\frac{g_\gamma}{2^{D-R}}$. We order the vector of coefficients from coarse to fine, so $\mathbf{G}$ will be a strictly lower triangular matrix with all zeros on the diagonal. Thus, all eigenvalues of $\mathbf{G}$ are zero.

Because each fine-scale wavelet coefficient depends on only one other coefficient when we use our restricted

disjoint domain pool, the iterative technique of our proof yields a fast decoding algorithm which requires $O(1)$ operations per pixel. This iterative decoding algorithm also yields a fast decoding algorithm for the larger domain pool $\mathcal{D} = \{(K, L) : 2^{D-R}K, 2^{D-R}L \in \mathbf{Z} \text{ and } 0 \leq K, L < 2^{N-D}\}$ for Haar wavelets since the matrix $\mathbf{G}$ has a band structure.

The above reconstruction theorem generalizes to allow adaptive image encoding. Using the disjoint domain pool, we can recover an image using a fast algorithm provided that for each self-quantized subtree we store its associated scaling function coefficient. Equivalently, we can recover the image provided we know all coarse-scale wavelet coefficients not contained in the range subtrees.

4.2 Implications

Loosely speaking, standard fractal compression schemes entail the quantization of "fine-scale" features using "coarse-scale" features. The above theorem shows we can make this notion of scale rigorous with a particular class of domain pools. Features of a particular scale constitute a detail space of a multiresolution analysis [14]. Fractal decoding can be seen as a cascading of information from coarse wavelet coefficients to fine. Fractal compression has been motivated by the theory of iterated function systems [2], which involves the construction of a strictly contractive map from the image to itself. However, the scale-extending maps described above are not in general contractive in any of the l^p norms. It is the flow from coarse to fine and not the contractivity of the map which matters.

The detail space of resolution 2^j of a multiresolution approximations is invariant under translations of 2^{N-j} pixels, but *not* under unit pixel translations. It is precisely this lack of translation invariance which causes convergence problems when we expand the disjoint domain pools studied above to include finer translates of domain subtrees. When we approximate range subtrees using fine translates of domain subtrees, we introduce dependencies of fine-scale wavelet coefficients on coefficients from the same or finer scales. Information no longer flows strictly from coarse to fine under the map. Dependency loops from fine-scales to fine-scales permit the growth of unstable eigenvectors unless these loops are damped by restricting the scaling coefficients g_Γ. These dependency loops are the reason scaling factors are limited to be less than 1 in magnitude in conventional fractal coders.

The reproducing kernel for the wavelet basis characterizes the overlap of arbitrary translates of the detail spaces. The restriction on the scaling factors (7) embodies this measure of overlap. We can obtain a translation invariant domain pool with unconditional convergence by switching to a basis of sinc wavelets, since the detail spaces for this basis correspond to translation invariant frequency bands. The sinc wavelet basis possesses a number of disadvantages, however. The basis elements do not have compact support, so locally self-similar features can only be approximately isolated by the subtrees. In addition, computations are very slow.

Our analysis sheds light on several other aspects of fractal coding. Our convergence proof shows that images are reconstructed from the stored self-similarity relation by cascading information from coarse to fine scales. This suggests that the decoding process will be more sensitive to errors in coarser scales than fine. In [5] we examine a weighted l^2 error metric which we use to select which domain subtree to use for quantization.

Our convergence proof also gives insight into the mechanism underlying fractal interpolation or super-resolution of images. Each iteration of our decoding process generates a new level of wavelet coefficients. By continuing to decode after convergence, we can generate additional fine-scale coefficients. When we invert the resulting transform, we obtain a larger image than we originally encoded with detail that has been interpolated using our self-similarity map.

5 IMPLEMENTATION

Traditional fractal coders have typically been implemented using top-down refinement schemes. An initial set of large subblocks are self-quantized using a domain pool drawn from larger subblocks. Subblocks for which the quantization error exceeds a predetermined threshold are subdivided into smaller subblocks. The smaller blocks are then quantized and the quantization error is again tested against the threshold. Refinement proceeds until either the error is less than the threshold or the size of the subblocks is smaller than some minimum value. Bit allocation is determined experimentally. Although these schemes have proven effective in practice, they are not optimal in a rate-distortion sense.

Our wavelet analog of fractal compression possesses a structure very similar to a number of recently introduced hierarchical wavelet coders [15][17][11]. The basic problem is to quantize a tree of wavelet coefficients using some combination of subtree quantization and scalar quantization. In addition the resolution of the scalar quantizers must be adaptively allocated. [17] describes an algorithm which optimally allocates bits between a set of scalar quantizers and subtree quantizers, and we use a slightly modified version of this algorithm for our experiments. We sketch the algorithm and our modifications here and refer the reader to [17] for details.

The set of wavelet coefficients to be quantized forms a tree T; each leaf of the tree contains the triple of wavelet coefficients $< \mathcal{I}, \psi^j_{\omega,k,l} >$ for $\omega \in \Omega$. Each node and its descendents constitute a wavelet subtree. We traverse the tree from the root to the leaves. At each node we have three options. We can quantize the subtree at this node via self-quantization, we can quantize the subtree as a zerotree, or we can scalar quantize the coefficients at the node and recursively repeat the decision process for the children of the node. Nodes at which subtrees are self-quantized or quantized to zerotrees are called terminal nodes. This set of choices generates a decision tree S. We must in addition decide how we will quantize wavelet coefficients for different scales and orientations. We choose a vector $\mathbf{q}$ of quantizer resolutions from an admissible set Q.

Our goal is to minimize the encoding distortion $D(\mathbf{q}, S)$ for a particular decision tree and set of quantizers. We have a constraint on the total number of bits used, so total bits used by S and $\mathbf{q}$ must satisfy $R(\mathbf{q}, S) \leq R_0$. We can transform this constrained problem into an unconstrained problem via Lagrange multipliers. We seek to minimize the Lagrangian cost $J(\mathbf{q}, S) = D(\mathbf{q}, S) + \lambda R(\mathbf{q}, S)$ where λ here depends on our constraint $\mathcal{R}_0$.

We initiate the algorithm with a fixed λ and an S such that the entire tree is scalar-quantized. The algorithm proceeds iteratively in two stages. First, given S and λ, we find an optimal allocation of quantizer resolutions $\mathbf{q}$ using a Lagrangian algorithm [16] (we assume uniformly spaced quantization). We next perform an optimization step on our decision tree. For each parent of a terminal node in the tree, we determine whether a change in the decision tree at that point will reduce the Lagrangian cost $J(\mathbf{q}, S)$ for the tree, and we modify the node accordingly. We repeat this process for the parents of the parents of terminal nodes, and so on, working up the tree until all nodes have been checked. This process creates the optimal balance of scalar quantization at the resolutions in $\mathbf{q}$ and subtree quantization. The restructuring of the decision tree affects the performance of the scalar quantizers (because it alters the distribution of the coefficients), so after the decision tree step we perform another cycle of optimization $\mathbf{q}$ followed by optimization of S and so on. The process terminates when no further changes are made to the decision tree S. Once the final decision tree has been determined, the wavelet coefficients, self-quantization parameters, and the choices in the decision tree are entropy coded using an adaptive arithmetic coder.

6 RESULTS

Figure 6 compares the peak signal to noise ratios of the 512×512 Lena image compressed by standard fractal compression methods and by our self-quantization of subtrees (SQS) scheme. Figure 5 shows the resulting

Figure 2: The top left shows the 512×512 Lena image compressed at 60.6:1 (PSNR = 24.9 dB) using a disjoint domain pool and the quadtree coder from Fisher. The top right image has been compressed at 65.0:1 (PSNR = 28.2 dB) using our SQS scheme. This SQS scheme uses exactly the same domain pool as the quadtree, but our analysis of the SQS scheme enables us to make much more efficient use of bits. The bottom left image has been compressed at 63.2:1 (PSNR = 29.9) using the same SQS scheme but with a smooth wavelet basis. Blocking artifacts have been completely eliminated. The bottom right image has been compressed at 63.1:1 (PSNR = 30.05) using a hybrid SQS/wavelet coder. The gain from increased flexibility is offset by increased decision tree cost.

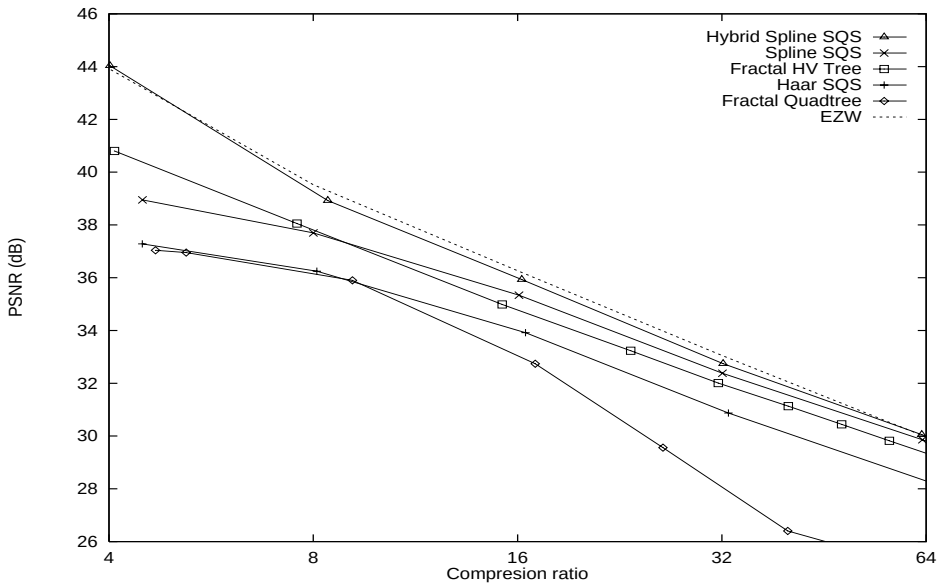

Figure 3: Peak signal to noise ratios for the 512×512 Lena image compressed by the quadtree fractal coder of Fisher, an adaptive Haar SQS scheme, an adaptive HV tree fractal coder of Fisher and Menlove, an adaptive spline SQS scheme, a hybrid spline/wavelet SQS scheme, and Shapiro's embedded zerotree wavelet coder.

decompressed images. The lowest line in Figure 6, 'Fractal Quadtree', is from the quadtree block coder listed in [8] using a disjoint domain pool as described in section 4.1 constructed from domain blocks of size 8×8 up to 128×128. (We note that quadtree schemes perform much better for larger domain pools; the point is to illustrate that our SQS scheme makes more effective use of existing resources.) The next line, 'Haar SQS', is from our adaptive SQS scheme using Haar wavelets. For comparison purposes, all range blocks in this SQS scheme have been coded either via self-quantization or by zerotrees. As we see from the decompressed images in Figure 5, the SQS scheme produces dramatically improved results using exactly the same domain pool.

There are several reasons for the SQS scheme's improved performance. First and foremost, our improved understanding of our compression mechanism enables us to partition bits efficiently between wavelet coefficients and subtrees. Indeed, our SQS scheme's similarity to existing wavelet transform coders allows us to use off-the-shelf near-optimal bit-allocation schemes. Secondly, smooth regions in the image contain subtrees consisting almost entirely of near-zero coefficients. We can store such subtrees compactly using zerotrees. Finally, our storage of the coarse-scale portion of the image in transformed form is very efficient. For low compression ratios, few zerotrees are used, and the decrease in error due to optimal bit alloction for the coefficients is small, so the improvement obtained from our algorithm is smaller.

The performance of fractal block coders is substantially improved by enlarging the domain pool. The third line from the bottom of Figure 6, 'Fractal HV Tree', shows the PSNR for a fractal block encoding of rectangular range blocks using rectangular domain blocks [7]. The use of rectangular blocks introduces an additional degree of freedom in the construction of the domain pool and gives increased flexibility to the partitioning of the image. The reconstructed images in [7] show the coding to be of high quality. In fact, the authors claim that their algorithm gives the best results of any fractal block coder in the literature. The enlarged domain pool results in high computational complexity for coding, however. Encoding times for the plotted points are as high as 46 CPU-hours on a Silicon Graphics Personal IRIS 4D/35.

The fourth line in Figure 6, 'Spline SQS', represents an alternative method of improving compressed image fidelity: we change bases. The Haar basis performs poorly for image compression because quantization introduces

horizontal and vertical discontinuities into the decoded image. Such artifacts yield poor subjective image quality since the human visual system is especially sensitive to horizontal and vertical lines. We replace the Haar basis with a biorthogonal spline basis designed for image compression (the spline variant with less dissimilar lengths) from [1] and use the same disjoint domain pool as with the Haar SQS scheme. There is a substantial improvement in perceived image quality over the Haar SQS scheme. In particular, blocking artifacts, a hallmark of fractal block coding schemes, have been completely eliminated. An additional advantage of this biorthogonal basis over the Haar is that the increased number of vanishing moments yields more zerotrees which can be coded very cheaply.

Allowing the Lagrangian bit allocator the additional option of coding subtrees by storing their wavelet coefficients yields the top solid line, 'Hybrid Spline SQS'. This increased flexibility yields improved coding for subtrees which are not well-represented by self-quantization. The improvement in performance is offset slightly by the increased cost of storing the decision tree for the compressed image. We note that [17] obtains some savings in the cost of storing the decision tree by predicting decisions conditioned on the values of the parent wavelet coefficients. Similar predictions are possible for this hybrid wavelet/SQS coder but have not been used here. We see that the performance of this hybrid coder is comparable to that of Shapiro's embedded wavelet zerotree encoder [15].

7 ACKNOWLEDGMENTS

This work was supported in part by DARPA as administered by the AFOSR under contract DOD F4960-93-1-0567.

8 REFERENCES

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, Vol. 1, No. 2, 205-221.

[2] M.F. Barnsley and A. Jacquin, "Application of recurrent iterated function systems to images." *Proc. SPIE*, vol. 1001, pp. 122-131, 1988.

[3] Z. Baharav, D. Malah, and E. Karnin, "Hierarchical interpretation of fractal image coding and its application to fast decoding," in *Proc. Digital Signal Processing Conference*, Cyprus, July 1993.

[4] T. Bell, J. G. Cleary, and I. H. Witten, *Text Compression*, Prentice Hall, Englewood Cliffs, NJ, 1990.

[5] G. Davis, "Self-Quantization of Wavelet Subtrees," in *Proc. SPIE Wavelet Applications II*, Orlando, April 1995, Vol. 2491, pp. 141-152.

[6] Y. Fisher, B. Jacobs, and R. Boss, "Fractal Image Compression Using Iterated Transforms," in *Image and Text Compression*, J. Storer (ed), Kluwer Academic, 35-61, 1992.

[7] Y. Fisher and S. Menlove, "Fractal Encoding with HV Partitions", in Y. Fisher, *Fractal Compression: Theory and Application to Digital Images*, Springer Verlag, New York, 1994.

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

[9] A. Jacquin, "Fractal image coding based on a theory of iterated contractive image transformations," *Proc. SPIE Visual Comm. and Image Proc.*, pp. 227-239, 1990.

[10] B. Jawerth and W. Sweldens, "An overview of wavelet based multiresolution analyses," *SIAM Review*, 36:3, 377-412, 1994.

[11] A. S. Lewis and G. Knowles, "Image compression using the 2-D wavelet transform," *IEEE Transactions on Image Processing*, Vol. 1, No. 2, pp. 244-250, April 1992.

[12] L. M. Lundheim, *Fractal Signal Modelling for Source Coding.* Ph.D. thesis, The Norwegian Institute of Technology, November 1992.

[13] G. Øien L^2-*Optimal Attractor Image Coding with Fast Decoder Convergence.* Ph.D. thesis, The Norwegian Institute of Technology, November 1992.

[14] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, 674-693, July 1989.

[15] J. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, pp. 3445-3462.

[16] Y. Shoham and A. Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. 36, pp. 1445-1453, September 1988.

[17] Z. Xiong, K. Ramchandran, and M. Orchard. "Joint Optimization of Scalar and Tree-structured Quantization of Wavelet Image Decompositions," *Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, November 1993.