Merging Fractal Image Compression and Wavelet Transform Methods

Axel van de Walle
Department of Applied Mathematics
Faculty of Mathematics
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
e-mail: axel@links.uwaterloo.ca

August 17, 1995

Abstract

Fractal image compression and wavelet transform methods can be combined into a single compression scheme by using an iterated function system to generate the wavelet coefficients.

The main advantage of this approach is to significantly reduce the tiling artifacts: operating in wavelet space allows range blocks to overlap without introducing redundant coding. Our scheme also permits reconstruction in a finite number of iterations and lets us relax convergence criteria. Moreover, wavelet coefficients provide a natural and efficient way to classify domain blocks in order to shorten compression times.

Conventional fractal compression can be seen as a particular case of our general algorithm if we choose the Haar wavelet decomposition. On the other hand, our algorithm gradually reduces to conventional wavelet compression techniques as more and more range blocks fail to be properly approximated by rescaled domain blocks.

1 Introduction

The most visible artifact of fractal image compression is the "tiled" aspect of the compressed images. A possible solution to this "tiling" artifact is to allow range blocks to overlap. This scheme reduces the compression ratio because some parts of the image are coded more than once. This multiple coding can be avoided if, instead of minimizing collage distance for each block independently, we minimize the global collage distance for all blocks simultaneously at the expense of solving a large system of equations.[1] The minimization of collage

[1] We can then use the block overlap to obtain a better fit.

distance when using overlapping range blocks was studied by Forte and Vrscay in [4].

This approach, however, can be easily improved. It is possible to filter the content of each range block so that the data it contains is orthogonal to the data of all neighboring overlapping blocks. In this way, neighboring blocks contain independent information which can be coded independently and without redundancy. Moreover, the collage distance can be minimized independently for each block without solving a large system of equations.

The filtering introduced in the previous paragraph is cumbersome to perform in the conventional pixel representation of the image. However, the use of an orthogonal wavelet representation (see [6]) makes this filtering completely natural, and has many other computational advantages. We shall treat the one dimensional case only, for the sake of clarity, but it can be easily generalized to two-dimensional data by using bases of separable wavelets.

2 Generalizing the Concept of Block

We can represent a function $f(t)$ as a partial wavelet decomposition where all *detail levels* below some level k are represented by a linear combination of translates of the scaling function ϕ associated with a wavelet ψ.

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

where

$$\begin{aligned} \psi_{ij}(t) &= 2^{i/2} T_{ij} \, \psi(t) \ , \\ \phi_{ij}(t) &= 2^{i/2} T_{ij} \, \phi(t) \ , \\ T_{ij} f(t) &= f(2^i t - j) \ . \end{aligned}$$

The constants $\langle f, \psi_{ij} \rangle$ and $\langle f, \phi_{ij} \rangle$ are called, respectively, the wavelet coefficients and the scaling function coefficients.

We then define the block operator B_{kl} that clips a function f to a block covering an interval of the form $[2^{-k}l, 2^{-k}(l+1)]$:

$$B_{kl} f(t) = \langle f, \phi_{kl} \rangle \, \phi_{kl}(t) + \sum_{i=k}^{\infty} \sum_{j=2^{i-k}l}^{2^{i-k}(l+1)-1} \langle f, \psi_{ij} \rangle \, \psi_{ij}(t) \ .$$

Figure 1 illustrates which coefficients are selected by operator B_{kl}. In this figure, the wavelet coefficients are represented by rectangles whose horizontal extent loosely represent the spatial extent of the corresponding wavelet. The vertical extent is related to the frequency selectivity. The same applies for scaling function coefficients.

Notice that if we use the Haar wavelet, the block operator B_{kl} reduces to the usual clipping by a block. However, if one uses a smooth wavelet instead

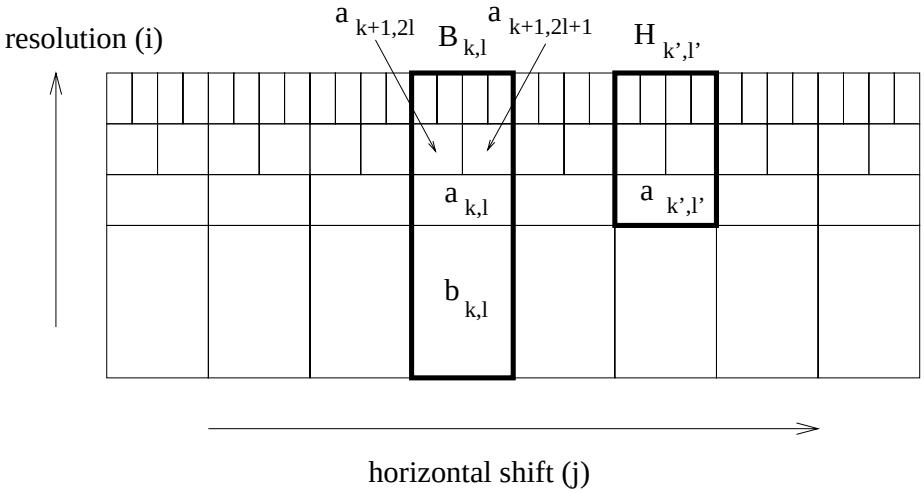

Figure 1: Action of Block Operators in the Wavelet Representation. $(a_{ij} = \langle f, \psi_{ij} \rangle, b_{ij} = \langle f, \phi_{ij} \rangle)$.

of the Haar wavelet, the clipping of f by two adjacent blocks B_{kl} and $B_{k,l+1}$ overlap but still enjoy the property that

$$\langle B_{kl} f, B_{k,l+1} f \rangle = 0 \ .$$

Also note that the DC component of a block in the Haar representation becomes the scaling function in the case of an arbitrary wavelet decomposition.

3 Fractal Compression in the Wavelet Representation

Let us momentarily go back to the case of the Haar wavelet to express conventional fractal compression using the wavelet point of view in order to make the use of *generalized blocks* easier. Proofs of the propositions used in this section can be found in reference [9].

To perform fractal compression, one usually tries to approximate the content of a given range block $B_{ij} f(t)$ by a rescaled and translated version of a bigger domain block $B_{kl} f(t)$ where $k < i$:

$$B_{ij} f(t) \approx B_{ij} f^*(t) = s_j \ T_{ij} T_{kl_j}^{-1} B_{k,l_j} f(t) + c_j T_{ij} \phi(t) \ .$$

The "offset" term $c_j T_{ij} \phi(t)$ is, as usual, constant within a block and zero outside. For simplicity, we let k and i be constant across the whole image (that is, domain and range blocks are constant in size). Thus, for each j, we only have to search the optimal l_j by computing the s_j and c_j that minimizes the RMS distance

between $B_{ij}f$ and $B_{ij}f^*$. If the mapping M defined as

$$Mf(t) = \sum_{j=-\infty}^{\infty} s_j \; T_{ij} T_{kl_j}^{-1} B_{kl_j} f(t) + c_j T_{ij} \phi(t)$$

is a contraction in some complete metric space, we can use the contraction mapping theorem to reconstruct the image from the l_j, s_j and c_j by iterating

$$f_{n+1}(t) \quad = \quad Mf_n(t) = \sum_{j=-\infty}^{\infty} s_j \; T_{ij} T_{kl_j}^{-1} B_{kl_j} f_n(t) + c_j T_{ij} \phi(t) \; , \qquad (1)$$

starting with some arbitrary $f_0(t)$.

The offset term $c_j T_{ij} \phi(t)$ can however be treated in a different but equivalent way. A simple mean square error minimization lets us find the optimal offset term c_j for a fixed s_j:

$$c_j = m_{ij} - s_j \; m_{kl} \; ,$$

where we have dropped the subscript of l_j and

$$m_{ij} = 2^i \int_{2^{-i}j}^{2^{-i}(j+1)} f(t) dt = 2^{i/2} \left\langle f, \phi_{ij} \right\rangle \; .$$

(In other words, m_{ij} is the mean gray level of $B_{i,j}f$ on its support.) Substitution of this expression of c_j in Equation (1) and a few manipulations yields

Proposition 1

$$B_{ij} \left(f_{n+1}(t) - \langle f, \phi_{ij} \rangle \, \phi_{ij}(t) \right) \quad = \quad s_j B_{ij} T_{ij} T_{kl}^{-1} \left(f_n(t) - \langle f, \phi_{kl} \rangle \, \phi_{kl}(t) \right) \; . (2)$$

Essentially, this equation expresses that one can subtract the DC component of domain and range blocks and still preserve the equality. The problem with Equation (2) is that we need to know f in order to compute f (because of the $\langle f, \phi_{ij} \rangle$ term). Fortunately, by choosing a particular initial condition f_0 we have the following consequence.

Proposition 2 *If*

$$f_0(t) = \sum_{j=-\infty}^{\infty} m_{i'j} T_{i'j} \phi(t) = \sum_{j=-\infty}^{\infty} \langle f, \phi_{i'j} \rangle \, \phi_{i'j}(t)$$

then

$$\langle f_n, \phi_{ij} \rangle = \langle f, \phi_{ij} \rangle$$

for all $n \geq 0$ *for any* $i \leq i'$ *and for any* j.

This result lets us replace all f in Equation (2) by f_n

$$B_{ij} \left(f_{n+1}(t) - \langle f_n, \phi_{ij} \rangle \, \phi_{ij}(t) \right) \quad = \quad s_j B_{ij} T_{ij} T_{kl}^{-1} \left(f_n(t) - \langle f_n, \phi_{kl} \rangle \, \phi_{kl}(t) \right) \; (3)$$

We can then define a new block operator

$$\begin{aligned} H_{kl} f(t) &= B_{kl} \left(f(t) - \langle f, \phi_{kl}(t) \rangle \, \phi_{kl}(t) \right) \\ &= \sum_{i=k}^{\infty} \sum_{j=2^{i-k}l}^{2^{i-k}(l+1)-1} \langle f, \psi_{ij} \rangle \, \psi_{ij}(t) \end{aligned}$$

that lets us write Equation (3) as

$$H_{ij} f_{n+1} = s_j T_{ij} T_{kl}^{-1} H_{kl} f_n \ . \tag{4}$$

Figure 1 illustrates the effect of operator $H_{k'l'}$ on the wavelet coefficients.

This new way of iterating has a very simple interpretation. The operator H_{ij} extracts all components of block B_{ij} except the scaling function ϕ_{ij}. Hence, once all scaling function coefficients have been set by the particular initial condition f_0 we chose, they will remain unchanged as we iterate Equation (4). We store all the components having a frequency too low to be contained in a block and we extrapolate the high frequency component by copying and rescaling operations.

This scheme can be completely expressed in term of the wavelet coefficients. One can easily convert the Equation (4) to an IFS on the wavelet coefficients.

Proposition 3

$$H_{ij} f_{n+1} = s_j T_{ij} T_{kl}^{-1} H_{kl} f_n$$

is equivalent to

$$a_{n+1,i+i',2^{i'}j+j'} = 2^{(k-i)/2} s_j \, a_{n,k+i',2^{i'}l+j'}$$

for $i' \geq 0$ *and* $0 \leq j' < 2^{i'}$, *where* $a_{nij} = \langle f_n, \psi_{ij} \rangle$. *(This relation between wavelet coefficients is illustrated in Figure 2.)*

We can also express the initial condition f_0 in term of wavelet coefficients instead of scaling function coefficients:

$$f_0 = \sum_{i=-\infty}^{i'-1} \sum_{j=-\infty}^{\infty} \langle \psi_{ij}, f \rangle \, \psi_{ij} \ .$$

Since the RMS norm is preserved under unitary transformations such as the orthogonal wavelet transform, minimizing the RMS distance between wavelet coefficients is equivalent to minimizing the RMS distance between the actual image blocks. We can then perform the whole algorithm in the wavelet representation as shown in Figure 2. All wavelet coefficients below some level i are stored without further transformation. Coefficients above level i are then approximated by copying operations (say, that map region B_{kl} to region B_{ij} in Figure 2).

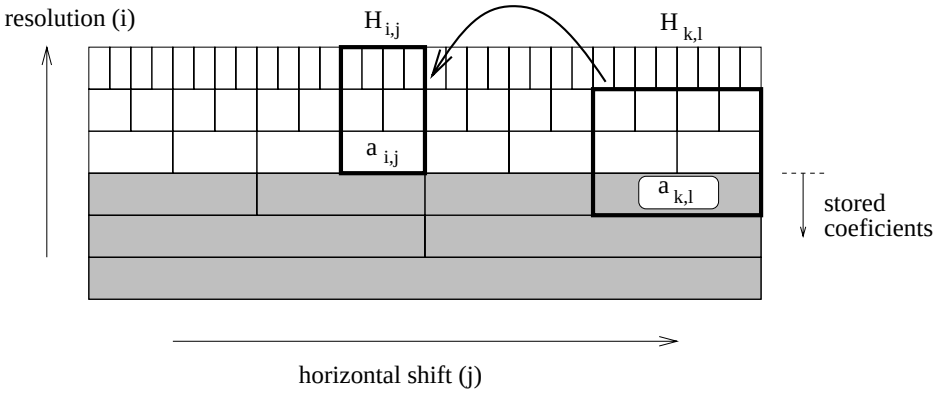

Figure 2: Illustration of the Compression Algorithm Using Pyramidal Ordering of the Wavelet Coefficients. *Filled rectangles represent stored coefficients, empty rectangles, approximated coefficients. Heavy rectangles show a mapping from a domain to a range block* ($a_{ij} = \langle f, \psi_{ij} \rangle$).

4 Advantages of the Wavelet Representation

4.1 Blockless Transform

Despite its expression in term of wavelets, the algorithm presented above is still equivalent to the usual fractal transform if we use Haar wavelets. However, being expressed in this wavelet form, the fractal transform algorithm can now be easily generalized to the case where range blocks overlap in an orthogonal way. We merely need to choose a smooth and orthogonal wavelet basis in place of the Haar wavelet basis. We thus have eliminated the "tiling" artifact without increasing the order of complexity of the algorithm and without affecting the compression ratio.

Figure 3 shows the "Lena" image compressed by our algorithm using the Haar wavelet decomposition while figure 4 shows the effect of using a smooth wavelet basis. This basis was generated with Adelson and Simoncelli's 9-tap filter (see [1]). The RMS distance between the original image and the compressed one is almost the same for both the Haar and the smooth wavelet decomposition. Yet, the artifacts are much less objectionable for the algorithm that uses a smooth wavelet.

The advantages of the use of an IFS on wavelet coefficients go beyond the elimination of the "tiling" artifact, as we shall see in the next sections.

4.2 Natural Quadtree Partitioning

The wavelet representation makes quadtree[2] partitioning of the image very natural. If one of the blocks of level i (say, $B_{ij} f(t)$) cannot be approximated

[2] "binary tree" in our one dimensional example

Figure 3: The 256x256 "Lena" Image Compressed with Wavelet Aided Fractal Compression. *The Haar wavelet was used. Compression ratio: 43:1, RMS error: 14.8.*

Figure 4: The 256x256 "Lena" Image Compressed with Wavelet Aided Fractal Compression. *Adelson and Simoncelli's "9-tap" filter was used. Compression ratio: 42:1, RMS error: 14.2.*

satisfactorily, we store the first layer of wavelet coefficients of this block and the try to approximate the two blocks $B_{i+1,2j}f(t)$ and $B_{i+1,2j+1}f(t)$. The splitting can be repeated if needed. Once the algorithm is completed, the wavelet coefficients are thus separated into two categories: those that are stored and those that are approximated. In this way, the cutoff between explicit storage and approximation of the coefficients varies from place to place, adapting to the local complexity of the image (as shown in Figure 5).

In the quadtree scheme presented above, it is assumed that it is always more efficient to store a range block as a copy of a domain block rather than to store its wavelet coefficients. If one does not perform any entropy coding, this assumption generally holds. However, wavelet coefficients can be entropy coded

much more efficiently than the parameters needed to represent a block copy operation. This being taken into account, we observed that some range blocks may well be more compactly represented by their wavelet coefficients than by a copy of a domain block. Our algorithm could thus be improved by performing entropy estimates in order to select the most compact representation for each range block.

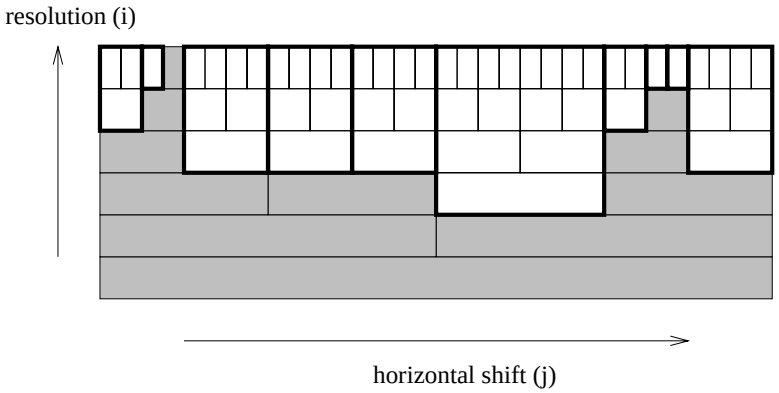

Figure 5: "Quadtree" Partitioning. *It is achieved by allowing cutoff between explicit storage and approximation to adapt locally.*

Our quadtree scheme has an interesting property. If the image is such that very few good matches can be found, more wavelet coefficients need to be stored. In the limit where no matches can be found we simply obtain the usual wavelet compression. In fact, we have a perfect interpolation between fractal and transform methods which automatically selects the most appropriate method for each part of the picture. Generally, at low compression ratio (*i.e.* high picture quality), few matches can be found and most wavelet coefficients have to be stored. Fractal compression only starts to play a significant role at high compression ratio. Indeed, in figure 6 we see that, at low compression ratio, our new algorithm tends to wavelet compression while conventional fractal compression does not perform well. On the other hand, in the range where fractal compression performs well, our algorithm performs just as well.

The results shown in figure 6 deserve some comments. First, it should be noted that Fisher's program does not perform any entropy coding while our method uses arithmetic coding. At high compression ratio, the difference between our algorithm and Fisher's can essentially be explained by this fact alone. At low compression ratio, however, the difference is genuine: conventional fractal compression is unable to approach perfect reconstruction as the compression ratio tends to 1. It should also be noted that the rapid degradation of the EPIC scheme at high compression ratio is most probably due to the use of at most 4 level in the wavelet decomposition.

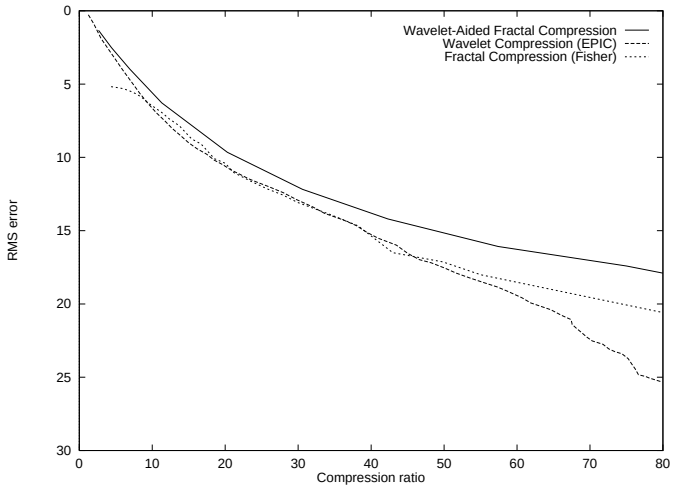

Figure 6: Comparison Between the Efficiency of Various Methods. *The 256x256 "Lena" image was used. Wavelet compression was performed using EPIC, Adelson and Simoncelli's wavelet compression program (see [2]). Fisher's quadtree program (see [3]) was used to generate the curve for conventional fractal compression (the parameters used were* `-m 3 -M 6 -f -d 2 -D 2`*).*

4.3 Reconstruction in a Finite Number of Steps

If the rescale and copy operations involve domain blocks that are bigger than range blocks, we can guarantee convergence in a finite number of steps during the reconstruction process.[3]

To prove this, let us suppose that all wavelet coefficients below some level i are known at some stage of the reconstruction process. Since range blocks are smaller than domain blocks, all wavelet coefficients of level i can be expressed as the rescaled values of some coefficients located below level i. Hence, after one iteration all wavelet coefficients below level $i+1$ are known. Since each iteration reveals one more layer of wavelet coefficients, the number of step needed is determined by the resolution level of the lowest non-stored wavelet coefficient. For example, in Figure 5 we would need at most 4 steps to reconstruct the signal.

4.4 Less Restrictive Convergence Criterion

The fact that reconstruction is accomplished in a finite number of steps lets us relax the usual convergence criterion $|s_j| < 1$. We first note that the reconstruction procedure only involves copying coefficients from the bottom of the pyramid to the top, a procedure which is guaranteed to stop whether each

[3]Assuming, of course, a finite-resolution image.

copying operation is contractive or not.

However, to guarantee that the reconstructed image is close to the original image, we need the Collage Theorem, which relies on the contractivity of our iterated mapping. Again, we can relax this requirement by rewriting the Collage Theorem for the case of a finite number of steps.

Theorem 4 *Let X be a complete normed space and let $F : X \mapsto X$ by a contractive mapping with contraction factor s and fixed point g^*. If, for some $n \in \mathbb{N}$, we have*

$$F^{\circ n}(g) = g^*,$$

for any $g \in X$, then

$$\|g - g^*\| \leq \frac{1 - s^n}{1 - s} \|g - F(g)\|$$

Because of this result, we do not strictly need s to be smaller than 1. Still, the smaller s is, the tighter the bound provided by the Collage Theorem.

A different way to look at the problem is to construct a norm in which F is a contraction.

Proposition 5 *The IFS on the wavelet coefficients introduced earlier (Proposition 3) is a contraction with respect to the norm*

$$\|f\|_\gamma = \sup_{i,j} \left(2^{\gamma i} |\langle \psi_{ij}, f \rangle|\right),$$

provided all gray level scaling factors (s) satisfy

$$|s| < 2^{\Delta i \left(\frac{D}{2} - \gamma\right)},$$

where

- Δi *is the difference in resolution indices between domain and range blocks;*
- D *is the spatial dimension;*
- γ *controls the strength of the norm.*

We can always find a γ small enough so that this inequality is satisfied. The only problem is that the resulting norm might not have good properties. In particular, in the limiting case of an image of infinite resolution, the norm $\|\cdot\|_\gamma$ can be very weak so that convergence to very ill-behaved functions is allowed.

It is interesting to note that quantizing the wavelet coefficients $\langle \psi_{ij}, f \rangle$ with a step size proportional to $2^{-\gamma i}$ corresponds to performing a vector quantization minimizing the norm $\|\cdot\|_\gamma$.

4.5 Limitations

There are three main drawbacks to using the wavelet representation:

- We are restricted to horizontal scaling ratios that are powers of 2.

- Usually, domain blocks are not only scaled, but also flipped or rotated. These operations are straightforward to perform in the wavelet representation only if we use a symmetric or antisymmetric wavelet.
- Not every block can be used as domain block, but only those that cover intervals of the form $[2^{-k}l, 2^{-k}(l+1)]$ with $k, l \in \mathbb{Z}$.

The last limitation is a significant one if we want to compress small images because we are forced to use a small domain pool, which reduces the likelihood of obtaining a match. For this reason, some 256x256 images can still be compressed slightly more efficiently using conventional fractal compression.

5 Speeding up the Search for Matching Blocks

The variance of the numerous high frequency wavelet coefficients is much lower than the variance of the few low frequency coefficients (see [9]). Since low frequency coefficients have a higher variance, they have the biggest influence on the RMS distance between two blocks. We can thus have a good estimate of the actual RMS distance of two blocks by just comparing their few lowest frequency coefficients. In this way we can rapidly discard blocks that have no chance of matching a given range block, saving a lot of computations.

Another way of speeding up fractal image compression is to classify image blocks into categories and only search domain blocks which are in the same category as the target range block (see [8]). The data structures used to store and perform such rapid searches are a fairly standard and well studied subject in computer science (see [5], for example). We shall therefore concentrate our efforts in finding good classification criteria.

We can represent the task of matching domain and range blocks as follows. The range block r is a point in an n-dimensional space where the axes represent the gray level values of all n pixels of the block (Figure 7 illustrates the case of $n = 2$). Let ϵ denote the maximum acceptable RMS collage distance. We now seek a scaling s that will map a domain block d within the ball of radius[4] ϵ centered on r. This can only be done if

$$\alpha \leq \arcsin\left(\frac{\epsilon}{\|r\|}\right) ,$$

where

$$\alpha = \frac{\langle r, d\rangle}{\sqrt{\langle r, r\rangle \langle d, d\rangle}}$$

is the angle between vector r and d and $\|\cdot\|$ denotes the RMS norm. (We assume that $\epsilon < \|r\|$ because otherwise, r can be approximated by the zero vector with an error less than ϵ.)

[4]The radius is measured with the RMS norm and is $\sqrt{n}$ times less than the Euclidian radius.

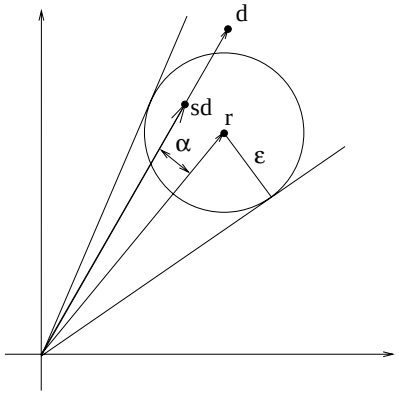

Figure 7: Geometrical Interpretation of the Process of Matching Domain and Range Blocks..

Ideally, we would thus like to convert the n pixel values into generalized spherical coordinates consisting of $(n-1)$ angles and one radius. We then use the angles as indices for a multi-dimensional classification scheme (for example, a tree structure). We could then only search among domain blocks which have the good "orientation". But relating the range of angles through which to search with the matching tolerance ϵ is not an easy task.

An alternative approach is to use the angles $\theta_i(v)$ between a vector v and all axes (the arccosines of the direction cosines). We can then generalize the two dimensional example very easily, at the expense of using n angles instead of $(n-1)$. To select all domain blocks d that have a chance of matching a given range block r with tolerance ϵ, we simply take all domain blocks d such that

$$|\theta_i(d) - \theta_i(r)| \leq \Delta\theta = \arcsin\left(\frac{\epsilon}{\|r\|}\right) \text{ for all } i\ .$$

Figure 8 illustrates this scheme for $n = 3$. In practice, one might want to chose a smaller $\Delta\theta$ to accelerate the search, at the risk of lowering the probability of obtaining a match.

In order to make the classification manageable, we do not want to use all direction cosines. But which one should be chosen? If the coordinates are the pixel values, we have no reason to prefer one angle over another. But if we use wavelet coefficients as the coordinate values, the choice is obvious. We should choose angles associated with wavelet coefficients a_{ij} having a small index i, since they have the highest variances and thus the largest influence on the RMS distance.

It is interesting to note that a classification scheme introduced by Yuval Fisher [3] is related to the method presented above. A part of his classification consists in splitting a block into four quadrants and sorting this block according to the relative average gray level of each quadrant. In the Haar wavelet representation, this essentially amounts to a classification based on the lowest frequency

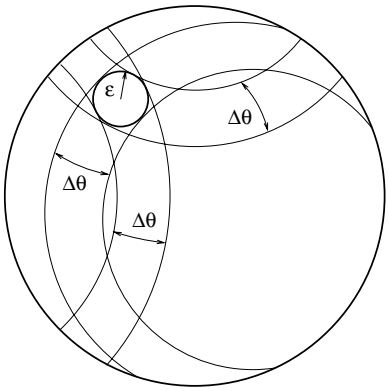

Figure 8: Relating the Span of Searched Angles with Matching Tolerance ϵ. $(\Delta\theta = \arcsin\left(\frac{\epsilon}{||r||}\right).)$

components. The relation between the gray level of each quadrant (*i.e.* which one is the darkest, second darkest, ...) can be interpreted as angular relations in our n-dimensional space.

Classification according to direction cosines has many advantages over this simple scheme:

- we are free to choose other wavelet bases which better concentrate the energy in the low frequency components than the Haar basis;
- we are free to classify the domain blocks in any number of classes;
- we know *exactly* in which classes the matching domain blocks are.

The use of "angles" in an n-dimensional space as classification indices is also very closely related to the *feature vectors* approach of Dietmar Saupe (see [7] and [8]). His classification is based on the value of the inner products of each domain block[5] with a small number of fixed orthogonal unit vectors. The values of all these inner products gives the so-called feature vector. The components of this vector are nothing but the direction cosines in some orthogonal basis. Saupe's scheme differs from the one described in this section only by the following facts:

- We have chosen the arccosines of the direction cosines because this makes the relation between RMS error and classification keys more intuitive and simple.
- We have chosen the set of orthogonal unit vectors to be a truncated wavelet basis. In this way, the feature vector captures most of the variance, that is, the classification keys give us more information about the domain blocks.

[5]normalized to have a zero mean and unit variance

6 Conclusion

In this article, we provided ways to address the various weaknesses of fractal image compression. We developed a new wavelet-aided fractal compression scheme which has some interesting properties.

- Tiling effects can be overcome with little overhead by allowing range blocks to overlap and by filtering them in a way that preserves the orthogonality of the contents of different blocks. Such a filtering can be more easily performed if we express fractal compression as an IFS on the wavelet coefficients. This new algorithm essentially amounts to the storage of all wavelet coefficients below some detail level and the extrapolation of high frequency coefficients by repeated copying and rescaling of the low frequency coefficients.
- "Quadtree" partioning of the image is implemented by simply allowing the threshold between stored and extrapolated coefficient to adapt to the local complexity of the image.
- Reconstruction is achieved in a finite number of steps.
- Strict contractivity is *not essential* to guarantee that the fixed point is close to the original image. (But the error bound is lower if the contractivity factor is small.)
- Efficient indices for classification of the domain blocks can be easily computed from the low frequency wavelet coefficients.
- Even if relatively few good matches are found between domain and range blocks, this scheme "degrades gracefully". As fewer and fewer matches are found, this algorithm gradually tends to wavelet compression.
- If one uses a Haar wavelet decomposition, this algorithm simply reduces to conventional fractal compression.

Acknowledgments

This work was supported by an "1967" Scholarship from the Natural Sciences and Engineering Council of Canada (NSERC), which are hereby gratefully acknowledged. The author also wishes to thank Professor E.R. Vrscay, Applied Mathematics Department, University of Waterloo, whose NSERC Operating Grant partially supported the travel to this conference.

References

[1] Edward H. Adelson, Eero Simoncelli, and Rajesh Hingorani. Orthogonal pyramid transform for image coding. *SPIE*, 845:50–58, 1987.

[2] Edward H. Adelson and Eero P. Simoncelli. Subband image coding with three-tap pyramids. Picture Coding Symposium, Cambridge, MA, 1990. Available through ftp://whitechapel.media.mit.edu/pub/epic.

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

[4] Bruno Forte and Edward R. Vrscay. Solving the inverse problem for function/image approximation using iterated function systems II. algorithm and computations. *Fractals*, 2(3):335–346, 1994.

[5] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1997.

[6] Stéphane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(7):674–693, July 1989.

[7] Dietmar Saupe. Accelerating fractal image compression by multi-dimensional nearest neighbor search. In James Storer, editor, *IEEE Data Compression Conference Proceedings.* IEEE Computer Society Press, March 1995.

[8] Dietmar Saupe and Raouf Hamzaoui. Complexity reduction methods for fractal image compression. In J. M. Blackledge, editor, *I.M.A. Conference Proceedings on Image Processing: Mathematical Methods and Applications.* Oxford University Press, September 1994.

[9] Axel van de Walle. Relating fractal image compression to transform methods. Master's thesis, University of Waterloo (Canada), 1995. Available through http://links.uwaterloo.ca/~axel.