Fast Fractal Image Compression with Triangular Multiresolution Block Matching

D. J. Hebert and Ezekiel Soundararajan
Department of Mathematics, University of Pittsburgh
djh+@pitt.edu, eso+@pitt.edu

Abstract

This paper presents an approach to fractal image compression whereby we reduce the computation to the one dimensional case using a triangular Sierpinski scan path algorithm which results in a binary triangulation tree structure. The scanned image is naturally partitioned into range blocks of contiguous intensities where each block maps to a triangle in the two dimensional image. A preliminary compression is achieved by pruning the binary triangulation tree by combining similar adjacent triangles with a compatibility constraint to ensure that the tree levels of the triangles do not differ by more than one. A wavelet based fractal image compression algorithm is applied to the pruned tree and the parameters are saved for the construction of iterative function systems (IFS) whose fixed point will approximate the pruned image.

1. Introduction

Fractal image compression as developed by Barnsley[2][3] and Jacquin[1] is based on the construction of an iterative function system whose fixed point approximates the image. Jacquin partitions the image into rectangular blocks and searches for rectangular domain blocks, which match the range blocks for the construction of the best local affine map. In this paper we construct iterative function systems for which the partitioning is triangular rather than rectangular. Indeed we use a binary triangulation whose computations are simplified by quadtree locational codes, binary tree indexing, and the automatic reduction of all computation and storage to the one dimensional periodic case (c.f. [6][7][8]). This is quite different from the Delaunay triangulation used by Davoine et al [4]. For block matching we use a method similar to the rectangular Haar-Wavelet pyramid block matching of Zhang and Chen[9] who perform a fast wavelet transform on each domain and range block and match the pattern of wavelet coefficients in the blocks. We construct wavelets in the manner of Donoho's interpolation wavelets and Swelden's lifting scheme[5] in which the wavelet coefficient is the difference between a finer level intensity and a prediction based on interpolation or smoothing.

2. Triangular Scanning

First we reduce computation and storage to the one-dimensional case by scanning the image along a triangular Sierpinski scan path which induces a cyclic ordering on the pixels. This path naturally partitions the image into triangular blocks. For example, if the image is of size $2^m \times 2^m$, then a partitioning of the scanned pixel list by 2^n consecutive blocks of size 2^k (where $k+n=2m$) is equivalent to a uniform binary triangulation of the image consisting of 2^n congruent triangles. The triangular scan algorithm is based on a natural connection between the dyadic partitioning of a Sierpinski scan and the binary triangulations whose triangles are located by quadtree locational codes and binary tree indexing. After the triangular scanning, our computer algorithm partitions the image into triangular range blocks and searches for matching triangular domain blocks of twice the size for the construction of an iterative function system.

3. Block Matching

Instead of directly matching the pixel values of domain and range blocks the program performs a one dimensional fast wavelet transform on the individual range and domain blocks checking for matching of wavelet coefficients, beginning at low resolution(cf[10]). The wavelets are one-dimensional wavelets such as the triangular Haar wavelets based on the dyadic partitionings into triangular blocks. Any standard one-dimensional wavelet will work, but wavelets with two dimensional smoothness and support on the triangular blocks are of special interest. Interpolation wavelets based on linear and polynomial prediction on binary triangulations are particularly convenient for computation. Although our pyramid algorithms are computationally one-dimensional, the wavelets are non-separable from the two-dimensional

point of view. We apply a method which Zhang and Chen used in the case of rectangular partitioning to obtain a phenomenal speed-up by rejecting any match that fails at a low-resolution level. The one dimensional pyramid algorithm transforms the domain and range blocks into $\{L_n(R), H_n(R), H_{n-1}(R), H_{n-2}(R),\ldots, H_0(R)\}$, $\{L_n(D), H_n(D), H_{n-1}(D), H_{n-2}(D),\ldots, H_0(D)\}$ where $L_n(R)$ and $L_n(D)$ are each associated with single values which are the overall averages of the range-block and the domain-block respectively, and that $H_i(R)$, $H_i(D)$ represent the sets of highpass (wavelets) coefficients for the range and domain blocks, respectively.

Let $E(\mu)$ represents the energy of the imagevector μ, i.e. $E(\mu) = 1/N\sum_{i=0}^{N-1} \mu[i]^2$ and define

$$E_k(R\text{-}D) = E(L_n(R) - L_n(D)) + \sum_{i=n}^{k} E(H_i(R) - H_i(D)).$$

In an iterative fashion, we process $E_k(R\text{-}D)$. While it is above a given threshold value reject the match and move on to the next level until we reach the topmost level, at which point we consider the pair matched. Otherwise R and D are not matched and we move on to the next possible pairing. This may produce several possible matches of which we select the best match as the one where the energy is minimum.

Our matching of triangular partition blocks using the triangulation pyramid algorithm has the additional advantage of automatic comparison of rotated, reversed, and quincunx matrix translated, triangular blocks with no geometric computations. Instead of storing the parameters of two-dimensional translation, scaling, intensity offsets, and matrix rotations, the triangulation tree structure allows us to store only dyadic linear offsets, binary tree level, and coarsest scaling coefficient the rotations are automatic and require no computation or storage. The storage of these mapping parameters requires only 20 bits per range block.

4. Tree Pruning

Further improvements in image compression are made by introducing a preliminary pruning of the binary triangulation tree before the construction of the IFS mapping. We use two methods of pruning: For triangular Haar wavelets we combine adjacent triangles whose intensities differ by less than a threshold value. For higher order interpolation wavelets we combine adjacent triangles whose common vertex value is within a threshold value of its prediction. The binary tree structure of the triangulation places an ordering on the triangular cells (which are leaves of a binary tree). This ordering is preserved by the pruning operation.

We impose a compatibility constraint on the triangles, similar to that used in finite-element analysis, namely that a triangle shares an edge with no more than one other triangle. This condition insures that tree levels of adjacent triangles differ by no more than one. If we traverse the leaves of the tree keeping track of the leaf level difference at each step we can reconstruct the tree structure from the list of leaf level differences. Since the differences are -1,0, or 1, we may record each difference as an 3-ary digit. The pruned binary tree is then stored as a linearly ordered memory sequence whose structure is completely represented by the list of 3-ary digits which define an integer value called a leaf level difference key. This pruned tree produces a preliminary compressed one-dimensional representation of the image. A wavelet-based fractal image compression algorithm similar to the one described above is then applied to the pruned image.

5. Results

5.1 Performance Criteria

We define compression ratio as:

$$\frac{\text{The number of bits in the original image}}{\text{The number of bits in the compressed image}}$$

One way to compare two imagevectors μ, μ' is to determine the normalized mean-square error.

$$d^2_{nrms}(\mu,\mu') = \frac{\sum_{i=1}^{i=N}(\mu[i]-\mu'[i])^2}{\sum_{i=1}^{i=N}\mu[i]^2}$$

The Signal-to-Noise-Ratio (SNR) in decibels (dB) corresponding to the above error is:

$$SNR_{dB} = 10\times\log_{10}\frac{1}{d^2_{nrms}(\mu,\mu')}dB$$

We confine our measurements to 8 bits per pixel (bpp) grayscale images, so the peak signal-to-noise-ratio (PSNR) is computed as

$$PSNR = 20\log_{10}\frac{255}{d^2_{nrms}(\mu,\mu')}$$

5.2 Computation of Bit Rates

The complete description of the fractal code is essential for the evaluation of compression ratios. Since we are using triangular partitions, no storage is necessary. Also since range-blocks are referred to by an index, no range-block information need be stored. Domain-blocks are also indexed and referred by this index so one byte is used to store the domain-block index, five bits are used to store the scaling, and seven bits for offset values. Therefore, (8+5+7=20) twenty bits per range block are required for storage of the transformation parameters.

5.3 Without Pruning

For an example of the IFS construction without pruning we divide the one-dimensional imagevector into 256 non-overlapping range blocks of size 256 pixels and 255 overlapping domain blocks of size 512 pixels each. We set a threshold value and find a set of mappings of the domain blocks to the range blocks. The collection of maps is used to decode the image. Figure 1 shows the original image and Figure 2 the decoded image. The result is surprisingly very good. The original image required 65,536 bytes of storage, whereas, the transformations required 640 bytes, giving a compression ratio of 102:1.

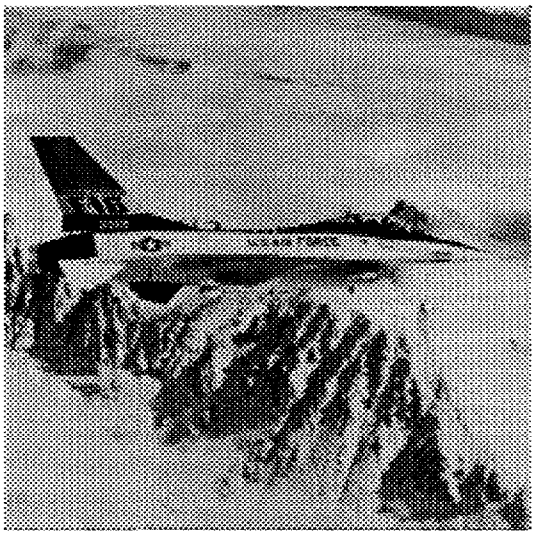

Figure 1

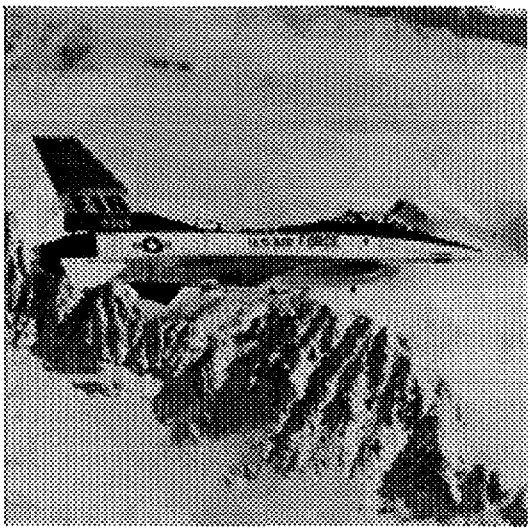

Figure 2

Values of the PSNR between the original image and the successive terms of the reconstructed sequence just described are listed in table 1.

5.4 Example with Pruning

Figure 3 shows the reconstructed image obtained from a similar compression method with pruning. The threshold value for Haar wavelet pruning is 60 pixel intensity levels and the range block size is 64. The compression ratio is 252:1. The severe triangular blocking in the smooth areas is due to the large threshold value, but the edges are better with pruning than without. Figure 4 shows results from the same algorithm with a threshold of 40 and a range block size of 128. The compression ratio is 420:1.

Table 1.

Iter#	1	2	3	4
PSNR	23.92	23.97	23.99	24.00

Iter#	5	6	7	8
PSNR	24.01	24.01	24.01	24.01

Figure 3

Figure 4

5.5 Other tests

Prototype programs written in Java, running on 100-200MHz Pentium processors, without optimization, compress the image within a few minutes, including all preprocessing, and reconstruct using 3-7 iterations of the IFS mapping within a few seconds. Current experiments are concerned with eliminating triangular blocking by using higher-order wavelets which require additional computing time. Since some wavelets give better visual results than others, we seek a compromise between smoothness of large area triangle interpolation as determined by wavelet characteristics and computational speed adjusted by parameters such as range block size and threshold values.

6. References

[1] A. E. Jacquin. Image Coding Based on a fractal theory of Iterated Contractive Image Transformation. IEEE Transaction on Image processing, 1991,1, 1.

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

[3] M. F. Barnsley. Fractals Everywhere. Academic Press, 1993

[4] F. Davoin, M. Antonin, J. M. Chassery and Barlaud. Fractal image compression based on Delaunay triangulation vector quantization. IEEE Trans. On Image Processing, 5:338-346 1996

[5] Wim Sweldens and Peter Shroeder. Building Your Own Wavelets at Home. In Wavelets in Computer Graphics, ACM SIGGRAPH Course Notes, 1996, 15-87.

[6] D. J. Hebert. Cyclic interlaced quadtree algorithms for quincunx Multiresolution. Journal of Algorithms, 27, 97-128.

[7] D. J. Hebert and HyungJun Kim. Image Encoding with Triangulation Wavelets. Proc. SPIE, Wavelet Applications in Signal and Image Processing, 2569, 1995, 381-392.

[8] D. J. Hebert and HyungJun Kim. A Fast Wavelet Compass Edge Detector. 1996, Proc. SPIE, Wavelet Applications in Signal and Image Processing, 2825, 432-442.

[9] Dan Zhang and Gang Chen. A Fast Fractal Image Coding Based on Multiresolution Analysis. Preprint, 1997.

[10] D.J.Hebert and Ezekiel Soundararajan. Fast Fractal Image Compression with Triangulation Wavelets. Proc. SPIE Conf. on Wavelet Applications in Signal and Image Processing,1998,toappear.