

Comparison Study of Compression techniques

By: Navya Sree Kolluri

**Fractal Image Compression & Wavelet compresson: A Comparision Study.**

**Kolluri , Navya Sree**

**Georgia State University**

**Department of Computer Science**

**1.Introduction:**

With the advancement of information technology there is great need for fast communication. There is great need to reduce the memory needed to store the image. These facts justify the efforts, to find image compression algorithms. In this project I would like to compare the image quality and file size of various image compression techniques.

Fractal image compression were first explored during the 1980’s, they were considered to be the future of image processing. But recently, advancements in wavelet –based compression techniques have all but rendered current fractal methods obsolete. In this paper in addition to understanding the history, concepts of development of fractal-based compression and wavelet compression, I will be comparing the results of compression obtained from both these techniques. The history of fractal based image compression taken together with the results of the study, will help us understand why fractal compression methods have not been successful at becoming image compression standards.

**2.Background of Fractal Image Compression & Wavelet Compression:**

Fractal Compression was first promoted by M.Barnsley, who founded a company based on fractal image compression technology but who has not released details of his scheme. Michael Barnsley, a leading researcher from Georgia Tech, wrote the popular book “Fractals Everywhere”. The book presents the mathematics of Iterated Functions Systems (IFS), and proves a result known as the Collage Theorem. The Collage Theorem states what an Iterated Function System must be like in order to represent an image.

Barnsley, however, armed with his Collage Theorem, thought he had it solved. He applied for and was granted a software patent and left academia and found Iterated Systems Incorporated. Barnsley announced his success to the world in the January 1988 issue of BYTE magazine. The images were given suggestive names such as "Black Forest" and "Monterey Coast" and "Bolivian Girl" but they were all manually constructed but . Barnsley's patent has come to be derisively referred to as the "graduate student algorithm." Though these manually constructed images achieved a compression ratio of 1000:1 and requires about 100 hours for each image to be encoded and 30minutes to decode.

Ironically, it was one of Barnsley's PhD students that made the graduate student algorithm obsolete. In March 1988, according to Barnsley, he arrived at a modified scheme for representing images called Partitioned Iterated Function Systems (PIFS). The algorithm was not sophisticated, and not speedy, but it was fully automatic. This came at price: gone was the promise of 10,000:1 compression. A 24-bit color image could typically be compressed from 8:1 to 50:1 while still looking "pretty good".

Let’s investigate the history of wavelet compression. A wavelet is a mathematical function useful in digital signal processing and image compression. The use of wavelets for these purposes is a recent development, although the theory is not new. The principles are similar to those of Fourier analysis, which was first developed in the early part of the 19th century.

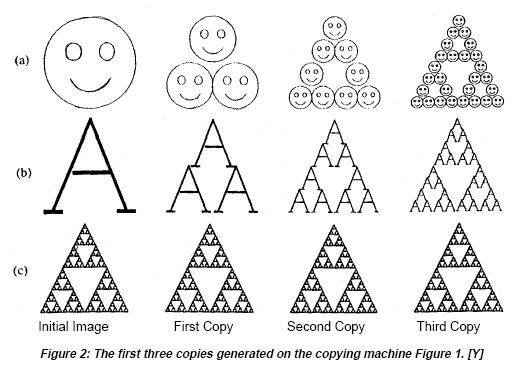
The modern wavelet transform was first developed in 1981 by Jean Morlet and Alex Grossman. Since then a complete theory has been developed and many applications have been explored. Solving partial differential equations, analyzing Brownian motion , artist identication ,are just a few applications that can involve wavelets.

**3.Concept of Fractal compression:**

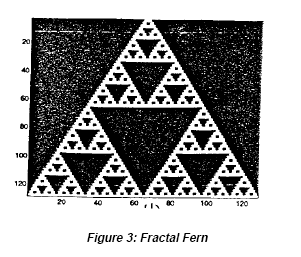
Fractal image compression is a lossy compression method. The final image is accurate enough to be acceptable with the original image. This method reduces the amount of data needed to store an image by exploiting the similarities with in that image. There are other compression techniques but however fractal compression claims to have better performance in that it produces an approximation that is closer to the original image. This technique uses a mathematical concept called Iterated function system. This system is a finite set of S of function being associated with probability. So every point in the image is applied with the one function from the set based on its probability. This is the basic idea of the implementation of fractal image compression.

Barnsley suggested that perhaps storing images as collections of transformations could lead to image compression. In practice, choosing transformations of the form is sufficient to generate interesting transformations called affine transformations of the plane. Each can skew, stretch, rotate, scale and translate an input image.

A common feature of these transformations that run in a loop back mode is that for a given initial image each image is formed from a transformed (and reduced) copies of itself, and hence it must have detail at every scale. That is, the images are fractals. This method of generating fractals is due to John Hutchinson.



Barnsley suggested that perhaps storing images as collections of transformations could lead to image compression. His argument went as follows: the image in Figure 3 looks complicated yet it is generated from only 4 affine transformations.

Each transformation wi *i*s defined by 6 numbers, *ai, bi, ci, di,*

**3.1 Why The Name Fractal:**

The scheme will encode an image as a collection of transforms that are very similar to the copy machine metaphor. Just as the fern has detail at every scale, so does the image reconstructed from the transforms. The decoded image has no natural size; it can be decoded at any size. The extra detail needed for decoding at larger sizes is generated automatically by the encoding transforms.

**Properties of Fractals:**

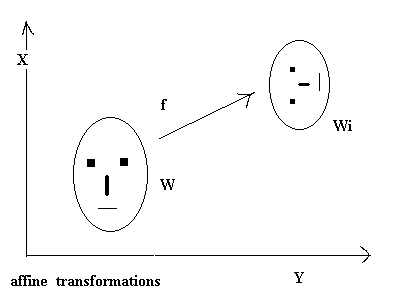
A set F is said to be a fractal if it possesses the following properties.

* F is found to contain detail at every scale.
* F is self-similar.
* The fractal dimension of F is greater than it’s topological dimension.
* F has got a simple algorithmic description.

**Affine Transformations:**

An affine transformation is any [transformation](http://mathworld.wolfram.com/Transformation.html) that preserves [collinearity](http://mathworld.wolfram.com/Collinear.html) and ratios of distances. Geometric contraction, expansion, dilation, reflection, rotation, shear, similarity transformations, spiral similarities, and translation are all affine transformations, as are their combinations. In general, an affine transformation is a composition of rotations, translations, dilations, and shears. While an affine transformation preserves *proportions* on lines, it does not necessarily preserve angles or lengths.

These are combinations of rotation, scaling and translation of the co-ordinate axis in an N-dimensional space. The figure shows an example of an affine transformation W which moves towards W(f)-that is it is a contractive transformation.



**3.2 Encoding Images:**

The theorem tells us that transformation *W* will have a unique fixed point in the space of all images. That is, whatever image (or set) we start with, we can repeatedly apply *W* to it and we will converge to a fixed image.

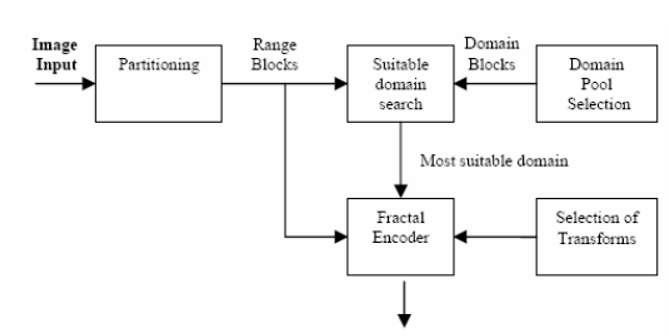
Suppose we are given an image *f* that we wish to encode. This means we want to find a collection of transformations *w1, w2, ...,wN* and want *f* to be the fixed point of the map *W*. In other words, we want to partition *f* into pieces to which we apply the transformations *wi* , and get back the original image *f* .

A typical image of a face does not contain the type of self-similarity like the fern. The image does contain other type of self-similarity. Figure 5 shows regions of Lena identical, and a portion of the reflection of the hat in the mirror is similar to the original.

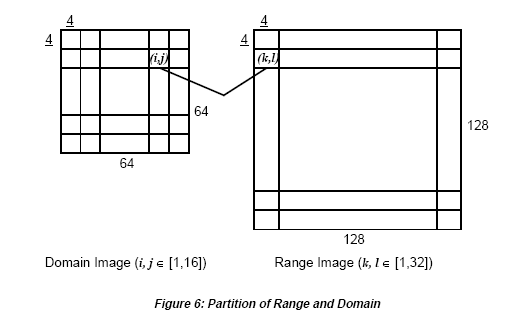
Original Image Self Similarity portions of the image

**3.3. Proposed Algorithm:**



**3.4 Fractal Encoding:**

The following example suggests how the Fractal Encoding can be done. Suppose that we are dealing with a 128 x 128 image in which each pixel can be one of 256 levels of gray. We called this picture Range Image. We then reduce by averaging (down sampling and lowpass-filtering) the original image to 64 x 64. We called this new image Domain Image. We then partitioned both images into blocks 4 x 4 pixels (see Figure 6).



We performed the following affine transformation to each block:

In this case we are trying to find linear transformations of our Domain Block to arrive to the best approximation of a given Range Block. Each Domain Block is transformed and then compared to each Range Block *Rk,l* . The exact transformation on each domain block, i.e. the determination of a and *to* is found minimizing

where *m, n, Ns* = 2 or 4 (size of blocks).

Each transformed domain block G*(Di,j*) is compared to each range block *Rk,l* in order to find the closest domain block to each range block. This comparison is performed using the following distortion measure.

Each distortion is stored and the minimum is chosen. The transformed domain block which is found to be the best approximation for the current range block is assigned to that range block, i.e. the coordinates of the domain block along with its a and *to* are saved into the file describing the transformation. This is what is called the Fractal Code Book.

**3.5 Fractal Decoding:**

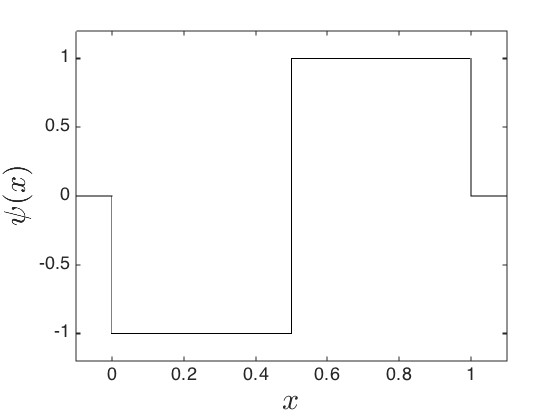
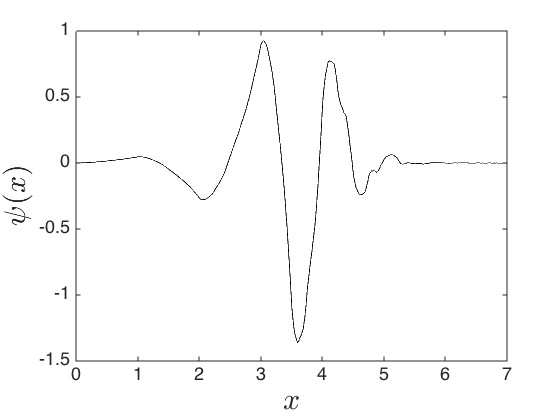
The reconstruction process of the original image consists on the applications of the transformations describe in the fractal code book iteratively to some initial image Winit*,* until the encoded image is retrieved back. The transformation over the whole initial image can be described as follows:

can be expressed as two distinct transformations:

represents the down sampling and low-pass filtering of an image to create a domain image e.g. reducing a 128x128 image to a 64x64 image as we describe previously. represents the ensemble of the transformations defined by our mappings from the domain blocks in the domain image to the range blocks in the range image as recorded in the fractal. *n* will converge to a good approximation of *orig* in less than 7 iterations.

**4. Concept of Wavelet Compression:**

From a given image, the goal of the compression is to minimize the sequence of bits needed to represent it, while preserving information of acceptable quality. Wavelets address this problem by decomposing an image in a manner that acts like the human visual system.



1. **Haar Mother wavelet**  **b) Daubechies 4 mother wavelet**

The function ψ(x) is a mother wavelet and {ψj,k }j,k∈Z is called a wavelet basis. Examples of two different mother wavelets are given in figure 1. The Haar mother wavelet is given by ψ(x) := −χ[0,1/2) (x) + χ[1/2,1) (x) and is shown in figure 1a. The Haar wavelet is the simpliest known wavelet. The Daubechies 4 mother wavelet is shown in figure 1b. These Daubechies wavelets will be used in the numerical experiments in this paper.

**4.1 Wavelet based image compression technique**

There are three main components to image compression: **transform, quantize and encode**. Applying a transform should remove some of the redundancy in the data by hopefully resulting in mostly zero values. Quantization reduces the precision of the transformed data, which leads to rounding errors. The final step of encoding aims to store the quantized data compactly.

It has been suggested that the wavelet transform acts similar to human visual system. A wavelet transform results in many zero or near zero grayscale values. A thresholding step is usually applied to the transformed data, that is any coefficient smaller than some prescribed value is set to zero. Thresholding is acceptable since many values near zero represent insignificant data to the human visual system. The level of thresholding, T , is a parameter that can be adjusted by the user.

Quantization is the only step that loses information. By reducing the precision of the wavelet transform rounding errors are incurred and information is lost. This step is of course necessary for effective compression of the data. One simple approach to quantization is to represent the wavelet transform with nearest integer values. Integers are stored using a smaller number of bits than floating point numbers. The amount of storage is therefore decreased.

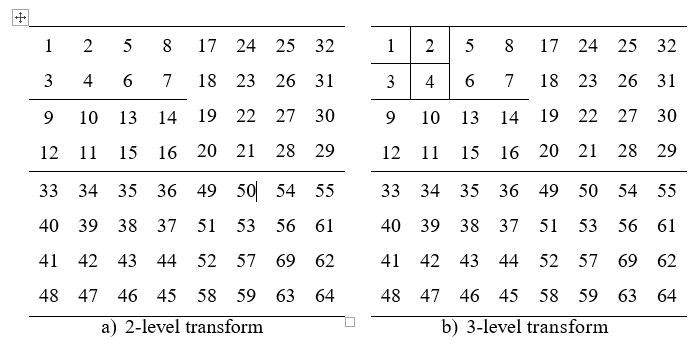
Location of nonzero values in the quantized data is of importance in compression. Encod- ing the data tries to represent the location of nonzeros compactly. The relationship between the quantized data and encoding is the crucial aspect of wavelet based image compression. Each wavelet based compression algorithms explores this relationship differently.

**4.2 Embedded ZeroTree Wavelet Algorithm:**

The embedded zerotree wavelet (EZW) algorithm is more complicated then the baseline above. The description here follows that of Walker and Nguyen. The complexity of the EZW algorithm is justified since it remedies the deficiencies of the baseline algorithm. That is, the EZW algorithm has the ability to specify a compression rate or a desired error. The EZW algorithm can also progressively transmit successive resolutions of the image.

The EZW algorithm was one of the first wavelet based compression algorithms to show the power of using wavelets. Shapiro first described the EZW algorithm in 1993. The method introduces fundamental concepts that more superior methods utilize. One now scans through the wavelet coefficients in a strategic manner when determining s(j, k). Namely, zigzag through approximation spaces and diagonal details, column scan through vertical details and row scan through horizontal details. Figure 3 depicts the scan order for 2-level and 3-level transforms of an 8 by 8 pixel image.

The next step is quantization. Quantization is accomplished using an embedded coding. An embedded coding allows for progressive transmission of the coefficients W f (j, k). Using an embedded coding for quantization allows the user to recieve more accuracy in W f (j, k) without running the algorithm again from the start. The embedded coding used by the EZW algorithm is called bit-plane coding.



Example:

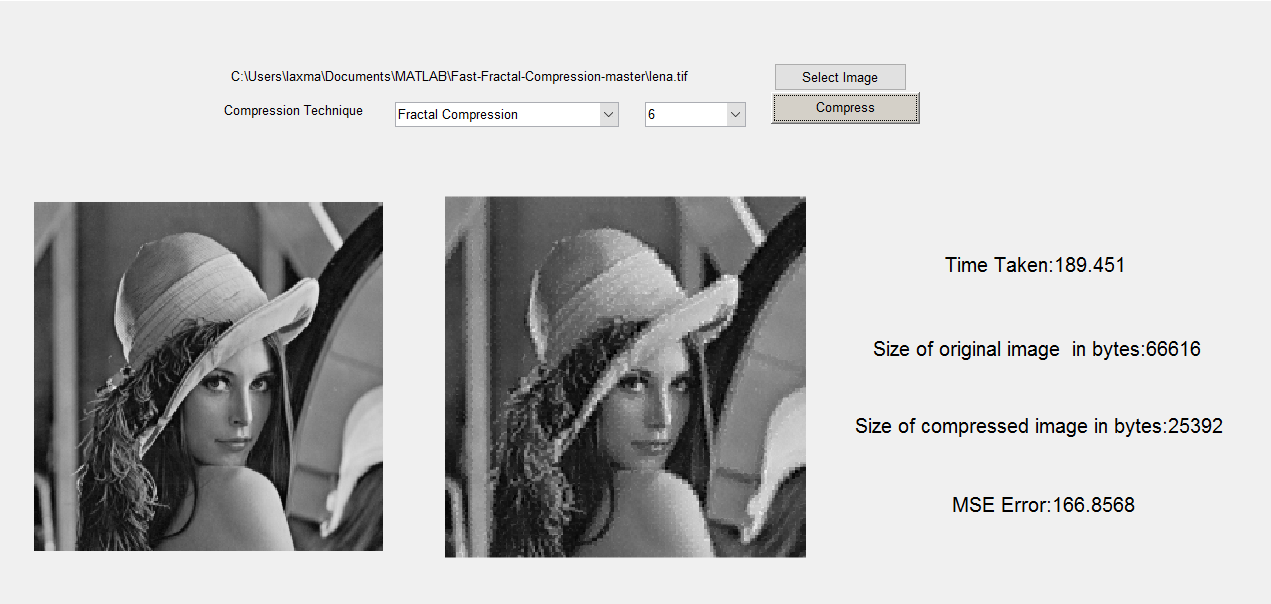
Suppose we have just two coefficients w = −9.5 and v = 42. Consider setting the initial threshold as T0 = 64. During the first loop T1 = 32, the output is the sign of v and quantized coefficient magnitudes wQ = 0 and vQ = 32. Second loop has T2 = 16, the refinement pass produces the bit 0. No output is obtained from the significance pass of the second loop. The quantized coefficient magnitudes are still wQ = 0 and vQ = 32. The third loop with T3 = 8, the significance pass outputs the sign of w. The refinement pass outputs the bit 1 because v ∈ [32 + 8, 32 + 16). The quantized coefficient magnitudes are wQ = 8 and wQ = 40.

Zerotrees can give a very compact description of the locations of insignificant coefficients. It is only necessary to encode one symbol, say R, to denote the root of a zerotree. The de- coder can then locate all other insignificant coefficients in the zerotree. Obviously these zerotrees are only useful if they occur frequently. With the wavelet trans- form of natural scenes, it is fortunate that the multiresolution structure produces many zerotrees. The zerotrees are incorporated in the EZW compression algorithm through the significance pass.

**5. Experimental Results for fractal and Wavelet Compression:**

**Fractal Image Compression for Lena:**

Now that the methods have been explained, some numerical examples of a real image are given. I have developed GUI with matlab to compare the compression of images using both the techniques. Its called compression.m. On executing of that file GUI would look like image given below. Select image button helps us with the selection of image for compression. Select fractal compression and select compress to perform fractal image compression. Similarly wavelet compression can be performed. These numerical experiments are carried out using the popular example in image compression, Lena.



1. **Fractal compression of Lena . Compression ratio 2.5:1**

As it was already explained above . Fractal Image compression takes really long time. It took above 189 seconds for the compression to take places.

Now lets perform EZW wavelet compession on Lena for various iterations. Wcompress function has a parameter for the selection of wavelet compression method it can either be EZW,SPIHT and baseline. Another parameter to choose the mother wavelet it can either be Haar wavelet, Daubechies 4 wavelet. In this experiment we use Daubechies 4 wavelet(db2).

Following are the images after application of wavelet compression from 8th iteration



a)Original image b) 8 iterations c) 9 iterations d) 10 iterations

size=66kb size = lkb size =2kb size = 6k

mse = 341 mse = 148 mse =62.5

e) 11 iterations f) 12 iterations

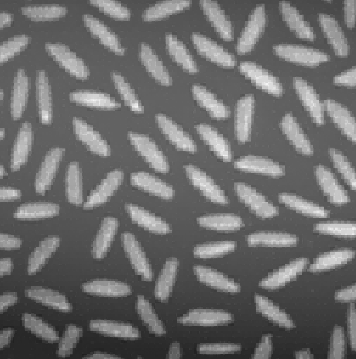
size =11kb size = 19kb

mse = 23.7 mse = 9.4

Comparing fractal compression and wavelet comparision . wavelet is better than fractal compression as it takes very less time to perform fractal compression, with minimum error and obtains better compression ratio. These were the reasons why fractal compression has become obsolete. Fractal compression is not very flexible it works with square images. Smaller images take less time for compression.

Let compare with another picture.

**Fractal Compression**

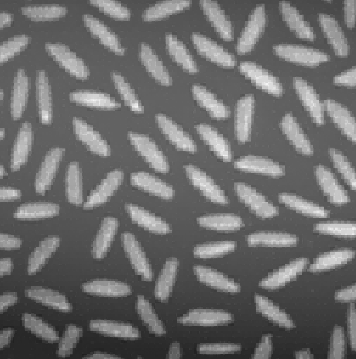
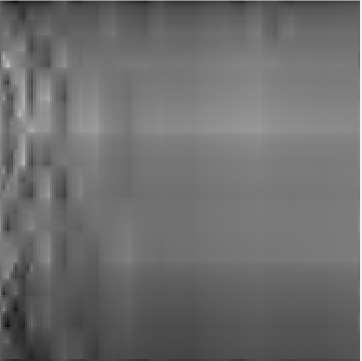
 

1. Original b) Compressed

Size = 9kb size = 2k

Mse = 9.4

**EZW Wavelet compression with Rice for various iterations**



original Iterations = 6 Iterations = 7 Iterations = 12

Size = 276b size = 450b size = 2k

Mse = 1382.25 Mse = 1361 Mse = 50

EZW Wavelet compression using cameraman: For all the iterations it didn’t take more than 3 seconds



Original Image 8 iterations 9 Iterations 10 Iterations

Size = 12k Size = 3k Size =8k Size = 18k

Mse = 335 Mse = 150.16 Mse = 62.7

**6. Conclusion:**

I have used a new fast fractal encoding algorithm using variance of image blocks. The experimental results showed that the numbers of searched domain blocks are reduced to less than a quarter of that of the full search while maintaining the quality of the decoded image exactly the same.

Image compression is a successful application of wavelet. The EZW algorithm have more accurate quantization that can be progressively transmitted. These algorithms outperform the fractal image compression technique in terms of time for processing, compression ratio and error compared with the original.