International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 1, Issue 4(December 2012), PP.07-15
www.irjes.com

Fractal Image Compression With Spiht Algorithm

Mr.Pratyush Tripathi, Ravindra Pratap Singh

[1]*(Assistant Prof. Electronics&Communication Engg,Deptt.)*
Kanpur Institute Of Technology, Kanpur
[2]*(Research Scholar) Kanpur Institute Of Technology, Kanpur*

I INTRODUCTION

Fractal image compression is a comparatively new technique which has gained considerable attention in the popular technical press, and more recently in the research literature. The most significant advantages claimed are high reconstruction quality at low coding rates, rapid decoding, and "resolution independence" in the sense that an encoded image may be decoded at a higher resolution than the original. While many of the claims published in the popular technical press are clearly extravagant, it appears from the rapidly growing body of published research that fractal image compression is capable of performance comparable with that of other techniques enjoying the benefit of a considerably more robust theoretical foundation.

So called because of the similarities between the form of image representation and a mechanism widely used in generating deterministic fractal images, fractal compression represents an image by the parameters of a set of affine transforms on image blocks under which the image is approximately invariant. Although the conditions imposed on these transforms may be shown to be sufficient to guarantee that an approximation of the original image can be reconstructed, there is no obvious theoretical reason to expect this to represent an efficient representation for image coding purposes. The usual analogy with vector quantisation, in which each image is considered to be represented in terms of code vectors extracted from the image itself is instructive, but transforms the fundamental problem into one of understanding why this construction results in an efficient codebook.

The signal property required for such a codebook to be effective, termed "self-affinity", is poorly understood. A stochastic signal model based examination of this property is the primary contribution of this dissertation. The most significant findings (subject to some important restrictions) are that "self-affinity" is not a natural consequence of common statistical assumptions but requires particular conditions which are inadequately characterised by second order statistics, and that "natural" images are only marginally "seif-affine", to the extent that fractal image compression is effective, but not more so than comparable standard vector quantisation techniques.

1.1 Fractal compression

The fundamental principle of fractal coding is the representation of a signal by the parameters of a transform under which the signal is approximately invariant. This transform is constructed so that it is contractive (D); Banach's fixed point theorem guarantees that an approximation to the original signal, called the fixed point of the transform, may be recovered by iterated application of the transform to an arbitrary initial signal. Although a more accurate description would be "fixed point coding" this form of coding is termed "fractal" since the iterative decoding process creates detail at finer scales on each iteration, with the result that the fixed point signal is, in principle at least, a fractal.

.

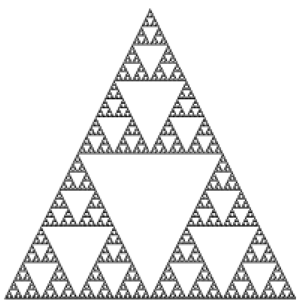


Figure 1.1 : The Sierpinski Gasket.

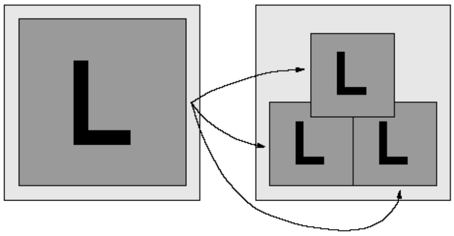


Figure 1.2 : IFS for the Sierpinski Gasket

The origins of fractal coding may be traced to Barnsley's work with Iterated Function Systems (IFS) for image modelling. An IFS is, to simplify somewhat, a collection of contraction mappings which are all applied to the same objects in a metric space. The collection of mappings taken together constitute a "super-mapping", which being contractive, has a unique fixed point. In the simplest examples, a binary image is represented by the set of all pixel coordinates of non-zero pixels, and the individual mappings are affine mappings in the Euclidean plane. The Sierpinski Gasket shown in Figure 1.2 is the fixed point of an IFS consisting of the three mappings of the image as a "collage" of transformed versions of itself. The Sierpinski Gasket is in fact a fractal, since every iteration of its generating IFS adds smaller triangles as a finer scale; the image in Figure 1.3 is only a finite-resolution approximation to the real Sierpinski Gasket.

An IFS generating a desired image may be found by "covering" sections of the image by transformed version of the entire image, resulting in a set of transforms which leave the image approximately invariant. The collage theorem implies that the fixed point of the IFS composed of these transforms will be close to the original image. Another well known example of an IFS-generated image is Barnsley's fern, displayed in Figure 1.3 which is the fixed point of an IFS consisting of four affine mappings

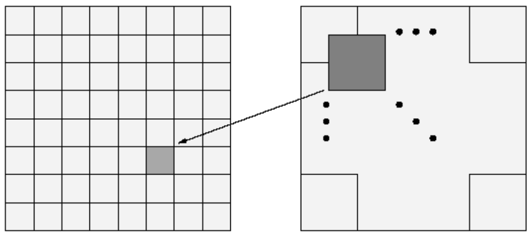
Figure 1.3 : Barnsley's fern.

The success of IFS modelling of natural images (eg. Barnsley's fern) in conjunction with the compactness of the resulting image representation prompted Barnsley to investigate the use of IFSs for image coding. Despite claims of 10000:1 compression ratios the decoded images in question are more appropriately described as the result of image modelling than image coding. In addition, all of the images were "coded" by a human operator assisted process, with no known automatic procedure for the "inverse problem".

Most current fractal coding schemes are based on representation by a Partitioned IFS (PIFS) a solution to the inverse problem of which was first published by Jacquin and subsequently patented by Barnsley. A PIFS differs from an IFS in that the individual mappings operate on a subset of the image, rather than the entire image. Instead of each iteration of the transform copying transformed version of the entire image to a new image, each transform operates only on a subregion of the image, commonly referred to as "domain blocks" due to their role in the mappings. The image subregions to which the domain blocks are mapped are called "range blocks" for similar reasons. In coding of greyscale (as opposed to binary) images, the image is represented as function on the Euclidean plane, where the height of the surface at each point represents the local pixel intensity. In this representation a transform on a domain block may separately transform the block support and the block intensities'.

The first step in a simple implementation is to tile the image by non- overlapping range blocks (eg. 8×8) and larger (eg. 16×16), possibly overlapping domain blocks. A set of admissible block transforms is defined, consisting of a contraction of the block support by a factor of two on each side by averaging neighbouring pixels, followed by the application of one of the eight rotations and reflections (see Figure 1.4) making up the isometries of a square, and finally an affine transform on the pixel intensities (see Figure 1.5-1.8).

The encoding phase (once again utilising the collage theorem) consist of finding for each range block a domain block for which the pixel values can be made close to those of the range block by the application of an admissible transform. Care must be taken in selecting these transforms so that their union is a contractive transform on the image as a whole. The pool of domain blocks is often referred to as the self- or virtual codebook, since collage theorem based encoding is equivalent to Mean Removed Gain Shape VQ, encoding with a codebook consisting of domain blocks extracted from the image to be encoded. The distortion measured during VQ encoding, resulting from the errors in covering the image with codebook blocks, is the same as the actual distortion obtained on decoding. This is not the case for fractal coding, since any error in covering a particular range block modifies the domain blocks with which it intersects, which is not taken into account during the usual encoding process.



Partition Scheme (Range Blocks) Virtual Codebook (Domain Blocks)

Figure 1.4 : Domain and range blocks in PIFS coding.

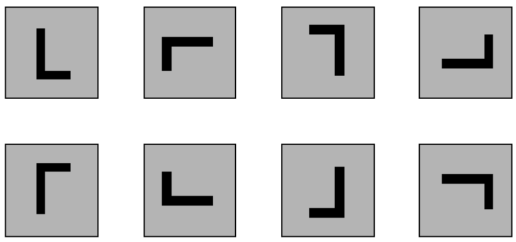


Figure 1.5 : The square isometries.

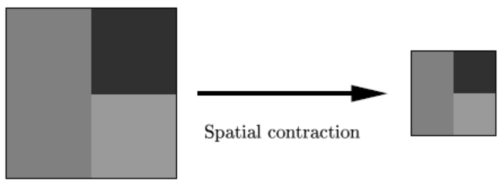
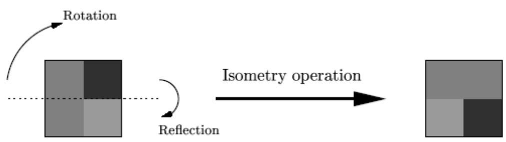


Figure 1.6 : Spatial contraction of a domain block.



Contracted domain block

Rotated and contracted domain block

Figure 1.7 : An isometry applied to a domain block.

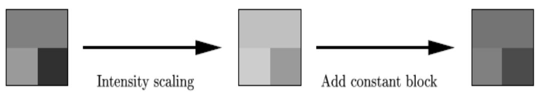


Figure 1.8 : An affine transform applied to a domain block.

The collage theorem nevertheless guarantees that the actual error on decoding may be made arbitrarily small, by making the collage error in covering each range by a transformed domain sufficiently small.

Once encoding is complete, the image is represented by a list containing the selected domain block and transform parameters for each range block. The image is decoded by iteratively transforming an arbitrary initial image using the transform consisting of the union of the transforms for each range block.

Fractal image compression is described in considerably greater detail in the following chapter, in which a broad overview of the fractal compression literature is presented.

II WAVELET IMAGE COMPRESSION

Generic wavelet based image compression techniques exploit the fact that the wavelet transform concentrates most of the energy of the image in a relatively small number of coefficients. The strategy is as follows: An optimal threshold for the coefficients is computed in such a way that a certain percentage of the energy of the image is preserved after compression. Then, coefficients with values below the threshold are deemed to be insignificant and forced to zero, while the rest of the coefficients are quantized and encoded in a refined fashion. For typical images, most of the energy of the image is generally localized in a relatively few coefficients, hence most of the coefficients can be insignificant and discarded, resulting in a some degree of compression. However, more sophisticated wavelet compression techniques can outperform this generic approach. These methods exploit the characteristics and structure of the wavelet decomposition tree in order to locate the significant coefficients.

Locating the Significant Coefficients

The discrete wavelet transform attempts to produce coefficients that are decorrelated with most of the energy of the image localized in a relatively few coefficients, as compared to the spatial distribution of the pixels in the original image. For a typical real-world image, the image is composed of mainly “trends” or relatively smooth areas where neighboring pixels are highly correlated. However, the most important features of the image in terms of the human perception lie in the edges and boundaries of the image. These features have lower cumulative energy than the rest of the image, however they contain perceptual significance that is far greater than their numerical energy contribution to the image. The wavelet transform attempts to separate these two main features of the image and localize them at various scales and in three different subbands. Typically, most of the energy of the image is localized in the lowest frequency components of the image (top left-corner of the wavelet decomposition tree), whereas most of the edge information or high frequency components of the image are scattered in the higher scales of the wavelet decomposition tree. Thus, the fine details or the high frequency components (edges) of the image constitute the most important perceptual characteristics of the image and they are often scattered among a large number of insignificant coefficients. Hence, if not done efficiently, this may represent a problem for wavelet-based image coding methods, as most of the bit budget may be spent in representing and coding the position of those few coefficients corresponding to significant edges or fine details. The challenge in wavelet-based image coding methods is how to efficiently locate these high-information coefficients and representing the positions of the significant wavelet coefficients.

There are many wavelet-based image compression methods, but most of them only differ in the way they locate and encode the significant coefficients. Two of the most efficient wavelet based image coding methods are the Embedded Zerotrees of Wavelet (EZW) method and the Set Partitioning in Hierarchical Trees (SPIHT) scheme, which are discussed briefly next.

Efficient Wavelet Image Coding Schemes

Over the past decade, many efficient wavelet-based image compression schemes have been developed. Two of the best wavelet image compression schemes, widely known as the Embedded Zerotrees Wavelet (EZW) and the Set Partitioning in Hierarchical Trees (SPIHT) algorithm. In 1993, Shapiro proposed the use of a special structure called zero tree for the purpose of locating and encoding the significant wavelet coefficients. The embedded zerotree wavelet algorithm (EZW) is a simple yet remarkably effective image compression algorithm, having the property that the bits in the bit stream are generated in order of importance, yielding a fully embedded code. This highly efficient wavelet-based image compression scheme is based on the following significance hypothesis:

If a wavelet coefficient at a coarse scale is insignificant with respect to a threshold then all of its descendants are also insignificant.

The embedded code represents a sequence of binary decisions that distinguish an image from the “zero” image.

In 1996, Said and Pearlman proposed an enhanced implementation of the EZW algorithm, known as the Set Partitioning in Hierarchical Trees (SPIHT). Their method is based on the same premises as the EZW algorithm, but with more attention to detail. The public domain version of this coder is very fast, and improves the performance of the EZW by 0.3-0.6 dB. Next, the main features of the SPIHT scheme are summarized and its performance is assessed.

The Main Features of the SPIHT Algorithm

In summary, the SPIHT algorithm partitions the wavelet coefficients into three sets: list of significant pixels, list of significant sets, and list of insignificant sets. By using this structure and conditionally entropy encoding in these symbols, the coder achieves very good rate-distortion performance. In addition, the SPIHT coder also generates an embedded code. Coders that generate embedded codes are said to be have progressive transmission or successive refinement property. Successive refinement consists of first approximating the image with a few bits of data, and then improving the approximation as more and more information is supplied. An embedded code has the property that for two given bit rates: $R_1 \geq R_2$, the rate R_2 code is a prefix to the rate R_1 code. Such codes are of great practical interest for the following reasons:

- The encoder can easily achieve a precise bit-rate by continuing to output bits until it reaches the desired bit-rate.
- The decoder can cease decoding at any given point, generating an image that is the best representation possible with the decoded number of bits. This is of practical interest in many applications, including broadcast applications where multiple decoders with varying computational, display and bandwidth capabilities attempt to receive the same bit-stream. With an embedded code, each receiver can decode the passing bit-stream according to its particular needs and capabilities.
- Embedded codes are also useful for indexing and browsing, where only a rough approximation is sufficient for deciding whether the image needs to be decoded or received in full. The process of screening images can be sped up considerably by using embedded codes.

The SPIHT method generates an embedded code by using a bit-slice approach. First the wavelet coefficients of the image are indexed into a one-dimensional array, according to their order of importance. This order places lower frequency bands before higher frequency bands since they have more energy, and coefficients within each band appear in a raster scan order. The bit-slice code is generated by scanning this one-dimensional array, comparing each coefficient with a threshold T. This initial scan provides the decoder with sufficient information to recover the most significant bit slice. In the next pass, new information about each coefficient is refined to a resolution of T/2, and the pass generates another bit slice of information. This process is repeated until there are no more slices to code.

The SPIHT algorithm is indeed embedded, progressive and computationally efficient. Figure 2.1 illustrates some typical SPIHT representation of the test image compressed at pre-determined bit-rates as well as the rate distortion performance of the SPIHT method.

In this section, a brief outline of the practical implementation of the DWT for the purpose of image compression is given. In particular, the main features of the SPIHT method, which is one of the most effective wavelet-based image codec, are described. Next, the hybrid fractal-wavelet scheme which combines the fractal and the wavelet transforms studied so far, is studied.

(a) RMSE = 5.61, PSNR=33.15 DB, C_R=40:1 Execution time ≈ 55 secs

(a) RMSE = 3.93, PSNR=36.24 DB, C_R=20:1 Execution time ≈ 51 secs

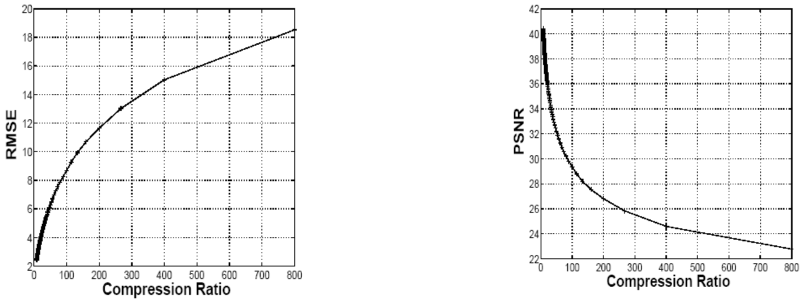


Figure 2.1 : Results of the SPIHT compression algorithm for the image of "Lenna": (a)-(b) illustrate SPIHT compressed images and (c)-(d) illustrate the rate distortion performance of the SPIHT

2.1 Generalized 2D Fractal-Wavelet Transforms

Fractal-wavelet transforms, discovered independently by a number of researchers to name only a few), were introduced in an effort to reduce the blockiness and computational complexity that are inherent in fractal image compression. Their action involves a scaling and copying of wavelet coefficient subtrees to lower subtrees, quite analogous to the action of fractal image coders in the spatial domain.

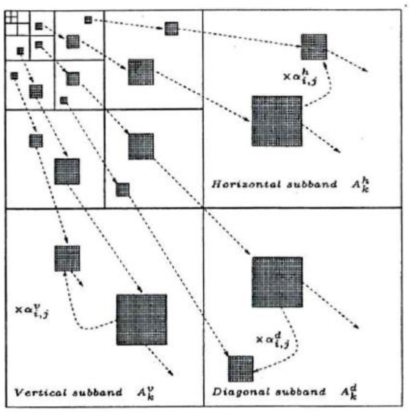


Figure 2.2 : The FW transform

The FW Transform

For the fully decomposed wavelet tree, let A_k^h, A_k^v, A_k^d denote the horizontal, vertical and diagonal sub-blocks of wavelet coefficients at decomposition level k, 0 ≤ k ≤ K, respectively. Each of these sub-blocks contains $\alpha_{kij}^h, \alpha_{kij}^v, \alpha_{kij}^d$ coefficients; respectively. The three collections of blocks

$$A^h = \bigcup_{k=1}^{K} A_k^v, \quad A^v = \bigcup_{k=1}^{K} A_k^v, \quad A^d = \bigcup_{k=1}^{K} A_k^d,$$

comprise the fundamental horizontal, vertical and diagonal subtrees of the coefficient tree, respectively. Now consider any wavelet coefficient α_{kij}^λ , λ∈ {h, v, d} in this matrix and the unique subtree, with this element as its root, this subtree will be denoted by A_{kij}^λ

The two-dimensional fractal-wavelet transforms involve mappings of "parent" subtrees of wavelet coefficients to lower "child" subtrees. For simplicity in presentation and notation, we consider a particular case in which the roots of all parent quadtrees appear in a given block and the roots of all child quadtrees appear in another given block. Select two integers, the parent and child levels, k_1^* and k_2^*, respectively, with $1 \le k_1^* \le k_2^*$. Then for each possible index $1 \le i, j \le 2^{k_2^*}$ define the three sets of affine block transforms:

$$\mathcal{W}_{ij}^{\lambda} \; : \; A^{\lambda}_{k_1^*, i^\lambda(i,j), j^\lambda(i,j)} \to A^{\lambda}_{k_2^*, i, j},$$
$$A^{\lambda}_{k_2^*, i, j} = \alpha^{\lambda}_{ij} A^{\lambda}_{k_1^*, i^\lambda(i,j), j^\lambda(i,j)}, \quad \lambda \in \{h, v, d\}.$$

Note how the child subtrees at level k_2^* are replaced by scaled copies of parent subtrees from level k_1^*. This procedure is illustrated in Figure 3.14. These block transforms will comprise a unique fractal-wavelet (FW) operator W. The use of the indices i^h, j^h, etc. emphasizes that the parent quadtrees corresponding to a given set of child quadtrees $A^h_{k_2^*,i,j}$, $A^v_{k_2^*,i,j}$, $A^d_{k_2^*,i,j}$ and need not be the same. As well, the scaling coefficients $\alpha^h_{kij}, \alpha^v_{kij}$ and α^d_{kij} are independent.

The "fractal code" associated with the generalized FW operator W consists of the following:

1. The parent-child index pair (k_1^* , k_2^*), generally k_2^* = k_1^* + 1.
2. The wavelet coefficients in blocks B_o, and A^{λ}_{kij} , $\lambda \in$ {h, v, d} for $1 \le k \le k_2^*$— 1, a total of $4^{k_2^*}$ coefficients.
3. The scaling factors α^{λ}_{ij} and parent block indices, (i^λ(i,j), j^λ(i,j), for all elements α^{λ}_{ij} in each of the three blocks At.. The total number of parameters:
   - 3 × $4^{k_2^*}$ scaling factors,

   2 × 3 × $4^{k_2^*}$ indices.

It has been shown [57, 70] that, under certain conditions, the fractal-wavelet transform VV is contractive in an appropriate complete metric space (12-type square summable sequences) of wavelet coefficients. For the special transform given in Eq. (2.16), contractivity is guaranteed when

$$c_Q = 2^{k_2^* - k_1^*} \max_{\lambda,i,j} |\alpha^{\lambda}_{ij}| \; < \; 1,$$

where $\lambda \in$ {h, v, d} and $0 \le$ i, j $\le$ $2^{k_2^*}$ — 1. From the Contraction Mapping Theorem, the condition $C_Q < 1$ guarantees the existence of a unique fixed point of the operator W, that is, a unique wavelet coefficient tree, $\bar{c}$ such that W($\bar{c}$) = . Moreover, the wavelet tree ë may be generated by iteration of W.

The standard FW scheme, as described in [18, 47], is a special case of the generalized FW scheme, where it assumes that common parents and common scaling factors are used for the various subbands, that is

$$\begin{aligned} i^h(i,j) &= i^v(i,j) = i^d(i,j) \\ j^h(i,j) &= j^v(i,j) = j^d(i,j) \\ \alpha^h_{ij} &= \alpha^v_{ij} = \alpha^d_{ij}. \end{aligned}$$

In other words, the Ii, v and d subbands are not treated independently.

Next, a few FW schemes that differ only in whether the three subbands (horizontal, vertical and diagonal) of the wavelet tree are combined together or treated independently, are described and implemented.

III COMPARISONS AND CONCLUDING REMARKS

In this thesis, several fractal, wavelet and fractal-wavelet image coding methods for the purpose of image compression were discussed and implemented. Some of the advantages of developing adaptive fractal-based image compression methods include performing content-dependent image compression at pre-determined bit rates, compression ratios or fidelity precisions and generating rate distortion curves. Generating rate distortion curves for these fractal-based schemes provided a comparison of their performance to each other as well to other image compression methods, such as the SPIHT method.

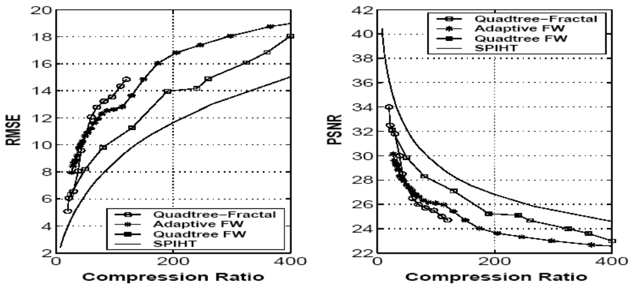


Figure 3.1 : Rate distortion curves generated by the various adaptive image compression methods studied in this chapter, namely the quadtree-based standard fractal, the SPIHT, the adaptive FW and the quadtreebased FW schemes.

Figure 3.1 illustrates a comparison between the various adaptive fractal and wavelet-based image compression methods covered in this chapter, namely the quadtree-based standard fractal, the SPIHT, the adaptive FW and the quadtree-based FW schemes. Clearly, the SPIHT performs best. However, when comparing the various fractal-based methods to each other, note that fractalwavelet based methods perform better than the standard fractal schemes, applied in the spatial domain of the image, for higher compression ratios. However, for lower compression ratios (i.e. less than 50:1), the quadtree-based standard fractal scheme starts to perform better than some of the FW methods.

In this thesis a brief review of the theory and application of various adaptive fractal and wavelet based image compression methods was presented. Rate distortion curves of these adaptive image compression schemes were generated and their performance was compared. While the SPIHT method performs considerably better than the best fractal-based schemes, fractal-based schemes were shown to be competitive especially at low compression ratios. Algorithms for making fractal-based schemes adaptive were also discussed. The fractal-wavelet schemes perform better than standard fractal schemes, especially for high compression ratios. Furthermore, fractal-wavelet schemes overcome the computational complexity and the disturbing blockiness artifacts that are evident when using the generic spatial-based fractal schemes.

In the following the application of these various fractal and fractalwavelet based image coding schemes for the purpose of image restoration and enhancement be investigated.

3.1 Future Research Directions

Some of the research directions that may stem from the work presented in this thesis can be outlined as follows:

- It was shown that the use of the quadtree-based fractal and fractal-wavelet predictive schemes for image denoising yields results that are significantly better than using standard fractal and fractal-wavelet schemes. However, whenever using the quadtree partitioning algorithm for the purpose of fractal image coding, one has to choose a threshold for the decomposition criterion. The determination of a reasonable, image independent strategy for selecting such a threshold is still an open question. This threshold can be viewed as a denoising fine-tuning parameter that measures the trade-off between suppressing the noise and reconstructing the high frequency content and important features of the image.
- In practice, one is often constrained with a bit-budget. Thus, developing image denoising methods that only aim for getting the best quality of the denoised image without also paying any attention to the compression ratios and bit-rate limitation may not be very practical. Thus, there a great need to develop effective schemes that perform not only image denoising but also image compression. The Rissanen’s Minimum Description Length (MDL) principle has recently been effectively used for the purpose of designing wavelet thresholding methods for the purpose of simultaneous image compression and denoising. The use of the MDL principle may also be applied for the purpose of developing effective fractal-based techniques that are capable of performing simultaneous denoising and compression of noisy images. Fractal-based methods have been shown to be effective lossy image compression methods. In this thesis, it was shown that fractal-based schemes are also effective image denoising methods. Thus, the development of fractal-based joint image compression and denoising would combine these capabilities of the fractal methods. These schemes would allow us to generate rate distortion curves that exhibit the trade-off between the quality of a fractally denoised image and the bit rate required to store this denoised image. Simultaneous image compression and denoising schemes are important in many applications where simultaneous compression and denoising is needed, for instance, when images are acquired from a noisy source and storage or transmission capacity is severely limited, such as in some video coding applications use adaptive thresholds. There are two aspects to the adaptivity of the thresholding operators: The first is related to the

selected threshold) where adaptive thresholds perform better than the universal one. The second adaptivity aspect is related to the manner the thresholding operators are applied. In this work, it was shown that better results were achieved by applying adaptive and localized thresholding operators instead of the conventional hard and soft thresholding point operators. While the selection of adaptive thresholds have been investigated in the literature, making the thresholding operators themselves more adaptive seem to have been overlooked.

REFERENCES

[1.] You-Lon Lin &Won-Lin-Chin; 2012.Fast search for Fractal Image Compression Journal of Information Science & Engineering; vol. 28, 17-30 (2012).
[2.] Hai Wang; 2010.fast Image Fractal Compression with Graph-Based Image Segmentation Algorithm. International Journal of Graphics; vol.1, No.1, November 2010.
[3.] Dr. Fadil Salman Ahmed; 2012. A Proposed Encoding & Hiding Text in an Image by using Fractal Image Compression. International Journal of Science & Engineering (IJSE); vol. 4, No.1, Jan.2012, ISSN 0975-3397.
[4.] Lifng Xi & Liang bin Zhang; 2007. A Study of Fractal Image Compression Based on an Improved Genetic Algorithm. International Journal Nonlinear Science; vol.3 (2007), No.2. pp. 116-124.
[5.] M.F. Barnsley, and S. Demko, “Iterated function systems and the global construction of fractals,” Proc. Roy. Soc. Lond., vol. A399, pp. 243-275, 2009.
[6.] M.F. Barnsley, and L.P. Hurd, Fractal Image Compression., Massachusetts: A.K. Peters, Wellesley, 1995
[7.] C. S. Burrus, R. A. Gopiiiath, and H. Guo, Introduction to Wavelets and Wavelet Transforms: A Primer Prentice Hall. New Jersey, 1999