Fractal Image Compression on a Pseudo Spiral Architecture

Huaqing Wang, Meiqing Wang, Tom Hintz, Xiangjian He, Qiang Wu

Faculty of Information Technology,
University of Technology, Sydney
PO Box 123, Broadway 2007, Australia

{huwang,meiqing,hintz,sean,wuq}@it.uts.edu.au

Abstract

Fractal image compression is a relatively recent image compression method which exploits similarities in different parts of the image. The basic idea is to represent an image by fractals and each of which is the fixed point of an Iterated Function System (IFS). Therefore, an input image can be represented by a series of IFS codes rather than pixels. In this way, an impressive compression ratio 10000:1 can be achieved. The application of fractal image compression presented in this paper is based on a novel image structure, Spiral Architecture, which has hexagonal instead of square pixels as the basic element. In the paper evidence would suggest that introducing Spiral Architecture into fractal image compression will improve the compression performance in compression ratio with little suffering in image quality. There are also much research could be done in this area to further improve the results.

Keywords: fractals, image compression, image encoding, Spiral Architecture, hexagonal structure

1 Introduction

Needless to say, visual information is of vital importance if human beings are to perceive, recognize and understand the surrounding world. With the tremendous progress that has been made in computer power, the corresponding growth in the multimedia market and the advent of the World Wide Web, it is becoming more than ever possible for images to be widely utilized in our daily life. In general, an image file contains much more data than a text file. An image with a large amount of data requires much memory to store, takes longer to transfer, and is complex to process. For example, a grey scale image with 256 × 256 pixels requires about 65 Kilobytes of memory space and more than 18 seconds to transfer at 28.8kb/s. As a consequence, image compression becomes necessary due to the limited communication bandwidth, CPU speed and storage size. Image compression has been one of the most challenging fields in the image processing research.

Fractal image compression is a relatively recent image compression method which exploits similarities in different parts of the image. For example, with a picture of a fern (Fig. 1) one can see where these similarities lie: each fern leaf resembles a smaller portion of the fern. This is known as the famous Barnsley fern (Barnsley & Demko 1985). During more than two decades of development, the IFS (Iterated Function System) based compression algorithm stands out as the most promising direction for further research and improvement (Barnsley & Hurd 1993). The basic idea is to represent an image by fractals and each of which is the fixed point of an IFS. An IFS consists of a group of affine transformations(Fisher 1995). Therefore, an input image can essentially be represented by a series of IFS codes. In this way, a compression ratio 10000:1 can be achieved (Barnsley & Sloan 1988). In short, for fractal image compression an image is represented by fractals rather than pixels. Each fractal is defined by a unique IFS that consists of a group of affine transformations. Therefore the key point for this algorithm is to find fractals which can best describe the original image and then to represent them as affine transformations.

Fig.1 A fern leaf (Barnsley & Sloan 1988)

The application of fractal image compression presented in this paper is based on a novel image structure, Spiral Architecture (Sheridan, Hintz & Moore 1991), which is inspired from anatomical considerations of the primate's vision (Schwartz 1980). On The Spiral Architecture, an image is a collection of hexagonal elements (Sheridan, Hintz & Alexander 2000). In the case of the human eye, these elements (hexagons) would represent the relative position of the rods and cones on the retina. Each pixel on The Spiral Architecture is identified by a designated positive number, called Spiral Address as shown in Fig.2. The numbered hexagons form the cluster of size 7^n. The hexagons tile the plane in a recursive modular manner along the spiral direction (He 1999). Any hexagonal pixel has only six neighbouring pixels which have the same distance to the centre hexagon of the seven-hexagon unit of vision.

This paper is organized as follows. Beginning with a review of fractal image compression in Section 2, an introduction of the Spiral Architecture is presented in Section 3. In Section 4, we describe the procedure of adapting the fractal image compression algorithm on The Spiral Architecture and the experimental results are supplied in Section 5 with some quantified analysis. We conclude in Section 6 by summarizing the opportunity of better performance for fractal image compression on the Spiral Architecture.

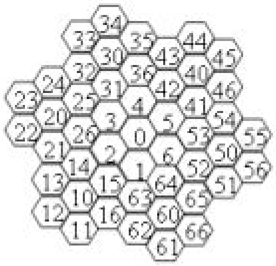


Fig.2 A collection of $7^2 = 49$ Hexagons with labelled addresses

2 Concepts of Fractal Image Compression

In the following section, the basic concepts of fractal image compression on the traditional square structure would be introduced. Before delving into details, there are some highlights of fractal image compression.

- It is a promising technology, though still relatively immature.
- The fractals are represented by Iterated Function Systems (IFSs).
- It is a block-based lossy compression method.
- Compression has traditionally been slow but decompression is fast.

2.1 Theory and Math Background

The fundamental principle of fractal image compression consists of the representation of an image by an iterated function system (IFS) of which the fixed point is close to that image. This fixed point is so called '***fractal***' (Fisher 1995). Each IFS is then coded as a contractive transformation with coefficients. Banach's fixed point theorem guarantees that, within a complete metric space, the fixed point of such a transformation may be recovered by iterated implementation thereof to an arbitrary initial element of that space (Kreyszlg 1978). Therefore, the encoding process is to find an IFS whose fixed point is similar to the given image. The usual approach is based on the ***collage theorem***, which provides a bound on the distance between the image to be encoded and the fixed point of an IFS (Fisher 1995). A suitable transformation may therefore be constructed as a 'collage' from the image to itself with a sufficiently small 'collage error' (the distance between the collage and the image) guaranteeing that the fixed point of that transformation is close to the original image (Wohlberg & Jager 1999).

In the original approach, devised by Barnsley, this transformation was composed of the union of a number of affine mappings on the entire image (Barnsley & Hurd 1993). While a few impressive examples of image modelling were generated by this method (Barnsley's fern, for example (Barnsley 1988)), no automated encoding algorithm was found. Fractal image compression became a practical reality with the introduction by Jacquin of the partitioned IFS (PIFS) (Jacquin 1993), which differs from an IFS in that each of the individual transformation operates on a subset of the image, rather than the entire image. Since the image support is tiled by 'range blocks', each of which is mapped from one of the 'domain blocks' as depicted in Fig. 3, the combined mappings constitute a transformation on the image as a whole. The transformation minimizing the collage error within this framework is constructed by individually minimizing the collage error for each range block, which requires locating the domain block which may be made closest to it under an admissible block mapping. This transformation is then represented by specifying, for ach range block, the identity of the matching domain block together with the block mapping parameters minimizing the collage error for that range block.

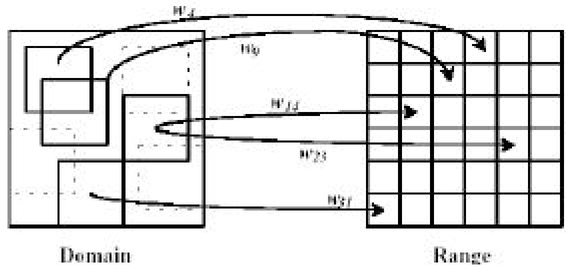


Fig. 3 Each range block is constructed by a transformed domain block

2.2 Basic Fractal Image Encoder

The encoder has to solve the following problem: for each range block R the best approximation

$$R \approx sD + oI \qquad 2.1$$

needs to be found, where D is a codebook block transformed from a domain block to the same size as R. The coefficients s and o are called ***scaling*** and ***offset***. We work out this problem with the Euclidean norm. That is, to minimize

$$E(D,R) = \min_{s,o} \left\| R - (sD + o\mathbf{1}) \right\| \qquad 2.2$$

we can use the well known method of least squares to find the optimal coefficients directly as follows.

Given a pair of blocks R and D of n pixels with intensities $r_1,\ldots, r_n$ and $d_1,\ldots, d_n$ we have to minimized the quantity

$$\sum_{i=1}^{n} (s \cdot d_i + o - r_i)^2 . \qquad 2.3$$

The best coefficients s and o are

$$s = \frac{n\left(\sum_{i=1}^{n} d_i r_i\right) - \left(\sum_{i=1}^{n} d_i\right)\left(\sum_{i=1}^{n} r_i\right)}{n\sum_{i=1}^{n} d_i^2 - \left(\sum_{i=1}^{n} d_i\right)^2} \qquad 2.4$$

and

$$o = \frac{1}{n}\left(\sum_{i=1}^{n} r_i - s\sum_{i=1}^{n} d_i\right) . \qquad 2.5$$

With *s* and *o* given the square error is

$$E(D,R)^2 = \frac{1}{n}\left[\sum_{i=1}^{n} r_i^2 + s\left(s\sum_{i=1}^{n} d_i^2 - 2\sum_{i=1}^{n} d_i r_i + 2o\sum_{i=1}^{n} d_i\right) + o\left(on - 2\sum_{i=1}^{n} r_i\right)\right] \quad 2.6$$

If the denominator in equation 2.4 is zero then

$$s = 0 \quad 2.7$$

and

$$o = \sum_{i=1}^{n} r_i / n . \quad 2.8$$

In summary the baseline fractal encoder with fixed block size operates in the following steps.

1. **Image segmentation.** Segment the given image using a fixed block size, for instance, 4× 4. The resulting blocks are called ranges R_i.
2. **Domain pool and codebook blocks definition.** By stepping through the image with a step size of l pixel(s) horizontally and vertically create a set of domain blocks, which are four times as the size of range blocks. By averaging the intensities of four neighbouring pixels each domain blocks shrinks to match the size of the ranges. This produces the codebook blocks D_i.
3. **The search of best *s* and *o*.** For each range block R_i an optimal approximation $R_i \approx sD_i + oI$ in the following steps:

   a) For each codebook block D_i compute an optimal approximation $R_i \approx sD_i + oI$ in three steps:

      i. Perform the least squares optimization using formulas 2.13 and 2.14, yielding a real coefficient scalar *s* and an offset *o*.

      ii. Quantize the coefficients using a uniform quantizer.

      iii. Using the quantized coefficients *s* and *o* compute the error $E(R_i, D_i)$.

   b) Among all codebook blocks D_i find the block D_k with minimal error

   $$E(R_i, D_k) = \min_i E(R_i, D_i).$$

   c) Output the code for the current range block consisting of indices for the quantized coefficient *s* and *o* and the index *k* identifying the optimal codebook block D_k.

3 Spiral Architecture and Image Representation

A digital image contains thousands of pixels to represent the real world and when we touch the term 'pixel' so far, that means a rectangular box in an image. Almost all the previous image processing and image analysis research is based on this traditional image structure. However, we do have a relatively new image structure called ***Spiral Architecture*** (SA) (Sheridan 1996). Spiral Architecture is inspired from anatomical considerations of the primate's vision (Schwartz 1980). From the research about the geometry of the cones on the primate's retina (See Fig.4) we can conclude that the cones' distribution has inherent organization and is featured by its potential powerful computation abilities. The cones with the shape of hexagons are arranged in a Spiral clusters. This cluster consists of the organizational units of vision. Each unit is a set of seven hexagons compared with the traditional rectangular image architecture using a set of 3× 3 vision unit as shown in Fig.5.

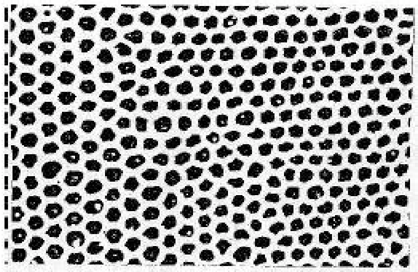
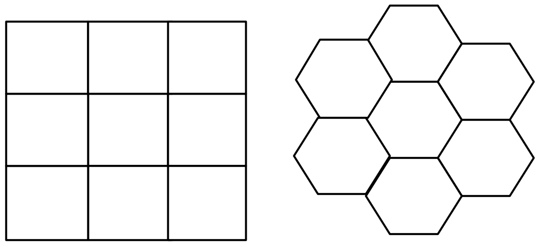


Fig.4 Distribution of Cones on the Retina



(a) Rectangular Architecture (b) Spiral Architecture

Fig.5 Unit of vision in the two image architectures

3.1 Spiral Addressing

The first step in SA formulation is initially labelling each of the individual hexagons with a unique ***address***. The addresses of these hexagons will then be simply referred to as the hexagons. This is achieved by a process that is initially applied to a collection of seven hexagons. Each of these seven hexagons is labelled consecutively with addresses 0, 1, 2, 3, 4, 5 and 6 as displayed in Fig.6.

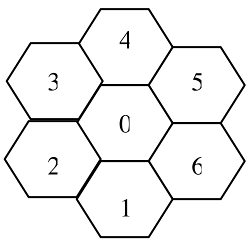


Fig.6 A collection of seven hexagons with unique addresses

Dilate the structure so that six additional collections of seven hexagons can be placed about the addressed hexagons, and multiply each address by 10. For each new collection of seven hexagons, label each of the hexagons consecutively from the centre address as we did for the first seven hexagons (see Fig.7).

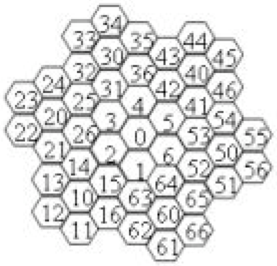


Fig.7 A collection of $7^2 = 49$ hexagons with labelled addresses

The repetition of the above steps permits the collection of hexagons to grow in powers of seven with uniquely assigned addresses. It is this pattern of growth of addresses that generates the ***Spiral***. Furthermore, the addresses are consecutive in base seven.

The important aspect of each hexagon is that it has six neighbouring hexagons. This establishes the property that for all hexagons, the centre of each hexagon has a constant distance from every one of its six neighbours. According to (Umbaugh 1996), the difference of light intensities between pixels is highly related to the distance between them: the closer they are, the less difference observed. Hence, the light intensity of a hexagonal pixel can be considered being equally affected by the light intensities of its six neighbouring pixels (He 1999). Moreover, each set of seven hexagons may enjoy very similar light intensities and the difference between the centre and others would be quite small. This idea is the foundation stone when considering image compression on SA.

3.2 Spiral Counting

Spiral Counting (Sheridan 1996) is an algorithm that designates a sequence of hexagons in SA. It can be considered as a Spiral movement that given a commencing hexagon, counts for a pre-determined number and terminates at another certain hexagon. Any hexagon in an image can be reached by Spiral counting from any other given hexagon in the same image. When applying Spiral counting, it is strictly dependent on a pre-determined key define by Sheridan in (Sheridan, Hintz & Moore 1991). A key is the first hexagon to be reached in an instance of a Spiral counting, which determines two important parameters: the distance and the orientation. For instance, given a Spiral address 15, the key of 15 can determine two values. One is the distance between the given hexagon 15 to the hexagon 0; the other is the orientation of hexagon 15 from hexagon 0. We could use the angle ? to represent the orientation (see Fig.8).

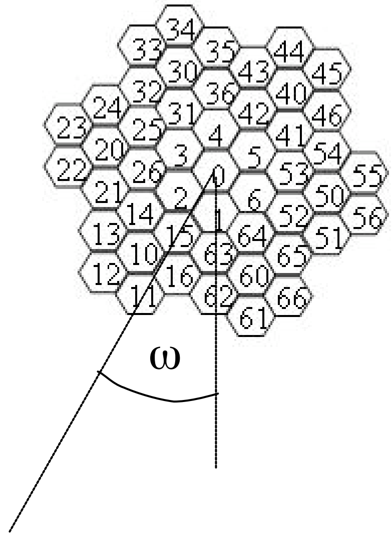


Fig.8 The key of hexagon 15

Spiral counting is used to define two operations in the SA, which are ***Spiral Addition*** and ***Spiral Multiplication*** (Sheridan, Hintz & Moore 1991). Let a and b be Spiral addresses of two arbitrarily chosen hexagons in SA. Then,

- Spiral addition of a and b, denoted by a + b, is the Spiral address of the hexagon found by Spiral counting b hexagons in the key of Spiral address 1 from the hexagon with Spiral address a;
- Spiral multiplication of a and b, denoted by a x b, is the Spiral address of the hexagon found by Spiral counting b hexagons in the key of Spiral address a from the hexagon with Spiral address 0.

Spiral Architecture together with the operations of Spiral Addition and Spiral Multiplication is a Euclidean Ring (Sheridan, Hintz & Moore 1991). These properties are necessary to locate and transform hexagonal pixels when further implementing SA for image compression.

3.3 Pseudo Model

Although SA has many advantages in image processing and machine vision, there is no available image capture or display device yet to support this structure. Hence, in order to implement our theoretical results, it is necessary to construct or mimic the SA from the existing image structure, on which the traditional image representation is based. There are several different methods available so far – Mimic model by He, Pseudo model by Sheridan and Visual model by Wu (He 1999; Sheridan, Hintz & Alexander 2000; Wu, He & Hintz 2004). Because of the less computational complexity and pixel-to-pixel representation, in our paper, we choose the Pseudo model.

- Representation of hexagonal pixels

In the Pseudo model, we are using only one rectangular pixel to represent a hexagonal pixel. The basic cluster of seven pixels with Spiral addresses 0~6 are represented in the following figure (see Fig.9).

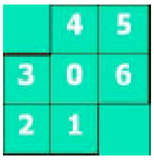


Fig.9 Distribution of 7 pixels constructed from rectangular pixels

Following the same labelling scheme, we could extend the previous cluster to get the first 49 Pseudo hexagons with Spiral addresses 0 to 66. (See Fig.10)

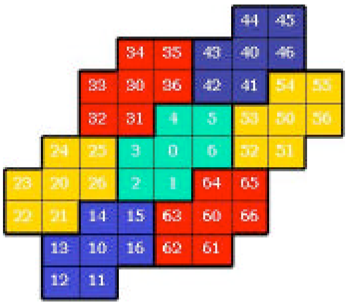


Fig.10 the first 49 Pseudo pixels with labelled spiral addresses

If we repeat the same scheme we are able to cover any image on traditional structure and label each of the pixels with a unique Spiral address. In order to be consistent with the important property of hexagonal distribution that each such pixel has exactly six surrounding pixels, we only consider six of the eight neighbours for the centre pixels. The election of these six neighbours is as shown Fig.9 so that we can keep the properties of Spiral Architecture to the greatest extent.

- Finding Pseudo Pixels

Again, the central pixel is labelled with Spiral address 0 and we correspondingly set the Cartesian coordinates of this pixel as (0, 0). Suppose that each square pixel has the length of 1 as edges. Then, the hexagon with Spiral addresses 1, 2, 3, 4, 5 and 6 have corresponding Cartesian coordinates (0, -1), (-1, -1), (-1, 0), (0,1), (1, 1) and (1, 0) respectively.

The rules to find the Cartesian coordinates for given Pseudo pixel with a labelled Spiral address are exactly the same as on the Mimic model that we mentioned previously. Following the formulas 3.1~3.4, given any Spiral address we can easily find the physical location, i.e. the Cartesian coordinates, of the pixel on the image.

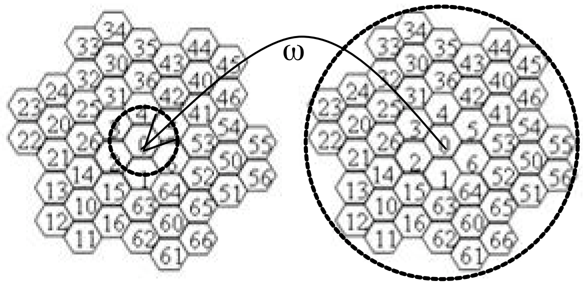
- Image Representation

To represent images on the Spiral Architecture by Pseudo model is not more than to select a certain set of pixels marked with a range of spiral addresses and to ignore the remaining pixels. The result is shown in Fig.11.

Fig.11 Boat in Square and Spiral Architecture in Pseudo Model

4 Fractal Image Compression on Spiral Architecture

In this preliminary research on adopting fractal image compression into Spiral Architecture, we follow the same idea applied on square structure, i.e. PIFS as described earlier in section 2. Firstly we separate the image into range blocks of seven pixels and define the domain blocks of seven times more, i.e. 49 pixels (see Fig. 12). Each pixel in the image can be the centre of domain block. Then we include the first 48 pixels around it based on Spiral counting to form a domain block unless any pixel of this domain block is out of the given image.



range block domain block

Fig.12 Range and domain blocks in Spiral Architecture

A number of researchers have noticed a tendency for a range block to be spatially close to the matching domain block, (Barthel & Voye 1994; Beaumont 1990), based on the observed tendency for distributions of spatial distances between range and matching domain blocks to be highly peaked at zero (Jacquin 1993; Woolley & Monro 1995). Motivated by this observation, the domain pool for each range block may be restricted to a region about the range block (Jacquin 1990), or a spiral search path may be followed outwards from the range block position (Barthel & Voye 1994; Beaumont 1990). Therefore, in order to reduce the computational complexity, for each range block we only search for up to 343 domain blocks, which are around this range block. Each of those range blocks has at most 343 domain blocks in the domain pool and the centres of domain blocks in the pool are the first 343 pixels counting from the centre of range block through the Spiral direction.

5 Experimental Results

We use the same algorithm mentioned before on square and Spiral Architecture for four popular images: a building, a boat, a toy duck and a house. The following figures show the experimental results and we summarize them in two tables.

Fig.13 Original and compressed 'building' in square structure

Fig.14 Original and compressed 'boat' in square structure

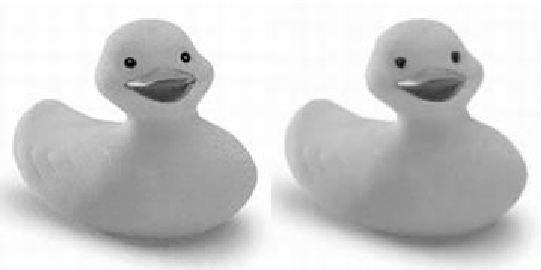


Fig.15 Original and compressed 'toy duck' in square structure

Fig.16 Original and compressed 'house' in square structure

Fig.17 Original and compressed 'building' in Spiral Architecture

Fig.18 Original and compressed 'boat' in Spiral Architecture

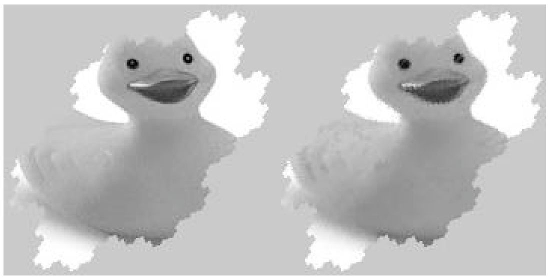


Fig.19 Original and compressed 'toy duck' in Spiral Architecture

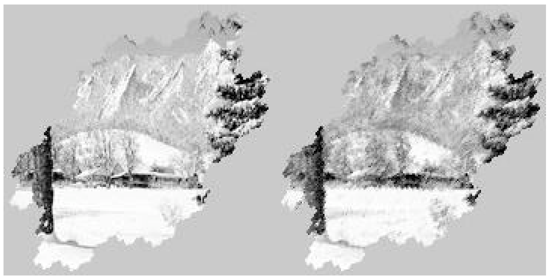


Fig.20 Original and compressed 'house' in Spiral Architecture

Image	Compression ratio	PSNR
Building	3.37	23.40
Boat	3.37	26.56
Toy duck	3.37	37.20
House	3.37	22.41

Table 1. Summary for images on square structure

Image	Compression ratio	PSNR
Building	16.01	22.73
Boat	16.01	24.27
Toy duck	16.01	29.80
House	16.01	20.10

Table 2. Summary for images on Spiral Architecture

From the above experimental results, it is obvious that adopting fractal image compression on Spiral Architecture would be able to achieve higher compression ratio with little trade-off in the image quality.

6 Conclusions

The fundamental principle of fractal image compression consists of the representation of an image by fractals that are ‘collaged’ together to approximate the original image. Each fractal is the fixed point of an Iterated Function System, which involves a number of contractive transformations. Then the coefficients of those transformations are saved as the compressed file. In this way, an image is able to b e represented by a series of codes rather than pixels. This feature distinguishes fractal image compression from any other method. However, at this stage the major problem is how to find a set of IFSs efficiently and effectively so that the fix points of which can collage together and best resemble a given image. Spiral Architecture is a novel image structure, which has hexagonal but not square pixels as elements. It has been proved that Spiral Architecture has two advantages in image compression: locality of pixel intensity and uniform partitioning. Therefore, adapting fractal image compression into Spiral Architecture should also see some better compression performance. Following the similar idea in square architecture we take a cluster of seven hexagons as a range block and 49 hexagons as a domain block. Then we define the domain pool as the 343 domain blocks around the given range block unless the domain block is out of the given image. According to the experimental results, we have found that Spiral Architecture has a great potential in improving fractal image compression. It will improve the compression ratio with little trade-off in image quality. Furthermore, considering the advantages offered by Spiral multiplication in searching the best match between range blocks and domain blocks, similar or even better image quality could be achieved and the encoding time is expected to be reduced as well.

7 References

Barnsley, M 1988, *Fractal Everywhere*, New York: Academic.

Barnsley, M & Demko, S 1985, 'Iterated Function Systems and the Global Construction of Fractals', *Royal Soc.*, vol. A399, London, pp. 243~75.

Barnsley, M & Hurd, LP 1993, *Fractal Image Compression*, AK Peters. Ltd.

Barnsley, M & Sloan, AD 1988, 'A better way to compress images', *BYTE*, January, pp. 215-23.

Barthel, KU & Voye, T 1994, 'Adaptive fractal image coding in the frequency domain', *Proc. Int. Workshop Image Processing*, pp. 33~8.

Beaumont, JM 1990, 'Advances in block based fractal coding of still pictures', *Proc. IEE Colloq.: The Application of Fractal Techniques in Image Processing*, pp. 3.1~3.6.

Fisher, Y 1995, *Fractal Image Compression: Theory and Application*, Springer-Verlag New York, Inc., New York.

He, X 1999, '2D-object Recognition with Spiral Architecture', University of Technology, Sydney.

Jacquin, AE 1990, 'Fractal image coding based on a theory of iterated contractive image transformations', *Pro. SPIE: Vis. Commun. Image Processing*, vol. 1360, pp. 227~39.

Jacquin, AE 1993, 'Fractal image coding: a review', *Proceedings of the IEEE*, vol. 81, no. 10, pp. 1451-65.

Kreyszlg, E 1978, *Introductory Functional Analysis with Applications*, New York: Wiley.

Schwartz, E 1980, 'Computational Anatomy and Functional Architecture of Striate Cortex: A Spatial Mapping Approach to Perceptual Coding', *Vision Research*, vol. 20, pp. 645-69.

Sheridan, P 1996, 'Spiral Architecture for Machine Vision', PhD thesis, University of Technology, Sydney.

Sheridan, P, Hintz, T & Alexander, D 2000, 'Pseudo-invariant image Transformations on a hexagonal lattice', *Image and Vision Computing*, vol. 18, pp. 907-17.

Sheridan, P, Hintz, T & Moore, W 1991, 'Spiral Architecture in Machine Vision', *Australian Occam and Transputer Conference*, Amsterdam.

Umbaugh, SE 1996, *Computer Vision and Image Processing: A Practical Approach Using CVIP tools*, Prentice Hall.

Wohlberg, B & Jager, Gd 1999, 'A Review of the Fractal Image Coding Literature', *IEEE Transaction on Image Processing*, vol. Vol. 8, viewed 1999.

Woolley, SJ & Monro, DM 1995, 'Optimum parameters for hybrid fractal image coding', *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, pp. 2571~4.

Wu, Q, He, X & Hintz, T 2004, 'Virtual Spiral Architecture', *International Conference on Parallel and Distributed Processing Techniques and Applications*, vol. 1, pp. 399~405.