VSA-based Fractal Image Compression

Huaqing Wang[1], Meiqing Wang[2], Tom Hintz[1], Qiang Wu[1], Xiangjian He[1]

[1] Faculty of Information Technology, University of Technology, Sydney
PO Box 123, Broadway 2007, Sydney, NSW, Australia

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

[2] College of Mathematics and Computer Sciences, Fuzhou University
No.502 Gong Ye Road, Fuzhou, Fujian, China, 350002

wangmeiqing@hotmail.com

ABSTRACT

Spiral Architecture (SA) is a novel image structure which has hexagons but not squares as the basic elements. Apart from many other advantages in image processing, SA has shown two unbeatable characters that have potential to improve image compression performance, namely, *Locality of Pixel Density* and *Uniform Image Partitioning*. 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 as fixed points of *Iterated Function Systems* (IFS). Therefore, an input image can be represented by a series of IFS codes rather than pixels. In this way, an amazing compression ratio 10000:1 can be achieved. The application of fractal image compression presented in this paper is based on Spiral Architecture. Since there is no mature capture and display device for hexagon-based images, the experiments are implemented on a newly proposed mimic scheme, called *Virtual Spiral Architecture* (VSA). The experimental results in the paper have shown that introducing Spiral Architecture into fractal image compression will improve the compression performance in image quality with little trade-off in compression ratio. A lot of research work exists in this area to further improve the results.

Keywords

Fractals, image compression, image encoding, Virtual 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 intricate to process. For example, a

grey scale image with 256×256 pixels requires about 64 KB of memory space and more than 18 seconds to download using a 28.8K Dialup Modem. 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 (Figure 1) one can see easily where these similarities lie: each fern leaf resembles a smaller fern. This is known as the famous Barnsley fern [Barnsley1985]. During more than two decades of development, the *Iterated Function System* (IFS) based compression algorithm stands out as the most promising direction for further research and improvement [Barnsley1993]. The basic idea is to represent an image as the fixed points of IFSs. An appropriately chosen IFS consists of a group of affine transformations [Fisher1995]. Therefore, an input image can virtually be represented by a series of IFS codes. In this way, a compression ratio 10000:1 can be achieved

[Barnsley1988]. In short, for fractal image compression an image is represented by fractals rather than pixels. Each fractal is defined by a unique IFS consists of a group of affine transformations. Therefore the key point for this algorithm is to find fractals which can best approximate the original image and then to represent them as a set of affine transformations.

Figure 1. A fern leaf

The application of fractal image compression presented in this paper is based on a novel image structure, Spiral Architecture [Sheridan1991], which is inspired from anatomical considerations of the primate's vision [Schwartz1980]. On The Spiral Architecture, an image is a collection of hexagonal elements [Sheridan2000]. In the case of 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 Figure 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 [He1999]. Any hexagonal pixel has only six neighboring 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 adopting 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 and by mentioning areas for future research.
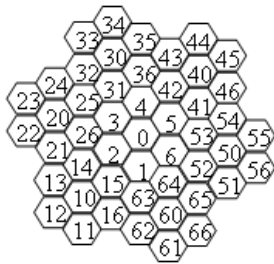

Figure 2. A collection of $7^2 = 49$ Hexagons with labelled addresses

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.

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 named as '*fractal*' [Fisher1995]. 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 [Kreyszlg1978]. Therefore, the encoding process is to find an IFS whose fixed point is close 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 (more details please refer to [Fisher1995] chapter 2). 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 [Wohlberg1999].

In the original approach, devised by Barnsley, this transformation was composed of the union of a number of affine mappings on the entire image [Barnsley1993]. While a few impressive examples of image modelling were generated by this method (Barnsley's fern, for example [Barnsley1988]), no automated encoding algorithm was found. Fractal image compression became a practical reality with the introduction by Jacquin of the partitioned IFS (PIFS) [Jacquin1993], 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 Figure 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 each range block, the identity of the matching domain block together with the block mapping parameters minimizing the collage error for that range block.
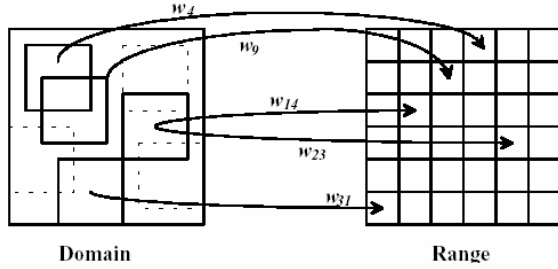

Figure 3. Each range block is constructed by a transformed domain block

Basic Fractal Image Encoder

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

$$R \approx sD + oI \tag{2.1}$$

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

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

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 $r_1,\dots, r_n$ and $d_1,\dots, d_n$ we have to minimized the quantity

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

The best coefficients s and o are

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

and

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

With s and o given the square error is

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

If the denominator in equation 2.4 is zero then $s = 0$ and $o = \sum_{i=1}^{n} r_i / n$

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 neighboring 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 2.4 and 2.5, 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) [Sheridan1996]. Spiral Architecture is inspired from anatomical considerations of the primate's vision [Schwartz1980]. From the research about the geometry of the cones on the primate's retina (See Figure 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 Figure 5.
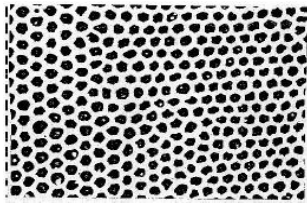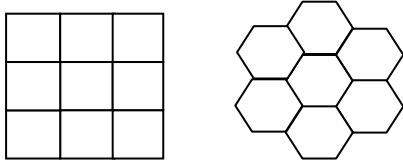

Figure 4. Distribution of Cones on the Retina

(a) Rectangular Architecture   (b) Spiral Architecture

Figure 5. Unit of vision in the two image architectures

Spiral Addressing

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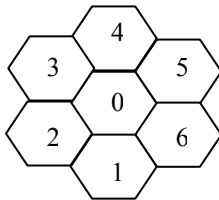

Figure 6. A collection of seven hexagons with unique addresses

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 Figure 7).
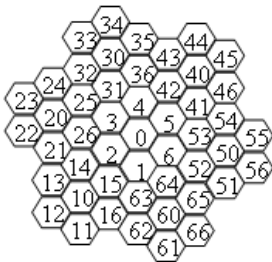

Figure 7. A collection of $7^2 = 49$ hexagons with labelled addresses

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 neighboring hexagons. This establishes the property that for all hexagons, the centre of each hexagon has a constant distance from every one of its six neighbors. According to Umbaugh [Umbaugh1996], 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 neighboring pixels [He1999]. 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.

Spiral Counting

Spiral Counting [Sheridan1996] 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 [Sheridan1991]. 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 ω to represent the orientation (see Figure 8).
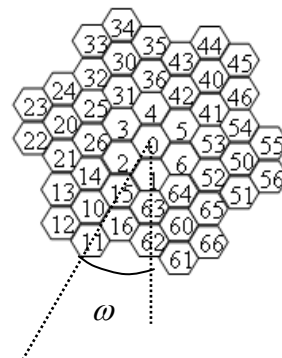

Figure 8. The key of hexagon 15

Spiral counting is used to define two operations in the SA, which are *Spiral Addition* and *Spiral Multiplication* [Sheridan1991]. 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 [Sheridan1991]. This property is necessary to further implement SA for image compression.

Virtual Spiral Architecture

SA has two unbeatable characters that are expected to improve image compression performance: *Locality of Pixel Intensity* and *Uniform Image Partitioning* [Hintz2003]. However due to the lack of capture and display devices, SA has not yet been widely used in image processing. In order to make SA applicable on the current available devices, Wu constructed a mimic scheme called *Virtual Spiral Architecture* (VSA) [Wu2004], with which images on rectangular structure can be smoothly converted to SA.

VSA mimicking scheme is so called 'virtual' because it only exists on computer memory during the procedures of image processing. The processing result will still be displayed on the traditional rectangular structure (see Figure 9).
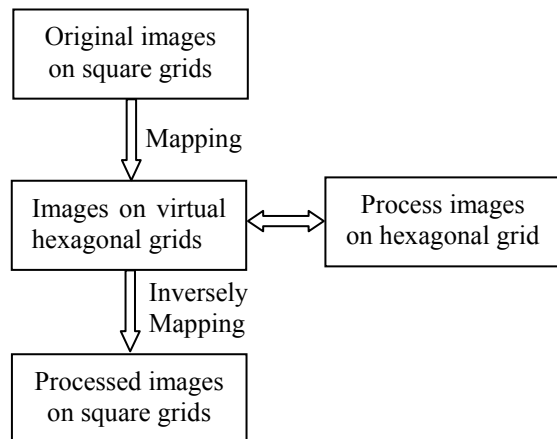
Original images on square grids

↓ Mapping

Images on virtual hexagonal grids ↔ Process images on hexagonal grid

↓ Inversely Mapping

Processed images on square grids

Figure 9. Flowchart of image processing on virtual Spiral Architecture

In order to keep the resolution, hexagonal and square pixels are defined as the same size, i.e. 1 unit area. Then if we map the SA on a traditional image and then let N denote the number of square pixels covered by a hexagonal pixel and let s_i represent the size of overlapped area in a certain square pixel i (See Figure 10), so the contribution of gray level given by this square pixel to the hexagonal pixel is measured by the percentage of the overlapped area, i.e. p_i.

$$p_i = s_i / 1 \times 100\% \qquad (3.1)$$

Therefore the grey value of this hexagonal pixel is

$$g_{Hex} = \sum_{i=1}^{N} (g_{Squ_i} \cdot p_i), \qquad (3.2)$$

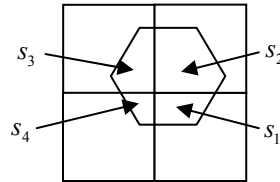where g_{Squ_i} is the grey level of the i square pixel.

Figure 10. The relationship between a virtual hexagonal pixel and overlapped square pixels.

As a result, the grey level information for SA is now available during the procedure of image processing and the experiment result can be displayed back on a traditional square-structure-based device following the similar mapping method (see Figure 11)

Figure 11. Boat in Square Structure and Virtual Spiral Architecture Displayed on Normal Device

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. Firstly we separate the image into range blocks of seven hexagonal pixels and define the domain blocks of seven times more, i.e. 49 pixels (see Figure 12). Each pixel in the image can be the centre of a domain block. Then we include the neighboring 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.
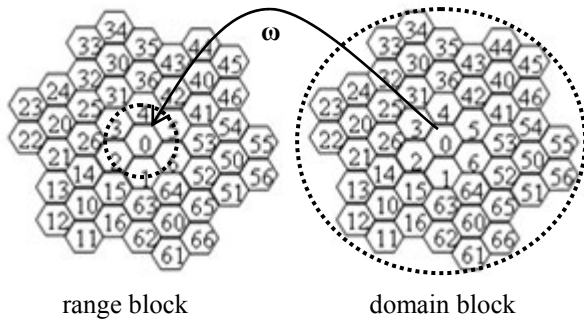

range block   domain block

Figure 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, [Beaumont1990; Barthel1994], based on the observed tendency for distributions of spatial distances between range and matching domain blocks to be highly peaked at zero [Jacquin1993; Woolley1995]. Motivated by this observation, the domain pool for each range block may be restricted to a region about the range block [Jacquin1990], or a spiral search path may be followed outwards from the range block position [Beaumont1990; Barthel1994]. 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 centers of domain blocks in the pool are the first 343 pixels counting from the centre of range block through spiral direction.

5. EXPERIMENTAL RESULTS

We use the same algorithms mentioned before on square and Spiral Architecture for four popular images: a building, a boat and a house. Figures 13 through 18 show the experimental results and we summarize them in two tables.

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

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

Figure 15. Original and compressed 'house' in square structure

Figure 16. Original and compressed 'building' in Spiral Architecture

Figure 17. Original and compressed 'boat' in Spiral Architecture

Figure18. Original and compressed 'house' in Spiral Architecture

Image	Compression ratio	PSNR
Building	3.37	23.40
Boat	3.37	26.56
House	3.37	22.41

Table 1. Summary for images on square structure

Image	Compression ratio	PSNR
Building	2	25.43
Boat	2	29.73
House	2	26.20

Table 2. Summary for images on Spiral Architecture

As the range block on SA is of 7 pixels (compare with 16 pixels in square structure), the compression ratio is slightly lower but the quality of decompressed image has increased.

6. CONCLUSIONS AND FUTURE WORK

According the experiments done so far, we have found that Spiral Architecture has a great potential in improving fractal image compression. Knowing the fact that there have been a large number of methods found to optimize fractal image compression on traditional image structure, we would try some of them on Spiral Architecture. Moreover, we may take advantage of spiral multiplication to find out the self-similarity in an image with less computational complexity. The following are some proposed methods:

1. Apply spiral multiplication to have a number of sub-images with 7^n pixels as range blocks. Define domain blocks as the sub-images with 7^{n+1} pixels obtained by spiral multiplication to form the domain pool. This method is expected to take advantage of the self-similarity introduced by spiral multiplication so that the time to search pairs between range and domain blocks will reduce significantly.

2. In order to have a more accurate domain pool, instead of averaging the neighboring seven pixels intensities to scale a domain block to be a codebook block, the medium value of these seven pixels could be used to represent their intensity.

3. Based on lots of experimental results, the larger errors between fractals and the original images always happen along the contour or edge of objects in the image. We are able to classify the range blocks into three categories by their frequency in intensity – shade, edge and midrange. During the search process, we then can enlarge the domain pool for range blocks with higher frequency.

In short, with the implement results it can be seen that introducing Spiral Architecture into fractal image compression has great future in improving the compression performance and a lot of researches exist in this area.

7. REFERENCE

[Barnsley1988] Barnsley, M., Fractal Everywhere, New York: Academic,1988.

[Barnsley1985] Barnsley, M. and S. Demko, Iterated Function Systems and the Global Construction of Factals, Royal Soc., London.

[Barnsley1993] Barnsley, M. and L. P. Hurd, Fractal Image Compression, AK Peters. Ltd,1993.

[Barnsley1988] Barnsley, M. and A. D. Sloan,A better way to compress images, BYTE: 215-223.1988.

[Barthel1994] Barthel, K. U. and T. Voye Adaptive fractal image coding in the frequency domain Porc. Int. Workshop Image Processing: 33~38,June 1994.

[Beaumont1990] Beaumont, J. M. Advances in block based fractal coding of still pictures Proc. IEE Colloq.: The Application of Fractal Techniques in Image Processing: 3.1~3.6,Dec, 1990.

[Fisher1995] Fisher, Y., Fractal Image Compression: Theory and Application, New York, Springer-Verlag New York, Inc.,1995.

[He1999] He, X., 2D-object Recognition with Spiral Architecture, PhD. Thesis, Faculty of Information Technology, University of Technology, Sydney1999.

[Hintz2003] T. Hintz and Q. Wu, Image Compression on Spiral Architecture, The International Conference on Imaging Science, Systems and Technology, Las Vegas, Nevada, USA.

[Jacquin1990] Jacquin, A. E. Fractal image coding based on a theory of iterated contractive image transformations Pro. SPIE: Vis. Commun. Image Processing 1360: 227~239,1990.

[Jacquin1993] Jacquin, A. E. Fractal image coding: a review Proceedings of the IEEE 81(10): 1451-1465,1993.

[Kreyszlg1978] Kreyszlg, E., Introductory Functional Analysis with Applications, New York: Wiley,1978.

[Schwartz1980] Schwartz, E. Computational Anatomy and Functional Architecture of Striate Cortex: A Spatial Mapping Approach to

Perceptual Coding Vision Research 20: 645-669,1980.

[Sheridan1996] Sheridan, P., Spiral Architecture for Machine Vision, PhD. Thesis, Faculty of IT, University of Technology, Sydney1996.

[Sheridan2000] Sheridan, P., T. Hintz, et al. Pseudo-invariant image Transformations on a hexagonal lattice Image and Vision Computing 18: 907-917,2000.

[Sheridan1991] Sheridan, P., T. Hintz, et al. Spiral Architecture in Machine Vision Australian Occam and Transputer Conference,1991.

[Umbaugh1996] Umbaugh, S. E., Computer Vision and Image Processing: A Practical Approach Using CVIP tools, Prentice Hall,1996.

[Wohlberg1999] Wohlberg, B. and G. d. Jager, A Review of the Fractal Image Coding Literature, IEEE Transaction on Image Processing.

[Woolley1995] Woolley, S. J. and D. M. Monro Optimum parameters for hybrid fractal image coding Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing 4: 2571~2574,1995.

[Wu2004] Wu, Q., X. He, et al., Virtual Spiral Architecture, The International Conference on Parallel and Distributed Processing Techniques and Applications.