Randomized Algorithm for Scaling Factors in Fractal Image Compression

Dinesh Rao B
Associate Professor
MCIS
Manipal

Ganesh Kamath
MS Student
MCIS
Manipal

Niranjan U C
Director Research & Training
Manipal Dot Net
Manipal

ABSTRACT
Modern fractal compression methods are based on iterative function systems (IFS), which was developed by Barnsley [1] and Jacquin [2]. During compression, the algorithm partitions the image into a set of square blocks (domain blocks). After this a new partition is made into smaller range blocks [2]. The domain blocks are generally double the size of range blocks. For every range block the nearest domain block is identified among all domain blocks after applying a set of transformations on the domain blocks. Smaller sized images are obtained by storing the information about these transformations alone. The transforms store the domain number, scaling constant, offset etc. This method of compression is called the partitioned iteration function system (PIFS). This paper explores the use of variable scaling factors for the transformation from domain to the range blocks. This variable factor has been generated using a pseudo-random number generator. The results show comparable ratios of compression and RMS error with PIFS (Partitioned iterated function systems) based fractal compression.

General Terms
Image Compression, Fractal, Non-linear, random, Range, Domain.

Keywords
Fractal image compression, IFS.

1. INTRODUCTION
Image compression makes image storage and transmission efficient. Excellent results have been obtained with fractal image compression in the past few years. Barnsley[1] promoted the most important fractal image compression algorithm based on Iterated Function System (IFS) called Partitioned Iterated function systems.

An iterated function system is a collection of contractive affine transformations. A basic theorem in fractal geometry states that each IFS, i.e. each set of contractive affine transformations, defines a unique image what is called fractal [2]. Fractal image compression is the inverse problem. Instead of generating an image from a given formula, the aim in fractal image compression is to find a set of transformations that can represent a given image.

The basic implementation of the PIFS compression method has tiled the image into B x B blocks in the image as the set of range blocks, and all (overlapping) 2B x 2B blocks in the image as the set of domain blocks [2]. The set of transformations applied consists of a spatial contraction, followed by one of the eight square symmetry operations (4 rotations and 4 reflections) followed by a contractive affine transformation on the grayscale value. The domain block is first brought down to the size of the range blocks. Each domain pixel is then multiplied by the scaling factor S and an offset O is added to it to get the corresponding range pixel. The following equation represents the transformation of domain pixels to range pixels.

$$R_i = S * D_i + O \qquad \text{equation(1)}$$

Where $-1<S<1$ guarantees contraction. In the above equation R_i represents the range pixel and D_i represents the domain pixel. Here S and O are so chosen that the RMS error between the domain and the range is a minimum. Search is made on the whole domain pool to identify the closest domain for each range. The union of these mappings defines the image.

Efforts have been made to reduce the search time for the closest domain by reducing the domain space based on polynomial approximation [5]. Spatial correlations [9] in both range and domain pools are utilized to reduce search space. In [10], mean and variance is used to classify the domain blocks improving the search speed for finding the nearest domain for a range block. Domain blocks can also be searched randomly [11] for every range to improve the speed of encoding. The decompression algorithm is improved by taking an initial seed value [6] other than arbitrary values or zero. If the closeness of the range and domain is not within a limit, quad-tree splitting can be made to reduce the size of range and domains [7] and increase accuracy at the cost of size of compressed image.

In this paper, we will discuss the usage variable scaling factors for transforming the domain pixels to range pixels. In PIFS, the scaling factor S (equation (1)) is a constant for a range-domain pair. These variable scaling factors are generated by the use of a pseudo random generator.

2. PIFS METHOD OF FRACTAL IMAGE COMPRESSION

In this section, we will first briefly discuss the method of PIFS [3] compression and decompression. We then investigate the methods of compression using the difference and non-linear methods [8].

2.1 Encoding procedure

The encoding procedure for PIFS:

1. Partitioning of the original image (Figure 1) into N non-overlapping range blocks.
2. Tiling of the image into M (possibly overlapping) domain blocks.

The following procedure is repeated for all range blocks.

1. Choose a range block R_i.
2. From all combinations of transformations for domain blocks, based on equation (1), choose the best transformation which minimizes the RMS distance between the range block and domain.

When the best pair has been found, only the transform detail for each range is stored. This transformation contains information about the positional description of the domain block D_j associated with a given range, the number of rotation operation, scaling (S) and offset (O) parameters.

2.2 Decoding procedure

The decoding procedure is as follows:

1. An initial image X (Figure 2) is chosen at random (usually a uniform gray image). A transformed image is created from the following transformation.
2. To get a range block, apply the transformation on its corresponding domain. The domain number is stored in the transformation.
3. When all range blocks are exhausted, the resulting image will contain the transformed version of the starting image.
4. In the next step we will transform the resulting image again starting from step 2.
5. Due to the contractive nature of the mappings, the resulting image will converge towards a final image after a few iterations (Typically 9 iterations are sufficient).

Figure 1: Original Image

Figure 2: Compressed image using PIFS technique

3. Randomized PIFS method

In randomized PIFS method, we store image using a pseudo random seed value and offset for every transformation. If the size of a range is 16, then, using the seed, 16 pseudo random numbers are generated in the range 0 to 1. These 16 values are considered to be the scale values for multiplying the domain pixels. A fixed number of different seed values are considered and an optimal seed value is stored along with the offset for every transformation. Different domains are not searched for optimal values of scaling and offset. Instead, only the first domain is considered for all ranges. Hence, it is not required to store the domain number in every transformation.

3.1 Transforms

The set of transforms are of the form

$$R_i = s_i * D_i + O \qquad \text{equation(2)}$$

Where R_i is the Range value of i^{th} pixel, D_i is the Domain value of i^{th} pixel, O is the offset value, s_i is the scale value that is a pseudo random number between 0 and 1 ($0 < s_i < 1$) for a particular seed.
The values stored for each transform are:

1. Seed for pseudo random generator.
2. Offset which minimizes the RMS error between the range block and domain block.
3. Symmetry.

3.2 Encoding Procedure

1. Partitioning of the original image into N non-overlapping range blocks $\{R_i\}N_{k=1}$

Repeat the following procedure for all range blocks:

1. Choose a range block.
2. Choose domain block number 1.
3. For all possible seed values, choose a seed value that minimizes the RMS error between the domain block and range block.
4. Store the seed value, offset and symmetry for each transform of the compressed image (Figure 3) .

Decoding Procedure

1. Create a random image.
2. Apply the transformation for each range block.
3. Repeat step 2 till the image converges. This normally happens in about 9 iterations.

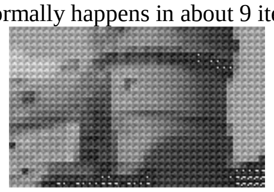

Figure 3: Compressed Randomized PIFS Image

4. Randomized non-linear PIFS

In this case, we compress the image using a power factor instead of scaling factor for every transform [8]. This makes the transforms non-linear.

4.1 Transform

The set of transforms are of the form

$$R_i = D_i^{P_i} + O \qquad \text{equation(3)}$$

Where R_i is the Range value of i^{th} pixel, D_i is the Domain value of i^{th} pixel, O is the offset value, P_i is the power value that is a pseudo random number between 0 and 1 ($0 < P_i < 1$) for a particular seed.
The values stored for each transform are

1. Pseudo-random seed which minimizes the RMS error between the domain and the range.
2. Offset which minimizes the RMS error between the range block and domain block.
3. Symmetry.

4.1 Encoding Procedure
The encoding procedure involves

1. Partitioning of the original image into N non-overlapping range blocks $\{R_i\}N_{k=1}$
2. Selection of a pseudo random seed for the power factor that minimizes the distance between the domain and the range.

Repeat the following procedure for all range blocks.

a. Choose a range block.
b. From all available seeds for generating pseudo random numbers, choose a seed which minimizes the distance between the only domain (first) and the range.
c. Store the seed value, symmetry and offset (O) that minimizes the distance between the domain and the range between the compressed (Figure 4) and the initial image.

4.2 Decoding Procedure
1. Create a random image.
2. Apply the transformation (equation 2) for each range block.
3. Repeat step 2 till the image converges. This normally happens in about 9 iterations.

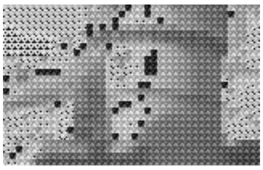

Figure 4: Cmpressed Randomized non-linear image

5. Difference based Randomized compression
Here difference based compression [8] is implemented using randomized scaling factors using pseudo random generators. Here, the difference between range and domain pixel is scaled. Pseudo random numbers are used for scaling rather than a fixed scaling factor.

5.1 Transform
The set of transforms are of the form

$$R_i = D_i + s_i * (R_i - D_i) + O \qquad \text{equation(4)}$$

Where R_i is the Range value of i^{th} pixel, D_i is the Domain value of i^{th} pixel, O is the offset value, s_i is the scaling factor that is a pseudo random number between 0 and 1 ($0 < s_i < 1$) for a particular seed.

5.2 Encoding Procedure
The encoding procedure involves

1. Partitioning of the original image into N non-overlapping range blocks $\{R_i\}N_{k=1}$
2. Selection of a pseudo random seed for determining the scaling factors that minimizes the distance between the domain and the range.
3. Storing the initial seed value for image that minimizes the RMS error between the compressed and decompressed image and the initial image.
4. Storing the number of iterations that minimizes the RMS error between the compressed image(Figure 5) and the initial image

Repeat the following procedure for all range blocks.

a. Choose a range block.
b. From all available seeds for generating pseudo random numbers, choose a seed which minimizes the distance between the only domain (first) and the range.
c. Store the seed value and offset (O) that minimizes the distance between the domain and the range.

5.3 Decoding Procedure
1. Create an image initialized with seed value.
2. Apply the transformation (equation 4) for each range block.
3. Repeat step 2 as many times as the number of iterations.

Figure 5: Compressed Difference based-Randomized image

6. Difference based Non-linear Randomized compression
6.1 Transforms
The set of transforms are of the form [8].

$$R_i = R_i + (D_i - R_i)^{P_i} + O \qquad \text{equation(5)}$$

Where R_i is the Range value of i^{th} pixel, R_n is the Range value of n^{th} pixel, D_i is the Domain value of i^{th} pixel, O is the offset value, P_i is the power value that is a pseudo random number between 0 and 1 ($0 < P_i < 1$) for a particular seed.

6.2 Encoding Procedure
The encoding procedure involves

1. Partitioning of the original image into N non-overlapping range blocks $\{R_i\}N_{k=1}$.
2. Selection of a pseudo random seed for the power factor that minimizes the distance between the only domain and the range.
3. Storing the initial seed value for image that minimizes the RMS error between the compressed, decompressed image and the initial image.
4. The number of iterations that minimizes the RMS error between the compressed image(Figure 6) and the initial image.

Repeat the following procedure for all range blocks.

a. Choose a range block.
b. From all available seeds for generating pseudo random numbers, choose a seed which minimizes the distance between the only domain (first) and the range.
c. Store the seed value, symmetry and offset (O) that minimizes the distance between the domain and the range.

6.3 Decoding Procedure

1. Create an image initialized with seed value.
2. Apply the transformation (equation 5) for each range block.
3. Repeat step 2 as many times as the number of iterations.

Figure 6: Compressed Difference based non-linear randomized Image

7. DISCUSSION

Instead of using constant scaling and power factors for compression using fractal techniques, we can use variable scaling and power factors. Since the pixels in a range are generally different, different scaling factors would help in getting better compression ratios. We have used randomized scaling factors being used for four different techniques and achieved reasonable compression ratios with good picture quality. Looking at the table, we see that difference based techniques perform better than the normal PIFS (randomized) techniques. Looking at the table 1, we see that difference based techniques perform better than the normal PIFS (randomized) techniques. Table 1 shows the comparison between the different image compression schemes experimented and the different RMS errors associated with them.

8. RESULTS

Table 1 shows the comparison between the different image compression schemes experimented and the different RMS errors associated with them when the image was compressed to 7.81 Kb each from an original image of size 15.6 Kb.

Table 1: Image compression results

Method	Compressed Size	RMS Error
PIFS	11.7 Kb	222
PIFS - Random	7.81 Kb	7350
Difference based Random	7.81 Kb	2591
Nonlinear Random	7.81 Kb	3448
Nonlinear Difference-based Random	7.81 Kb	6964

9. CONCLUSIONS

Compression can be achieved with randomized techniques. A fixed domain is sufficient to achieve reasonable levels of compression with good accuracy. The compressed images have smaller size when compared to other methods of fractal compression. The space required to address a large number of domains is replaced by a seed value which takes only 8 bits. Improvement in quality is not seen with increase in the number of tried seed values.

10. REFERENCES

[1] M. F. Barnsley and L. P. Huard, “Fractal Image Compression”, AK Peters. Ltd. (1992).

[2] Ning Lu, Fractal Imaging, Academic Press, 1997.

[3] Arnaud E. Jacquin, “Fractal Image Coding: A Review”, Proceeding of the IEEE, Vol. 81, No. 10, October 1993.

[4] Dan C. Popescu, Alex Dimca, and Hong Yan, “A Nonlinear Model for Fractal Image Coding”, IEEE Transaction on image processing, Vol. 6, No. 3, March 1997.

[5] Zhuang Wu, Bixi Yan, “An effective fractal image compression algorithm”, IEEE International conference on ICCASM, 2010, pp.139-143.

[6] Hosseini, Shookooh, Shahhosseini, Beizaee, “Speeding up fractal image de-compression”, IEEE International conference on ICCAIE, 2010, pp.521-526.

[7] Hui Yu, Li Li, Dan Liu, HongyuZhai, Xiaoming Dong, “Based on Quadtree Fractal Image Compression Improved Algorithm for Research”, IEEE Trans, 2010, pp.1-3.

[8] Dinesh Rao B., Ganesh Kamath and Arpitha K. J. “Difference based Non-linear Fractal Image Compression”. International Journal of Computer Applications 38(1):41-44, January 2012. Published by Foundation of Computer Science, New York, USA.

[9] T.K Truong, C.M Kung, , J.H Jeng, M.L Hsieh, “Fast fractal image compression using spatial correlation”, ScienceDirect, Vol 22, Issue 5, December 2004, Pages 1071–1076.

[10] Yung-Gi Wu, Ming-Zhi Huang, Yu-Ling Wen, "Fractal image compression with variance and mean", IEEE, vol.1, 18 August 2003.

[11] S. K. Ghosh, Jayanta Mukherjee, P. P. Das, “Fractal image compression: a randomized approach”, Pattern Recognition Letters archive, Volume 25 Issue 9, 2 July 2004, Elsevier Science Inc. New York, NY, USA.