

# Difference based Non-linear Fractal Image Compression

Dinesh Rao B.  
Associate Professor  
MCIS  
Manipal

Ganesh Kamath  
MS Student  
MCIS  
Manipal

Arpitha K. J.  
MS Student  
MCIS  
Manipal

## ABSTRACT

Good fractal compression methods rely on iterative function systems (IFS), which have been designed by Barnsley [1] and Jacquin [2]. The main principle is that an image can be built by combining parts which are similar to the other parts. When encoding an image, the image is partitioned into possibly overlapping square blocks called domain blocks. Subsequently a new partition is made into smaller range blocks [2]. For each range block the closest domain block is searched among all domain blocks by applying a set of transformations on the domain blocks. Compression is obtained by storing only the information about these transformations. The information stored includes domain number, scaling constant, offset etc. This method of compressing images is called Partitioned Iterated Function System (PIFS). This paper explores the use of difference between range and domain elements to find the scaling constants and offset. This paper also explores the implications of mapping IFS blocks on a non-linear basis between range and domain blocks. Further, difference based non-linear mapping is also looked upon. The results show comparable ratios of compression and RMS error with PIFS based fractal compression.

## General Terms

Image Compression, Fractal, Non-linear, Difference Based, Range, Domain.

## Keywords

Fractal Image Compression, IFS.

## 1. INTRODUCTION

The major application of compressing images lies in efficient storage and transmission. Good results have been achieved with fractal image compression in the recent past. Barnsley[1] promoted Iterated Function System (IFS) which forms a key technique on which many fractal compression techniques rely on.

An iterated function system is a collection of contractive affine transformations. A fundamental theorem in fractal geometry is that each IFS, that is each set of contractive affine transformations, defines a unique image or what is called fractal. The Fractal image compression technique 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 IFS compression method has tiled the image into  $B \times B$  blocks in the image as the set of range blocks, and all (overlapping)  $2B \times 2B$  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  $d$  the corresponding range pixel. The following equation represents the transformation of domain pixels to range pixels .

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

Where  $-1 < S < 1$  guarantees contraction. In the above equation  $R_i$  represents the range pixels and  $D_i$  represents the domain pixels. Here  $S$  and  $O$  are so chosen that the rms error between the domain and the range is minimum. Search is made on the whole domain pool to identify the closest domain for each range. The union of 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]. 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 discuss about construction of transforms based on the difference between the range and domain pixels. Linear fractal compression has proved to significantly able to reduce the size of images. We intend to present our findings on taking a non-linear approach in hopes to optimize upon compression. This paper further discusses the results comprising of a combination of difference based approach and a non-linear approach combined as one to create a compressed image. Attempts have been made to implement non-linear models of image compression using complicated transforms, Dan C. Popescu[3], we have used simple function, power for non-linearization.

## 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 method.

### 2.1 Encoding procedure

The following steps describe the encoding procedure:

1. Partitioning of the original image 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.

3. Choose a range block  $R_i$
4. 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 is chosen at random (usually a uniform gray image). A transformed image, is created from the transformation as follows:
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).

## 3. DIFFERENCE METHOD

This method relies on applying transforms on the difference between the range and the domain pixel values.

### 3.1 Transforms

The set of transforms are of the form

$$R_i = D_i + S * (R_i - D_i) + O \quad \text{equation(2)}$$

where  $R_i$  is a range pixel and  $D_i$  is a domain pixel. Here S is a scaling factor between 0 and 1 and O the offset.

The values stored for each transform are

- Scaling factor which scales the difference between the domain pixel and range pixel.
- Offset which minimizes the RMS error between the range block and domain block.
- The domain number.
- Initial seed value.
- Number of iterations.

### 3.2 Encoding Procedure

1. Partitioning of the original image into N non-overlapping range blocks.
2. Tiling of the image into M (possibly overlapping) domain blocks..  
Repeat the following procedure for all range blocks.
3. Choose a range block.
4. Consider all domain blocks for this range block and choose a domain and its transformation which minimizes the rms error.

The image is decoded starting with a seed value for a certain number of iterations. The optimal seed value, number of

iterations and the set of transformations which has least RMS difference with the original image is stored in the encoded image. The image does not contract to a single image but keeps changing based on different seed values and the number of iterations. The size of the compressed image is two bytes more than the one for PIFS method.

## 3.3 Decoding Procedure

1. Initial image S has seed value.
2. Apply the stored transformation to each range block.
3. Repeat step 2 as many times as the number of iterations.

Though the Compressed image (Figure 2) appears slightly blurrier than the original image (Figure 1), it still gives the overall details present in the image. Further research may improve the quality of the decompressed image.



Figure 1: Original Image



Figure 2: Compressed Image (difference method)

## 4. NON-LINEAR METHOD

Generally, fractal compressions give a good result to linear compression. However mapping of sound has shown that non-linear approaches may be better suited to some methods than linear methods. Based on the type of the image, certain images may be best suited for non-linear compression.

### 4.1 Transforms

The set of transforms are of the form

$$R_i = D_i^p + O \quad \text{equation(3)}$$

Where  $D_i$  is a domain pixel and  $R_i$  is a range pixel. Here, O is the offset and p is a real number between 0 and 1.

The values stored for each transform are

- The domain number D from the pool of domains which reduces the rms error from the range after applying transformation in equation (3).
- Power p in the range between 0 and 1. This helps in the convergence of the image.
- Offset O which minimizes the RMS error between the range block and domain block.

## 4.2 Encoding Procedure

The encoding procedure is as follows.

- Partitioning of the original image into N non-overlapping range blocks.
- Tilting of the image into M (possibly overlapping) domain blocks.

Repeat the following procedure for all range blocks.

- Choose a range block.
- Consider all domain blocks for this range block and choose a domain and its transformation which minimizes the rms error.

## 4.3 Decoding Procedure

The decoding procedure is as follows.

1. Initial image S has seed value of 0.
2. Apply stored transformation to each range block.
3. Repeat step 2 till the image converges. The image converges because of p being a value between 0 and 1.

Figure 3 shows the compressed and decompressed image using this technique.



Figure 3: Compressed Image (Non-linear method)

## 5. NON-LINEAR DIFFERENCE BASED METHOD

A combination of differenced based and non-linear approach may lead to a better image quality in some types of images.

### 5.1 Transforms

The set of transforms are of the form

$$R_i = D_i + (R_i - D_i)^p + O \quad \text{equation(4)}$$

Where  $R_i$  is a range pixel and  $D_i$  is a domain pixel. Here p is the power factor which is a fraction between 0 and 1 and O the offset.

The values stored for each transform are

- The domain number D from the pool of domains which reduces the rms error from the range after applying transformation in equation (3).
- Power p in the range between 0 and 1. This helps in the convergence of the image.
- Offset O which minimizes the RMS error between the range block and domain block.

The compressed image also stores

- Initial seed value.
- Number of iterations.

## 5.2 Encoding Procedure

The encoding procedure is as follows.

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

Repeat the following procedure for all range blocks.

- Choose a range block.
- Consider all domain blocks for this range block and choose a domain and its transformation (equation 4) which minimizes the rms error between the two.

The image is decoded starting with a seed value for a certain number of iterations. The optimal seed value, number of iterations and the set of transformations which has least RMS difference with the original image is stored in the encoded image. The image does not contract to a single image but keeps changing based on different seed values and the number of iterations. The size of the compressed image is two bytes more than the PIFS method.

## 5.3 Decoding Procedure

1. Initial image S has seed value stored in the compressed image.
2. Apply the stored transformation to each range block.
3. Repeat step 2 as many times as the number of iterations.

Figure 4 relates to the compressed and decompressed image following the above technique.



Figure 4: Compressed Image (Non-linear difference method)

## 6. DISCUSSION

Difference based image compression is sensitive to initial value and number of iterations. We need to figure out optimum values for these two quantities. If the seed value is changed, the resultant image also changes. The image does not converge to the same value as the number of iterations increase.

In all the above implementations, the scaling factor S has been kept as 0.75 and power p is also assigned 0.75. Varying these values give small changes in RMS error values.

The pattern of image changes based on seed values and the number of iterations are interesting. The RMS error decreases and then increases with the change in seed values.

Non-linear methods can be used to compress images. Convergence can be achieved by keeping the exponent to a value between 0 and 1. Other simple functions like logarithm, sine and cosine can also be used to achieve convergence.

## 7. 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 from 15.6 Kb to 11.7 Kb each.

**Table 1: compression results**

Method	Size	RMS Error
PIFS	11.7 Kb	222
Nonlinear PIFS	11.7 Kb	1022
Difference	11.7 Kb	2763
Difference-based Non-linear	11.7 Kb	48000

## 8. CONCLUSIONS

We can use the difference method to achieve fractal image compression. Here, we have used linear transforms for compression of images based on difference between range and domain pixel values, performed non-linear transform on the image as well as performed a difference based non-linear transform. Non linear transforms can be used to improve the compression factor. Study of the change in image due to seed change and change in iterations seem to be interesting.

There is a lot of scope for improving these methods and finding out the right type of transform may hold the key to developing the best image compression method.

## 9. 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, Beizae, “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.