Fractal Image Compression and IFS

Ritu Shrivastava(1), S. C. Shrivastava(2)
(1) Govt. Naveen College Risali, Durg, (C.G.), India.
(2) Department of Applied Mathematics, Rungta College of Engineering and Technology,Bhilai (C.G.), India.

ABSTRACT
Image compression is a process of reducing or eliminating all redundant or irrelevant data. The new compression technique called Fractal image compression. This scheme works by partitioning an image into blocks and using contractive mapping to map each range blocks to its matched domains. In present study we review the work on encoding and decoding to compress the image by fractal image compression with IFS and also study the role of affine transformation and methodology of affine transformation in fractal image compression.

Keywords: Affine Transformation, Contractive mapping. Fractal encoding and decoding, IFS, Self-similarity.

1. INTRODUCTION

Day by day, the demands for higher and faster technologies are rapidly increasing for everyone. Now a days before purchasing computers, customers are concerned about two things (1) the speed of the CPU and (2) the storage and memory capacity. Image compression helps to reduce the memory capacity and to have faster transmission rate [1].

There are two types of image compression is present. They are lossy and loss-less method.

(1) In lossy method:- The reconstructed image contains degradation relative to the original and lossy technique image quality degradation in each compression step. Lossy compression technique lead to loss of data with higher com-pression ratio. (2) Lossless method:- The reconstructed image after compression is numerically identical to the original image. Lossless compression gives good quality of compressed images [2].. Digital images are claiming an increasingly larger portion of the information world. The growth of the Internet and continuing advances in the technology of digital cameras, scanners and printers, has led to the widespread use of digital imagery. As a result there is renewed interest in improving algorithms for the compression of image data. Image compression is a process of reducing or eliminating redundant or irrelevant data. In general, lossy techniques provide for greater compression ratios than lossless techniques i.e. Lossless compression gives good quality of compressed images, but yields only less compression where as the lossy compression techniques lead to loss of data with higher compression ratio. The approaches for lossless image compression include variable-length encoding, Adaptive dictionary algorithms such as LZW, bit-plane coding, lossless predictive coding, etc. The approaches for lossy compression include lossy predictive coding and transform coding. Transform coding, which applies a Fourierrelated transform such as DCT and Wavelet Transform such as DWT are the most commonly used approach. Compression is important both for speed of transmission and efficiency of storage.

In this paper, we will study the technique fractal image compression and basic concepts to keep in mind for the fractal image coding and some speed up techniques for encoding.

2. FRACTAL IN IMAGE COMPRESSION

Lossy image coding by partitioned iterated function system (PIFS), popularly known as Fractal Image Compression, has recently become an active area of research. Fractal theories are totally different from the others [1].

The idea fractal image compression is to find subspaces (or sub images) of the original image space, which can be regenerated using IFS. Where possible, if on IFS can be used in place of several IFS's which reproduced similar sub images, it is more efficient in terms of storage space to use that one IFS. It is more likely that an image will require more than one IFS to reproduce a compressed image, which closely resembles the original [3].

Fractal image compression is also called as fractal image encoding because compressed image is represented by contractive transforms and mathematical functions (iterated functions) required for reconstruction of original image[4], [5]. A. E. Jacquin suggested to have the range and domain blocks to be always in the shape of a square and the domain size to be twice the size of the range [1].

In fractal compression system the first image is partitioned to form of range blocks then domain blocks are selected. This choice depend on the type of partitioned scheme used then set of transformation are selected which are applied on domain blocks to range blocks and determines the convergence properties of decoding [1].

Figure 1 shows the whole process of fractal image encoding. In any Fractal compression system the first image is partitioned to form of range blocks. Then domain blocks are selected. This choice depends on the type of partition scheme used. The domain pool in fractal encoding is similar to the codebook in vector quantization (VQ), referred as virtual codebook or domain codebook. Then set of transformation are selected which are applied on domain blocks to form range blocks and determines the convergence properties of decoding.

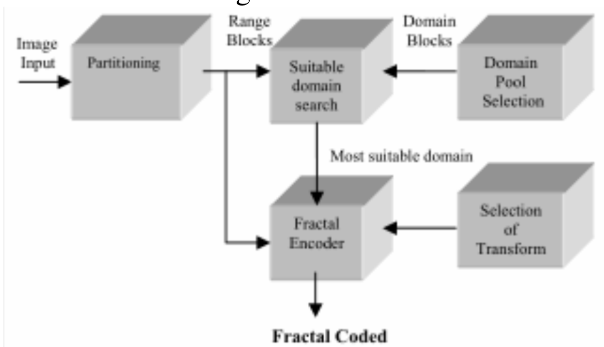


Figure 2 Block diagram of Fractal Image Compression

2.1 Fractal Image compression has the following features [1]:

- Compression has high complexity.
- Fast image decoding.
- High compression ratio can be achieved.`
- It is resolution independent.

2.2 Some Basic Terms for Fractal Image Compression

Self-similarity

Subsets of fractals when magnified appear similar or identical to the original fractal and to other subsets. This property is called self-similarity and it makes fractals independent of scale and scaling. Thus there is no characteristic size associated with a fractal. A typical image does not contain the type of self-similarity found in fractals. But, it contains a different sort of self-similarity. The figure 2 shows regions of Lena that are self-similar at different scales. A portion of her shoulder overlaps a smaller region that is almost identical, and a portion of the
reflection of the hat in the mirror is similar to a smaller part of her hat.

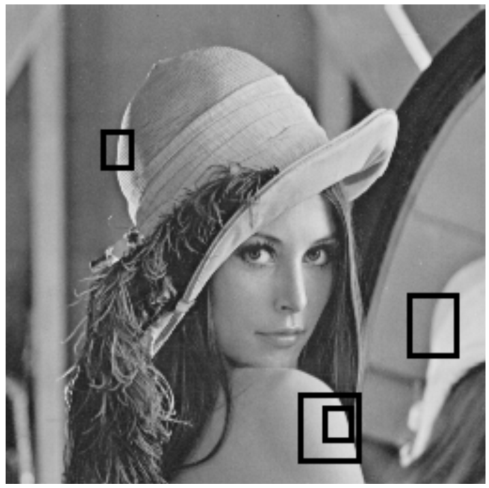


Figure 2 Self Similarity in Lena image

The difference here is that the entire image is not self-similar, but parts of the image are self- similar with properly transformed parts of itself. Most naturally occurring images contain this type of self-similarity. It is this restricted redundancy that fractal image compression schemes attempt to eliminate.

2.3 Fractal Image Compression has three basic steps [1] :

- Partition the image.
- Encoding.
- Decoding.

3. FRACTAL IMAGE COMPRESSION ALGORITHM

3.1 Encoding

The image should be used in this compression are of the square size. The image is partitioned into non-overlapping square blocks R_n (range blocks) of size $B * B$ and large over lapping square blocks D_n(domain blocks) of size $2B * 2B$. That means, the pixels in the domain are average in group of four so that the domain is reduced to the size of range [2].

For a range block R_n, we would find the best domain block D_n with the corresponding mapping T_n . Apply IFS transformation from domain block to range block. To use IFS to reproduce images by partitioning an image into blocks, typically $8 * 8$ or $16 * 16$ pixels, it becomes possible to map small portions of an image to large portions. The smaller portions are reproduced by using affine transformations. These transformation effectively map squares to parallelograms through translation , scaling, skewing, rotation etc [6].

The affine transformation of the pixel values is found that minimizes the rms difference between the transformed domain pixel values and range pixel value. Select best domain with best transformation (compare each range block with whole domain blocks to find the best match). In this way an image can be stored as a collection of affine transformations that can be used to reproduce a near copy of original image [2].

Once the best matching domain block is obtained, the reconstruction error is estimated. If it is small than the threshold, a fractal code is generated, otherwise the range block is split into four sub blocks of size $B/2 * B/2$. Pixels to be considered in the next level of the quadtree decomposition. The partitioning process finishes when a minimum block size is reached [7].

3.2 Decoding

The decoding in fractal compression is much faster compressed with the encoding, here the time depends on the number of iterations, however, we will see that only a few iterations are required to reach the fixed point or attractor [1].

Load the saved coefficients. For decoding, an image consists of iteration T from any initial image. In every iteration, for each range R_n , the domain D_n that maps to it shrunk by two in each dimension by averaging non-overlapping groups of pixels and stored the fractal codes information $\{Dxi;\ Dyi;\ si;\ oi;\ Ui\}$ that is location of the domain block in the image space, contrast factor, brightness and type of affine transformation. The shrunk domain pixel values are multiplied by s_i added to and o_i placed in the location in the range determined by the orientation information. This is one decoding iteration

We can define $T_i\ : F \rightarrow F$ operating on image f(x, y) by

$$T_i(f)(x,y) = s_i\, f\big(T(x,y)\big) + o_i$$

Provided T_i is invertible and $(x,y) \epsilon R$

The decoding step is iterated until the fixed point is approximated [2].

4. Affine Transformation

The use of homogeneous coordinates is the central point of affine transformation which allow us to use the mathematical properties of matrices to perform transformations. So to transform an image, we use a matrix $T \in M_3(R)$ providing the changes to apply

$$T = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ p_x & p_y & 1 \end{bmatrix}$$

The vector $[T_x, T_y]$ represent the translation vector according the canonical vectors. the vector $[P_x, P_y]$ represents the projection vector on the basis. The square matrix composed by the a_{ij} elements is the affine transformation matrix[3], [5].

An affine transformation $T:\ R^2 \rightarrow R^2$ is a transformation of the form $T: Ax + B$ defined by

$$T\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ 1 \end{pmatrix}$$

Where the parameter $a_{11}, a_{12}, a_{21}, a_{22}$ form the linear part which determines the rotation, skew and scaling and the parameters t_x, t_y are the translation distances in x and y directions, respectively[1], [2].

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = T\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = x\, a_{11} + y\, a_{12} + t_x$$

$$y' = x\, a_{21} + y\, a_{22} + t_y$$

The general affine transformation can be defined with only six parameter:

θ : the rotation angle.

t_x : the x component of the translation vector.

t_y : the y component of the translation vector.

S_x : the x component of the scaling vector.

S_y : the y component of the scaling vector.

Sh_x : the x component of the shearing vector.

Sh_y : the y component of the shearing vector[3].

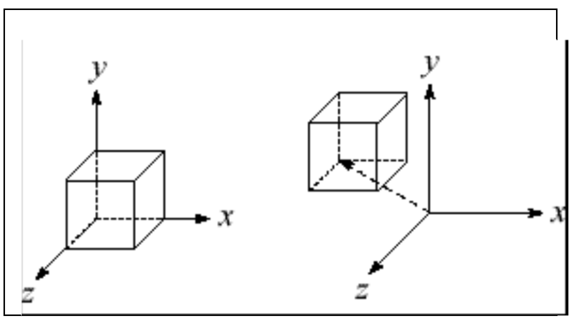
In the other words, The Fractal is made up of the union of several copies of itself, where each copy is transformed by a function T_i, such a function is 2D affine transformation, so the IFS is defined by a finite number of affine transformation which characterized by Translation, scaling, shearing and rotation.

4.1 Translation

A translation is a function that moves every point with constant distance in a specified direction by the vector $T = [T_x, T_y]$ which provide the orientation and the distance according to axis x and y respectively[3].

The mathematical transformation is the following[4], [6]:

$$T' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
$$x' = x + t_x$$
$$y' = y + t_y$$



4.2 Transvection

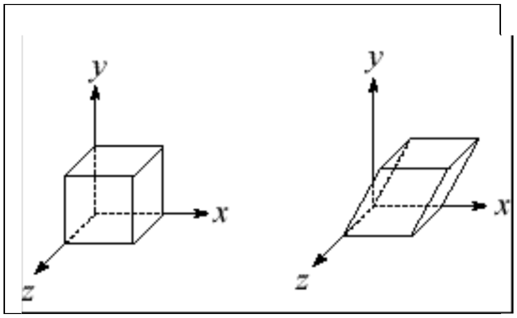
A transvection is a function that shifts every point with constant distance in a basis direction x or y[3].

The mathematical vertical and horizontal transvection respectively is the following[4], [6]:

$$T' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
$$x' = x + y\, s_v$$
$$y' = y$$

$$T' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
$$x' = x$$

$$y' = x\,s_h + y$$



5. Iterated Function System and Attractor

According to great mathematician Cauchy "A convergent sequence converges to a unique value." Suppose we have a sequence defined by f (n) = n/2. It is obvious that this sequence will converge to zero whatever may be the initial value of n. On the same lines we can say that a contractive mapping converges to a unique fixed point irrespective of the shape and size of initial image fed to be contracted. If such mappings are applied to an image then the final image that we get is called ***Attractor***. Like the unique value of a sequence, the shape of attractor is independent of the shape of initial image but is dependent upon their position and orientation[8].
□□□It reduces the initial image by half. □□It places the three copies of the image in a triangular configuration.
There is a provision of giving the output of one copy operation as the input for the second copy operation i.e. machine can function in an iterative fashion.

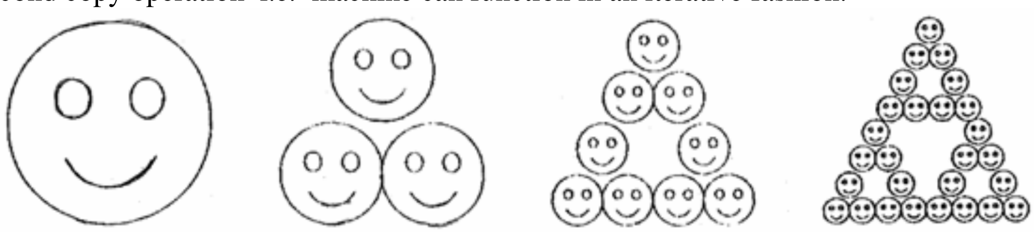


Figure 3 Contracting transformations by Iterated Function System

An Iterated Function System (IFS) is a collection of contractive mappings .Let IFS be denoted by W and the
collection of transforms by $w1, w2, w3\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots wn.$

$$W(X) = w1\,(X)\,U\,w2\,(X)\,U\,w3(X)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots U\,wn(X)$$

Where X is the grayness level of the image upon which the transform is applied.
If w's are contractive in plane then W will be contractive in space. Let Xw be an attractor resulted by the repeated
applications of W on an initial image, then after applying further transformation it will converge to Xw only.

Fractals reproducing realistic shapes, such as mountains, clouds, or plants, can be generated by the iteration of one or more affine transformations. An affine transformation is a recursive transformation of the type

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

Each affine transformation will generally yield a new attractor in the final image. The form of the attractor is given through the choice of the coefficients *a* through *f*, which uniquely determine the affine transformation. To get a desire shape, the collage of several attractors may be used (i.e. several affine transformations). This method is referred to as an Iterated Function System (IFS)[8].

An example of an iterated function system is the black spleenwort fern. It is constructed through the use of four affine transformations (with weighted probabilities):

Transformation	Translation	Probability
$\begin{pmatrix} +0.00 & +0.00 \\ +0.00 & +0.16 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	1%
$\begin{pmatrix} +0.85 & +0.04 \\ -0.04 & +0.85 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1.6 \end{pmatrix}$	85%
$\begin{pmatrix} +0.20 & -0.26 \\ +0.23 & +0.22 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 1.6 \end{pmatrix}$	7%
$\begin{pmatrix} -0.15 & +0.28 \\ +0.26 & +0.24 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0.44 \end{pmatrix}$	7%

The resulting image is:

Figure 4 spleenwort fern

This image is infinitely complex — it is a self-similar fractal on all scales. What is truly amazing is that only 28 numbers are necessary to generate this infinitely complex image: four 2 x 2 transformation matrices, four 2 x 1 translational vectors, and four weighted probabilities for the transformations (each attractor).

6. CONCLUSION

In this paper, we have described the nature of image compression system based on a fractal theory of iterated contractive image transformations. The advantage of using fractal image compression is that for each range block we have to save only few coefficients, which will give the ability of obtaining a very high compression ratio.

REFERENCES

[1] B. Sankaragomathi, L. Ganesan, S. Arunmugan(2007), Encoding video sequences in fractal based compression, June

[2] Meenu Bansal, Sukhjeet K. Ranade(2012), A review on fractal image compression, International Journal of advances in computing and information technology, July.

[3] Hilashi, Gaganpreet kaur, Sugandha Shrma(2012), Fractal image compression A review, International Journal of advanced research in computer science and software engineering, Vol2, Issue 2, February.

[4] A.E. Jacquin(1990), A novel fractal block coding technique for digital images, ICASSP, International conference on Acoustics, speech and signal processing.

[5] A.E. Jacquin(1922), Image coding based on a fractal theory of iterated contractive image transformation, IEEE Transactions on image processing, Vol1, No1,January.

[6] Vijyayshri Chaurasia and Ajay Somkumar(2010), Approximation error based suitable domain search for fractal compression, International Journal of Engineering science and Technology, Vol2, pp 104-108.

[7] William Robson Schwartz, Helio Pedrant(2011), Improved fractal image compression based on Robust feature descriptors, Institute of computing, Universityof campinas.

[8] M. Bransely(1988), Fractal Everywhere, NewYork academic.