

FRACTAL IMAGE COMPRESSION USING QUADTREE DECOMPOSITION AND HUFFMAN CODING

Veenadevi.S.V.¹ and A.G.Ananth²

¹Department of Electronics & Communication Engineering, RVCE, Bangalore, INDIA
veenadevi@yahoo.co.in

²Department of Tele Communication Engineering, RVCE, Bangalore, INDIA
antisro@gmail.com

ABSTRACT

Fractal image compression can be obtained by dividing the original grey level image into unoverlapped blocks depending on a threshold value and the well known techniques of Quadtree decomposition. By using threshold value of 0.2 and Huffman coding for encoding and decoding of the image these techniques have been applied for the compression of satellite imageries. The compression ratio (CR) and Peak Signal to Noise Ratio (PSNR) values are determined for three types of images namely standard Lena image, Satellite Rural image and Satellite Urban image. The Matlab simulation results show that for the Quad tree decomposition approach shows very significant improvement in the compression ratios and PSNR values derived from the fractal compression with range block and iterations technique. The results indicate that for a Lena image C R is 2.02 and PSNR values is 29.92, Satellite Rural image 3.08 and 29.34, Satellite urban image 5.99 and 28.12 respectively The results are presented and discussed in this paper.

KEYWORDS

Quadtree, fractal, Huffman encoding, decoding, image compression, PSNR.

1. INTRODUCTION

Image storage and transmission have created an increasing importance for the compression techniques. The redundancy and similarity among different regions of images makes compression feasible. In such cases Fractal compression of images is an efficient method of compression. A Fractal is a rough or fragmented geometric shape that can be split into parts, each of which is a reduced-size copy of the whole, a property called self-similarity. A mathematical fractal is based on an equation that undergoes iteration, a form of feedback based on recursion [1]. Fractals are easily found in nature. These objects display self-similar structure over an extended, but finite, scale range [2]. Examples include clouds, snowflakes, crystals, mountain ranges, lightning, river networks, and systems of blood vessels and pulmonary vessels. Trees and ferns are fractal in nature and can be modelled on a computer by using a recursive algorithm [5]. Fractal method based on block forming was first suggested by Jaquin in 1990. In fractal compression the coder first divides the image into a number of square blocks called range. Further the image is divided into bigger square blocks, called domain blocks, which are usually four times larger than the range block [10]. The domain blocks with their rotations are searched for the best match for every range block. For every range block the number of the appropriate domain and relevant information needed to retrieve that range are stored. Hence the compression is achieved in place of storing a range block only the parameters are stored. The decoder performs a number of iterative operations in order to reconstruct the original image. Fractal compression is very beneficial due to high Compression ratio, the decoding stage of the algorithm is independent of

the reconstructed image and the reconstructed image is of good quality [11]. PSNR is used to compute the quality of the reconstructed image compared with the original image by measuring the differences between the two images.

2. FRACTAL IMAGE COMPRESSION

The interest in fractal theory has focused on indexing and compression technique and developing several post processing strategies for this approach. In spite of the manifold advantages offered by fractal compression, such as high decompression speed, high bit rate, and resolution independence, the greatest disadvantage is the high computational cost of the coding phase. At present, fractal coding cannot compete with other techniques, if compression is considered. However, most of the cases of fractal coding are due to its applications in image database indexing and face recognition. These applications utilize some type of coding and have a good discriminating power even when the PSNR is high for the coding module. The Fractal coding is often the computational bottleneck, so it is desirable to make it as efficient as possible [12]. The way the input image is transformed determines the final result when running the copy machine in a feedback loop. However we must constrain these transformations, with the limitation that the transformations must be contractive that is, a given transformation applied to any two points in the input image must bring them closer in the copy. This technical condition is quite logical, since if points in the copy were spread out the final image would have to be of infinite size. Except for this condition the transformation can have any form [1]. Choosing transformations of the form,

$$w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}$$

Sufficient to generate interesting transformations called affine transformations of the plane. Each can skew, stretch, rotate, scale and translate an input image. A common feature of these transformations that run in a loop back mode is that for a given initial image each image is formed from a transformed copies of itself, and hence it must have detail at every scale. That is, the images are fractals. An affine transformation maps a plane to itself. The general form of an affine transformation is

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} ab & 0 \\ cd & 0 \\ 00 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e \\ f \\ 0 \end{bmatrix}$$

Where the parameters a, b, c, d form the linear part, which determines the rotation, skew and scaling, and the parameters s and o are the contrast and brightness adjustments for the transformation.

2.1. Contractive Transformations

A transformation w is said to be contractive if for any two points P_1, P_2 , and the distance $d(w(P_1), w(P_2)) \leq s d(P_1, P_2)$

For some $s < 1$. This gives the application of a contractive map always brings points closer together [1]. This definition is applying to any space on which we can define function $d(P_1, P_2)$, so that if the points have coordinates $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, then

$$d(P_1, P_2) = \text{SQRT}((x_2 - x_1)^2 + (y_2 - y_1)^2)$$

3. QUADTREE DECOMPOSITION

Fractal Geometry has become an important branch of modern mathematics and nonlinear science, it has been widely used covering many branches of science and engineering. At present, among the studies of fractal compression encoding, there are two research focuses on the application of fractal on the field of image compression. The main problem is that the fractal encoding is taking too much time. Many approaches to reduce the encoding time has bad affection on the image quality after iteration, therefore the hybrid encoding method of combining fractal coding and other coding methods becomes an important direction of fractal methods. The Quad tree approach divides a square image into four equal sized square blocks, and then tests each block to see if meets some criterion of homogeneity. If a block meets the criterion it is not divided any further, and the test criterion is applied to those blocks. This process is repeated iteratively until each block meets the criterion. The result may have blocks of several different sizes [3][4][6].

3.1 Huffman Encoding

The Huffman encoding algorithm starts by constructing a list of all the alphabet symbols in descending order of their probabilities. It then constructs, from the bottom up, a binary tree with a symbol at every leaf. This is done in steps, where at each step two symbols with the smallest probabilities are selected, added to the top of the partial tree, deleted from the list, and replaced with an auxiliary symbol representing the two original symbols [8]. When the list is reduced to just one auxiliary symbol (representing the entire alphabet), the tree is complete. The tree is then traversed to determine the code words of the symbols.

3.2 Huffman Decoding

Before starting the compression of a data file, the encoder has to determine the codes. It does that based on the probabilities of frequencies of occurrence of the symbols. The probabilities or frequencies have to be written, as side information, on the output, so that any Huffman decoder will be able to decompress the data. This is easy, because the frequencies are integers and the probabilities can be written as scaled integers. It normally adds just a few hundred bytes to the output. It is also possible to write the variable-length codes themselves on the output, but this may be awkward, because the codes have different sizes. It is also possible to write the Huffman tree on the output [7], but this may require more space than just the frequencies. In any case, the decoder must know what is at the start of the compressed file, read it, and construct the Huffman tree for the alphabet. Only then can it read and decode the rest of its input. The algorithm for decoding is simple. Start at the root and read the first bit off the input (the compressed file). If it is zero, follow the bottom edge of the tree; if it is one, follow the top edge. Read the next bit and move another edge toward the leaves of the tree. When the decoder arrives at a leaf, it finds there the original, uncompressed symbol, and that code is emitted by the decoder. The process starts again at the root with the next bit.

4. THE PROPOSED ALGORITHM

The algorithm steps are as follows.

1. Divides the original image using Quadtree decomposition of threshold is 0.2, minimum Dimension and maximum dimension is 2 and 64 respectively.
2. Record the values of x and y coordinates, mean value and block size from Quadtree Decomposition.
3. Record the fractal coding information to complete encoding the image using Huffman coding and calculating the compression ratio.
4. For the encoding image applying Huffman decoding to reconstruct the image and calculating PSNR. Figure 1. Shows the proposed Fractal Compression Technique.
- 5.

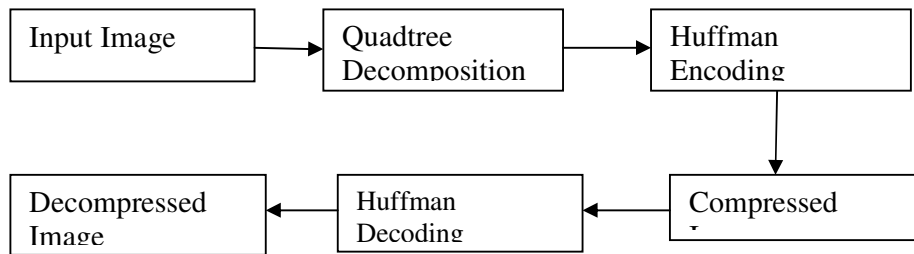


Figure 1. The proposed Fractal Compression Technique

5. RESULTS AND DISCUSSION

The algorithm realized in Matlab to code and to decode the satellite image of Urban of size 1377 X 955 , Rural image of size 995 X 571 and Lena image of size 512 X 512. But all these Images are resized to 256 X 256. Threshold value is 0.2, maximum dimension is 64 and minimum dimension is 2. The compression ratios and the PSNR values obtained for the reconstructed Lena Image, Satellite Rural Image and Satellite Urban Image is listed in Table 1. The original image and the decompression image after Quadtree decomposition, Huffman encoding and decoding is shown in Figure 2, 3 and Figure 4.

It can be seen from the Table 1 that for all the three images the PSNR values are greater than 28, the compression ratios achievable different . The Urban image with PSNR ~28.12 shows the maximum compression ratio ~5.99. The Rural image with PSNR ~29.34 shows high compression ratio ~3.08 and the Lena image with PSNR~ 29.92, the compression ratio ~2.02. The higher compression ratio for satellite Urban image can be explained in terms of the occurrence of fractals due to repetition of urban features at regular intervals for ex :Buildings, Roads and Parks etc, which are manmade and artificial repeated patterns seen in a urban scene. These features can be easily compressed by fractal compression and Quadtree decomposition. Where as the Rural image contains fractals they are less occurring compared to urban images as the rural images contain only natural features such as vegetation, water bodies, dry soil etc which are randomly distributed and not have much of repetition in their occurrence. Therefore one can achieve only lesser compression ratios for rural images compared to the urban images. However the compression ratio achieved for Lena images being the lowest brings out the fact that there are no fractals in the Lena image therefore it is not possible to achieve significant compression for such images.

TABLE 1. THE COMPRESSION RATIOS AND PSNR VALUES DERIVED FOR SATELLITE IMAGERIES

TEST IMAGES	TIME TAKEN FOR COMPRESSION (SEC)	COMPRESSION RATIO	TIME TAKEN FOR DECOMPRESSION (SEC)	PSNR IN DB
Lena image	15.86	2.02	20.20	29.92
Satellite Rural image	19.43	3.08	30.93	29.34
Satellite Urban image	23.92	5.99	66.39	28.12

It is clearly evident from the table that for three types of images with reasonably good PSNR values clearly indicate that the compression ratio achievable for satellite imageries is much higher compared to the standard Lena image. A detailed comparison of the results with the compression ratios achieved using range block and iteration techniques [9] show much less compression ratios ~3.2 and much lower PSNR values for both satellite Urban and Rural imageries. The present analyses suggests that the fractal compression techniques with Quadtree decomposition is a better approach towards achieving maximum compression ratios and more suited for compression high resolution Urban satellite imageries

Original image Quadtree decompos Decompressed image

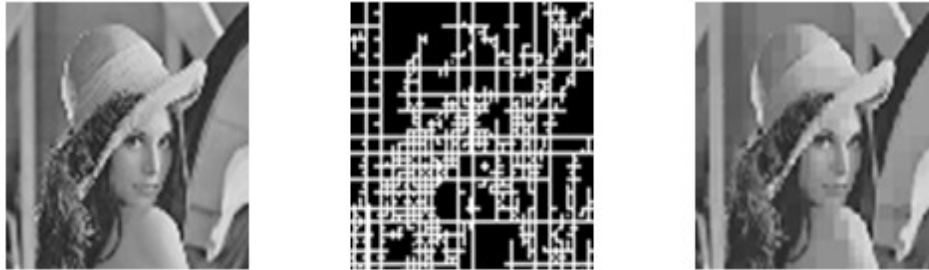


Figure 2. Lena image

Original image Quadtree decompos Decompressed image

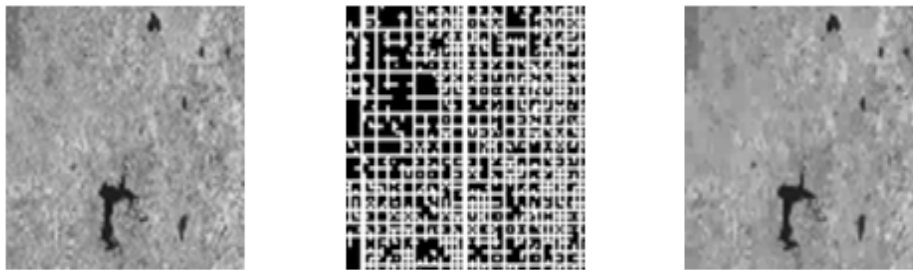


Figure 3. Satellite Rural image

Original image Quadtree decompos Decompressed image

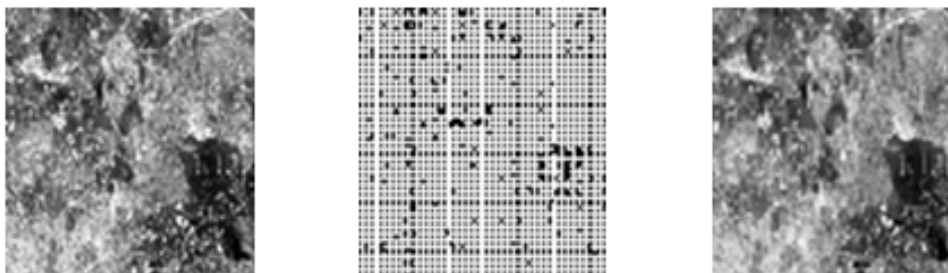


Figure 4. Satellite Urban image

The reconstructed images after fractal compression are shown in Figure 2, 3 and 4. It is evident for the Figures that, the reconstructed image quality is good as the PSNR values for the three images indicate significantly higher values >28 . The Figures clearly suggests that the fractal compression with Quadtree decomposition find very good applications for achieving high compression ratios with reasonably good PSNR values for satellite Urban and Rural Imageries.

6. CONCLUSIONS

From the analysis carried out in the paper the following conclusions can be drawn

- The Quadtree decomposition and Huffman techniques can be applied for achieving high compression ratios and better PSNR values for satellite Imageries.
- For satellite Urban images shows highest compression ratios 5.99 with PSNR values 28.12, the satellite Rural images shows compression ratios 3.08 with PSNR~29.34.
- For the Lena Image the compression ratio 2.02 with PSNR ~ 29.92 is the lowest compared to Satellite Rural and Urban images.

REFERENCES

- [1] Fisher Y, editor (1995) "Fractal image compression: theory and application", New York, Springer-Verlag.
- [2] Arnaud E. Jacquin, (1993) "Fractal image coding", Proceedings of IEEE VOL.81, pp. 1451-1465
- [3] Bohong Liu and Yung Yan, (2010) "An Improved Fractal Image Coding Based on the Quadtree", IEEE 3rd International Congress on Image and Signal Processing, pp. 529-532.
- [4] Hui Yu, Li Li, Dan Liu, Hongyu Zhai, Xiaoming Dong, (2010) "Based on Quadtree Fractal Image Compression Improved Algorithm for Research", IEEE Trans, pp.1-3.
- [5] Barnsley MF,(1993) "Fractal everywhere", 2nd ed, San Deigo Academic Press.
- [6] Dr. Muhammad Kamran, Amna Irshad Sipra and Muhammd Nadeem, (2010) "A novel domain Optimization technique in fractal image compression", IEEE Proceedings of the 8th World Congress On Intelligent Control and Automation, pp. 994-999.
- [7] Manoj Aggarwal and Ajai Narayan (2000) "Efficient Huffman Decoding", IEEE Trans, pp.936-939.
- [8] H.B.Kekre, Tanuja K Sarode, Sanjay R Sange(2011) " Image reconstruction using Fast Inverse Halftone & Huffman coding Technique", IJCA,volume 27-No 6, pp.34-40.
- [9] VeenaDevi.S.V and A.G.Ananth (2011) "Fractal Image Compression of Satellite Imageries", IJCA, Volume 30-No.3, pp.33-36.
- [10] Jinshu Han (2007) "Speeding up Fractal Image Compression Based on Local Extreme Points", IEEE Computer Society, pp. 732-737.
- [11] Narges Rowshanbin, Shadrokh Samavi and Shahram Shirani (2006) "Acceleration of Fractal Image Compression Using Characteristic Vector Classification", IEEE CCECE/CCGEI, pp.2057-2060.
- [12] Riccardo Distasi, Michele Nappi and Daniel Riccio (2006) "A Range/Domain Approximation Error Based Approach for Fractal Image Compression" IEEE Trans on Image Processing VOL.15, No1. pp. 89-97.