A FAST ENCODING FRACTAL IMAGE COMPRESSION USING QUANTIZED QUAD-TREE PARTITIONING TECHNIQUES

1Atish Khobragade, 2Ajay Meshram and 3Komal Meshram
1Dept. of Electronics Engineering, Yeshwantrao Chavan College of Engineering, India
2Dept. of BioChemistry, Datta Meghe Institute of Medical Science Wardha, India
3Department of Physiology, Datta Meghe Medical College, Nagpur, India
Corresponding Author Email: {atish_khobragade@rediffmail.com}.

Abstract—Fractal Image Compression is a lossy compression method for images based on fractals rather than pixels, which are best suited for texture and natural images. It works on self similarity property, relying on the fact that parts of an image often resemble other parts of the same image. It takes long encoding time and affects the image quality. This paper introduces an improved quantized quad tree technique for fractal image compression. Quantized quad tree method divides the original gray level image into various blocks depending on features presented in image. Simulation results show that with the quantized quad trees compression improves the compression ratio and PSNR without losing the quality of the image compression with range block, domain block and iterations technique.

Keywords: Fractals, Affine Transformation, Encoding Time, Compression Ratio, PSNR

1. Introduction
The concept of fractal was described by IBM mathematician Benoit Mandelbrot. He found that traditional geometry was inadequate to describe the structure of natural objects which are complex such as mountain, cloud, coastlines and tree. M. Barnsley suggests principle of fractal image compression in 1988. Affine transformation applied on fractal image compression is combination of transformations i.e. rotation, scale change, skew or translation. The image after applying affine transformation is an approximation of original image. In Jacquin method the image is partitioned in sub images called as Range blocks and Partitioned Iterated Function System is applied on sub-images, rather than the entire image to find out most suitable domain block. Basic steps in fractal image compression are: (1) Partition of image to be encoded to form range blocks (2) Selection of domain pool (3) Form a class of transform applied to the domain block (4) Searching most suitable domain block for formation of particular range block.

The aim of compression algorithm is to increase compression ratio and to minimize encoding time.

1.1 Overview of fractal image compression
The first decision regarding partition is the choice of the type of image partition used for the range blocks. Since domain blocks must be transformed to cover range blocks, this decision, together with the choice of block transformation described later, restricts the possible sizes and shapes of the domain blocks. A wide variety of partitions have been investigated, the majority being composed of rectangular blocks.

The following are the different types of partition scheme:

- a) Fixed block size
- b) Quadtree
- c) Horizontal Vertical
- d) Irregular partition

Fig 1 Types of Partitioning

- a) Triangular (3-side split)
- b) Triangular (3-side split)
- c) Delaunay triangulation
- d) Polygonal

Fig 2 Types of Partitioning
Fixed size square blocks partitioning: The simplest possible range partition consists of the fixed size square blocks Fig 1(a).

Quad tree partitioning: The quad tree partition employs the well known image partitioning technique based on a recursive splitting of selected image quadrants, enabling the resulting partition to be represented by a tree structure in which each non-terminal node has four descendents. The usual top down construction starts by selecting an initial level in the tree, corresponding to some maximum range block size, and recursively partitioning any block for which a match better than some preselected threshold is achieved Fig 1(b).

Horizontal-vertical partitioning: The Horizontal-Vertical (HV) partition like the quad tree produces a tree-structured partition of the image. Instead of recursively splitting quadrants, however, each image block is split into two by a horizontal or vertical line. Splitting positions may be constructed so that boundaries tend to fall along prominent edges Fig 1(c).

Irregular regions partitioning: A tiling of the image by right-angled irregular-shaped ranges may be constructed by a variety of merging strategies on an initial fixed square block or quad tree partition Fig 1(d).

Polygonal blocks partitioning: Polygonal partitions have been constructed by recursive subdivision of an initial coarse grid by the insertion of line segments at various angles, as well as by merging triangles, in a Delaunay triangulation, to form quadrilaterals Fig 2(a),(b),(c),(d).

Implementation of fixed fractal image compression

The work is divided into two parts 1) The fixed fractal image compressor 2) fractal image compressor based on quantized quad tree partitioning. Experimental result is the difference between the compression ratio, encoding time and PSNR between the proposed architecture i.e. quad tree and basic method (fixed fractal image compression).

2. Materials and Methods

Fractal Image coding involves partitioning of the image into non overlapping blocks called Range Blocks and Domain Blocks. Size of domain Blocks chosen were twice the size of the range Blocks.

Steps

1. Share the images of non overlapped range blocks.
2. Cover the image with sequence of the domain blocks which are overlapped and are of different sizes.
3. For every range block find the domain block and corresponding transformation which in the best way covers the range block. The Affine transformation is used.
4. Adjust parameter of transformation, such as contrast and brightness, for the best accordance.
5. If there is not enough exact accordance, we subdivide range blocks to smaller range blocks. We continue this process until we achieve comprehensible accordance.

2.1 Implementation of fractal image compression by using quantized quad tree partitioning technique

Following shows the block diagram of quad tree shown in Fig. 3 based fractal image compression technique. These are classified in two i.e. compression and decompression. Compression consists of domain block, quad tree partitioning, domain range classification & matching, transform, encoder. Decompression consists of decoder, quad tree demodulator module, and inverse transform.
2.2 Setting Domain Data

The domain data consist of domain size, location, number of domains and the classes of domains. Classification of domains is done using a scheme, which takes into account the brightness values of the domains.

2.3 Quad Tree Partitioning

In the quad-tree partitioning technique, the partitioning of two dimensional spaces like the image is executed repeatedly by subdividing the image into four quadrants or regions. This process is made to continue based on some threshold criteria. The shape of the partitioning regions may be of any arbitrary shapes. Quad-tree partitioning scheme is found analogous to a structure of a tree. The root of the tree is the initial image and each node of the tree is a square portion of the image which contains four sub nodes corresponding to the four quadrants of the square image.

Table 1: Applying Transformation

| ID | Transformation matrix | Meaning          |
|----|-----------------------|------------------|
| 0  | \[1 0\] \[0 1\]      | rotation 0°      |
| 1  | \[0 1\] \[-1 0\]     | rotation 90°     |
| 2  | \[-1 0\] \[0 -1\]    | rotation 180°    |
| 3  | \[0 -1\] \[1 0\]     | rotation 270°    |
| 4  | \[1 0\] \[0 -1\]     | X reflection     |
| 5  | \[-1 0\] \[0 1\]     | Y reflection     |
| 6  | \[0 1\] \[1 0\]      | Y=X reflection   |
| 7  | \[0 -1\] \[-1 0\]    | Y=X reflection   |

Eight possible orientations to each domain are applied to map the domain to the range (four rotation and flip with four rotations). The Domain Block is affine transformed and is compared with each of the Range block and a closest approximation of the domain block with that of the Range block is found as in equation based on the Root mean squared error calculated between the Range Blocks and the transformed domain Blocks. The square image is split up into four equal-sized sub squares recursively until the squares are small enough to be covered within some specified threshold value. In the implemented quad-tree encoding technique a square image size of 256*256 pixels is considered. The domain sizes are chosen to be twice the range sizes. The pixels in the domain are averaged in groups of four so that the domains are reduced to the size of the range for comparison purpose.

2.4 Domain, Range classification and matching

Range blocks are classified using the same classification scheme used for the domains, and then a search is carried out to determine the best domain-range match. The match operation is restricted to only those domains and ranges belonging to the same class, and the error between them (threshold) is computed. If this error value is less than the specified threshold, the transformation is calculated; else, the image is further partitioned. This process is repeated for the whole image. Thus, classification helps in reducing the search time and computational overhead involved during the matching operation.

2.5 Apply transform

An affine transformation can skew, stretch, rotate, scale, shear and translate an input image. The feature of these transformations that run in a loop back mode is that for a given initial image. Each image is formed from transformed copies of itself and hence it must have detail at every scale. These images are fractals. Store these images as collections of transformations lead to image compression.
2.6 Encoding: Finding $c_i$ and $b_i$ for each range and domain block pair

The coefficients $c_i$ and $b_i$ (contrast and brightness settings) are calculated for each $(R_i, D_i)$ by minimizing the sum of squared errors

$$E(c_i, b_i) = \sum_{k=1}^{N} (c_i d_k + b_i - r_k)^2$$

(2.1)

$$c_i = \frac{(\sum_{k=1}^{N} r_k d_k + \sum_{k=1}^{N} r_k) - (\sum_{k=1}^{N} d_k)^2}{N \sum_{k=1}^{N} d_k^2}$$

(2.2)

$$b_i = \frac{\sum_{k=1}^{N} r_k - c_i \sum_{k=1}^{N} d_k}{N}$$

(2.3)

Where $d_k, r_k$ are the pixel values in the domain and range blocks respectively.

N=number of pixels in range/domain block

The above coefficients are quantized and stored in a compressed file.

If the mean square error between the range and domain block exceeds a minimum mean square error specified by a parameter called $q$ (quality), the range block is further split into 4 more blocks and the best affine map is again computed for each block. This process is repeated recursively until range block size reduces to 4x4 pixels.

Step 1: First of all, the original image is segmented to determine $D_i$.

Step 2: Directly divide the $D_i$ into blocks, and find the right $R_i$ for each block.

Step 3: Exchange and sort the $D_i$ sub-blocks according to the block mean, and note the information of exchange. The block mean and exchange information are added as part of its code.

Step 4: Exchange $R_i$ according to the same method as step 3, and the exchange information is added as part of the code.

Step 5: Search the similar block of $D_i$ in $R_i$, if found, continue next, if not found, look for the most similar block $R_i$ and calculate the correctional value to construct the fictitious $R_i$, add the correctional value as part of the code.

Step 6: Repeat step 4 and 5 until it reaches the set of conditions.

2.7 Decoding

Step 1: Read the encoding file by sequences. First, read the block mean of $D_i$ and its exchange information. According to the exchange information, we can get the sub-block mean with the original order.

Step 2: Then read the correctional value and exchange information of $R_i$ to calculate the mean of $R_i$.

Step 3: Repeat step 2 until reaching the smallest block size.
2.8 Calculate objective evaluation parameter

MSE and PSNR, encoding time, decoding time of decompressed image is measured and noted. Decoding of an image is reverse of the encoding process. The range block is matched with the domain blocks and hence it is an iterative quad tree partitioning scheme to determine all the ranges in the image. $x$ is the original image and $z$ is the decompressed image. $B$ is the maximum pixel value in the image.

$$MSE = \frac{\sqrt{\sum (x - z)^2}}{mn}$$

$$PSNR = 10 \log_{10} \left( \frac{B^2}{MSE} \right)$$ (2.4)

3. Results and Discussions

FIC algorithm is designed using MATLAB simulation and the result of fractal image compression and Quad tree based fractal image compression are compared for various images.

| Table 3 Comparison of images |
|-------------------------------|
| **Input Image** | **Decompressed FIC** | **Decompressed FIC+QT** |
| ![Image 1] | ![Image 2] | ![Image 3] |
| ![Image 4] | ![Image 5] | ![Image 6] |
| ![Image 7] | ![Image 8] | ![Image 9] |
| ![Image 10] | ![Image 11] | ![Image 12] |

$$CR = \frac{\text{Number of bits in original image}}{\text{Number of bits in compressed image}}$$ (2.6)
The higher compression ratios for fractal images can be explained in terms of the occurrence of fractals due to repetition features at regular intervals. Quantized quad-tree decomposition and coding are applied on images of different types and sizes. The results showed significant improvement in compression ratios, PSNR and better encoding time. These results are derived from the fractal compression with range block and iterations technique, the results are shown in Table above.

4. Conclusion

In this paper, the performance of different types of FIC techniques employed in the present work, with regard to encoding time is summarized in Table. The following observations are drawn from the results.

1) The Quantized Quad tree based FIC performs better than the fixed partitioning FIC where the encoding time with quad tree based FIC is reduced from 5.537 sec to 1.9197 sec.
2) The quad tree based FIC exhibits a superior performance with regard to encoding time.
3) Also from the result conclude that the compression ratio of quad tree based FIC is higher than the fixed fractal image compression.
4) For example, as in lena image shown the compression ratio is increases in quad tree based fractal image i.e. 90.09%

References

[1] Brendt Wohlbery, G.D Jagor “A Review of the fractal image compression”, IEEE Transactions on Image Processing, 10.1109/83.806618 February 1999

[2] S. Padamshree, Rohini Nagapadma “Different Approaches for Implementation of fractal image compression of medical images” 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)), 978-1-5090-4697-3/16, IEEE 2016

[3] N. A. Koli, M. S. Ali "A survey on Fractal Image Compression key issue", Information Technology journal, Asian network for scientific Information, ISSN 1812-5638, 1085-1095, 2008

[4] Bolong Liu, Ying Yan, ”An improved Fractal Image coding based on Quad-tree”. 3rd International Congress on Image and Signal Processing (CISP2010), 978-1-4244-6516-3/10/ IEEE-2010

[5] Rasha Adel Ibrahim, Sherin M. Yousef, Saleh Mesbin Elkaffias “An Enhanced fractal image compression integrating quantized Quad tree and entropy encoding”, 11th International Conference on Innovations in Information Technology (IIT), 978-1-4673-8511-4/15 IEEE-2015

[6] Satish Lokkoju, Dinesh Reddl, “Fast Coding Unit Partition Search”, Samsung India Software Operations Pvt Ltd Bangalore, India 978-1-4673-5604-6/112, IEEE 2012

[7] S. Padmashree, Dr. Rohini Nagapadma, “Suitability Analysis of Fractal Compression Technique for Medical Image”, Ninth Int Conf. on Advanced Computing (ICoAC) IEEE 978-1-5386-4349-2017

[8] Aihua Zhang, Pei Yang, ”An Improved Algorithm for Fractal Image Encoding based on Relative Error”, 5th International Congress on Image and Signal Processing, IEEE 978-1-4673-0964-2012

[9] A. Martinez-Ramirez, A. Diu-Sanchez, M. Linares-Aranda, J. Vega-Pineda “Simple and Fast Fractal Image Compression for VLSI Circuits”, Proceedings of the 3rd international Symposium on Image and Signal Processing and Analysis (2003)

[10] Vijayshri Chaursias, Ajay Somkuwar, “Speed up Technique for Fractal Image Compression,” International Conference on Digital Image Processing IEEE-978-0-7695-3565-2009

[11] R. Starosolski, “A new approach for improvement of fractal image encoding”, International Journal on Computer Science and Engineering Vol. 02, No. 04, 1387-1394- 2010

Table 4: Output comparison of basic fractal image compression and Quantized Quad tree based fractal image compression

| Input Image     | Basic fractal image compression | Fractal image compression using Quad tree partitioning |
|-----------------|--------------------------------|-----------------------------------------------------|
|                 | ET (Encoding Time) | CR (Compression Ratio) | PSNR     | ET (Encoding Time) | CR (Compression Ratio) | PSNR     |
| Medical(USG)    | 6.3981            | 74.7581                | 67.3029 db| 5.1800            | 76.1984                | 66.1211db|
| Satellite image | 5.1346            | 70.2728                | 77.2108 db| 1.9314            | 90.1016                | 76.7052db|
| Natural Image   | 4.9126            | 70.0745                | 76.8017 db| 1.9365            | 90.5584                | 77.3193db|
| Lena image      | 5.5357            | 70.1950                | 66.4819 db| 1.9197            | 90.0970                | 66.8805db|