

Application of Quadtree Partitioning in Fractal Image Compression using Error Based Approach

Roshni S. Khedgaonkar

M.Tech Student Department of Computer Science and Engineering, YCCE, Nagpur, 441110 India

Shailesh D. Kamble

Department of Computer Technology YCCE, Nagpur, 441110 India

Abstract—Fractal image Compression is a lossy compression technique that has been developed in the early 1990s. Fractals can be an effective approach for several applications other than image coding and transmission: database indexing, texture mapping, and even pattern recognition problems such as writer authentication. Fractal image compression has received much attention from the research community because of some desirable properties like resolution independence, fast decoding, and very competitive rate-distortion curves. Despite the advances made, the long computing times in the encoding phase still remain the main drawback of this technique. In Fractal image compression we obtained self similarity in image. The basic scheme of fractal image compression is to partition a given image into non overlapping blocks of size $r \times r$, called range blocks and form a domain pool containing all of possible overlapped blocks of size $2r \times 2r$, called domain blocks associated with 8 isometries from reflections and rotations. We have proposed a method to reduce the complexity of the image coding phase by classifying the blocks according to an approximation error measure. It perform range\domain comparisons with respect to a preset block, in this instead of binary tree we have used B tree. The main application of B tree is the organization of huge collection of records into a file structure i.e., insertion, deletion and modification operation can be carried out perfectly and efficiently. Therefore, it is possible to reduce drastically the amount of operations needed to encode each range. For partitioning an original image into range blocks we have used quadtree partitioning method.

Keywords- Fractal image compression, classification, feature vector, approximation error, quadtree partitioning.

I. INTRODUCTION

An image is essentially a 2-D signal processed by the human visual system. The signals representing images are usually in analog form. However, for processing, storage and transmission by computer applications, they are converted from analog to digital form. A digital image is basically a 2-Dimensional array of pixels [1]. Images form application in many areas such as remote sensing, biochemical and video conferencing. The objective of image compression is to reduce

irregular and redundancy of an image data in order to be able to store and transmit data in an efficient form. The reduction in file size allow more images to be stored in a given amount of disk or memory space, It also reduces the time required for images to be sent over the internet or download from web pages.

There are various methods of compressing images and every method has three basic steps Transformation ,quantization and encoding.

A. Fractal Image Compression

Fractal theories are totally different from the others. M. Barnsley introduced the fundamental principle of fractal image compression in 1988 [2]. Fractal image compression is also called as fractal image encoding because compressed image is represented by contractive transforms and mathematical functions required for reconstruction of original image[16]. Contractive transform ensures that, the distance between any two points on transformed image will be less then the distance of same points on the original image [3]. These transforms are composed of the union of a number of affine mappings on the entire image, known as iterated function system (IFS).

Fractal image compression is based on contractive affine transformation is of the form:

$$w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} \quad (1)$$

is sufficient to generate interesting transformations called affine transformations of the plane. Each can skew, stretch, rotate, scale and translate an input image[13].

1) **Contractive Transformations:** A transformation w is said to be contractive if for any two points P_1, P_2 , the distance $d(w(P_1), w(P_2)) < s d(P_1, P_2)$ for some $s < 1$, where $d =$ distance. This formula says the application of a contractive map always brings points closer together.

2) **The Contractive Mapping Fixed Point Theorem:** This theorem says something that is intuitively obvious: if a transformation is contractive then applied repeatedly starting with any initial point, we converge to a unique fixed point. If X is a complete metric space and $W: X \rightarrow X$ is contractive, then W has a unique fixed point $|W|$.

3) **Fractal Encoding** : Fractal image Compression is a lossy compression technique that has been developed in the early 1990s. fractal image compression is based on partition a given image into non overlapping blocks of size rxr , called range blocks and form a domain pool containing all of possible overlapped blocks of size $2rx2r$, called domain blocks associated with 8 isometries from reflections and rotations [4]. For each range block, it exhaustively searches, the domain pool, for a best-matched domain block with the minimum rms error after applying a contractive affine transform to the domain block. The problems that occur in fractal encoding are the computational demands and the existence of best range-domain matches [5]. However searching the domain pool is highly computationally intensive. For an nxn image, the number of range blocks are (nxn/rxr) and the number of domain blocks are $(n-2r+1) \times (n-2r+1)$. The computation of best match between a range block and a domain block is $O(r^2)$. If r is constant, the computation complexity of entire search is $O(n^4)$ [15].

A.E. Jacquin gave first publication on Fractal image compression with partitioned IFS (PIFS) in 1990 [6], [7], [8]. In Jacquin's method the image is partitioned in sub images called as 'Range blocks' and PIFS are applied on sub-images, rather than the entire image [17]. Locating the range blocks on their respective position in image itself forms the entire image. Temporary images used to form range blocks are known as domain blocks.

The overall process of fractal image encoding includes four levels of decision making the flow is shown in the Figure 1. In any Fractal compression system the first decision is to choose the type of image partition for the design of range blocks. A wide variety of partitions have been investigated but we have used quad tree partitioning[18],[20],[21].Our partition method is based on a split/merge approach as introduced by Horowitz and Pavlidis in [19]. Domain pool is the second level of decision. This choice depends on the type of partition scheme used. In third step we form a class of transform applied to the domain block and in the final step searching most suitable domain block for the formation of particular range block. This step of fractal image compression is computationally very expensive, because it requires a large number of domain range comparisons. The attempts to improve encoding speed are addressed as speed-up techniques and focused on two areas such as domain classification based methods and feature vector based methods.

At present, fractal coding cannot compete with other techniques (wavelets, etc.) if compression per se is involved. However, most of the interest in fractal coding is due to its side applications in fields such as image database indexing [9] and face recognition [10]. So, our main aim is to short the encoding time, speed-up the coding phase of fractal image compression and to reduce the computational cost of

exhaustive search while still preserving a good image quality, we will apply classification method for fractal image compression called DRDC based on an approximation error

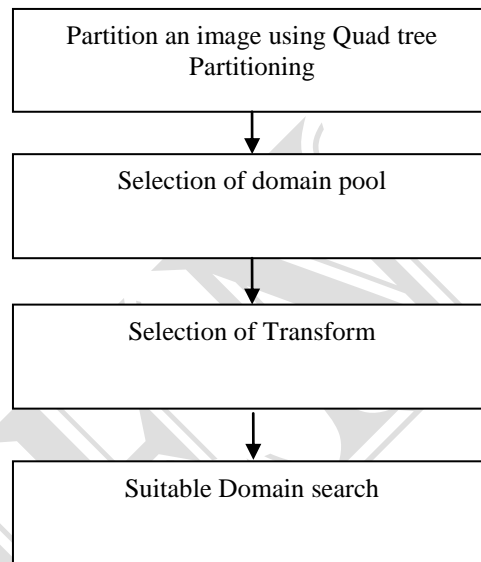


Figure 1. Decision making levels in fractal encoding[14].

measure. Which will used to perform range/domain comparisons with respect to a preset block, a preset block is used as a temporary replacement.

The coding process of DRDC is divided in two phases. In the first phase, where the domain codebook is created, all the domains are extracted from the image, then each of them is compared with the preset block by solving a mean square root problem. The *preset block/domain* approximation error is computed and stored in a B tree data structure. The main application of B tree is the organization of huge collection of records into a file structure. The organization should be in such a way that any record in it can be searched very efficiently i.e., insertion, deletion and modification operation can be carried out perfectly and efficiently. In this paper, we are extending the work of R. Distasi in [11] they have used domain tree for insertion, selection and nearest neighbor search operation where as we are using B tree for performing the same operation because B tree is an efficient technique, it guarantees at least 50% storage utilization and it gives good performance as compare to Domain tree as a binary tree.

In the second phase, the ranges have to be encoded; each one of them is compared with the preset block, thus obtaining the preset block/range approximation error, in the same way as we did for domains. Using this data, we find the domains that are likely to encode the current range with the best accuracy. This criterion is formally justified by the lemma in Section II in [11] which proves that a generic range block is accurately coded by domains with equal or similar approximation error. In this way, for each range we have to perform a much smaller

number of range/domain comparisons, and the time spent for coding is significantly reduced.

The paper is organized as follows: In Section II, we present the DRDC strategy and quadtree partitioning method, while Section III illustrates the experimental setup and results. Finally, Section IV gives conclusion.

II. PROPOSED APPROACH

In case of fractal coding, we have to reduce the search time, because full exhaustive search is prohibitively costly. Many different solutions have been proposed for this problem. For instance, modifying the partitioning process or providing new classification criteria or heuristic methods for the range/domain matching problem. All these approaches can be grouped in two classes[12]:

- Classification methods and
- Feature vectors.

The fundamental idea of this algorithm consists in deferring the comparisons between ranges and domains, utilizing a preset block \bar{d} , as a temporary replacement with which ranges and domains are compared.

After computing the preset block as the average block.

$$\bar{d} = \frac{1}{|R|} \sum_{r \in R} r$$

In the first phase, all domains are compared with \bar{d} , and the approximation error is computed and stored in a KD-tree or Domain tree, as shown in Figure 2. Domain tree is a binary tree, in which every node consists of two references, one to the left child and one to the right. Insertion, search and nearest-neighbor search operations are defined on it. The domain tree is structured so that every bit of the feature vector corresponds to one tree level, from the most significant in the root to the least meaningful one in the leaves.

In the second phase, all ranges to be encoded are compared with \bar{d} and the approximation error is computed but not stored: It is used as a feature vector and serves as the search key for locating the best fitting domain for the given range. This is illustrated in Figure3.

To see how it is done, consider a function $\psi : \mathcal{R}^n \times \mathcal{R}^n \rightarrow \mathcal{R}^n$, where $n = |r| = |d|$, and d represents the domain after contraction, as defined in[11].

In this paper, we have captured the colour image, convert it into gray scale image and partition that image using Quadtree partitioning method. The basic process of quadtree partitioning is shown in Figure 4. In quadtree the original image blocks are split into four different quadrant and the concept of threshold is used in which splits a block if the maximum value of the block elements minus the minimum value of the block elements is greater than THRESHOLD as

given in Algorithm1. This process is repeated iteratively until each block meets the criterion. The result may have blocks of several different sizes. THRESHOLD is specified as a value between 0 and 1.

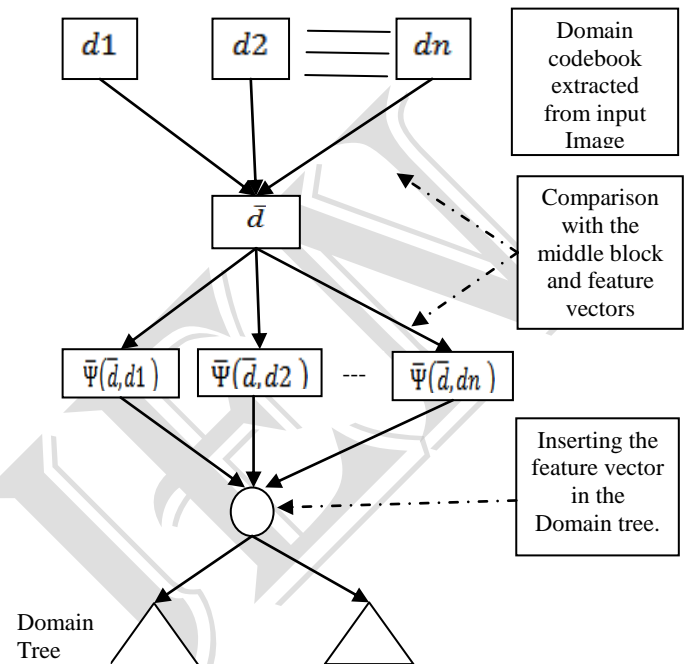


Figure2.First Phase of DRDC Process

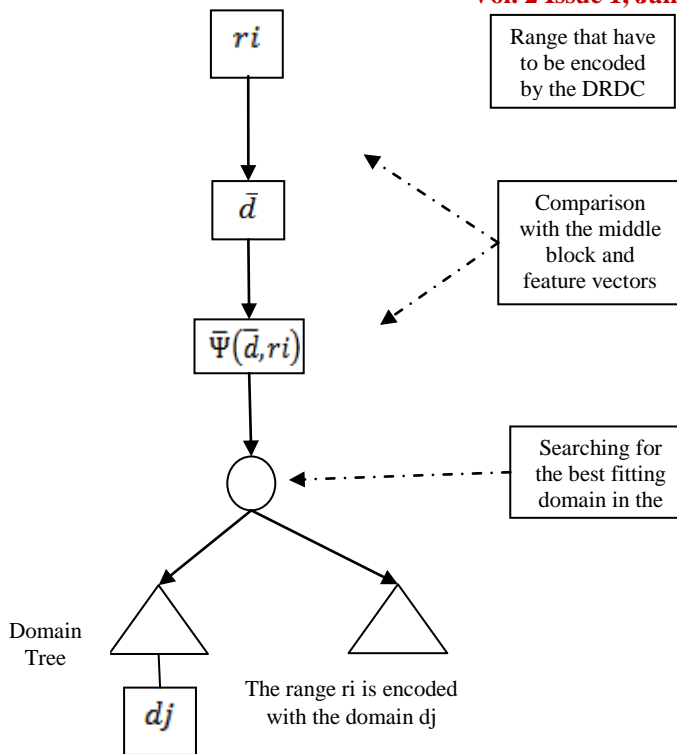


Figure3. Second Phase of DRDC Process

Resizing an image in standard size $n \times n$;
 if $(\max(B_i) - \min(B_i)) > \theta$ then
 Split B into $B_i, i=1, \dots, 4$;
 for $i=1, \dots, 4$ do
 $B_i = \text{Split}(B_i)$
 end for
 end if
 return Block B

III. EXPERIMENTAL SETUP AND RESULTS

In order to reduce the encoding time, we have tested the first module of our approach on the Lena colour image of size 225×225 and then converted into the greyscale image of standard size namely- 512×512 . We have partition the original image in 16×16 , 8×8 , and 4×4 pixel blocks using a quadtree partitioning with THRESHOLD value as 0.27. The quad tree partitioning of lena image is shown in Figure5. Experimentation has been performed using MATLAB 7.9 development tool on WINDOWS platform. Hardware requirement of proposed approach is general minimum configuration, Pentium III Processor and 256MB RAM.

IV. CONCLUSIONS

In this paper, we have adopted classification method for fractal image compression called DRDC. It is based on the

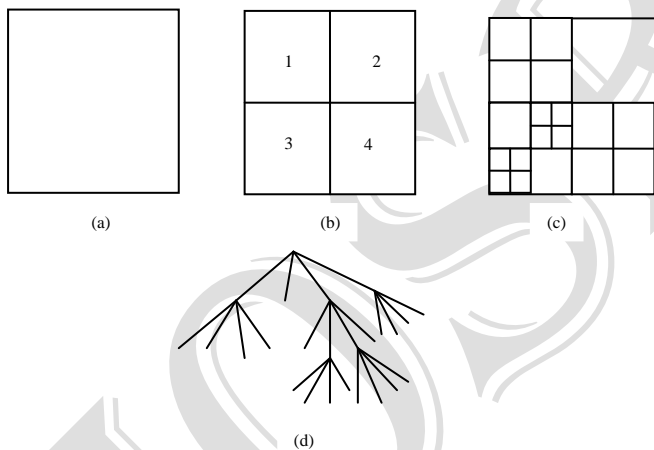


Figure4. Quad tree partitioning process: (a) Original Block (b) Original block partitioned into four different quadrant (c) Division of a block into sub blocks (d) Quad tree or decision tree.



(a)



Algorithm 1: Split(B)

Input: Colour image I of any size, THRESHOLD θ , Block B
 Output: Quadtree partitioning of original image.
 Original image, $\theta = \{0,1\}$;
 $B = \text{value}(I)$
 Convert I into grayscale image;

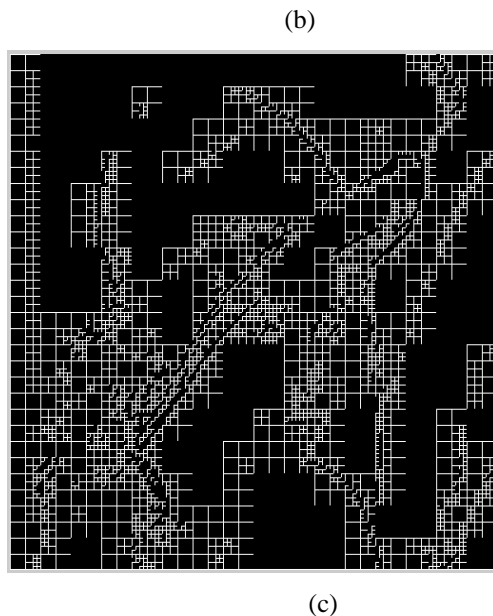


Figure 5. Lena image using quadtree partitioning method: (a) Original Lena image of size 225x225 (b) Gray Scale Lena image of size 512x512 (c) Quadtree partitioning of Lena image of size 512x512 with 0.27 Threshold value.

approximation error, which is computed deferring range/domain comparisons with respect to a preset block. We have implemented the partitioning of an original image using Quadtree partitioning. We have taken coloured input image convert it into gray scale image, divides a square image into four equal-sized square blocks, and then tests each block with respect to threshold value to see if it meets some criterion of homogeneity. If a block meets the criterion, it is not divided any further otherwise, it is subdivided again into four blocks. This process is repeated iteratively until each block meets the criterion. The result may have blocks of several different sizes.

In future work, We will implement a B tree data structure instead of Domain Tree as binary tree, for feature vectors management. It will show a significant reduction of the number of operations required to generate a good fractal code for a given image, and consequently a performance improvement of the coding process.

REFERENCES

- [1] Sonal, D.Kumar, "A Study of Various Image Compression Technique".
- [2] M.F. Barnsley and A.E. Sloan, "A better way to compress image," *BYTE Magazine.*, January 1988.
- [3] Miroslav Galabov, "Fractal Image Compression," International Conference on Computer System and Technology-CompSys'2003.
- [4] C. S. Tong and W. Man, "Adaptive Approximation Nearest Neighbor Search for Fractal Image Compression," *IEEE*

- Transactions on Image Processing*, vol. 11, No. 6, pp. 605-615, 2002.
- [5] B. Wohlberg and Gerhard de Jager, "A review of the Fractal Image Compression Literature," *IEEE Transactions on Image Processing*, vol. 8, No. 12, pp. 1716-1729, Dec. 1999
- [6] A.E. Jacquin, "A novel fractal block-coding technique for digital Images", *ICASSP International Conference on Acoustics, Speech and Signal Processing*, (1990).
- [7] A.E. Jaquin, "Image coding based on a fractal theory of iterated contractive image transformation", *IEEE Trans. On Image Processing*, **1**(1): (1992).
- [8] A.E Jaquin, "Fractal image coding: A review", *Proceeding of tile IEEE*, **81**(10): (1993)
- [9] R. Distasi, M. Nappi, and M. Tucci, "FIRE: Fractal Indexing with Robust Extensions for image databases," *IEEE Trans. Image Process.*, vol. 12, no. 3, pp. 373-384, Mar. 2003.
- [10] H. E. Komleh, V. Chandran, and S. Sridharan, "Face recognition using fractal," in *Proc. Int. Conf. Image Processing*, vol. 3, Oct. 2001, pp. 58-61.
- [11] R. Distasi, M. Nappi and D. Riccio, "A range/domain approximation error- based approach for fractal image compression", *IEEE Trans. Image processing*, **15**(1): 89-97(2006).
- [12] D. Saupe, R. Hamzaoui, "Complexity Reduction Methods for Fractal Image Compression," *I.M.A. Conference Proceedings on Image Processing: Mathematical Methods and Applications*, September 1994, J.M. Bleclledge(ed.), Oxford University Press, 1995, pp. 211-229
- [13] Y. Fisher, *Fractal Image Compression: Theory and Application*. New York: Springer-Verlag, (1994).
- [14] V. Chaurasia and A. Somkuwar; "Review of a novel technique: fractal image compression", *International Journal on Emerging Technologies* **1**(1): 53-56(2010)
- [15] V. Vaddella, R. Basu, "Fast Fractal Compression Based on Domain-Range Pixel Value Difference," *Global Gernal of Computer Science and Technology*, vol. 10, pp. 67-72, June 2010.
- [16] S. A. Curtis and C. E. Martin, "Functional Fractal Image Compression," International Conference on Computer Systems and Technologies, 2002
- [17] Jacquin, "A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding". *Doctoral Thesis, Georgia Institute of Technology*, 1989.
- [18] Tilo Ochotta and Dietmar Saupe, "Edge-Based Partition Coding for Fractal Image Compression," *Published by the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia*, Volume 29, Number 2C, Dec 2004.
- [19] S. L. Horowitz, T. Pavlidis, "Picture segmentation by a tree traversal", *Journal of the ACM*, **23** (2) (1976), pp. 368-388.
- [20] Y. C. Chang, B. K. Shyu, J. S. Wang, "Region-based fractal image compression with quadtree segmentation", *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, 4 (1997), pp. 3125-3128.
- [21] Y. C. Chang, B. K. Shyu, J. S. Wang, "Region-based fractal compression for still image", *Proceedings 8-th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media*, 2000.