Semiconductor Optoelectronics, *Vol. 42 No. 2 (2023), 1645-1663*
https://bdtgd.cn/

Semiconductor Optoelectronics
ISSN: 1001-5868

STUDY ON ENCODING TIME REDUCTION USING GRAPH BASED IMAGE SEGMENTATION TOWARDS FRACTAL IMAGE COMPRESSION

Dr. Shimal Das[1], Dr. Jhunu Debbarma[2] and Dr. Dibyendu Ghoshal[3]
[1,2]Department of Computer Science and Engineering, Tripura Institute of Technology, Narsingarh, Agartala,India
[3]Department of Electronics and Communication Engineering, National Institute of Technology, Agartala, India
*Corresponding author E-Mail: shimalcs.tit@gmail.com

Abstract
Fractal geometry based image compression is a very promising technique for image compression. Due to long encoding time and high computational complexity fractal image compression has not been widely used. 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. Many different solutions have been proposed for this problem, but there is not yet a standard for fractal coding. Based on adaptive threshold quad tree fractal compression approach, we consider image's semantic characteristic, and apply graph-based image segmentation to fractal image compression, separating the initial image into many logic areas, then encoding each area with fractal image compression method. According to the problem of long encoding time which exists in the typical fractal compression approaches, we have proposed a fast image fractal compression approach based on the combination of adaptive quad tree compression approach and graph-based image segmentation algorithm.
Keywords: Fractal coding, Affine Transformation, Equilateral Triangle Segmentation, Graph based image segmentation, Self-Similarity, Iterated Function System (IFS), Local Iterated Function System (LIFS), Partitioned Iterated Function System (PIFS).

I. INTRODUCTION

Fractal geometry based image compression has become very important in high definition television system common computer graphics and animation, inter networking. The main attraction of fractal geometry is to its capability to handle any object of irregular geometric shape. Main disadvantage of fractal based processing are computational complexity and the larger processing line [1].There are various research studies are found in journals and conference presentation all have tried to put sum effort to make some improvement in any of the related prosperities of the image .The basic prosperities of iterative functional system are to be done and for our purpose partition by at the image flow must be chosen in a nice way. Graph involve method have already been repeated for image segmentation and subsequently fractal compression technique is applied on them. The choice of segmented area places a vital rule for this process.

Fractal coding is based on the concept of fractional geometry which is used to describe irregular and fragmented objects or patterns. There are various objects like cloud, fire flame, snow fall, mountain, waves, and trees etc which are difficult to describe or deal with the help of other geometry.

Fractal coding based image compression method is originated from Barnsley's research for IFS system [1],[3] and the image fractal block coding proposed by Jacquin [2]. In 1988, In 1988, Barnsley applied fractal image compression based on IFS system to computer graphics, and compressed the aerial image, which made him get a compressed ratio 1000:1, but the approach requires manual intervention. Subsequently, Jacquin proposed a new fractal image compression method based on image block, and the method can conduct automatically without manual intervention. Therefore, Jacquin's method has become a typical representation for this research direction; fractal image compression has become practical since then. Currently, fractal image compression has got extensive attention by the research community because of its novel ideas, high compression ratio and resolution independent characteristics [4], and it is recognized as one of the most three promising next generation image compression technology [5], besides, fractal compression can also be applied to audio and video compression [6]-[8].

In the present work graph based image segmentation has been followed to get isosceles triangle shaped segmentation. An adaptive threshold based concepts was used for region based segmentation and for fractal iteration system. Based on adaptive threshold quad tree fractal compression approach, we consider image's semantic characteristic, and apply graph-based image segmentation to fractal image compression, separating the initial image into many logic areas, then encoding each area with fractal image compression method. According to the problem of long encoding time which exists in the typical fractal compression approaches, we have proposed a fast image fractal compression approach based on the combination of adaptive quad tree compression approach and graph-based image segmentation algorithm for higher compression ratio with better PSNR.

II. MATHEMATICAL BACKGROUND

II. A. 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 1 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 1: Self Similarity in Lena

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.

II. B. Iterated Function System and Attractor

According to great mathematician Cauchy "A convergent sequence converges to a unique value."A sequence defined by f (n) = n/2. It is obvious that these sequences will converge to zero whatever may be the initial value of n. On the same lines it can be told 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. 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 2 show the contracting transformations by iterated function system.

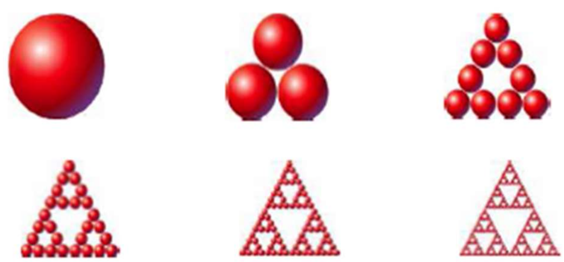


Figure 2: Contracting transformations by Iterated Function System.

An Iterated Function System (IFS) is a collection of contractive mappings .Let IFS be denoted by W and the collection of transforms by

$w_1, w_2, w_3 \ldots\ldots, w_n$. (1)

$W(X) = w_1(X) \cup w_2(X) \cup w_3(X) \ldots\ldots \cup w_n(X)$ (2)

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.

II. C. Partitioned Iterated Function System (PIFS)

Fractal image compression uses a special type of IFS called a partitioned iterated function system (PIFS). In figure 3 shows the Partitioned Iterated Function System. A PIFS consists of a complete metric space X ,a collection of sub domains Di $\subset$ X, i = 1,2,.....n, and a collection of contractive mappings

$$\tilde{w} \; : \; D_i \longrightarrow X, \; i = 1,2,.....n. \qquad (3)$$

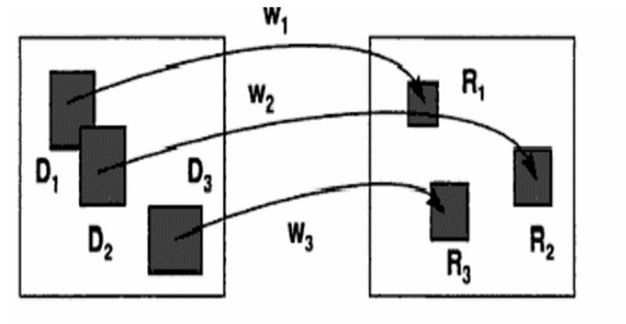


Figure 3: A Partitioned Iterated Function System.

II. D. Contraction Mapping

A transformation f: X ⟶X on a metric space (X, d) is called a contraction mapping if there is a constant s, $0 \le s \le 1$, such that

$$d(f(x_i),f(x_2)) \le s \, d(x_1,x_2) \; ….. \qquad (4)$$

for all x1,x2 X. The constant s is called the contractivity factor for f. Figure 4 shows an example of a contraction mapping f acting on a set of points in R2.

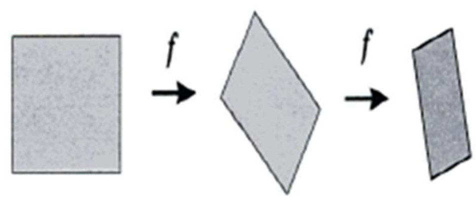


Figure 4: A contraction mapping f acting on a set of points in R2.

The above figure depicts a transformation that is applied more than once. That is, once f(x) is computed for a point x, the value f(f(x)) is computed by applying f to the result. The transformations obtained by applying f over and over again in this way are called iterates off.

Let f: X⟶ X be a contractive mapping on a complete metric space(X, d).Then f possesses exactly one fixed point

X_f € X, and for any x € X, the sequence $\{f^{on}(x) : n = 1,2,,.....\}$ converges to x_f, that is,

$$\lim_{n\to\infty} f^{on}(X) = x_f \text{, for all } x \in X. \qquad (5)$$

If W is a contraction on F(image space),then according to the theorem W has a unique fixed point f_w € F satisfying

$$W(f_w) = f_w \qquad (6)$$

Iteratively applying W to any starting image f_0 will recover the fixed point f_w.

II. E. The Collage Theorem

Barnsley (1993) [1] has derived a special form of the contraction mapping theorem applied to IFS's on H(X),h) called the Collage Theorem. For a grayscale image f, we can find a contractive transformation W such that

$$d_2(f, W(f)) \le \varepsilon,$$

then

$d_2(f, f_w) \leq \frac{\varepsilon}{1-\varepsilon}$ (7)

Where s is the contractivity factor of W, f_w is the fixed point and ε be > 0. This means we can start with any image g and iterate W on g to get an image that is close to f.

$W^{on}(g) \longrightarrow f_w \approx f.$ (8)

The Collage Theorem brings us one step closer to fractal image encoding. Decoding consists of iterating W on any starting image g to reform g to recover f_w.

II. F. Affine Transformation

The transformations used by Barnsley for his IFS's are the affine transformations. An affine transformation T: $R^2 \longrightarrow R^2$ is a transformation of the form.

An affine transformation maps a plane to itself. The general form of an Affine Transformation is

$$W \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \ldots\ldots \quad (9)$$

Where a, b, c, d, e, f Affine transformations can accomplish rotation, translation and contraction.

Let $\tilde{w}_i(x, y)$ be an affine transformation on $I^2 \longrightarrow I^2$, then

$$\tilde{w}_i(x, y) = A_i \begin{pmatrix} x \\ y \end{pmatrix} + b_i \quad (10)$$

for some 2 * 2 matrix A_i and 2 * 1 vector b_i

Let $D_i \subset I^2$ be some sub domain of the unit square I^2, and let R_i be the range of $\tilde{w}_i$ operating on D_i, that is $\tilde{w}_i(D_i) = R_i$.

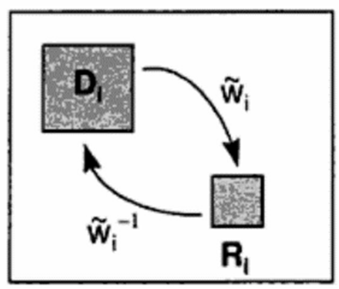


Figure 5: Spatial affine transformation and its inverse.

We can define w_i: F⟶F operating on images $f(x, y)$ by

$W_i(f)(x,y) = s_i f(\tilde{w}_i(x,y)) + o_i$ provided $\tilde{w}_i$ is invertible and $(x, y) € R_i$, s_i controls the contrast and o_i controls the brightness. A spatial affine transformation and its inverse system are shown in figure 5.

II. G. Basic fractal image encoding approach

The basic ideal of fractal image compression is as following: divide initial image into small image blocks with non-overlapping (Rang block, R block for short). For each R block, find an image block (Domain block, D block for short) which is the most similar to current R block under a certain transform, that is, use some image blocks' transformation to splice the initial image, and make the spliced image similar to the original image as much as possible[9].

Suppose the image which to be encoded is 256*256 size with 256 gray shade and R block is a 8*8 size block, so the whole image can be divided into 1024 R blocks, all the R blocks composite R pool. Suppose D block is four times larger as R block, so the number of D block is (256-2*8+1)2=58081, all the D blocks composite D pool. For each R block, find a D block from D pool which is the most similar to it.

The concrete steps are as following:

1) Shrink D block to the size of R block, marked D’ block, and the specific shrinking method is four neighborhoods regional method

2) Transpose, turn D’ block. Specifically, we can choose eight affine transformations which proposed by Jacquin, and the corresponding transformation matrix.

3) Compare each R block with all D’ blocks in D pool, and obtain the most similar D’ block. The similarity can be measured with average variance MSE, if we see each R block and D block as vectors.

4) For each R block, record the corresponding compression affine transformation W. All compression affine transformations constitute the whole image’s fractal code.

The whole above process is described in fig 6.

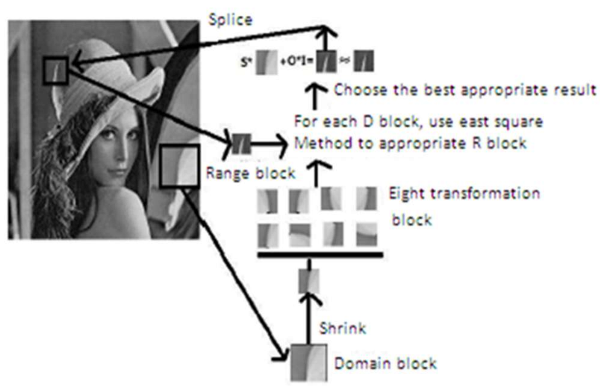


Figure 6: Show the schematic diagram of the fractal encoding process [9]

III. GRAPH-BASED SEGMENTATION

Consider a graph-based approach to segmentation. Let $G = (V, E)$ be an undirected graph with vertices $v_i \in V$, the set of elements to be segmented, and edges $(v_i, v_j) \in E$ corresponding to pairs of neighboring vertices. Each edge $(v_i, v_j) \in E$ has a corresponding weight $w((v_i, v_j))$, which is a non-negative measure of the dissimilarity between neighboring elements v_i and v_j. In the case of image segmentation, the elements in V are pixels and the weight of an edge is some measure of the dissimilarity between the two pixels connected by that edge [10] (e.g., the difference in intensity, color, motion, location or some other local attribute). In Sections 5 and 6 we consider particular edge sets and weight functions for image segmentation. However, the formulation here is independent of these definitions. In the graph-based approach, a segmentation S is a partition of V into components such that each component (or region) $C \in S$ corresponds to a connected component in a graph $G0 = (V, E\,0)$, where $E\,0 \subseteq E$. In other words, any segmentation is induced by a subset of the edges in E. There are different ways to measure the quality of segmentation but in general we want the elements in a component to be similar, and elements in different components to be dissimilar. This means that edges between two vertices in the same component should have relatively low weights, and edges between vertices in different components should have higher weights.

IV. PAIR-WISE REGION COMPARISON PREDICATE

This predicate is based on measuring the dissimilarity between elements along the boundary of the two components relative to a measure of the dissimilarity among neighboring elements within each of the two components.

The resulting predicate compares the inter-component differences to the within component differences and is thereby adaptive with respect to the local characteristics of the data. We define the internal difference of a component C ⊆ V to be the largest weight in the minimum spanning tree of the component, MST(C, E). That is, Int(C) = max e∈MST(C,E) w(e) .

(1) One intuition underlying this measure is that a given component C only remains connected when edges of weight at least Int(C) are considered. We define the difference between two components C1, C2 ⊆ V to be the minimum weight edge connecting the two components. That is, Dif(C1, C2) = min vi∈C1,vj∈C2,(vi,vj)∈E w((vi , vj)) .

(2) If there is no edge connecting C1 and C2 we let Dif(C1, C2) = ∞. This measure of difference could in principle be problematic, because it reflects only the smallest edge weight between two components. In practice we have found that the measure works quite well in spite of this apparent limitation. Moreover, changing the definition to use the median weight, or some other quantile, in order to make it more robust to outliers, makes the problem of finding a good segmentation NP-hard, as discussed in the Appendix. Thus a small change to the segmentation criterion vastly changes the difficulty of the problem.

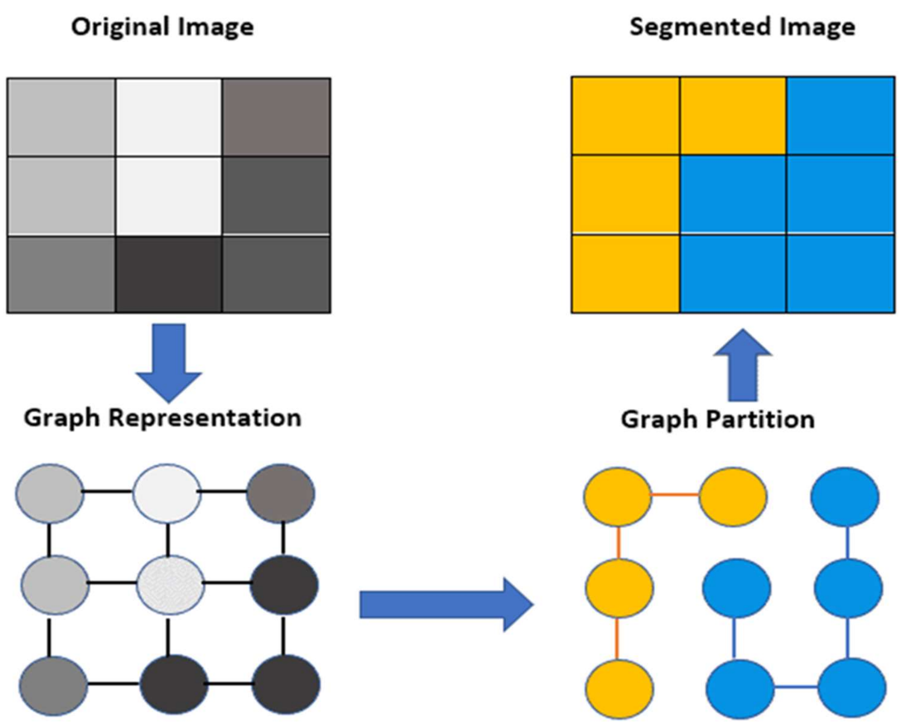


Figure 7: Schematic diagram of image partition based on Graph based segmentation

V. PROPOSED METHOD

In 2004 Pedro F. Felzenszwalb and Daniel P.Huttenlocher [11] proposed a graph-based image segmentation algorithm, and this algorithm overcomes the shortcomings of traditional image segmentation algorithm, i.e., excessive segmentation over image. At the same time, this algorithm can adjust its segmentation scale according to different images, thus, it can achieve better content-based image segmentation. In the fractal encoding process, most blocks which come from different image's contents have different textures, so blocks come from different image contents can't be the best-match block for each other in most cases, but for the blocks come from the same image content, they can easily match each other well. An example is shown in fig 8.

From fig 2, it can be seen that, blocks come from region 1 can easily find his best-match block in region 1, the situation for blocks come from region 3 is the same as that in region 1, but blocks come from region 1 can hardly find his best-match block in region 3.

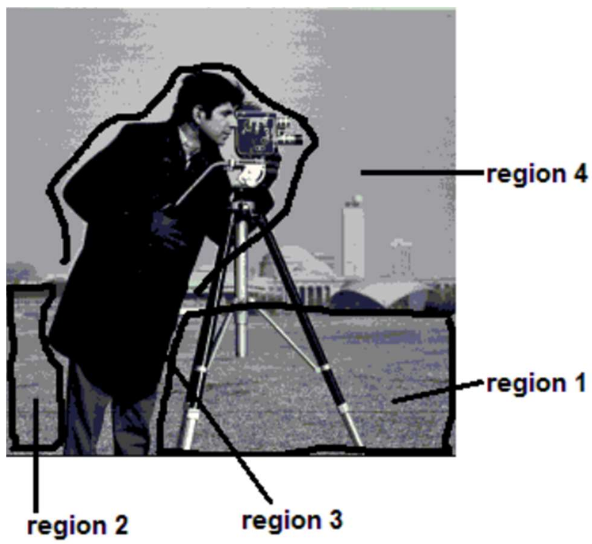


Figure 8: Schematic diagram of image content segmentation

Inspired by this fact, in 2010 Hai Wang [9] combined graph-based image segmentation algorithm and adaptive threshold quad tree fractal compression approach together, and proposed a new fractal image compression approach: used graph-based image segmentation approach to segment image at an appropriate segmentation scale, separate the original image into many different logic areas according to image content, and construct the corresponding search space for each logic area. Then encode each logic area using adaptive threshold quad tree approach. When the encoding for all logic areas in the whole image is finished, the whole image's encoding is completed.

But in 2010, Hai Wang [9] separates the initial image into several different logic areas, it is difficult to encode the irregular shape with fractal compression approach; in this paper Hai Wang adopted the following method:

After the initial image is separated into several different logic areas, they can set a same label for all pixels located in the same logic area, by doing so; they indicate that all those pixels are in the same logic area. When encoding, they can also set a label for each image block, and the label for image block is the equal to the labels of pixels located in the block, if there are several different pixels with different labels in one block, they can set the block's label as anyone of them, all the image blocks with the same label constitute a search space. As shown in fig 8, image block B can be either partitioned into the search space which A located in or partitioned into the search space which C located in.

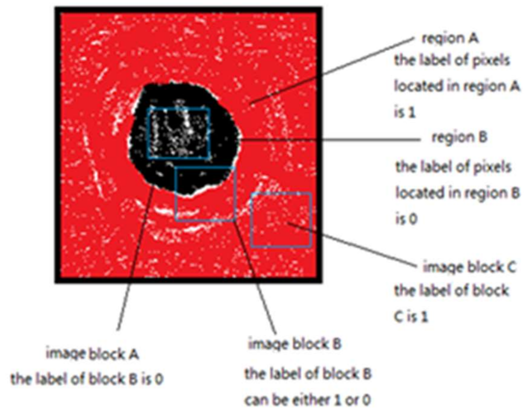


Figure 9: Schematic diagram of setting image block's label [9]

When encoding with adaptive threshold quad tree approach, for each R block, they don't need to find its best-match block within the entire image, but find it within the local search space which R block located in, thus, used local optimum to replace global optimum.

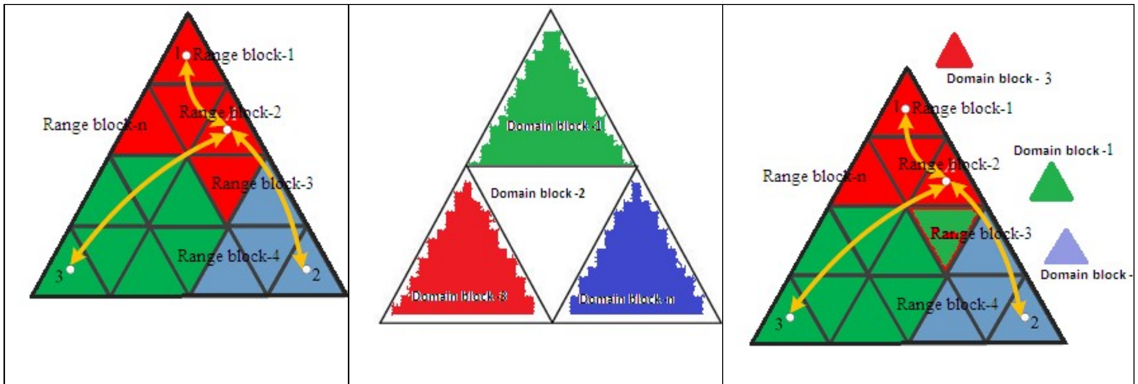
They obtained valid shorter encoding time on the premise of guaranteeing the recovered image quality. Compared with compression approaches they proposed in reference [12]-[14], their improved approach has following advantages: firstly, divide the image blocks into several different search space according to the image content, and overcomes the disadvantage that searching within the global search space in reference [13]; secondly, adaptively segment the image at an appropriate segmentation scale according to the image content, then divide all image blocks into several classes, and this overcomes the disadvantage that dividing all image blocks into a fixed kinds of categories in reference [14], for the image blocks with rich texture, their improved approach can speed up the encoding process more obviously; thirdly, for the classifications proposed in reference [12] and [14], those image blocks come from the adjacent class might have totally different textures, so once they can't find their best-match block in the search space which they belonged to, they couldn't expand their search into the adjacent class thus the recovered image has poor quality. But in 2004 Hai Wang [9], by the processing method of image block which is at the junction of different image contents, the image block can be divided into several different classes, once an R block can't find its best-match D block; it can expand its search to its adjacent class, so the decoded image has a higher quality.

Based on adaptive threshold quad tree fractal compression approach [9], considered image's semantic characteristic, and apply graph-based image segmentation to fractal image compression, separating the initial image into many logic areas, and then encoding each area with fractal image compression techniques. Proposed approach [9] can improve the recovered image's quality and compression ratio significantly, but encoding time lowering was very nominal.

To reduce the encoding time and increase the compression ratio and PSNR without degrading the image quality proposed a new method, fractal image compression based on graph theory. Proposed method has three steps as, first: input image segmentation based on graph theory [9], achieved better content-based image segmentation, in this technique we find the different region in fig.7, separate the original image into many different logic areas according to image content, and construct the corresponding search space for each logic area. Second: segmented image is partitioned based on triangle segmentation.

In this triangle segmentation the domain blocks and range blocks are divided into equilateral triangles rather than rectangles fig. 10 show the range bocks, domain bock and range and domain bock segmentation approach. This segmentation method is in favor of approaching the diagonal edge and using the self-similar relationship of the image which helps reconstruct the edge information of original image.

From fig. 9(a), hypotenuse length of non-overlapped range blocks R1, R2,…, Rn is 4, and R1, R2,…, Rn cover the whole image.



(a) (b) (c)

Figure 10: (a) is range block segmentation, (b) is domain block segmentation, (c) is segmentation scheme of range blocks and domain blocks

Figure 10(b) and figure 10(c) are the overlapped domain block segmentations with hypotenuse length of 10. Step length of domain blocks is L. In other words, number of pixels between two adjacent domain blocks is L. The pixel number of domain blocks is eight times more than that of range blocks. The step length of domain block would influence the efficiency and image quality of fractal coding directly. The shorter the step length, the more matching times between domain blocks and range blocks is needed. However, over-length step may result in mismatching between blocks leading to the reducing of the image quality.

For finding the best matching domain block, every domain block needs to do shrink transformation, isometric transformation, brightness transformation and brightness excursion as shown in figure 10. The most similar domain block of current range block is found after comparing range blocks with transformed domain blocks. Then record the corresponding coefficient of affine transformation. It is the coefficients which construct fractal code of current range block.

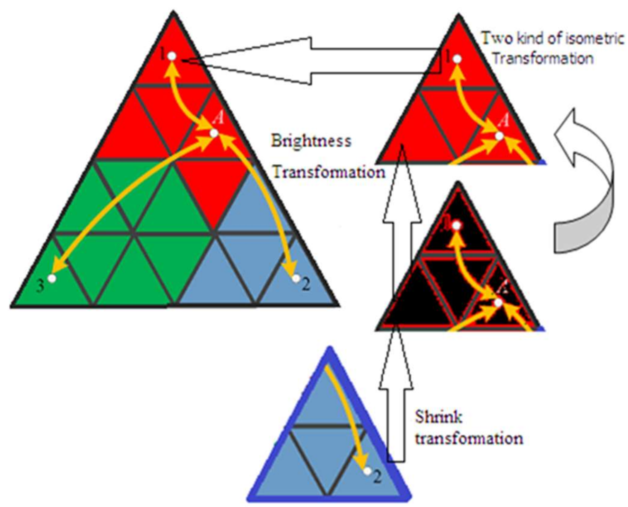


Figure 11: Matching process between domain and range blocks

In third used fractal based image encoding for getting the higher compression ratio and PSNR and decreasing the encoding time based on self- similarity. Figure 12 shows the flow diagram of the proposed encoding process.

Proposed algorithms encoding steps as following:

Step1: Segment the initial image with graph-based image segmentation algorithm, and set label for all pixels;

Step2: Divide initial image into different R blocks with non-overlapping, each R block is 8*8 size, and set label for each R block, classify R block based on triangle segmentation rather than rectangles/square;

Step3: Divide image into different D blocks allowing some overlapping, the size is four times as R block, and set label for each D block, classify D block based on triangle segmentation rather than rectangles/square;

Step 4: Shrink D block to R block's size;

Step 5: Calculate D block's variance, and for D blocks in the same class, sort them according to their variance;

Step 6: Choose one R block; calculate its variance and matching threshold;

Step 7: In the search space where R block located in, find a D block which has the closest variance to current R block;

Step 8: Define a searching window; the ratio between the number of blocks which located in the searching window and the number of total blocks is W%, and half of blocks in the searching window have a larger variance than the D block mentioned in step 7, others have a smaller variance.

Step 9: In the searching window, after affine transformation and gray migration, find the best-match D block;

Step 10: Calculate the matching error between best-match D block and R block, if the error is larger than threshold 'bias', divide R block into four sub-blocks, add them to the corresponding search space according to their labels, and slide certain steps, divide some new D blocks, add them to corresponding search space according to their labels too. If the error is smaller than threshold bias, storage affine transformation parameters; now, calculate the variance of range and domain blocks by the equation below. The variance of block I is defined as,

$$Var(I) = \sqrt{\left\{\frac{1}{n*n}\left(\sum_{i=1}^{n*n} X_i^2\right) - \left(\frac{1}{n*n}\left(\sum_{i=1}^{n*n} X_{i=1}\right)\right)^2\right\}}$$

Where n is the size of the block and Xi is the pixel value of the range blocks.
Step 11: For all R blocks, repeat step 6-10;
Step 12: Write out the compressed data in the form of a local IFS code;
Step 13: Apply a fractal compression algorithm to obtain a compressed IFS code.

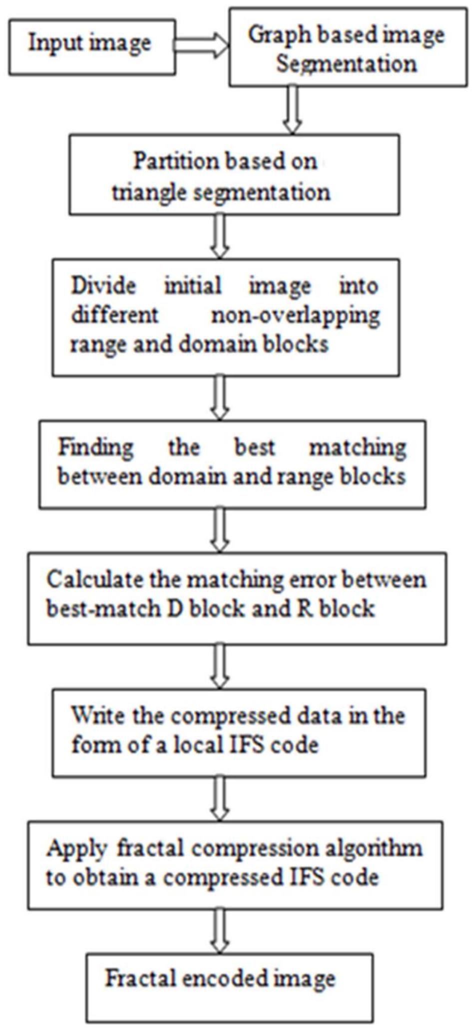


Figure 12: Fractal based encoding process

VI. IMPLEMENTATION, RESULTS AND DISCUSSIONS

The proposed technique is experimented and observed the simulation result. The simulation results are carried out Core I7 processor, Windows XP operating system. Typical results with six color images (namely, A street scene, a baseball scene, an indoor scene, Lena image, Leopard image and Tower image) of size 256×256×3 are presented here.
Image compression is based on fractal coding is lossy compression method. The quality of

image is measured by Peak Signal to Noise Ratio (PSNR).

$$\text{PSNR} = 10 \log \frac{255^2}{\left(\frac{1}{M\,X\,N}\right)\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{i,j}-\grave{x}_{i,j}\right)^2} \qquad (11)$$

Where, M X N is the size of the composite image, and x_(i,j) and x`_(i,j) are values of the original image and reconstructed image at position (i, j). Naturally higher PSNR value indicates much better image quality. And execution time of the program to calculated using tie-toe code of Matlab toolbox.

Compression ratio (CR) is a measure of the reduction of the detailed coefficient of the data. In the process of image compression, it is important to know how much detailed (important) coefficient one can discard from the input data in order to sanctuary critical information of the original data. Compression ratio can be expressed as:

$$CR = \frac{Decompressed\ image\ data\ size}{Orginal\ image\ data\ size} \qquad (12)$$

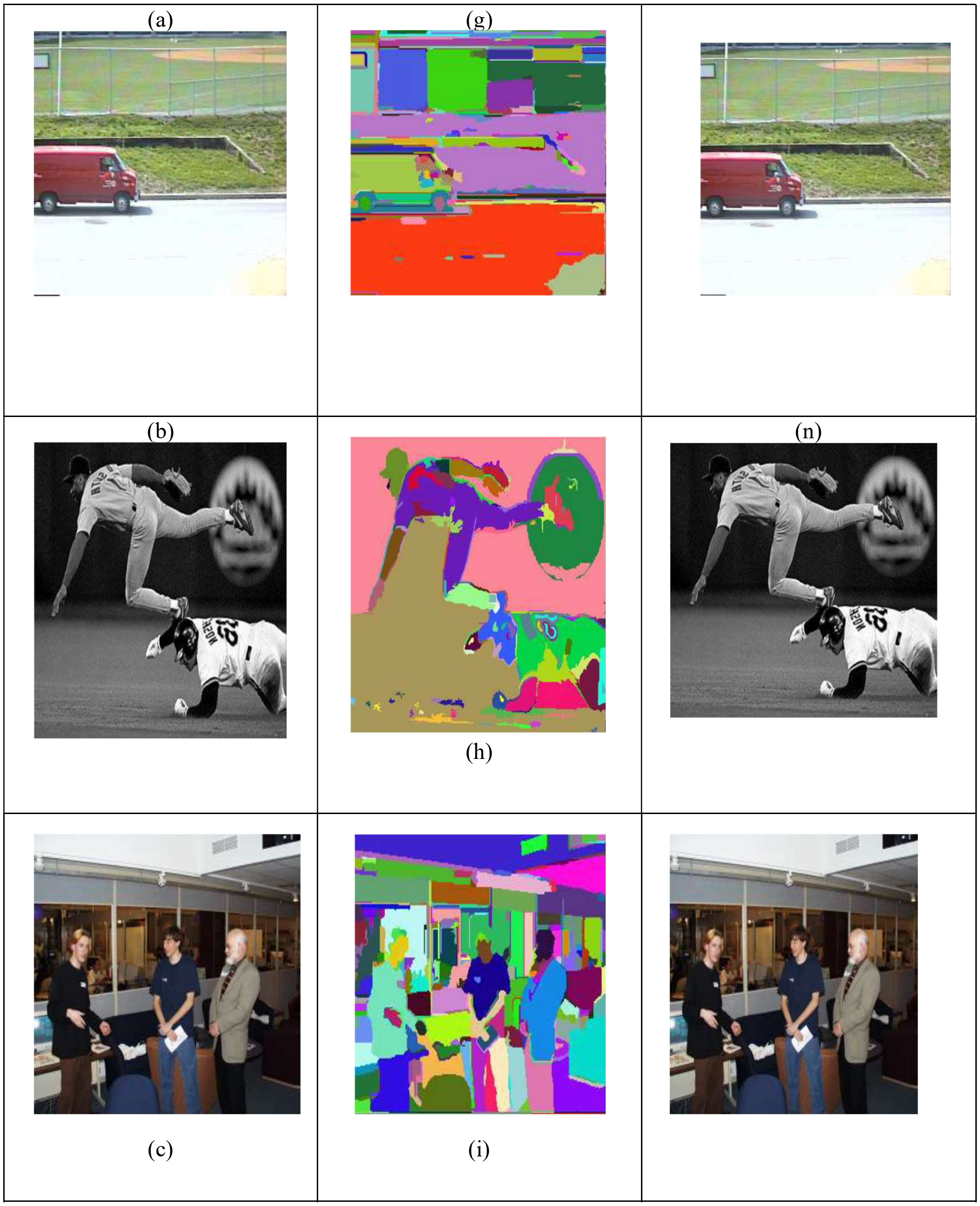
Where CR= Compression Ratio.

The quantization table (Q) and the scaling factor (SF) are the main controlling parameters of the compression ratio. Each element of the transformed data is divided by corresponding element in the quantization table (Qmatrix) and rounded to the nearest integer value by using round function in MATLAB. This process makes some of the coefficients to be zero which can be discarded [16].

In order to achieve higher compression ratio, the quantizer output is then divided by some scalar constant (SF) and rounded to nearest integer value. This process yields more zero coefficients which can be discarded during compression [17]. The CR can be varied to get different image quality. The more the details coefficients are discarded, the higher the CR can be achieved. Higher compression ratio means lower reconstruction quality of the image.

Fig.13 (a), (b), (c), (d), (e) and (f) show the original color images of A street scene, a baseball scene, an indoor scene, Lena image, Tower image and Leopard image respectively. Fig. 13 (g), (h), (h), (i), (j), (k) and (l) show the segmented images. In fig. 13 (m), (n), (o), (p), (q) and (r) show the decompressed images in our proposed method.

In the experiment, it also be implement the approaches proposed in reference [12], [13], [14], [9] and [16] under the same experimental environment, and run every compression approach 30 times, record their encoding time, recovered images PSNR, compression ratio every time, then calculate the average value for those 30 groups' data, get those six compression approaches. Table 1 shows the comparison between reference [12], [13], [14], [9] and [16] and proposed method with respects to PSNR (dB), encoding time (sec) and compression ratio (CR).


(a)
(g)
(b)
(h)
(n)
(c)
(i)

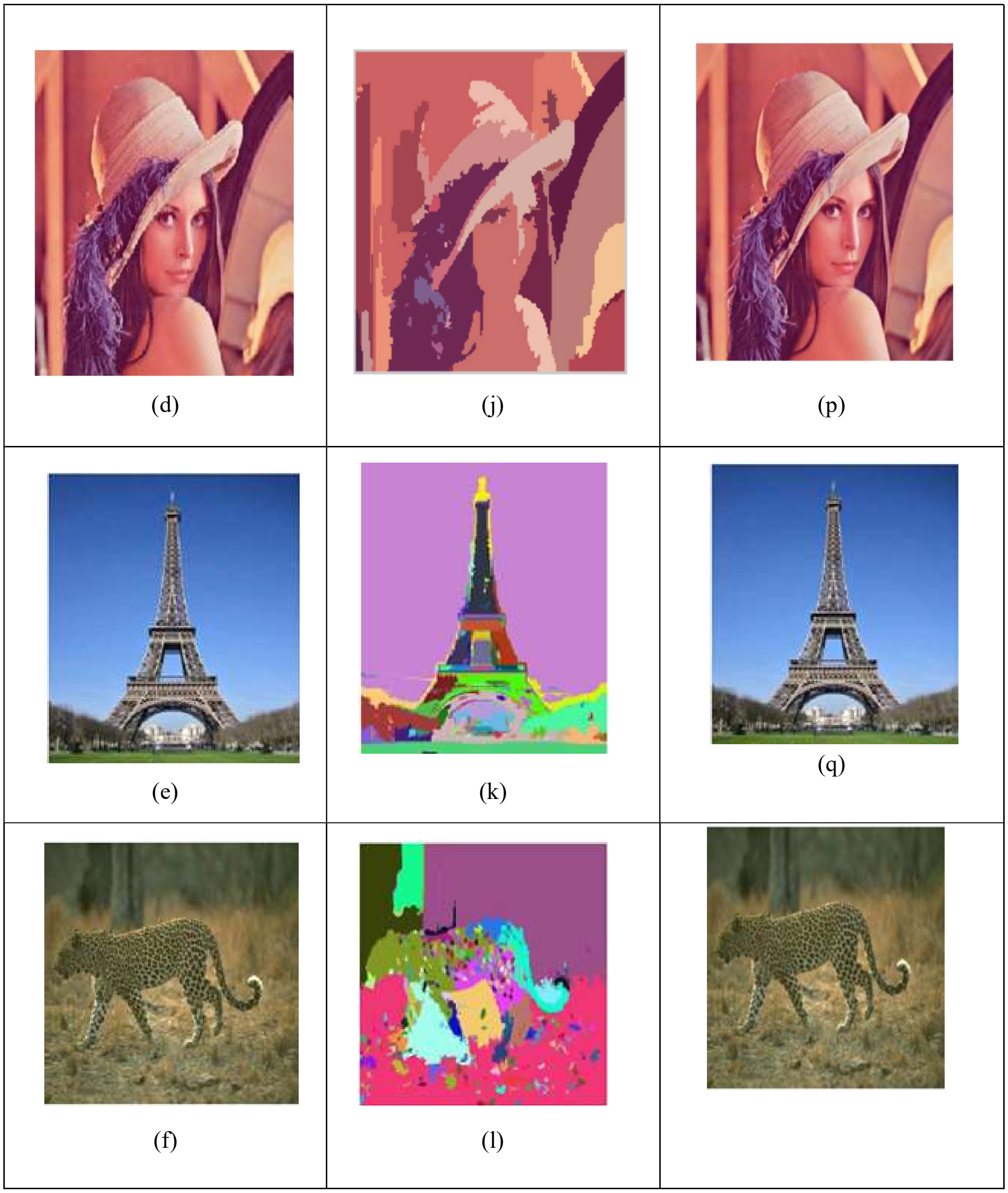


Figure 13: (a), (b), (c), (d), (e) and (f) shows the original images of A street scene, a baseball scene, an indoor scene, Lena image, Tower image and Leopard image and (g), (h), (i), (j), (k), and (l) show the segmented images and (m), (n), (o), (p), (q), (r) show the decoded images.

Table-1: Comparison between Ref [14], Ref [12], Ref [13], Ref [15], Ref [9] and Proposed Method with respect to PSNR, Encoding Time and Compression Ratio

Test images	Reference [14]			Reference [12]			Reference [13]			Reference [16]			Reference [9]			Proposed method		
	PSNR (dB)	Time (sec)	Compression ratio	PSNR (dB)	Time (sec)	Compression ratio	PSNR (dB)	Time (sec)	Compression ratio	PSNR (dB)	Time (sec)	Compression ratio	PSNR (dB)	Time (sec)	Compression ratio	PSNR (dB)	Time (sec)	Compression ratio
A street scen	30.9	2.2	8.9	32.8	3.4	9.0	32.6	3.2	9.2	35.16	42	4.88	34.9	2.3	12.2	40.01	2.2	30.2
A Baseball scene	31.01	2.4	8.8	32.6	3.7	8.9	32.4	3.1	9.0	35.12	42	4.88	34.9	2.4	12.3	41.01	2.2	31.14
An Indoor scene	30.8	2.3	8.7	32.4	3.7	9.01	32.5	3.2	9.4	35.19	43	4.84	34.5	2.4	12.01	39.71	2.1	34.88
Lena	31.01	2.3	8.8	31.8	3.3	9.1	32.8	3.2	9.3	35.11	41	4.89	34.8	2.5	11.7	39.41	2.0	33.91
Tower image	31.02	2.1	8.9	32.9	3.3	9.3	32.7	3.4	9.5	35.18	40	5.03	34.9	2.3	12.3	38.7	2.1	33.10
Leopard image	30.8	2.1	8.8	32.7	3.4	9.0	31.6	3.3	9.1	35.19	42	4.88	34.9	2.3	12.1	40.23	2.0	35.85

Table 1 represents some compared data on PSNR, coding time cost, compression ratio using the method proposed in this paper and Reference [14], Reference [12], Reference [13], Reference [15] and Reference [9]. From table 1, it has been observed that compared with other approaches, proposed compression approach can reduce the encoding time greatly, at the same time; there are also marginal increases in recovered image's quality, PSNR and compression ratio. Equilateral Triangle Segmentation based range block and domain blocks need less times of similarity matching, and the matching effect at edge is much better than square/rectangle blocks. To increasing the processing speed and compression ratio Equilateral Triangle Segmentation and graph based fractal compression has been proposed. Proposed method provide better image quality and improved time efficiency compared with other methods reference [14], [12], [13], [15] and [9].

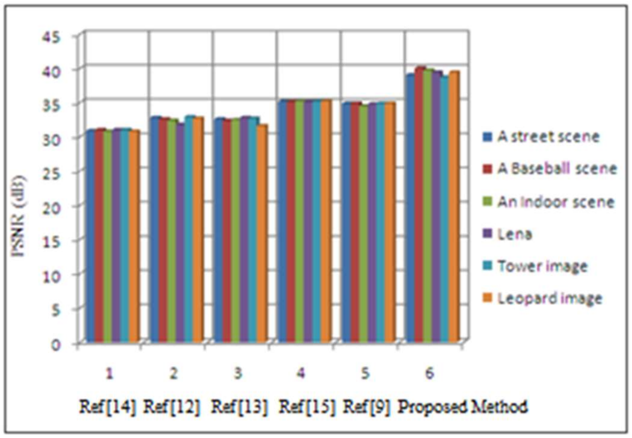


Figure 13: Showing the comparison of PSNR between Ref. [14], Ref. [12], Ref. [13], Ref. [15], Ref. [9] and Proposed method.

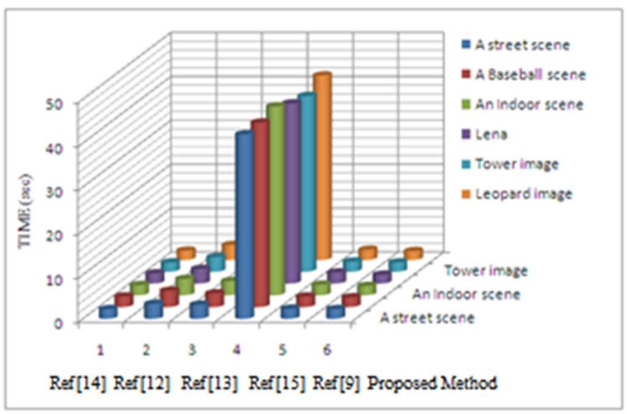


Figure 14: Comparison of TIME between Ref. [14], Ref. [12], Ref. [13], Ref. [15], Ref. [9] and Proposed method.

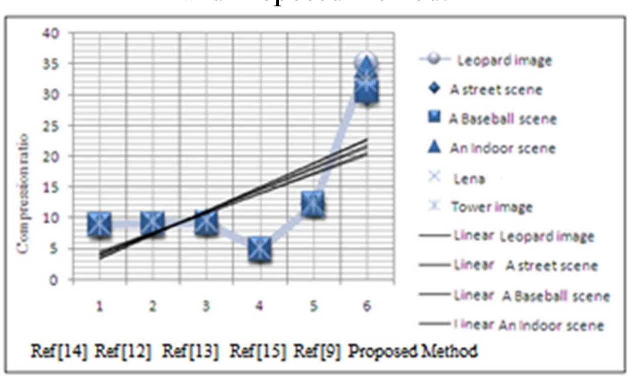


Figure 15: Comparison of compression ratio between Ref. [14], Ref. [12], Ref. [13], Ref. [15], Ref. [9] and Proposed method.

In figure 13 shows the comparison of PSNR between ref. [14], [12], [13], [15], [9] and proposed method respectively. From the comparison graph it has been noticed that value of PSNR increased in proposed method compared to others approaches. Figure 14 shows the comparison graph of encoding time and it has been seen that in proposed approached need less encoding time compared to others methods. And lastly in figure 15 shows the compression ratio is increased in proposed methods.

VII. CONCLUSIONS

Fractal geometry has created a history in dealing with the objects of irregular shapes and sizes. Most of the natural objects lie in that category. In the present study, six types of digital images have been considered. This paper briefly introduces the basic theory of image fractal compression, and discusses some typical fractal compression techniques. Based on triangle geometry fractal compression approach and applying graph-based image segmentation to fractal image compression, separating the initial image into many logic areas, then encoding each area with fractal image compression method, better of advantages have been achieved. In conclusion, graph-based segmentation is a powerful and flexible technique for image processing, presenting multiple applications in different areas. Graph-based segmentation can produce accurate and precise results by considering local and global information in an image. But, to do that, graph-based segmentation requires precise parameter tuning and can be computationally expensive. Constructing a graph using feature sets rather than pixel level information and finding an optimum partition that maximizes the dissimilarities across boundaries is also a study of interest to the research community. Proposed method can improve the recovered image quality and compression ratio significantly.

References

- M.F. Barnsley, Fractal everywhere [M], New York: Academic Press, 1988.
- A.E. Jaquin, Image coding based on a fractal theory of iterated contractive image transformations [J], IEEE Transactions on Image Processing, 1(1): 18-30, 1992.
- M.F. Barnsley and H.Rising III, Fractals Everywhere,2nd edition, Boston, MA:Academic Press Professional,1993
- B.Wohlberg and G.de Jager,A Review of the Fractal Image Coding Literature,IEEE Transactions on Image Processing,1999,8(12):234~245.
- L.Torres and M.Kunt.Video coding: the second generation approach.Kluwer AcademicPublisher,1996
- Hurd,L.P, Gustavus,M.A,Barnsley,M.F.etc.Fractal video compression. Compcon Spring '92. Thirty-Seventh IEEE Computer Society International Conference, Digest of Papers.
- Brijmohan, Yarish .Low bit-rate video coding using fractal compression of wavelet subtrees. IEEE AFRICON Conference, v 1, p 39-44, 2004
- Cochran, W.O,Hart, J.C,Flynn. P.J. Fractal volume compression. IEEE Transactions on Visualization and Computer Graphics, v 2, n 4, p 313-322, Dec 1996
- Hai Wang, Fast Image Fractal Compression with Graph-Based Image Segmentation Algorithm, International Journal of Graphics Vol. 1, No.1, pp.19-28, November, 2010.
- K. Santle Camilus, V. K. Govindan, A Review on Graph Based Segmentation, I.J. Image, Graphics and Signal Processing, 2012, 5, 1-13
- Pedro F. Felzenszwalb and Daniel P.Huttenlocher. Efficient Graph-Based Image Segmentation. International Journal of Computer Vision, 59(2) September 2004

- Shiping Zhu, Liang Yu, Kamel Bellouata. An improved fractal image coding algorithm based on adaptive threshold for quadtree partition. The International Society for Optical Engineering, v 7129, 2008
- Jinjiang Li, Da Yuan, Qingsong Xie, Caiming Zhang. Fractal Image Compression by Ant Colony Algorithm. The 9th International Conference for Young Computer Scientists
- B.Hurtgen,C.Stiler, Fast hierarchical codebook search for fractal coding of still images[J],Proceedings of EOS/SPIE Visual Communications PACS Medical Applications'93,Berlin,1993:397~408.
- Yuli Zhao, Zhiliang Zhu, Hai Yu, Fractal Color Image Coding Based on Isosceles Triangle Segmentation, International Workshop on Chaos-Fractal Theory and its Applications 2010
- Rao, K.R. and Yip, P., ―Discrete Cosine Transform‖ ,Academic Press, New York, 1990
- Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, Image quality assessment: From error visibility to structural similarity, IEEE Trans. on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004.