Statistical Analysis of Fractal Image Coding and Fixed Size Partitioning Scheme

Tai-Hoon Kim

Received: 11 December 2014 Accepted: 1 January 2015 Published: 15 January 2015

Abstract

Fractal Image Compression (FIC) is a state of the art technique used for high compression ratio. But it lacks behind in its encoding time requirements. In this method an image is divided into non-overlapping range blocks and overlapping domain blocks. The total number of domain blocks is larger than the range blocks. Similarly the sizes of the domain blocks are twice larger than the range blocks. Together all domain blocks creates a domain pool. A range block is compared with all possible domains block for similarity measure. So the domain is decimated for a proper domainrange comparison. In this paper a novel domain pool decimation and reduction technique has been developed which uses the median as a measure of the central tendency instead of the mean (or average) of the domain pixel values. However this process is very time consuming.

Index terms— fractal image compression, fishers classification, hierarchi-cal classification, median, DCT, IFS, PIFS, PSNR.

1 Introduction

major objective of image coding is to represent digital images with as few bits as possible while preserving the level of intelligibility, usability or quality required for the application. Fractal image coding has been used in many image processing applications such as feature extractions, image watermarking, image signatures, image retrievals and texture segmentation The theory of fractal based image compression using iterated function system (IFS) was first proposed by Michael Barnsley [2]. A fully automated version of the compression algorithm was first developed by Arnaud Jacquin, using partitioned IFS (PIFS) [8]. Jacquins FIC scheme is called the baseline fractal image compression (BFIC) [2,3]. This method exploits the fact that real world images are highly self-similar [4] i.e. diferent portions of an image resemble each other. Also there is self-similarity at every scale. Fractal compression is an asymmetric process. Encoding time is much greater compared to decoding time, since the encoding algorithm has to repeatedly compare a large number of domains with each range to _nd the bestmatch. Thus the Jacquin's Scheme lacks behind other image compression techniques like jpeg (DCT [12,22,24] based image compression) or wavelet based technique. Thus the most critical problem this technique faces is its slow compression step. A huge amount of research has been done to improve the performance of this technique which mainly includes:-Better partitioning scheme; Efective encoding scheme; Reducing the number of domains in the domain pool; Reducing number of domain and range comparison or better classification; II.

2 Fractal Image Compression a) Mathematics

The mathematical analogue of a partition copying machine is called a parti-tion iterated system (PIFS) [6]. The definition of a PIFS is not dependent on the type of transformations, but in this paper we will use affine transformations. There are two spatial dimensions and the grey level adds a third dimension, so the transformations W i are form, An affine transformation in Rn is a function consisting of a linear trans-formation and translation in Rn. Affine transformations in R2, for example, are of the form:-W (x; y) = (ax + by + e; cx + dy + f)

Where the parameters a, b, c, and d form the linear part, which deter-mines the rotation, skew, and scaling; and the parameters e and f are the translation distances in the x and y directions, respectively.

A domain and a range is compared using an RMS metric [6]. Given two square sub-images containing n pixel intensities, a 1 ; a 2 ;?,a n (from the domain)and b 1 ; b 2 ;?,b n (from the range), with contrast sA W i ? ? x y z ? ? = ? ? a i,1 a i,2 0 a i,3 a i,4 0 0 0 0 ? ? × ? ? x y z ? ? + ? ? d i,1 d i,2 o i ? ?**(1)**

and brightness o between them, the RMS distance between the domain and the range is given by This gives the settings for contrast scaling s and brightness o that make the affinely transformed a i values to have the least squared distance from the b i values. The minimum value of R occurs when the partial derivatives with respect to s and o are zero. Solving the resulting equations will give the coe_cients s and o as shown below in Eq. 4 and 5.

Detailed mathematical description of IFS theory and other relevant results can be found in (Barnsley, 1988; **??**arnsley and Hurd, 1993; **??**dgar, 2007, Falconer, 2013) [2,3,7].

3 b) The Pain

As mentioned in section 1, a very large number of domain-range comparisons is the main bottleneck of the compression algorithm [6]. For example, consider an image of size 512 x 512. Let the image be partitioned into 4 x 4 non-overlapping range blocks. There will be total 2 14 = 16384 range blocks. Let the size of domain blocks be 8 x 8 (most implementations use domain sizes that are double the size of range). The domain blocks are overlapping. Then, for a complete search, each range block has to be compared with 505 x 505 = 255025 domain blocks. The total number of comparisons will be around 232. The time complexity can be estimated as (2 n):

III.

4 Partition Schemes

The first decision to be made when designing a fractal coding scheme is in the choice of the type of image partition used for the range blocks [12]. The domain blocks need to be transformed to cover range blocks. Thus this 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.

5 a) Fixed Size Partitioning

This is the simplest of all partitioning schemes that consists of fixed size square blocks [5] depicted in Fig. 1(a). This type of block partition is successful in transform coding of individual image blocks since an adaptive quantization mechanism is able to compensate for the varying activity levels of diferent blocks, allocating few bits to blocks with little detail and many to detailed blocks [12].

6 Statistical Analysis of Fractal Image Coding and Fixed Size Partitioning

Scheme R = n i=1 (s.a i + o ? b i) 2 (3) s = [(n i=1 d i r i) ? (n i=1 d i)(n i=1 r i)] [n(n i=1 d 2 i) ? (n i=1 d i) 2] (4) o = 1 n [n i=1 b i ? s n i=1 a i]**(5)**

and The quadtree partition shown in Fig. 1(b) recursively splits of selected image quadrants, which enables the resulting partition to be represented by a tree structure in which each non-terminal node has four descendants. 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 not found.d rms (f ? (R i xI), w i (f))**(6)**

7 c) Horizontal-Vertical Partitioning

This is a variant of the quadtree partitioning scheme in which a rectangular image [26] is partitioned shown in Fig. 1(c) either horizontally or vertically to form two new rectangles. The partitioning repeats recursively until a covering tolerance is satis_ed, as in the quadtree scheme. This scheme is more exible, since the position of the partition is variable.

8 d) Triangular Partitioning

This is a specialization of the polygon partitioning scheme in which the image is partitioned recursively into triangular blocks shown in Fig. 1(d).

9 Problems of Exhaustive Search

As describe in section 1, a very large number of domain-range comparison is the main dificulty of the fractal encoding algorithm. Experiments on standard images, consider an image of size N x N. Let the entire image is partitioned into M x M non-overlapping range blocks. The total number of range blocks are given by Most implementation use the size ofdomain block is twice larger than the range block i.e. 2 x M. Let the total number of domain blocks are given by (N -2M + 1) 2 . The domain blocks are overlapping. In Algorithm 1, there are nested LOOP in the process and for every step we need to calculate the error defined by Eq. 6. The computation of best matching between a range block and a domain block is O(M 2). Considering M to be a constant, the Fig. ?? Domain search of a range computation complexity domain search for a range is O(N 4), which is approximately exponential time. Encoding time can be reduced by reducing the size of the domain pool [1,25].

V.

10 Fisher's Classification Scheme

The domain-range comparison step of the image encoding is very computationally intensive. We use a classification scheme in order to reduce the number of domains blocks compared with a range blocks. The classification scheme is the most common approach for reducing the computational complexity. In such classification schemes, domain blocks are grouped in to number of classes according to their common characteristics. For fractal image decoding, the decoding will be done in less number of comparisons, so that it would become the faster computations. While reconstructing, the pixels of each range with the average of their corresponding domain are sub-stituted. This provides a very high quality image in a few iterations withoutany change in compression Error Calculation After that it is also possible to rotate the subimage (domain or range) such that the Ai are ordered in one of the following three ways: These orderings constitute three major classes and are called canonical orderings. Under each major class, there are 24 subclasses consisting of 4 P 4 orderings of V i . Thus there are 72 classes in all. In this paper, we refer to this classification scheme as FISHER72.error = a k D + b l I ? R 2 (7) N M) 2

According to the fisher that the distribution of domains across the 72 classes was far from uniform [14]. So fisher went on to further simplify the scheme of 24 classes in the FISHER72 classification. Fisher concluded: the improvement attained by using 72 rather than 24 classes is minimal and comes at great expense of time [6]. In this paper, we refer to this modified form of FISHER72 as FISHER24 using this concepts a hierarchical classification is proposed by N. Bhattacharya et al. [14]. We simply take the advantages of hierarchical classification [14] of sub-images and combining with fixed size partition to reduce the encoding time.

11 VI. Proposed Hierarchical Classification Scheme

Fisher used values proportional to the mean and the variance of the pixel intensities to classify the domain and range image. In our proposed schemes Algorithm 2 [13], we use only the sum of pixel intensities of fixed parts of domain (8 x 8) or range (4 x 4) then classify those fixed part. According to the proposed Algorithm 2 [13] compression, at first the domain pool is being related data structures are defined as in the Fig. 3. Domains are first classified by their size, then into Level-I, according to pixel-value sum of 4 quadrants, and finally into Level-II, according to pixel-value sum of 16 sub quadrants. After two Levels of classification domain is place in list of point to array known as domain pool Fig. 3.

In the proposed compression algorithm, when searching the domain pool for a best-match with a particular range, only those domains that are in the same Level-II and same class. A i = n j=1 r i j (8) V i = n j=1 (r i j) 2 ? A i**(9)**

12 Year ( ) a) PROPOSED TECHNIQUE -I (P-I)

In the domain pool creation phase, Jacquin [10] selected squares cantered on a lattice with a spacing of one-half of the domain size. It is convenient to select domains with twice the range size and then to subsample or average groups of 2 x 2 pixels to get a reduced domain with same number of pixels as the range as shown in Fig. 4. In our proposed technique we calculate the median of the 2 x 2 pixel blocks instead of taking the average or mean of the pixels. It produces better results as median is a better measure (or statistic) of the central tendency of data. This is because the mean is susceptible to the inuence of outliers (i.e. an extreme value that difers greatly from other values). So, this will

13 Range Pool (R)

The image is partitioned into non-overlapping Fixed size range (4 x 4).

14 3:

15 Domain Pool (D)

The image is partitioned into overlapping Fixed size domain (8 x 8).

16 4:

17 Loop

Each range block is then divided into upper left, upper right, lower left and lower right each part is known as quadrant (S i).S i = n j=1 r i j (10) 5:

Thus we observe that there can be in total 4 P 4 (24) permutations possible, based on the relative ordering of the summation of pixel intensities and a corresponding class (class -1 to 24) is assigned to it.

18 6:

Each of the quadrant is further sub-divided into four sub-quadrants.

19 7:

The sum of pixel values S i,j (i = 0,1,2,3; j = 0,1,2,3) for each subquadrant are calculated.

20 8:

We again obtain the classes each of the sub-quadrants (class 1 to 24) i.e. for a particular a range /domain block we obtain 16 sub-quadrants or the domain pool can be classified into 24 4 = 331776 classes.

nullify the efect outlier pixel value among the four pixels and produce a value that is closer to the majority of pixel values.

The reduced domain pool thus contains the median values of the 2 x 2 blocks.

21 b) Proposed Technique -II (P-II)

This is an add-on to the Algorithm 2 [13] that has been proposed above, to reduce the number of domain-range comparisons.

Each of the four quadrants of a domain are assigned a number between 1 and 24 gives 244 =331776 cases in total shown in Fig. 5, for the entire sub-image. A number between 1 and 331776 that uniquely identifies this The main idea behind this procedure is to heuristically eliminate the null classes or the classes which don't contain any domain.

22 VII.

23 Results and Discussions a) Tools

Five standard 512 x 512 x 8 grayscale images have been used to test the proposed techniques 5 and also for comparison with FISHER24 classification scheme and modified Hierarchical classification [14].

The algorithm was implemented in C++ programming language running on a PC with following specifications: CPU Intel Core 2 Duo 2.0 GHz; RAM 4 GB; OS Ubuntu 14.4 64-bit.

24 b) Research Result

The Comparison of compression time for the five image files have been made in Table 1. The comparison of PSNRs for the same image are given in Table 2 while space saving are given in Table 3. The pictorial representation of compression times, PSNRs, space savings and decoding times are illustrated in Figures 6, 7, and 8 respectively. particular case is assigned to this sub-image [13]. Thus there are a lot of classes which are left empty (i.e. no domains are assigned to it).

25 c) Extended Experimental Result

In the previous proposed [13] technique we used the minimum domain block size of 8 x 8 pixels. The PSNR has been improved by reducing the minimum domain block size to 4 x 4 pixels (range blocks are 2 x 2). As a trade-of the encoding time is slightly increased. This is because, as the block domain size has been reduced, the no. of domains in the domain pool increases. But the overall efect on PSNR outweighs the increased encoding time. So this method is convenient. The results have been shown in the tables below based on the comparison of Fisher's method, P-I and P-II.

We test the extended technique proposed-I and proposed-II with standard Lenna image (512 x 512 x 8). For every range block, we use 3 bits to store the scaling parameter ai in Eq. 3 and 1 byte to store the mean of range block ~r. In Fixed size partitioning structure, we considered 2 levels which starts 4 X 4 domain block size and 2 x 2 range block size. We see that, P-I and P-II fractal coding technique is very fast, when PSNR = 30, it only takes only 1.371 s (P-I) and 1.370 s (P-II)

To compare our proposed technique with the result of fast method reported by Tong and Wong [27]. Tong and Wong improved the algorithm proposed by Saupe [17]. To comparison of Tong and Wong, Saupe and our method for Baboon(512 x 512 x 8) shown in Table **??** 7.

The Comparison of compression time for the six image files have been made in Table 4. The comparison of PSNRs for the same image are given in Table 5 while space saving are given in Table 6. The pictorial representation of compression times, PSNRs, space savings and decoding times are illustrated in Figures 10, 11, and 12 respectively. Figure 13 show the close up of Standard original images, decoded images after using existing as well as proposed P-I and P-II.

10

Figure 1: ? 10 Global

26 Conclusions

The proposed Fractal image encoding by using fixed size partition and hierarchical classification of domain and range improves the compression time [1] [2]

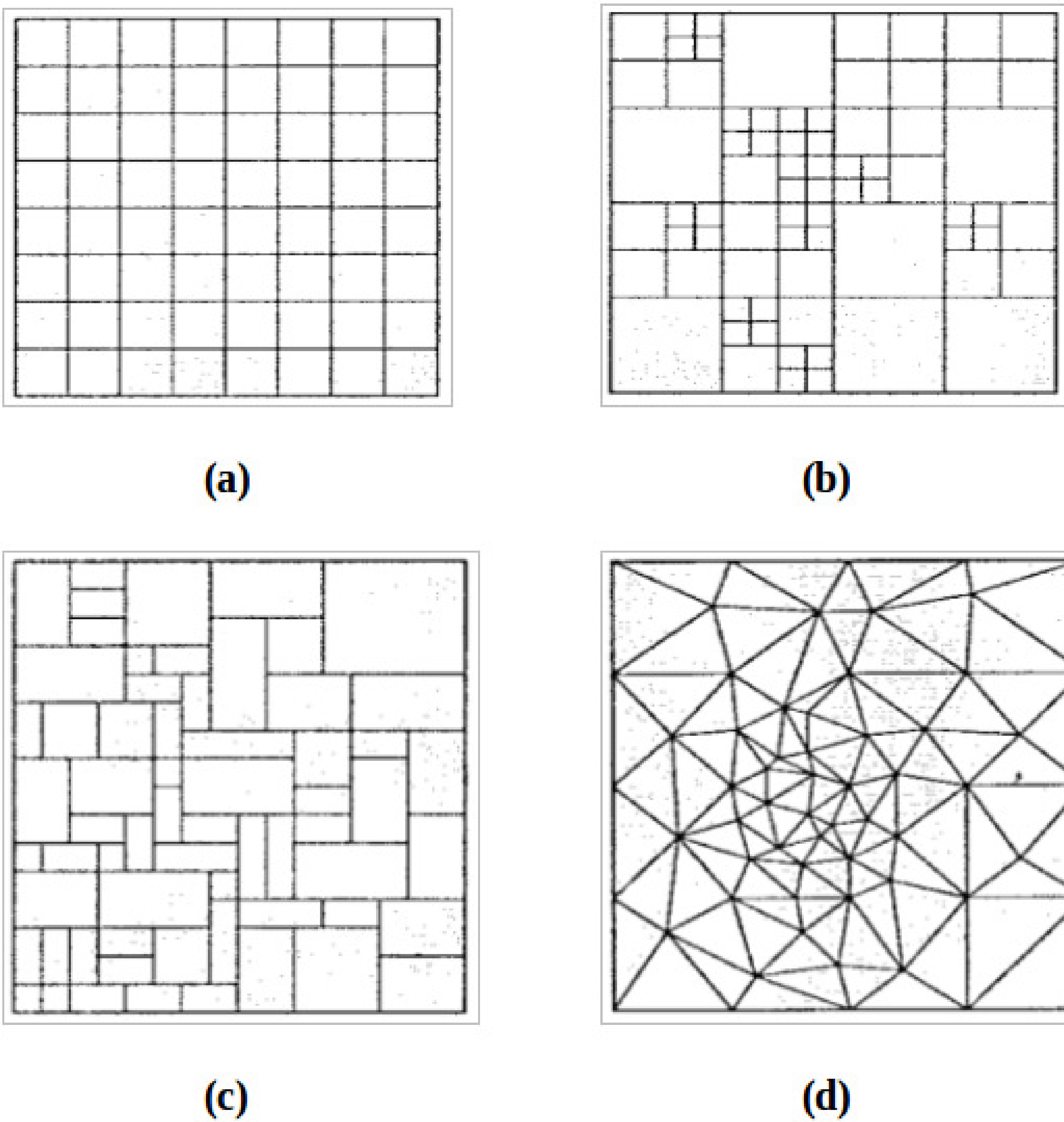
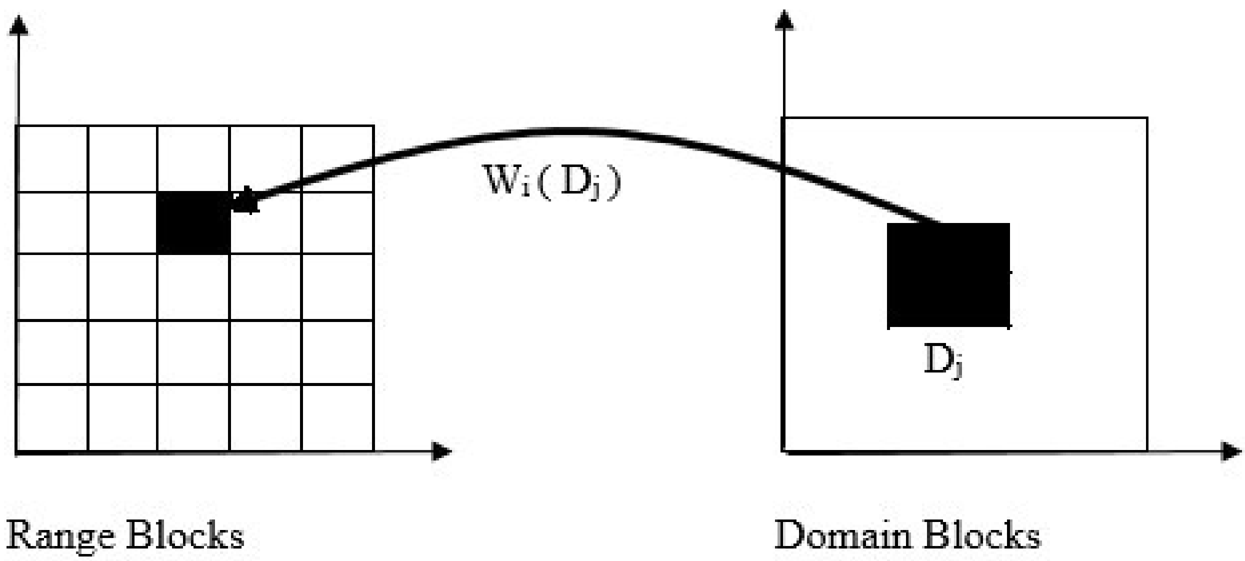
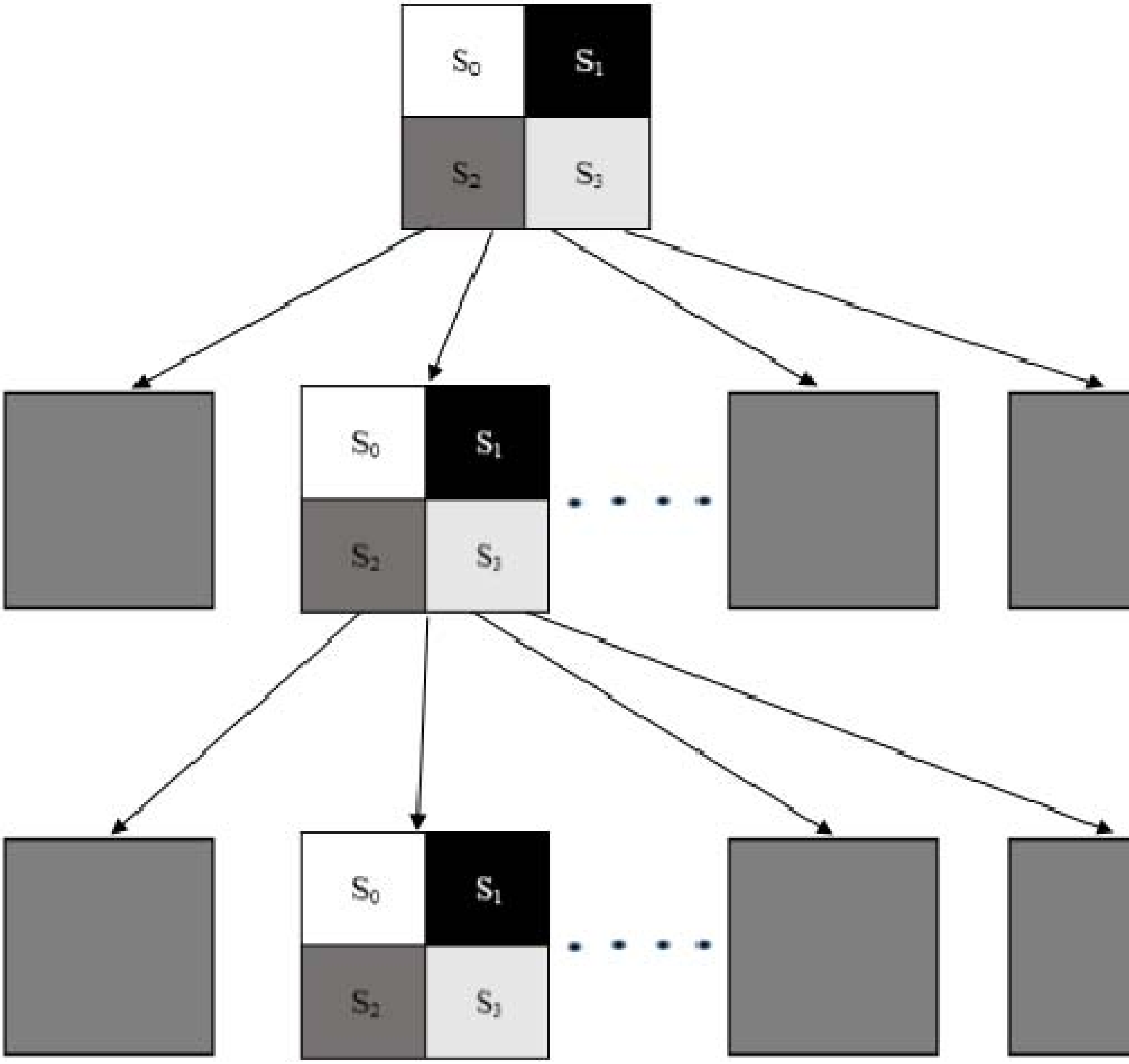


Figure 2: Statistical



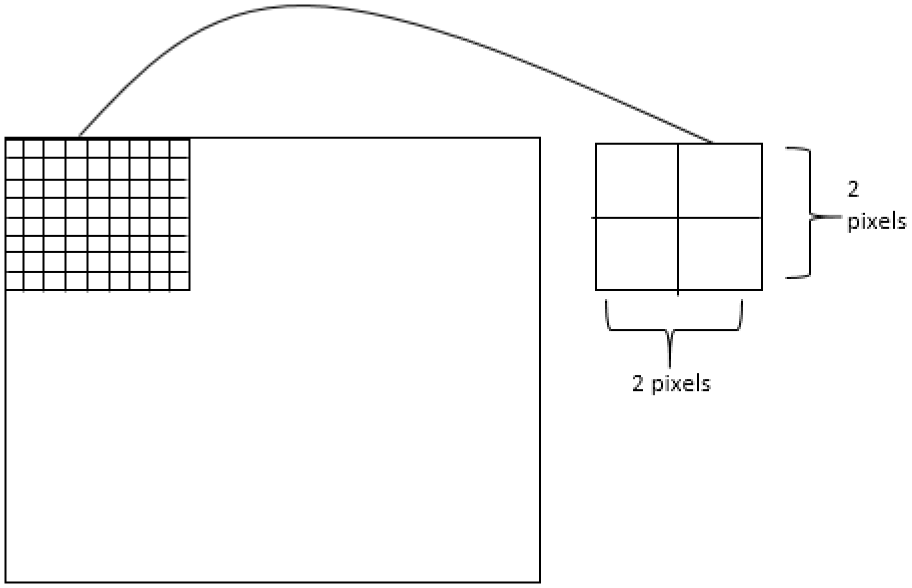
1

Figure 3: Figure 1 :



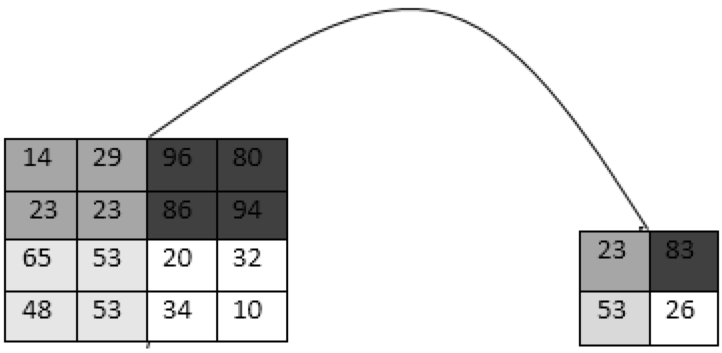
567

Figure 4: 5 : 6 : 7 :



212

Figure 5: Figure 2 : 12 Global



3

Figure 6: Figure 3 :

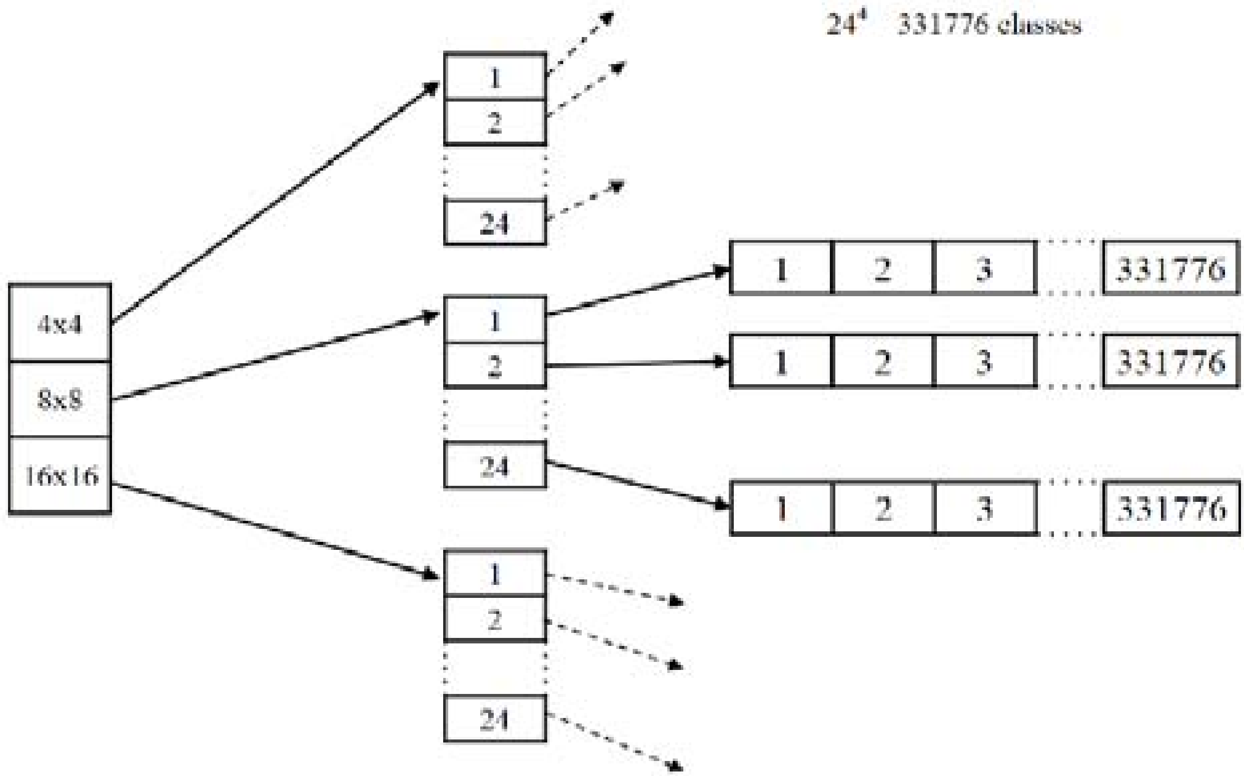
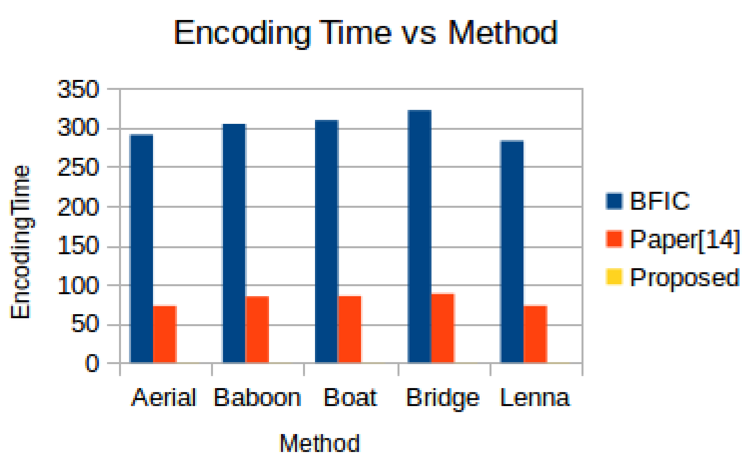
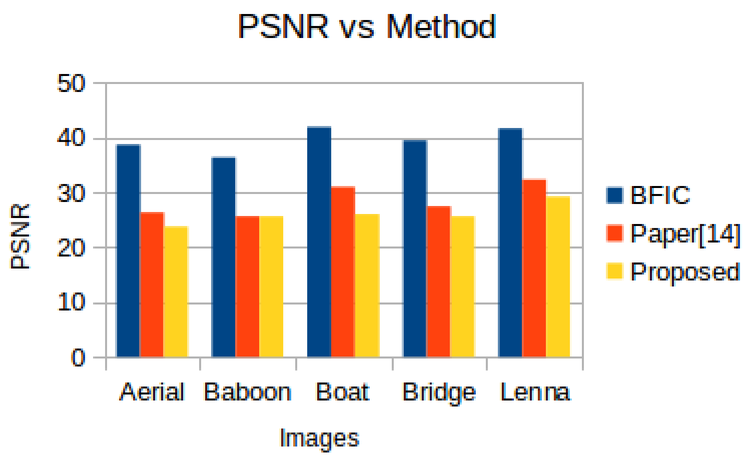


Figure 7: Statistical



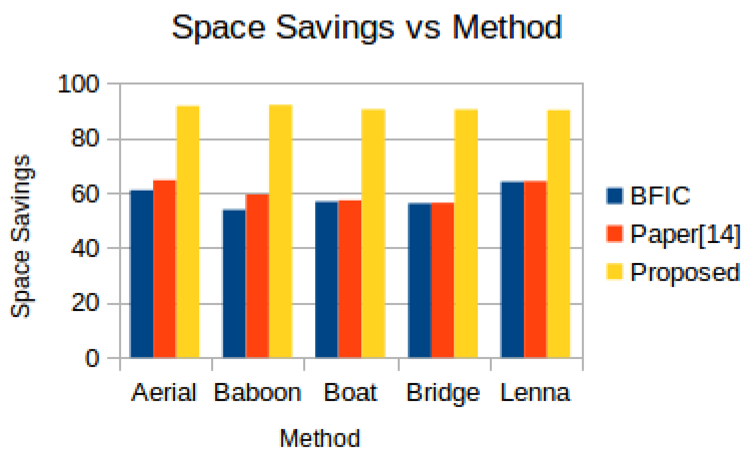
4

Figure 8: Figure 4 :



5

Figure 9: Figure 5 :



6

Figure 10: Figure 6 :

7

Figure 11: Figure 7 :

8

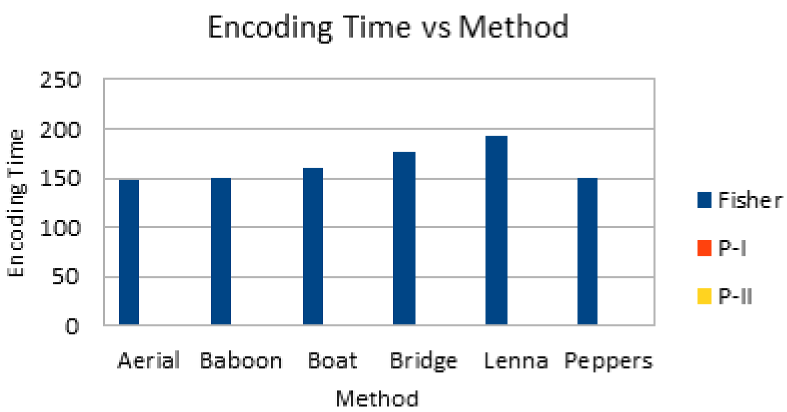
Figure 12: Figure 8 :

9

Figure 13: Figure 9 :

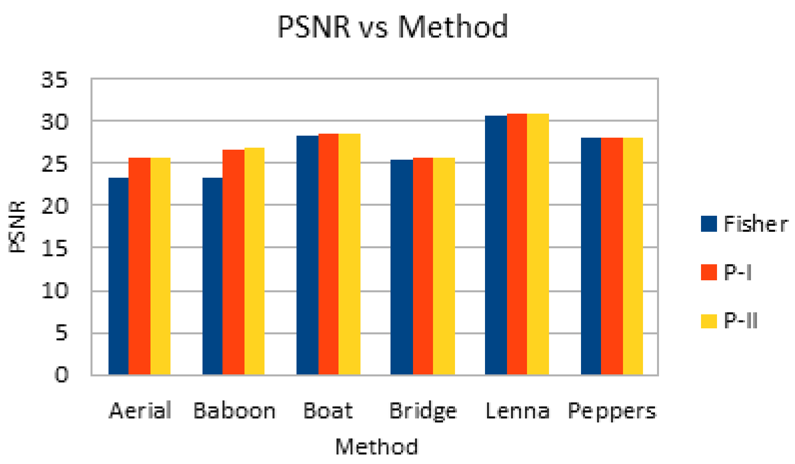
10 

Figure 14: Figure 10 :



12

Figure 15: Figure 12 :



11

Figure 16: Figure 11 :

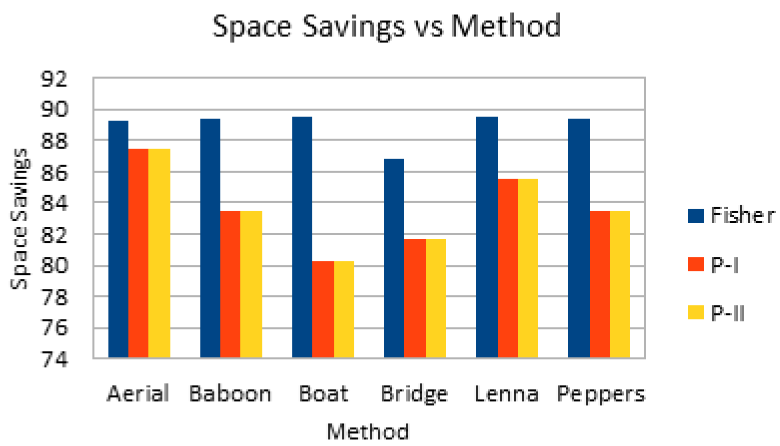


Figure 17:

1: procedure BFIC
2: Loop:
3: Range Block for every range block R i ,
i = 1,2,....,N R ,do
4: Loop:

Figure 18:

[Note: 1: procedure Proposed 2:]

Figure 19:

2

Figure 20: Table 2 :

1

Image data BFIC Paper [14] Proposed			
Aerial	291.081	72.781	0.451
Baboon	304.790	84.618	0.437
Boat	309.488	85.425	0.439
Bridge	322.336	88.303	0.441
Lenna	283.244	72.949	0.492

Figure 21: Table 1 :

3

Figure 22: Table 3 :

4

Image data BFIC Paper [14] Proposed				
Aerial		60.94	64.63	91.71
Baboon		53.80	59.36	92.07
Boat	Bridge	56.76	57.27	90.43
Lenna		56.12	56.34	90.40
		64.03	64.23	90.23

Image data Fisher Aerial 147.441 1.373 1.310 P-I P-II Baboon

Boat	160.219 1.910	2.098
Bridge	175.924 1.798	2.171
Lenna	193.066 1.370	1.371
Peppers	150.112 1.211	1.435

Figure 23: Table 4 :

5

Image data	Fisher	P-I	P-II
Aerial	23.22	25.63	25.66
Baboon	23.40	26.55	26.87
	28.44	28.46	28.50
Bridge	25.55	25.61	25.62
Lenna	30.60	30.95	30.95
Peppers	28.10	28.01	28.10

[Note: a. Original image b. Decoding result P-I c. Decoding result P-II d. Decoding result Fisher's [6]]

Figure 24: Table 5 :

6

Figure 25: Table 6 :

7

Method	PSNR(dB)	TIME(s)
Proposed-I (P-I)	26.55	2.211
Proposed-II (P-II)	26.87	1.988
Tong and Wong [27]	25.82	8
Saupe [17]	25.19	60

Figure 26: Table 7 :

[Trieu-Kien et al. (2000)] ‘A Fast Encoding Algorithm for Fractal Image Compression using the DCT Inner Product’. Trieu-Kien , J H Truong , I S Jeng , P C Reed , Alan Q Lee , Li . *IEEE Trans. On Image Processing* April 2000. 9 (4) .

[Furao and Hasegawa ()] ‘A Fast No Search Fractal Image Coding Method’. S Furao
mage Processing , O Hasegawa
mage Processing . *Signal Processing: Image Communication* 2004. 1990. 19 p. .

[Jacquin ()] ‘A Fractal Image Coding Based on a Theory of Iterated Contractive Image Transformations’. A Jacquin . *Proc. SPIE’s Visual Communications and Image Processing*, (SPIE’s Visual Communications and Image essing) 1990. p. .

[Jacquin (1989)] *A Fractal Image Coding Based on a Theory of Iterated Markov Operators with Applications to Digital Image Coding*, A Jacquin . Aug. 1989. Georgia Institute of Technology (PhD thesis)

[Thao (1996)] ‘A hybrid fractal-DCT coding scheme for image compression’. N T Thao . *Proc. IEEE Int. Conf. Image Processing*, (IEEE Int. Conf. Image essingLausanne, Switzerland) Sept. 1996. I p. .

[Roy et al. (2015)] ‘A Speeding Up Fractal Image Compression using Fixed Size Partition and Hierarchical Classification of Sub-Images’. S K Roy , S K Bandyopadhyay , A Mahato , Tai-Hoon Kim . *NAUN) International Conference on Mathematical, Computational and Statistical Sciences*, (Dubai) Feb. 2015. p. . 3rd North Atlantic University Union (

[Tong and Wong ()] ‘Adaptive approximate nearest neighbour search for fractal image compression’. C S Tong , M Wong . *IEEE Trans. Image Processing* 2002. 11 (6) p. .

[Zhou et al.] ‘An E_cient Fractal Image Coding Algorithm using Unified Feature and DCT’. Y Zhou , C Zhang , Z Zhang . *Chaos, Solutions and Fractals* 39 p. .

[Tong and Pi ()] ‘Analysis of hybrid fractal predictive coding encoding scheme’. C S Tong , M Pi . *Signal Processing: Image Communication*, 2013. 18 p. .

[Monoro and Dudbridge ()] ‘Approximation of Image Blocks’. D M Monoro , F Dudbridge . *Proceedings of the International Conference on Acoustics, Speech, Signal Process ing*, (the International Conference on Acoustics, Speech, Signal Process ing) 1992. 3 p. .

[Barnsley ()] M F Barnsley . *Fractals Everywhere*, (San Diego) 1988. Academic Press.

[Bridge and Peppers. (a) Original image. (b) Result of using FISHER24 classification. (c) Result of using proposed P-I technique. ‘Bridge and Peppers. (a) Original image. (b) Result of using FISHER24 classification. (c) Result of using proposed P-I technique. (d) Result of using proposed P-II technique’. *International Conference on Image Processing*, (Washington, DC) Oct. 1997. 13. (Test images and results -Baboon, Boat)

[Cormen et al. ()] T H Cormen , C E Leiserson , R L Rivest , C Stein . *Introduction to Algorithms*, (Cambridge, MA, U.S.A) 2009. MIT press.

[Duh et al. (2005)] ‘DCT based Simple Classification Scheme for Fractal Image Compression’. D J Duh , J H Jeng , S Y Chen . *Image and Vision Computing* May 2005. Elsevier. 23 p. .

[Falconer ()] K Falconer . *Fractal Geometry: Mathematical Foundations and Applications*, 2013. John Wiley & Sons.

[Tong and Pi ()] ‘Fast Fractal Image Encoding Based on Adaptive Search, Image Processing’. C S Tong , M Pi . *IEEE Trans. On Image Processing* 2001. 9 p. .

[Wohlberg and De Jager (1995)] ‘Fast Image Domain Fractal Compression by DCT Domain Block Matching’. B Wohlberg , G De Jager . *Electronics Letters* May 1995. 31 p. .

[Fisher ()] Y Fisher . *Fractal Image Compression: Theory and Applications*, (New York) 1995. Springer-Verlag.

[Jacquin (1993)] ‘Fractal Image Coding: A Review’. A E Jacquin . *Proc. IEEE*, (IEEE) Oct. 1993. 1 p. .

[Barnsley et al. (1992)] *Fractal Image Compression*, M F Barnsley , L P Hurd , Ltd Peters , Ma Wellesley . December 1992.

[Bhattacharya et al. (2015)] ‘Fractal Image Compression Using Hierarchical Classification of Sub-Images’. N Bhattacharya , S K Roy , U Nandi , S Banerjee . *10 th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, (Berlin) March 2015.

[Saupe ()] ‘Fractal Image Compression Via Nearest Neighbor Search’. D Saupe . *Conference on Proceedings of SPIE Electronic Imaging*, (San Jose) 1996. 2669.

[Jacquin] A E Jacquin . *Fractal Coding Based on a Fractal Theory of Iterated Contractive Transformations*, IEEE.

[Lasfar et al. ()] A Lasfar , S Mouline , D Aboutajdine , H Cheri_. *Content-besed Retrieval in Fractal Coded Image Databased in Proc. 15th Int. Conference on Pattern Recognition*, 2008. 1 p. .

[Ponomarenko et al. ()] 'Modified horizontal vertical partition scheme for fractal image compression'. N Ponomarenko , V Lukin , K Egiazarian , J Astola . *Proc. 5 th Nordic Signal Processing Symp*, (5 th Nordic Signal essing SympHurtigruten, Norway) 2002.

[Bani ()] 'Speeding Up Fractal Image Compression'. B Bani . *Proc. SPIE's: Still-Image Compression*, (SPIE's: Still-Image Compression) 1995. 2418 p. .

[Statistical Analysis of Fractal Image Coding and Fixed Size Partitioning Scheme Trans. Image Processing (1992)] *Statistical Analysis of Fractal Image Coding and Fixed Size Partitioning Scheme Trans. Image Processing*, Jan. 1992. 1 p. .

[Benoit and Mandelbrot (1982)] *The Fractal Geometry of Nature, Times Books*, B Benoit , Mandelbrot . 15 August 1982.

[Monoro and Wake_Eld] 'Zooming with Implicit Fractals'. D M Monoro , P D Wake_Eld . *Proceedings of the ICIP*, (the ICIP) IEEE.