EFFICIENCY AND CAPABILITY OF FRACTAL IMAGE COMPRESSION WITH ADAPTIVE QUARDTREE PARTITIONING

Utpal Nandi [1] and Jyotsna Kumar Mandal [2]

[1]Dept. of Comp. Sc. & Engg., Academy Of Technology, West Bengal, India
nandi.3utpal@gmail.com

[2]Dept. of Computer Sc. & Engg., University of Kalyani, West Bengal, India
jkm.cse@gmail.com

ABSTRACT

In this paper, efficiency and capability of an adaptive quardtree partitioning scheme of fractal image compression technique is discussed with respect to quardtree partitioning scheme. In adaptive quardtree partitioning scheme, the image is partitioned recursively into four sub-images. Instead of middle points of the image sides are selected as in quardtree partitioning scheme, Image contexts are used to find the partitioning points. The image is partitioned row-wise into two sub-images using biased successive differences of sum of pixel values of rows of the image. Biased successive differences of sum of pixel values of columns of the sub-images are used to partition each sub-image farther column-wise into two parts. Then, a fractal image compression technique based on the adaptive quardtree partitioning scheme is discussed. The comparison of the compression ratio, compression time and PSNR are done between fractal image compression with quardtree and adaptive quardtree partitioning schemes. The fractal image compression with adaptive quardtree partitioning scheme offers better rate of compression most of the time with comparatively improved PSNR. But, the time of compression of the same scheme is much more than its counterpart.

KEYWORDS

Fractal compression, Quardtree partition, Iterated Function Systems (IFS), Partitioned Iterated Function Systems (PIFS), Affine map compression ratio.

1. INTRODUCTION

There are several varieties of standard lossy image compression techniques. JPEG [14, 15, 18] is one of the well-known standard for lossy image compression. The compression technique is very effective at low compression ratios. High compression rates can only be achieved by eliminating high frequency components of images. As a result images become very blocky after decompression of the compressed image and image quality becomes poor for practical use with the increasing of compression ratio. Another drawback of this technique is that it is resolution dependent. Replications of pixels are done to zoom-in on a portion of an image and to enlarge the image that produce blockiness of image. This problem can be solved by saving the same image at different resolution. But, it wastes storage space. Currently, another method got its popularity is fractal image compression technique [2, 3, 5, 6, 10, 11, 12, 17, 18] that can solve the these problems. The technique can offer much better rates of compression than DCT-based JPEG compression technique and is resolution independent. Therefore, the technique works well not

DOI : 10.5121/ijma.2013.5404

only for offering high rates of compression , but also for enlarging a complete image by some scale factor from same compressed image and zooming on a portion of an image. First, the term fractal is used to represent objects that are self-similar at different scales such as clouds, trees, mountains etc. These real objects have details at every scale. The interest of fractal methods is grown for image compression after development of the theory of Iterated Function Systems (IFS) [16, 17]. An Iterated Function System (IFS) is collection of contractive transformations { $T_i : R_2 \rightarrow R_2$| i=1,2,……,n} which map the plane R_2 to itself. The number of contractive transformations defines a map T(.)=$\bigcup_{i=1}^{n} T(.)$. One important fact is that if the T_i are contractive in the plane, then T is a contractive in a space of subsets of the plane. Again, if a contractive map T is given on a space of images, then there is a special image called the attractor A_T, with three properties. Firstly, if T applied to the A_T, the output is equal to the input and A_T is called the fixed point of T. That is , $T(A_T)=A_T=T_1(A_T)UT_2(A_T)U\ldots\ldots\ldots UT_n(A_T)$. Second one is that for an input image M_0, $M_1=T(M_0)$, $M_2=T(M_1)=T(T(M_0))=T^{02}(M_0)$ and so on are satisfied. The attractor does not depend on the choice of initial input image M0 and is given by $A_T=\lim_{n\to\infty} T^{0n}(M_0)$. And the third one is that A_T is unique. These three properties are known as the contractive Mapping Fixed-Point Theorem. But, using IFS the significant compression ratios could be obtained only on images that are specially constructed with someone guiding the compression process. It is not possible to completely automate the compression process. Thus, image compression using IFS is not practical. The problem is solved with the development of the theory of Partitioned Iterated Function Systems (PIFS) [16, 17] where the image is partitioned into non-overlapping ranges, and a local IFS for each range is found. PIFS makes the problem of automating the compression process into a manageable task completely. PIFS are also termed as Local Iterated Function Systems (LIFS). In fractal compression with PIFS, the input image is used as its own dictionary during encoding. The compression technique first partitions the input image into a set of non-overlapping ranges. There are many possible partitioning scheme [16] that can be used to select the range R_i like quardtree, HV and triangular partition. The process of partitioning image repeats recursively starting from the whole image and continuing until the partitions are covered with a specified tolerance or the partitions are small enough. For each range, the compression technique searches for a part of the input image called a domain, which is similar to the range. The domain must be larger in size than the range to ensure that the mapping from the domain to the range is contractive in the spatial dimensions. To find the similarity between a domain D and a range R, the compression technique finds the best possible mapping T from the domain to the range, so that the image T(D) is as close as possible to the image R. The affine transformations are applied for this purpose. The best affine map from a range to a domain is found. This is done by minimizing the dissimilarity between range and mapped domain as a function of the contrast and brightness. The RMS metric or any other metric can be used to find the dissimilarity between range and mapped domain. The ways of partitioning image in this technique is one of the important factor determining the compression rate, compression time and quality of the image. One way to partition the image is quardtree partitioning scheme [17, 18] where the image is broken up into four equal-sized sub-images. The compression technique repeats recursively starting from the whole image and continuing until the squares are either covered with in some specified RMS tolerance or smaller than the specified minimum range size. But, there are some weakness of the quardtree based partitioning. It makes no attempt to partition the image in a image-context dependent way such that partitions share some self-similar structures. A way to remedy this problem i.e. an adaptive quardtree partitioning scheme is discussed [1]. The variable position of the partition based on context of image makes the partitioning scheme more flexible. The partitioning of the image is done in such a way that they share some self-similar structure. The detail of the partitioning scheme is discussed in section 2. Then, a fractal image compression technique is discussed based on the proposed adaptive quardtree partitioning scheme and termed as Fractal Image Compression with Adaptive Quardtree Partitioning (FICAQP) [1] . The compression technique is discussed in section 3. Results are given in section 4 and conclusions are done in section 5.

2. THE ADAPTIVE QUARDTREE PARTITIONING SCHEME

Let us consider, a RxC range have the pixel values $P_{i,j}$ for $0 \le i < R$ and $0 \le j < C$, where R and C are the number of rows and columns of the range respectively and the top-left point of the rectangular range is (x,y) as shown in Fig. 1. The partitioning scheme has three step. In the first step, the range of the image is partitioned horizontally. In order to partition the range horizontally, pixel value sum $\sum_{j=y}^{y+C-1} P_{i,j}$ is calculated for each pixel raw i, $x \le i \le x+R-1$. These sums are used to compute absolute value of successive differences between pixel value sum of ith and (i+1)th row i.e. $\left(\sum_{j=y}^{y+C-1} P_{i,j} - \sum_{j=y}^{y+C-1} P_{i+1,j} \right)$ for each pixel raw i, $x \le i \le x+R-1$. Then, a linear biasing function min(i, R - i - 1) is applied to each of these differences to multiply them by their distance from the nearest side of the rectangular range. This gives the biased horizontal differences between ith and (i+1)th row,

$$h_i = \min (i-x, R-i+x-1)\{|(\sum_{j=y}^{y+C-1} p_{i,j} - \sum_{j=y}^{y+C-1} P_{i+1,j})|\}, \quad (1)$$

for each pixel row i, $x \le i \le x+R-1$.

Then, the value of i (say Rp) is found such that h_i is maximal that is mathematically,

$$Rp = \text{Maximize } [\min (i-x, R-i+x-1)\{|(\sum_{j=y}^{y+C-1} p_{i,j} - \sum_{j=y}^{y+C-1} P_{i+1,j})|\}] \quad (2)$$

Subject to $x \le i \le x+R-1$

The partition is made by the row number Rp to produce upper sub-ranges REC1 and lower sub-range REC2 shown in Fig.2. In the second step, the sub-range REC1 is partitioned vertically. In order to partition the sub-range REC1 vertically, sum of pixel values $\sum_{i=x}^{Rp-1} P_{i,j}$ is calculated for each column j, $y \le j \le y+C-1$ of REC1. Similarly, the absolute value of successive differences between sum of pixel values of jth and (j+1)th columns i.e. $\left(\sum_{i=x}^{Rp-1} P_{i,j} - \sum_{i=x}^{Rp-1} P_{i+1,j} \right)$ are computed for each pixel column j, $y \le j \le y+C-1$ and the liner biasing function min(j-y, C-j+y-1) is applied to each of these differences to calculate the biased vertical differences between jth and (j+1)th columns,

$$v1j = \min (j-y, C-j+y-1) \{|(\sum_{i=x}^{Rp-1} P_{i,j} - \sum_{i=x}^{Rp-1} P_{i,j+1})|\}, \quad (3)$$

for each pixel column j, $y \le j \le y+C-1$.

Then, the value of j (say Cp1) is found such that v_{1j} is maximal that is mathematically,

$$Cp1 = \text{Maximize } [\min (j-y, C-j+y-1) \{|(\sum_{i=x}^{Rp-1} P_{i,j} - \sum_{i=x}^{Rp-1} P_{i,j+1})|\}] \quad (4)$$

Subject to $y \le j \le y+C-1$.

The partition at vertical position Cp1 of REC1 is made to create two sub-ranges REC1.1 and REC1.2 as shown in Fig.3. Similarly, in the third step, the sub-range REC2 is partitioned vertically. The pixel value sum $\sum_{i=Rp}^{R-1} P_{i,j}$ is calculated for each column j, $y \le j \le y+C-1$ of REC2 that are used to compute absolute value of successive differences $\left(\sum_{i=Rp}^{x+R-1} P_{i,j} - \sum_{i=Rp}^{x+R-1} P_{i,j+1} \right)$ for each pixel column j, $y \le j \le y+C-1$. Similarly, biased vertical differences are calculated as

$$v_{2i} = \min (j-y, C-j+y-1)\{|(\sum_{i=Rp}^{x+R-1} P_{i,j} - \sum_{i=Rp}^{x+R-1} P_{i,j+1})|\}, \quad (5)$$

for each pixel column j, y≤j≤y+C-1.

The value of j (say Cp2) is found such that v_{2j} is maximal that is mathematically,

$$Cp2 = \text{Maximize } [\min (j-y, C-j+y-1) \{ |(\sum_{i=Rp}^{x+R-1} P_{i,j} - \sum_{i=Rp}^{x+R-1} P_{i,j+1})|\}] \quad (6)$$

Subject to y≤j≤y+C-1

The partition at vertical position Cp2 of REC2 is made to create two sub-ranges REC2.1 and REC2.2 as shown in Fig.4. Finally, Rp x Cp1 sub-range REC1.1, Rp x (C-Cp1) sub-range REC1.2, (R-Rp) x Cp2 sub-range REC2.1 and (R-Rp) x (C-Cp2) sub-range REC2.2 are formed with their top-left points (x,y), (x,Cp1), (Rp,y) and (Rp,Cp2) respectively. The algorithm of the proposed adaptive quardtree partitioning scheme is given in section 2.1.

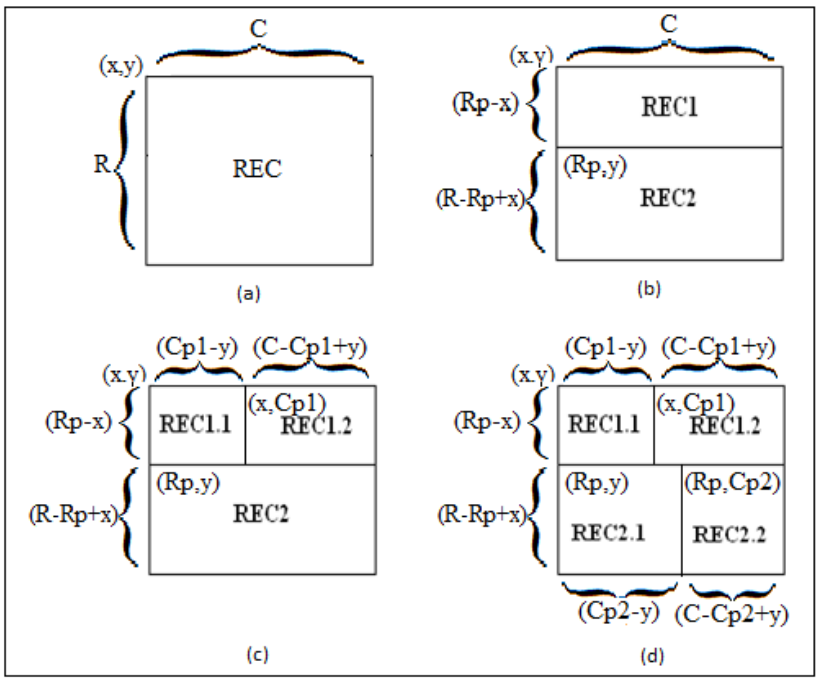

Figure1. (a) initial image, (b) after 1st step, (c) after 2nd step, (d) after 3rd step.

2.1 The algorithm of adaptive quardtree partitioning scheme

Let us consider a RXC range, REC with top-left point (x,y) where R and C are the number of rows and columns of the REC respectively. The detail algorithm of proposed adaptive quardtree partitioning scheme is shown in Fig. 2. At the beginning of the algorithm, the horizontal partitioning row number Rp is found by calculating biased absolute value of successive differences for each row and determining the row that maximize the same. Then, this RP is used to partition the REC horizontally into two rectangles REC1 and REC2 with top-left points (x,y) and (Rp,y) respectively. After that, the vertical partitioning column number Cp1 of range REC1 is found by determining the column that maximizes the biased absolute value of successive differences for each column of REC1. Then, the range REC1 is partitioned vertically into two rectangles REC1.1 and REC1.2 with top-left points (x,y) and (x,Cp1) respectively. Similarly, the range REC2 is partitioned into two rectangle REC2.1 and REC2.2 with top-left points (Rp,y) and (Rp,Cp2) respectively.

Step 1: Initially, set Hmax = Vmax1 = Vmax2 = 0;
Step 2: **/*Determine the horizontal partitioning row number Rp */**
For i = x to x+R-1, do
X1=X2=0;
For j = y to y+C-1, do
X1 = X1 + P[i][j]; X2 = X2 + P[i+1][j];
End for;
H = min (i-x,R-i+x-1) * ABS (X1 – X2);
If H > Hmax, then Hmax = H; Rp = i; End if;
End for;
Step 3: **T**he range REC is divided horizontally into two rectangles REC1 and REC2
using row number Rp with top-left points (x,y) and (Rp,y) respectively;
Step 4: **/*Determine the vertical partitioning column number Cp1 of range REC1 */**
For j = y to y+C-1, do
Y1 = Y2 = Y3 = Y4 = 0;
For i = x to Rp-1, do
Y1 = Y1 + P[i][j]; Y2 = Y2 + P[i][j+1];
End for;
V1 = min (j-y,C-j+y-1) * ABS (Y1 – Y2);
If V1 > Vmax1, then Vmax1 = V1; Cp1 = j; End if;
End for;
Step 5: The range REC1 is divided vertically into two rectangles REC1.1 and REC1.2 using column number Cp1 with top-left points (x,y) and (x,Cp1) respectively.
Step 6: **/*Determine the vertical partitioning column number Cp2 of range REC2 */**
For j = y to y+C-1, do
For i = Rp to x+R-1, do
Y3 = Y3 + P[i][j]; Y4 = Y4 + P[i+1][j];
End for;
V2 = min (j-y,C-j+y-1) * ABS (Y3 – Y4);
If V2 > Vmax2, then Vmax2 = V2; Cp2 = j; End if;
End for;
Step 7: The range REC2 is divided vertically into two rectangles REC2.1 and REC2.2 using column number Cp2 with top-left points (Rp,y) and (Rp,Cp2) respectively.
Step 8: Return four sub-ranges i.e. REC1.1, REC1.2, REC2.1 and REC2.2 with their corresponding top-left points (x,y), (x,Cp1), (Rp,y) and (Rp,Cp2) respectively.
Step 9: Stop.

3. THE FRACTAL IMAGE COMPRESSION WITH ADAPTIVE QUARDTREE PARTITIONING SCHEME

The image compression technique is quite similar to the existing fractal image compression technique except the used partitioning scheme. The detail algorithm of technique is shown in Fig. 3. Initially, a grey scale rectangular image F is taken as input for compression and a tolerance

level Q (Quality) is set where encoding with lower value of Q will have better fidelity. The minimum range size R_{min} is also set. A range with size R_{min} is mapped even if there is no such domain that covers the range with the specified tolerance level Q. Initially, the whole image is the initial range R_1 that is marked as uncovered. Then, another parameter i.e. domain density is also set that determines domain step (distance between two consecutive domains) and a domain pool is created where each domain must be larger than the ranges to ensure contractive mapping from domain to range in spatial dimension. For a density value of zero, the domains start on a boundary multiple of their size, thus the domains do not overlap. The domain step is divided by 2 for a domain density of 1and by 4 for a domain density of 2.The domain-range comparisons are computationally very intensive task. Therefore, classification scheme is used to minimize the number of comparisons. After creation of domain pool, it is classified. During encoding, each range is classified in similar way and compared with the domains belonging to the same class. In this way, a significant number of domain-range comparisons are reduced. There are several methods of such classification. Here, one simple way is used. A domain or range is divided into four quadrants (upper left, upper right, lower left and lower right). The brightness of each quadrant is calculated as Qi $= \sum_{k=0}^{\infty} Pk^i$, if the pixel values in quadrant i are $P_1^i, P_2^i \ldots\ldots P_n^i$ for i=1 to 4. The class of a domain or range is determined by the ordering of the brightness of the four quadrants. There are 24 ordering of the brightness of four quadrants, hence 24 classes. For each uncovered range R_i (starting with initial uncovered range R_1), the domain D_i is found from the similar class that covers R_i better i.e. that minimizes the expression $D_{rms}(F \cap (R_i \text{ X } I), T_i(F))$,where I means interval [0,1] and RMS(root mean squre) metric

$$D_{rms}(F1,F2)=\sqrt{\int (F1(x,y) - F2(x,y))dxdy}. \qquad (7)$$

and the corresponding affine map T_i is also found. If a domain is found that covers the range R_i within the specified value Q or the size of the range is less than or equal to R_{min}, the range R_i is marked as covered and the corresponding affine transformation Ti is written out to the output file. Otherwise, the range R_i is partitioned into four sub-ranges R_{i1}, R_{i2}, R_{i3} and R_{i4} using proposed adaptive quardtree partitioning scheme. Produced sub-ranges are marked as uncovered. This process is continued until all the ranges are covered. The above compression process creates a list of affine maps. To decode the compressed image, the Contractive Mapping theorem is applied. Starting from an arbitrary image, for example a completely black image, the decoding process can apply the Partition Iterated Function System (PIFS) iteratively. This process converges to the fixed point of the PIFS. To perform a iteration of the PIFS, the decoding process takes the list of all affine maps and applies each one in turn which transforms a set of domains into a set of ranges to create a new complete image. The decoding process is repeated until convergence is achieved. The number of iterations and the scale factor are taken as input during decompression. The scale factor determines the size of the decompressed image with respect to input image.

Step 1: Input a gray scale rectangular image F and a tolerance level quality (Q) where encoding with lower Q will have better fidelity.
Step 2: Select the minimum range size R_{min} and mark the initial range R_1 as uncovered.
Step 3: Choose domain density value and construct a domain pool D such that each domain must be larger than the range.
Step 4: All the domain are classified by the ordering of the image brightness in the four quadrants of the domain.
Step 5: While (There are uncovered ranges R_i), do
Step 5.1: Determine the class C_i of range R_i by the ordering of the image brightness in the four quadrants of the range.
Step 5.1: Determine the domain Di from domains of similar class C_i of the domain pool D and the corresponding affine map Ti that cover range R_i better than other i.e. that minimize the expression $D_{rms}(F \cap (R_i \times I), T_i(F))$,where I means interval [0,1] and RMS(root mean squre) metric

$$D_{rms}(F1,F2)=\sqrt{\int (F1(x,y) - F2(x,y))^2 dxdy}.$$

Step 5.2: If (($D_{rms}(F \cap (R_i \times I), T_i(F)) < Q$) OR ($SIZE(R_i) <= R_{min}$)),then
Step 5.2.1: Range R_i is marked as covered and the affine transformation T_i is written out to the output file.
Else
Step 5.2.2: Divide the range Ri into four sub- ranges R_{i1}, R_{i2}, R_{i3}, R_{i4} using Adaptive Quardtree Partitioning scheme. Mark Produced sub-ranges as uncovered.
End if;
End while;
Step 6: Stop.

Figure 3. Algorithm of fractal image compression using PIFS with Adaptive Quardtree Partitioning scheme.

4. RESULTS

The comparison of compression ratios, compression times and PSNRs among fractal image compression technique with quardtree partitioning scheme (FICQP) and the fractal image compression technique with adaptive quardtree partitioning scheme (FICAQP) have been made in table I using five gray scale images having size 64000 Bytes (200 X 320) where all images have been compressed with a domain density value of 1 and image quality 1. The grey-scale files have a file suffix of "GS," which identifies them as "non-formatted" grey-scale files. The graphical representations of the comparison of compression ratios and compression times of all five images are shown in Fig. 4 and Fig. 5 respectively. The proposed technique offers comparatively better rate of compression than its counterpart for all five images. But, the compression times taken by the proposed technique are much more than existing technique for all five images. The graphical representations of the comparison of PSNRs among both the techniques are shown in Fig. 6 for a particular value of quality and domain density. The PSNR between original image and image after compression & decompression cycle in proposed technique are improved significantly than the existing technique for all five images. The decompression times of both the techniques are almost same for all five images (close to 10 m. seconds for 64000 Bytes images). The effect on

compression ratio, compression time and PSNR for increasing the quality value Q in both FICQP and FICAQP for a particular domain density (i.e. D=1) and image (MOUSE.GS) have been given in Table II and the graphical representations of the same are shown in Fig. 7, Fig. 8 and Fig. 9 respectively. The compression ratios increase and the compression times decrease with the increasing value of quality Q in both the techniques. But, the PSNRs are reduced. The decompressed MOUSE.GS images that are compressed with quality values 1, 5 and 15 are shown in Fig. 10 ensures that encoding with lower Q will have better fidelity The visual inspections of MOUSE.GS image after 1st , 2nd, 3rd , 4th and 8th iterations during decompression are shown in Fig. 11 where an arbitrary image is taken to start the decompression process. After completion of each iteration, image is improved. The visual inspections of MOUSE.GS image after compression decompression cycle maintaining same compression ratio are done using both techniques. The original MOUSE.GS image is shown in Fig. 12 (a). The Fig. 12 (b) and Fig. 12 (c) show the images after compression decompression cycle of MOUSE.GS using existing and proposed compression techniques. Both the techniques blur slightly the original image. But, there is no such significant visual difference between two images. The resolution independence is a very useful feature of the fractal compression technique. MOUSE.GS decompressed with scale factor 4 by the proposed technique and original MOUSE.GS enlarged 4 times are shown in the Fig. 13 and Fig. 14 respectively. The image obtained through fractal decompression is much more natural-looking than enlarged original by generating artificial detail that is not present in the original image.

Table1. Comparison of compression ratios, times and PSNRs

File name	FICQP %comp	FICQP Time (m.sec)	FICQP PSNR (dB)	FICAQP %comp	FICAQP Time (m.sec)	FICAQP PSNR (dB)
CHEETAH.GS	82.56	140	27.50	82.76	310	27.93
LISAW.GS	82.59	170	38.91	82.91	360	41.24
ROSE.GS	87.45	130	28.97	87.50	290	29.25
MOUSE.GS	85.87	150	38.03	85.93	340	38.67
CLOWN.GS	82.82	190	29.20	83.13	420	29.72

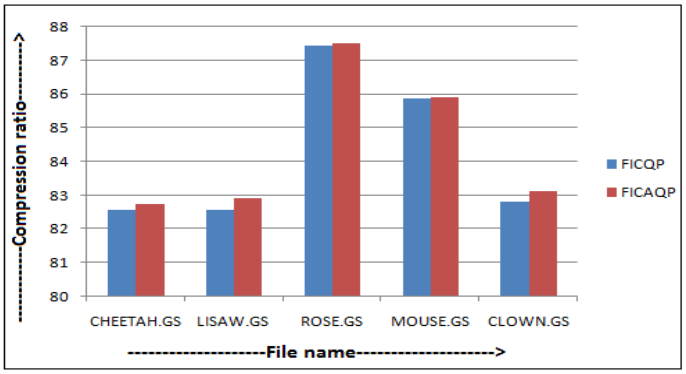

Figure 4. Comparison of compression ratio.

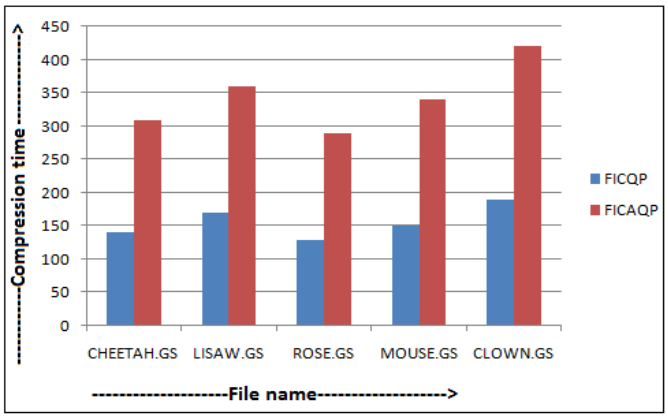

Figure 5. Comparison of compression time.

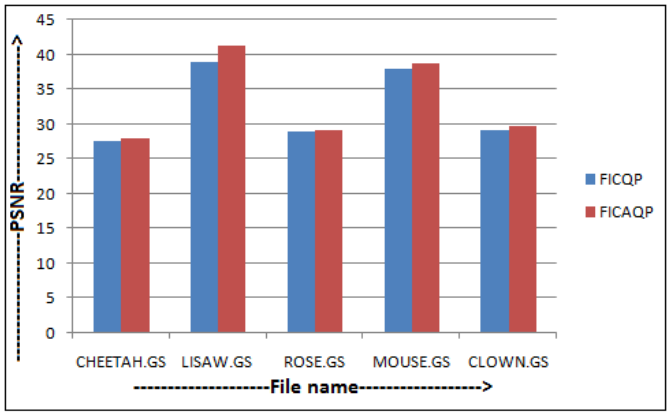

Figure 6. Comparison of PSNR

Table 2. Comparison of compression ratios, times and PSNRs with varying quality values.

Quality Factor(Q)	FICQP %comp	FICQP Time (m.sec)	FICQP PSNR (dB)	FICAQP %comp	FICAQP Time (m.sec)	FICAQP PSNR (dB)
1	85.87	150	38.03	85.93	340	38.67
3	92.06	90	37.66	92.19	200	37.94
5	94.37	50	36.86	94.48	130	36.93
7	96.10	40	35.71	96.36	110	35.85
10	97.45	20	34.08	97.31	60	34.13
12	97.96	20	33.07	97.95	40	33.15
15	98.30	20	32.20	98.44	50	32.21

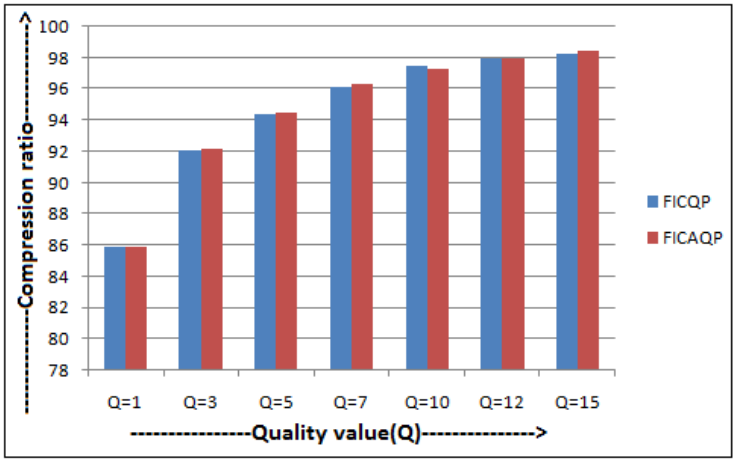

Figure 7. Quality value vs. compression ratio.

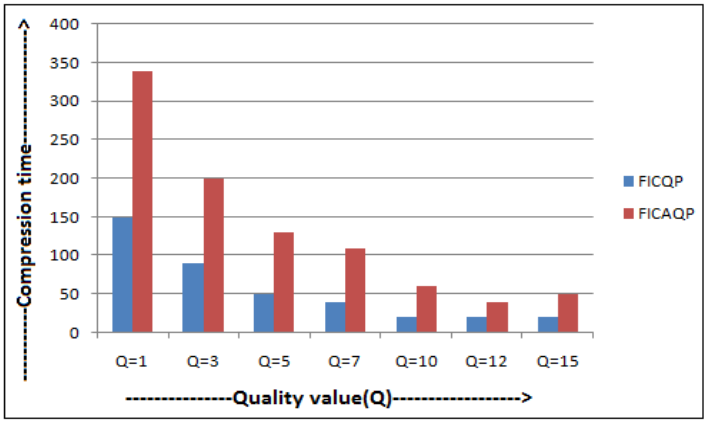

Figure 8. Quality value vs. Compression time.

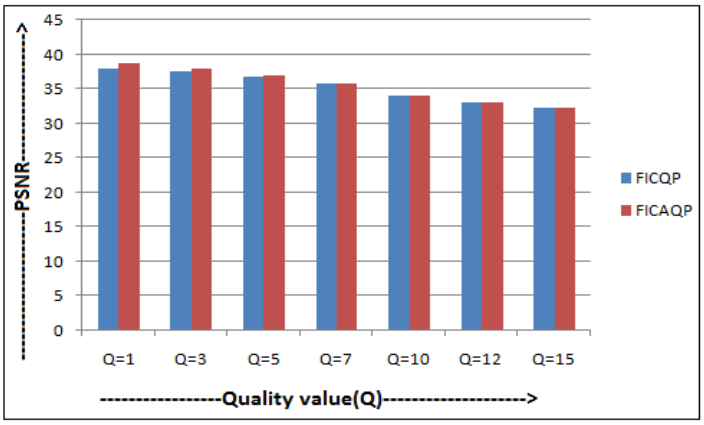

Figure 9. Quality value vs. PSNR

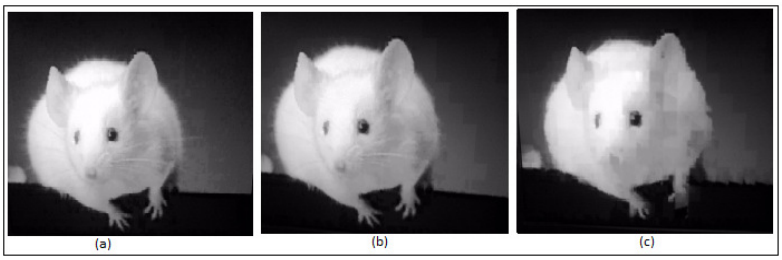

Figure 10. (a) quality 1 (b) quality 5 (c) quality 15

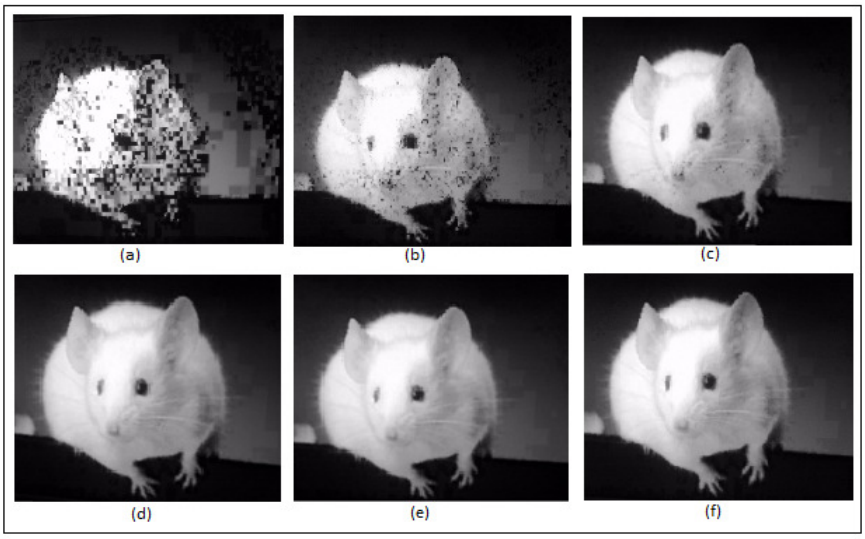

Figure 11. (a)1st iteration (b)2nd iteration (c)3rd iteration (d)4th iteration (e)5th iteration (f) 8th iteration.

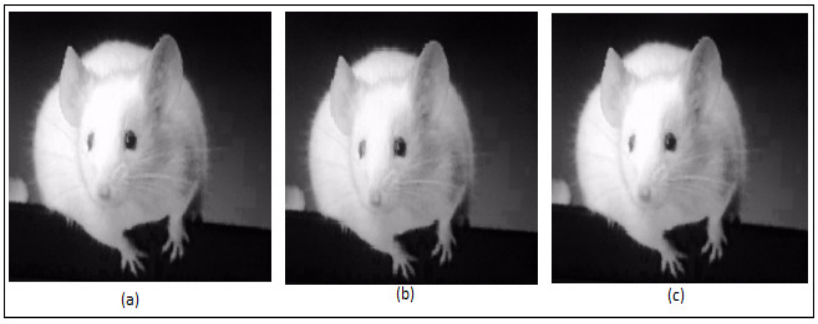

Figure 12. (a) Original image (b)FICQP (c) FICAQP.

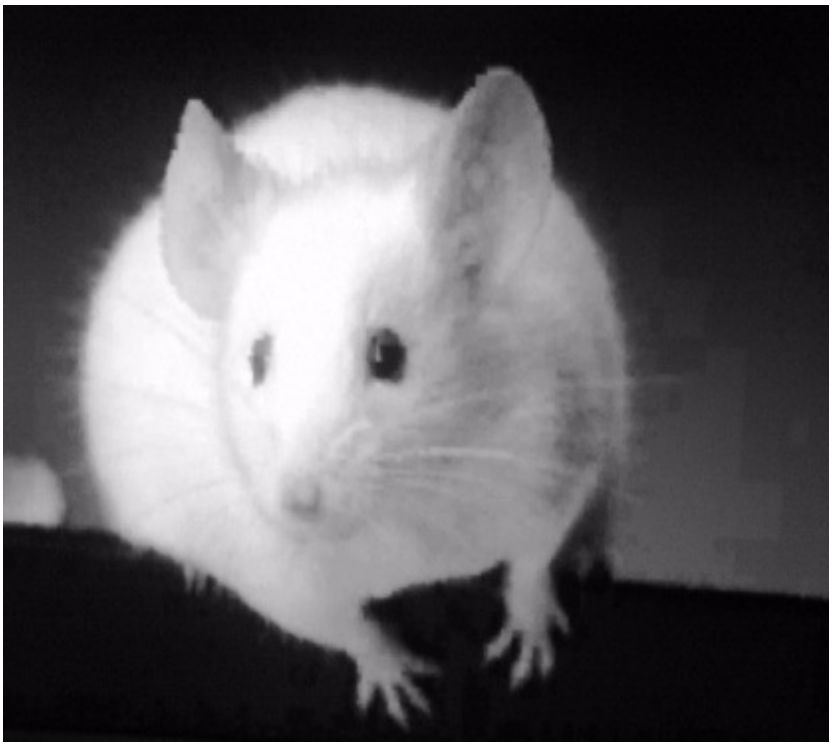

Figure 13. MOUSE.GS decompressed with scale factor 4 by proposed technique.

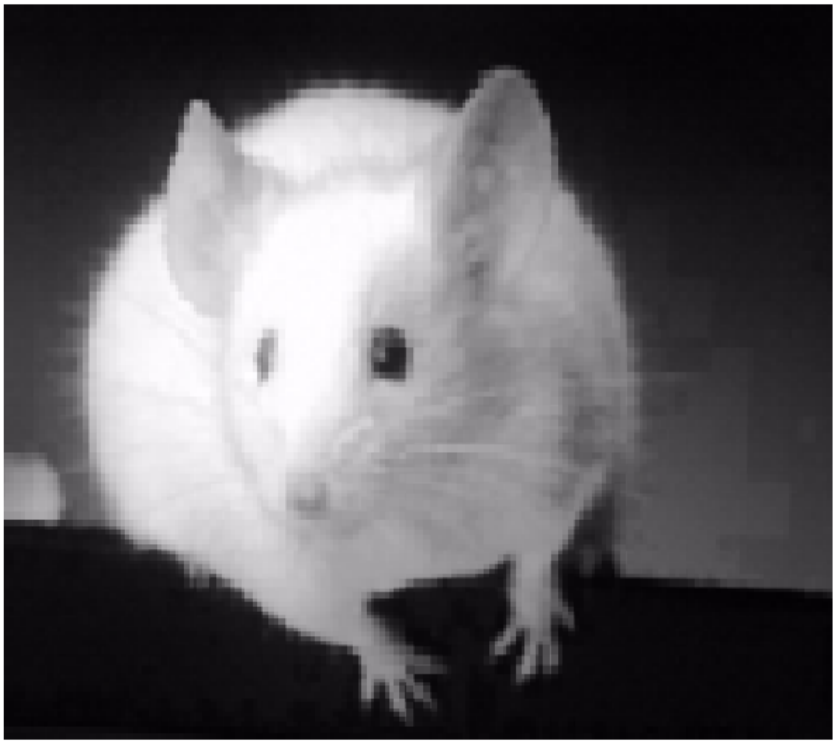

Figure 14. Original MOUSE.GS enlarged 4 times.

5. CONCLUSIONS

In adaptive quardtree partitioning scheme, the partitioning points are selected adaptively in a context-dependent way such that some self-similar structure are shared. Therefore, the fractal image compression with adaptive quardtree partitioning scheme is more efficient than the quardtree partitioning scheme in terms of compression ratio and PSNR. But, the compression time is much more in fractal image compression technique with adaptive quardtree partition than the fractal image compression technique with quardtree partition. Farther investigation is needed to reduce the compression time of the technique without loss of PSNR.

ACKNOWLEDGEMENTS

The authors extend sincere thanks to the department of Computer Science and Engineering and PURSE Scheme of DST, Govt. of India and Academy Of Technology, Hooghly, West Bengal, India for using the infrastructure facilities for developing the technique.

REFERENCES

[1] Nandi, U. & Mandal, J. K., (2012) "Fractal image compression with adaptive quardtree partioning", International conference on signal, image processing and patter recognization (SIPP- 2013),vol. 3, no. 6, pp. 289–296, 2013,Chennai, India.
[2] Fisher, Y.(1995) "Fractal Image Compression: Theory and Application", Springer Verlag, New York
[3] Cardinal, J,(2001) "Fast Fractal Compression of Greyscale Images", IEEE transactions on image processing, vol. 10, no. 1.
[4] Barnsley, M.F.(1993) " Fractals Everywhere", Academic Press,2nd edition, San Diego.
[5] Jacquin, A. (1990) " Fractal image coding based on a theory of iterated contractive image transformations" , Proc. SPIE Visual Communications and Image Processing, pages 227-239.
[6] Belloulata, K.., Stasinski, R., Konrad, J.,(1999) " Region based Image compression using fractals and Shape adaptive DCT ", in proceeding of IEEE international conference on image processing,1999, icip99, October 24-28, vol.2, ISBN : 0-7803-5467-2, pp. 815-819 (1999).
[7] Mandelbrot, B.B.(1983) "The Fractal Geometry of Nature ", W.H. Freeman and company, New York.
[8] Hutchinson, J.E., (1981) " Fractals and self-similarity ", Indiana University Mathematics Journal, Volume 3, Number 5, pages 713-747.
[9] Demko,S., Hodges, L., Naylor, B., (1985) "Construction of Fractal Objects with Iterated Function Systems", Computer Graphics 19(3):271–278.
[10] Saupe, D., (1995) " Accelerating fractal image compression by multi-dimensional nearest-neighbor search", in Proc. DCC'95.
[11] Kominek, J., (1995) "Algorithm for fast fractal image compression", in Proc. IS&T/SPIE 1995 Symp. Electronic Imaging: Science Technology, vol.2419.
[12] Wohlberg, B. E., G. de Jager, (1995) "Fast image domain fractal compression by DCT domain block matching : Electron. Lett., vol. 31, pp. 869–870.
[13] Lu, N., (1997) " Fractal Imaging", New York: Academic.
[14] Wallace, Gregory K., (1999) "The JPEG Still Picture Compression Standard ", Communications Of the ACM, Volume 34, No. 4, pp 31-44.
[15] Pennebaker, William B., Joan L. Mitchell, L.(1992) "JPEG Still Image Data Compression Standard", New York, Van Nostrand Reinhold.
[16] Salomon , D., (2002) "Data Compression, The complete reference ", ed. Third , Springer,.
[17] Fisher , Y. (2008) " Fractal Image Compression, Theory and application ", ed. Second , Springer.
[18] Nelson , M., (2008) "The Data Compression Book" ,ed. Second , India, BPB Publications.

Authors

Joytsna Kumar Mandal

M.Tech.(Computer Science, University of Calcutta), Ph.D. (Engg., Jadavpur University), Professor in Computer Science and Engineering, University of Kalyani, Nadia, West Bengal, India. Life Member of Computer Society of India since 1992. 25 years of teaching and research experiences. 8 Scholars awarded Ph.D.; 1 Scholars submitted Ph.D. and 7 scholars are pursuing Ph.D. Total number of publications 228.

Utpal Nandi

M.Sc.(Computer Science, Vidyasagar University),M.Tech.(Computer Science & Enggineering) from University of Kalyani, Nadia, West Bengal, India. Assistant Professor in Computer Science and Engineering, Academy Of Technology, Hooghly, West Bengal, India. Total number of publications 10.