The Nucleus **49**, No. 1 (2012) 21-27

The Nucleus
A Quarterly Scientific Journal of Pakistan Atomic Energy Commission
NCLEAM, ISSN 0029-5698

QUADTREE FRACTAL COMPRESSION FOR BRAIN MRI IMAGES

*J. MIR and G. RAJA

Department of Electrical Engineering, University of Engineering and Technology, Taxila, Pakistan

(Received November 10, 2011 and accepted in revised form March 06, 2012)

Significant attention has been given recently to medical image compression due to benefits in archival and transmission of medical images. This paper describes the Quadtree fractal scheme for compression of Brain MRI medical images by defining the Region of Interest (ROI). Set of contractive transformations are found by using Partitioned Iterated Function system (PIFS). Fixed value of contrast scaling factor is used to compute the modified error function in implemented method reducing the encoding time. Various simulations are conducted to test the performance of Quadtree fractal compression scheme using different brain MRI medical images. Results are compared with Uniform partitioning fractal compression technique. Simulation results show good quality near lossless images with high compression ratio and acceptable PSNR values well above 30dB.

Keywords: Fractal image compression, PIFS, PSNR, Quadtree partitioning

1. Introduction

Need for data storage and transmission in digital form is increasing tremendously due to computerization of different fields in our daily life. Every digital image like drawings, scanned documents, images from digital or video cameras, satellite images, medical images are to be stored for human needs. Solution to this ever growing storage capacity needs is compression. Each technique achieves suitable compression of digital images by targeting and removing redundancy present. In recent years significant attention has been diverted to medical image compression. Archives and data bases of medical images are formed for keeping the storage of case histories of different patients, growing rapidly on daily basis. Therefore, compression is required to ensure fast interactivity through large sets of images for searching context dependant images and for quantitative analysis of measured data [1]. Since human observers interpret medical images, diagnostic values needs to be preserved correctly in recreated images [2].

Several lossless image compression algorithms like lossless JPEG, JPEG-LS, JPEG 2000, PNG and CALIC are used for the applications of compressing medical images. Comparing all, JPEG-LS are the algorithm with best performance with compression ratio and compression speed [3]. Being lossless, compression ratio is limited in all these schemes. Requirement is to compress medical images at higher compression ratio with image quality preserved for diagnostic needs.

Fractals are basically objects which are formed by transformed copies of themselves or parts of themselves. They have self similarity and can be described by simple recursive definitions. Barnsley was the first person to think of using fractals for image compression [4]. Later on Jacquin published the fractal image compression scheme [5]. Fractal image compression scheme is based on assumption that the images can be treated as fractals and their recursive descriptions could be found. Fractals based compression schemes allows images to be stored by computers using less memory and bandwidth and are computationally simple to decode [6]. This paper describes implementation of fractal compression scheme for medical images by exploiting the self similarity present in the images. Rest of paper is organized as follows: Section 2 briefly reviews the fractal image coding. Section 3 describes the implemented technique. Simulation results are presented in section 4 and conclusions are given in section 5.

* Corresponding author : junaid.mir@uettaxila.edu.pk

2. Fractal Image Compression

Based on mathematical model called IFS (Iterated Function System), fractal compression was first introduced by John Hutchinson and developed by Michael Barnsley. An IFS consists of collection of contractive affine transformations $\left\{w_i : R^2 \rightarrow R^2 \middle| \ i=1,...,n\right\}$ mapping the plane R^2 to itself [7]. Fractal operator W (1) called Hutchinson's operator given in Eq.1 is formed from map which is defined by collection of transformations.

$$W(.)=U_{i=1}^{n} w_i(.) \tag{1}$$

Map is applied to collection of points which are basically the pixels of the image. Contractive transformations w_i are applied on these pixels which bring the point's P_1 and P_2 closer satisfying condition given in Eq.2 and are fed back to IFS. d is the distance between two points.

$$d\left(w(P_2), w(P_1)\right) < sd\left(P_1, P_2\right) \tag{2}$$

The contractivity factor s should always be limited to $|s|<1$ to ensure contractivity of transformations. After some iteration the unequivocal fixed point called attractor X_w defined by operator W is reached from any starting image [8]. Attractor is approximation of the image to be encoded. First fractal compression technique based on partitioned iterated function system (PIFS) was introduced by Jacquin in which partitioned self similarity was achieved [9]. Method was to partition image into non-overlapping range blocks and then finding transformations for most suitable domain blocks. The collage error which is the quantitative distance between the original and decoded image was used for selection of suitable domain blocks from domain pool for each range block. Later on Hierarchical approaches which were image adaptive techniques were introduced. The choice of partitioning scheme has great influence on fidelity of image. The best results are obtained when partitioning of the image adapts to the content of the image. Famous partitioning schemes are Quadtree partitioning and Horizontal-Vertical partitioning techniques. Polygon and Triangular partitioning did not have much attention because of difficulties in one to one pixel mapping. Split and merge approaches also had the drawback of coding complexities and long encoding time.

3. Implementation of Quadtree Fractal Compression

Quadtree partitioning scheme is used to partition the image into range blocks. It partitions image based upon content present [10]. In our implementation, medical image is divided into two regions as shown in Figure 1. Region of Interest (ROI) lies in the center of the image containing useful information which is preserved for correct diagnosis. Rest of the image is black portion present due to diagnostic machines and does not contain any information. Significant portion 'ROI' is divided into blocks according to Quadtree partitioning scheme. Rest of the image, the non significant portion is divided into blocks with maximum permissible block size.

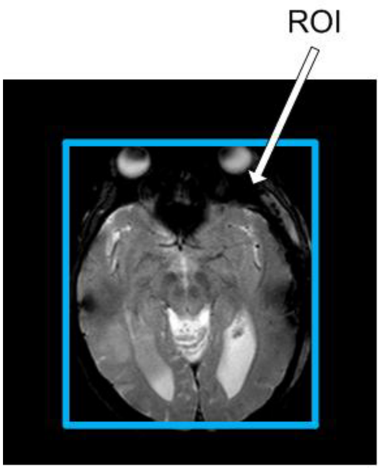


Figure 1. Regions

3.1. Encoding

Region of Interest (ROI) is encoded through Quadtree partitioning scheme. Figure 2 shows the block diagram of encoder. Input module takes the parameters which are to be defined by the user. Input image defines the image to be encoded. Min_depth defines the upper bound of block size and is the minimum block size when image is partitioned into blocks initially. The Max_depth is the lower bound of block size. It is the maximum size which block can achieve after partitioning. TF is the tolerance factor which needs to be set after evaluating image quality.

The function of block formation module is to partition the Region of Interest (ROI) into range blocks of size 32x32 specified by Min_depth parameter in first step and to make virtual domain pools for incoming range block size. A virtual domain pool contains the domain blocks of double the size of range blocks. The domain blocks are transformed by using standard isometric transformations [7] increasing the overall size of the domain pool by 8. These domain blocks are then contracted by taking average of 4 pixels to match the size of range blocks.

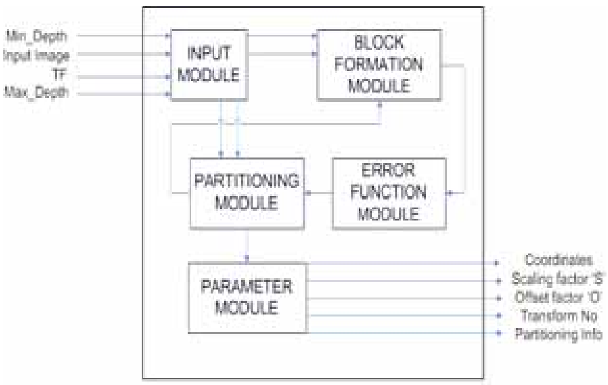


Figure 2. Encoder Block Diagram.

The square Euclidean distance given in Eq.3 is computed usually to find out best matched transformed domain block from domain pool.

$$E(R,D)=\left|\left|R-(s.D_i+oI\right|\right|^2 \quad (3)$$

It requires the pre calculation of contrast scaling factor 'S' and luminance offset factor 'O' and increases the encoding time and coding complexity. We have used fixed value of contrast scaling factor S=1. Error function module computes the error function by transformation given in Eq.4 [11].

$$E(R,D)=\left|\left|(R-r)-s_i.(D_i-d)\right|\right|^2 \quad (4)$$

r and d are average pixel intensities of range block and domain block respectively. For each range block R of the image, domain block D is selected from virtual domain pool which gives minimum value of error function. This gives the best matched domain block for each range block.

Partitioning module takes the decision of partitioning after computing chk value which determines if the range block needs to be partitioned further or not. We compute chk value given in Eq.5 for the range block and its respective selected best matched domain block.

$$\text{chk value}=\sqrt{\sum\left(E(R,D)\right)} \quad (5)$$

We compute chk value which will be zero only if exact match is found for range block or if we encounter a block of Uniform intensity called shade block. Tolerance factor 'TF' is set at zero for implemented technique. It can be varied if reconstructed image quality is not good [12]. If chk value is equal to TF then block is not partitioned otherwise block is partitioned into four sub blocks by partitioning module and procedure of finding best match from domain pool is repeated for each sub block. The maximum size which sub block can achieve is already defined by Max_depth parameter in beginning by user. Upon achieving maximum depth, best matched domain block giving minimum value of error function is selected.

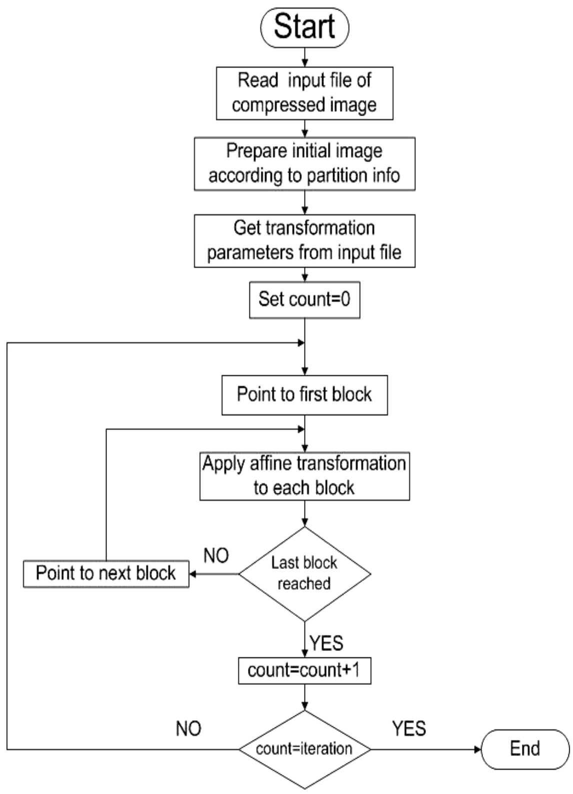


Figure 3. Flow chart for encoding process.

Parameter module stores the affine coefficients after the partitions and selection of domain blocks for each range block. Coordinates of best matched domain block are stored for each range block alongwith the isometry transformation number. Partitioning information is stored for decoder side for image to be partitioned accordingly. Figure 3 shows the flowchart for encoding process. Luminance offset factor is calculated using Eq.6.

$$o=r-s_i d \quad (6)$$

3.2. Decoding

Blank initial image of same size as original image is taken for decoding purpose. Figure 4 shows the block diagram of decoder for implemented Quadtree fractal compression technique. Range blocks are formed of initial image according to the partitioning information obtained from encoder side by block formation module. For each range block best matched domain block is extracted by using coordinates saved. Average of 4 neighboring pixel values is taken to rescale domain block size to match range block size [7].

Intensity calculator module multiplies the domain block with contrast factor and adds the luminance offset factor. These parameters were calculated at decoder side. Transformation module transforms the block according to the isometry transformation number. Block is reflected or rotated by some angle depending upon isometric transformation number. In the end the range block is replaced by this transformed block by image formation module. The number of iterations is 10 as decoded images generally are not improved after 10 iterations [6]. Flowchart for decoding process is shown in Figure 5.

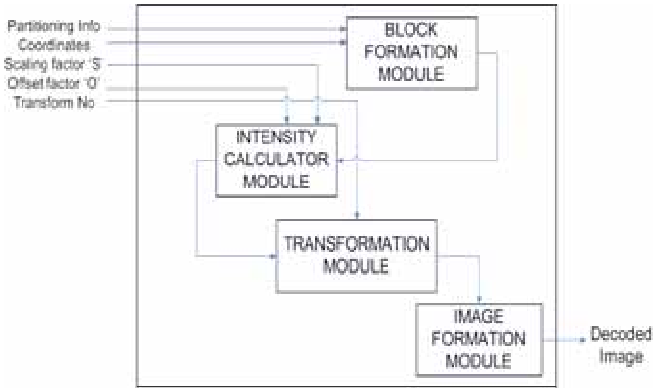


Figure 4. Decoder block

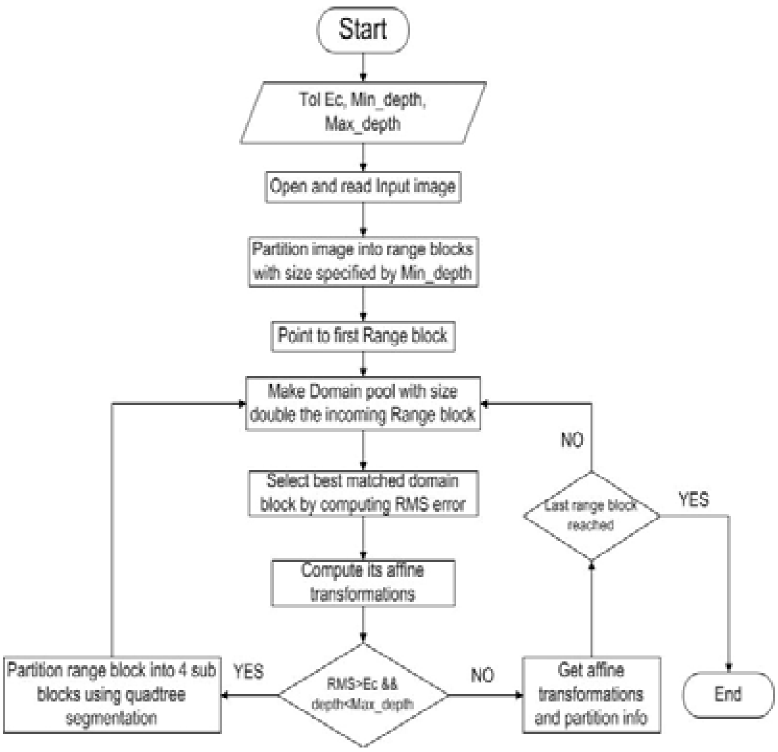


Figure 5. Flowchart for decoding process.

4. Simulation Results

We have taken set of brain MRI 256 x 256 pixel images. Images are taken from standard Digital Imaging and Communication in medicine (DICOM) databank which is standard used for handling, sharing and storing medical images [13]. Images are converted into standard bitmap format by viewing them through ezidicom viewer software. Technique is implemented in MATLAB and all simulations are performed on core i3, 2.40GHz with 4 GB RAM. The largest block size is set at 32x32 and smallest block size is set at 4x4. Tolerance factor (TF) is set at 0. All the images are encoded with contrast scaling factor of S=1. Images encoded with other values of scaling factor gives almost same results as scaling factor is multiplication of pixels with number from 0 to 1 and it does not change pixel position. Its value can be made fixed when error function given in Eq.4 is used. Peak Signal to Noise Ratio (PSNR) is calculated using Eq.7 to compute image quality.

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\mathrm{MSE}} \tag{7}$$

Where MSE is mean square error computed using relation

$$\mathrm{MSE} = \frac{1}{M \times N}\sum_{j=1}^{N}\sum_{i=1}^{M}\left[f(i,j) - g(i,j)\right]^2 \tag{8}$$

Table 1 shows the objective results obtained through implemented Quadtree fractal compression technique. PSNR values and compression ratio achieved are calculated for all the test samples. The results show that decoded image have good PSNR values with high compression ratio. Subjective results are shown in Figures 6 and 7 for visual inspection. Original images are compared with the decoded images obtained through implemented Quadtree fractal compression technique. Decoded images are visually lossless with preserved diagnostic values and good quality. PSNR values are computed for same set of brain MRI images using Uniform partitioning (UP) fractal compression technique for comparison with implemented Quadtree (QT) fractal compression technique. Figure 8 shows the comparison between PSNR values. Average PSNR value using implemented Quadtree technique for 20 brain MRI test images is 33.18dB while for

Table 1. Objective Results. PSNR values and Compression ratio (C.R) of decoded images.

No	Test Image	PSNR	C.R
1	Brain MRI No.01	33.89	8.6:1
2	Brain MRI No.02	35.19	9.1:1
3	Brain MRI No.03	32.91	9.1:1
4	Brain MRI No.04	36.37	9.0:1
5	Brain MRI No.05	34.77	10.5:1
6	Brain MRI No.06	30.31	9.2:1
7	Brain MRI No.07	31.73	9.5:1
8	Brain MRI No.08	29.94	8.7:1
9	Brain MRI No.09	33.10	9.9:1
10	Brain MRI No.10	32.54	9.4:1
11	Brain MRI No.11	31.84	8.8:1
12	Brain MRI No.12	33.98	8.6:1
13	Brain MRI No.13	33.21	8.5:1
14	Brain MRI No.14	34.74	8.5:1
15	Brain MRI No.15	30.52	8.6:1
16	Brain MRI No.16	33.74	8.8:1
17	Brain MRI No.17	32.70	9.7:1
18	Brain MRI No.18	34.07	10.0:1
19	Brain MRI No.19	33.45	9.4:1
20	Brain MRI No.20	34.69	9.2:1

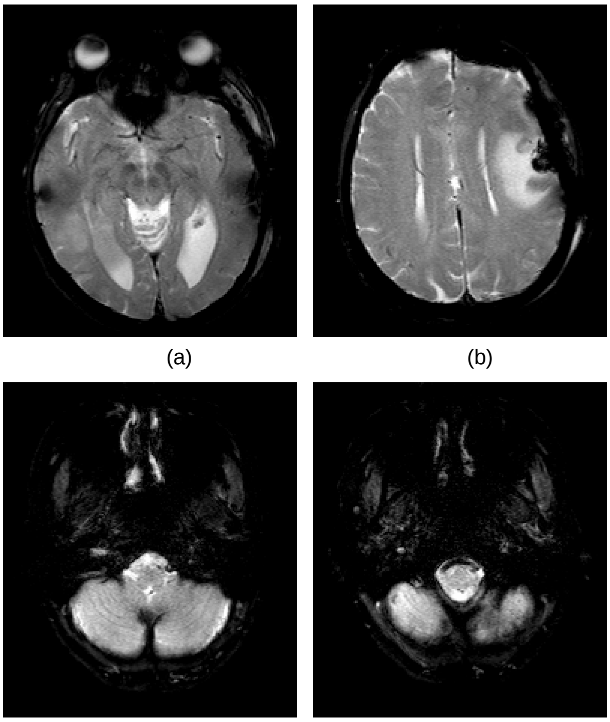
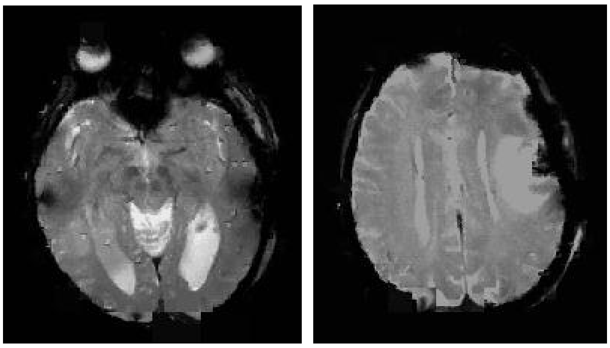
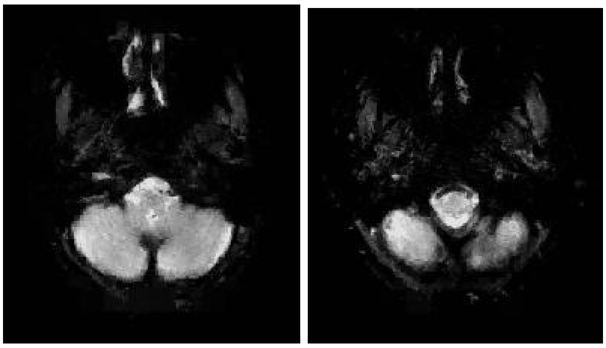


Figure 6. Subjective Results. Original Brain MRI Images.



(a) CR=8.6:1, PSNR=33.89dB (b) CR=9.6:1, PSNR=32.91dB



(c) CR=9.1:1, PSNR=35.19dB (d) CR=9.1:1, PSNR=36.37dB

Figure 7. Subjective Results. Decoded Brain MRI images using implemented Quadtree technique

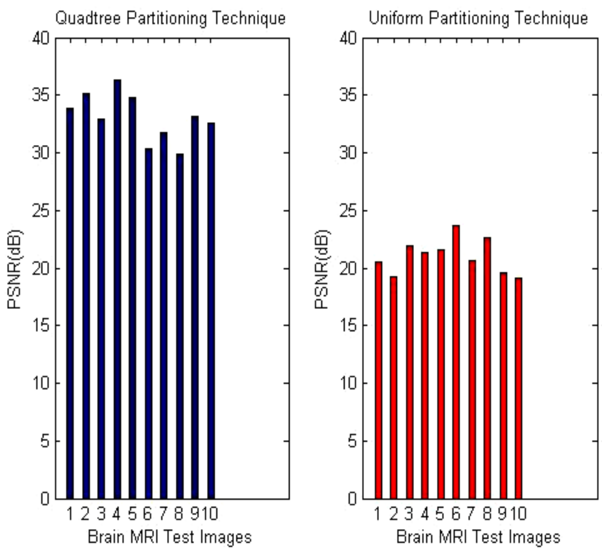


Figure 8. Subjective Comparison of PSNR values of Quadtree and Uniform partitioning technique.

Uniform partitioning technique is 20.57dB. Figure 9 and 10 show the comparative results of PSNR variation against compression and encoding time for Quadtree and Uniform partitioning scheme. These results show that implemented Quadtree fractal compression technique is superior to Uniform partitioning fractal compression technique in terms of compression ratio but lags behind in encoding time. Images decoded with Quadtree technique have high compression ratios at same PSNR than Uniform technique but encoding time is much higher due to coding complexity of Quadtree technique.

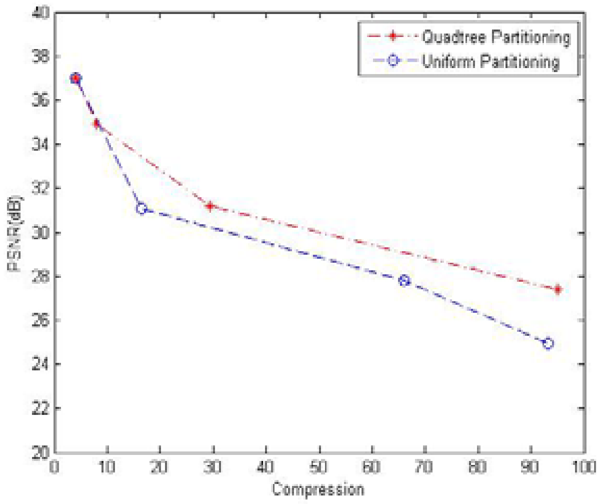


Figure 9. PSNR (dB) vs. Compression

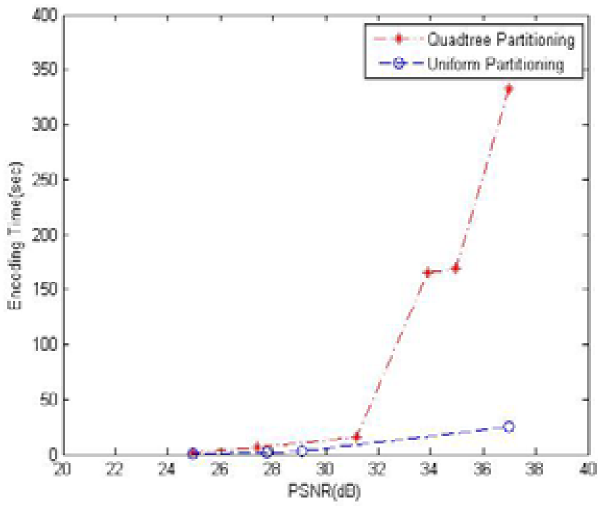


Figure 10. PSNR (dB) vs. Encoding Time.

5. Conclusion

In this paper, we have implemented the Quadtree fractal compression scheme for brain MRI. Results show that Quadtree fractal compression technique can be used to encode medical images and is much superior to Uniform fractal compression technique in terms of compression ratio. The use of ROI in encoding improves overall encoding time and compression ratio. Use of fixed scaling factor ‘S’ does not degrade the image quality. Images decoded using implemented technique with tolerance factor of zero results in visually lossless images with high compression ratio and good PSNRs which can be used for keeping databanks.

References

[1] S. Bhavani and K. Thanushkodi, International Journal on Computer Science and Engineering **2** (2010) 1429.

[2] W. E. Glenn, SMPTE Journal **102** (1993) 392.

[3] M. Ferni Ukrit, A. Umamageswari and G. R. Guresh, International Journal of Computer Applications **8** (2011) 31.

[4] M. F. Barnsley, Fractals Everywhere, Academic Press, New York, USA (1988), ISBN: 0-12-079062-9.

[5] A. E. Jacquin, Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding, Ph.D thesis, Georgia Institute of Technology, USA (1989).

[6] K. M. S. Soyjaudah and I. Jahmeerbacus, International Journal of Electrical Engineering Education **39** (2002) 71.

[7] A. E. Jacquin, IEEE Trans. Image Processing **1** (1992) 18.

[8] J. Kwiatkowiski and W. Walczak, Fractal Magnification of Medical Images, Information Tech. in Biomedicine, ASC 47, Springer-Verlag Berlin Heidelberg (2008) 85.

[9] J. Cardinal, IEEE Trans. on Image Processing **10** (2001) 159.

[10] Y. Fisher, E. W. Jacobs and R. D. Boss, Fractal Image Compression using Iterated Transforms, Image and Text Compression, Kluwer Intl. Series in Engg. and Comp. Sci. **176** (1992) 35.

[11] C. C. Wang, L. C. Lin and S. H. Tsai, Fast Fractal Encoding Algorithm using the Law of Cosines, 47th IEEE Intl. Midwest Symp. on Circuits and Systems **1** (2004) I–233.

[12] Y. Fisher, Fractal image compression – Theory and Applications, Springer-Verlag, New York, USA (1994).

[13] DICOM Test Medical Images available on-line at http://pubimage.hcuge.ch:8080/, accessed March 2011.