International Journal of Electrical and Electronics Research ISSN 2348-6988 (online)
Vol. 8, Issue 3, pp: (1-9), Month: July - September 2020, Available at: www.researchpublish.com

DEVELOPMENT OF FRACTAL IMAGE CODING TECHNIQUE FOR MEDICAL IMAGES

D. V. Shobhana Priscilla*

* Sr. Assistant *Professor, Dept of EIE, VNR Vignana Jyothi Institute of Engineering and Technology,*
Hyderabad.
shobana_dv@vnrvjiet.in

Abstract: **The domain pool design is one of the dominant issues, which affects the coding performance of the fractal compression. In this paper a new method called Block Averaging Method is used to design an effective domain pool. The key feature in this method is that the redundancy between the adjacent domain blocks is reduced. The generated domain pools are much more efficient than those generated by the conventional fractal coding schemes and thus providing improved coding performance. The proposed iteration-free fractal codec method provides high decoding speed and excellent image quality apart from improving the compression ratio and reducing the bit rate. In addition a Hybrid Model is also proposed which will still more reduce the computational time during the coding process.**

Keywords: **Block Average, Domain Pool, Fractal Image Compression, Iteration-Free.**

---

1. INTRODUCTION

The storage and transmission of medical images has made compression a necessity. In order to transmit a large number of medical images with less memory usage, the use of compression techniques have gained very much importance in our day-to-day life. The fractal image coding is different in which the image is made up of copies of properly transformed parts(domain pool) of itself. The reduction in the redundancy between the domain blocks in the domain pool is very important in fractal coding as the domain pool is the storehouse for the fractal codes by which the image can be reconstructed without any error failing which the reconstructed image will be different from the original image.

Arnaud E. Jacquin [1] proposed an independent and novel approach to image coding based on fractal theory of iterated transformation. The main characteristics of this approach are that i) it relies on the assumption that the image redundancy can be efficiently exploited through self-transformability on a block wise basis and ii) It approximates an original Image by a fractal Image. The feasibility of the design of an expert algorithm for the determination of optimal fractal block coding system parameters which would enable the coder to meet specific image fidelity and bit rate requirements for given type of images to encode is to be investigated. More over the coding and decoding phase of this method is still iterative and hence it involves high computational complexity which affects the memory requirement and also increases the computation time significantly.

Hsuan T. Chang et al [6], proposed the Linde, Buzo, and Gray(LBG) method of designing the domain pool in the conventional fractal-coding scheme. Moreover the training process requires iteration to obtain the minimum quantization error. The larger the codebook size, the bigger the iteration number. However it considers the encoding process only and it is not possible to obtain the identical codebook on the decoding phase unless an offline transmission is made. The other option is to apply Linde, Buzo, and Gray (LBG) to generate the domain pool in the decoding phase as well which will further prolong the decoding process.

The first decision to be made when designing a fractal-coding scheme is the choice on the type of image partition used for the Range Blocks. Fisher Y. et al [4], proposed a horizontal-vertical partition which produces a tree-structured partition of the image. In this, each image block is split into two by a horizontal or vertical line. However when partitioning using the image blocks that are not right angled requires some form of interpolation in performing the block transforms when there is no simple pixel to pixel correspondence between the domain and range block, which in turn increases the computational complexity of construction of domain blocks and hence the coding phase.

Novak M. [9] suggested a method of triangular partitioning where by the image is splitted into two main triangles by inserting a suitable diagonal and progressively smaller triangles were placed by three side split in which the existing triangle is split in to two by inserting a line from the vertex of the triangle to a point on the opposite side. Though this method claims to give better performance in terms of improved MSE, the computational complexity and the time taken in the coding phase makes us to rethink on this kind of block partitioning.

The type of block transform selected is a critical element of a fractal-coding scheme, since it determines the convergence property on decoding and its quantized parameter comprises the majority of information in the compressed representation. Barthel K. U. et al [2], proposed a method of performing the transformation in the frequency domain where the transformed block is a function of discrete cosine transforms (DCT). Though the performing DCT in general is going to remove the DC component from the image block, moving in to frequency domain is going to further increase the computational complexity and increases the hardware or memory requirement.

Davoine F. et al [3], suggested the use of non-rectangular blocks for image transforms like an affine mapping of the image support is sufficiently general to transform domain triangles to range triangles in triangular partitioning. These affine transforms are determined by requiring that the transformed vertices of the domain blocks match those of the range blocks. However it is evident form the previous section that the non-rectangular partitioning of image blocks has its own dis-advantages like computational complexity and since we are not going to use non-rectangular partitioning it is highly likely that non-rectangular transformation cannot be used.

The selection on the type of Domain Pool is vital in the case of fractal image compression. Kim K. et al [7], suggested a synthetic code book which has significant variation from that of the conventional fractal coding scheme. In this the domain pool is extracted from a low resolution image approximation which is coded independently rather from the image itself. However the concept of constructing the domain pool independent of the image have chances that not all images can be reproduced perfectly and this might affect the MSE/PSNR.

Hamzaoui R et al [5], suggested a Hybrid Codebook which allows range blocks to be represented either as a mapping from domain blocks or as a fixed VQ code books. Though the performance in terms of quality does seem to be improved, this method is going to further increase the computational cost of maintaining both the data and also affects the bit rate/compression ratio quiet significantly.

Kwon Kim [8], proposed a fractal coding algorithm for still images based on VQ and fractal approximation of residual block. This method employs a static codebook for low-frequency components of input image as well as adaptive codebook by fractal coding for residual signals. However the complexity of computation is higher thereby increasing the memory requirement.

2. ITERATION-FREE CODEC DESIGN

In order to obtain the same domain pool in both the encoder and the decoder without using an offline transmission, an iteration-free fractal image codec is proposed in which the information of the domain blocks are hidden in the fractal code. More over in order to reduce the redundancies between the generated domain blocks, the proposed Block Averaging Method can be applied.

2.1 Encoder

The basic flow chart for the design of the encoder for the proposed iteration-free scheme is shown in figure 1. The input image of size M x M is partitioned in to the so called Range Blocks (R) of size B x B and the Range Blocks are non-overlapping. First we will find the mean and the variance of all Range Blocks. We generate a mean image of size M/B x M/B with each pixel corresponding to the block mean. If the variance of the Range Block

$$\mathrm{Var}\{ R \} = (1 / B^2) \sum_{0 \le i, j < B} (r_{i,j} - \mu_R) \qquad (1)$$

(where $r_{i,j}$ denotes the (i, j)th pixel in the Range Block) is smaller than a threshold value, then the Range Block is coded by mean. Otherwise the Range Block is coded by Affine Transform. For coding with the Affine Transform, the mean image (where each pixel contains the mean value of the Range Blocks), the Domain Pool, the eight values of Contrast Scaling and the eight Isometries are required. The Domain blocks in the Domain Pool is designed by the Block Averaging Method as is described in section 2.2.1. The size of the Domain Block (D) is same as that of the Range Block and thus the contraction procedure in the conventional fractal coding scheme is eliminated. We therefore proceed with the non-contractive affine transformation between the range block and the domain block generated from the mean image.

The non contractive affine transformation can be expressed as

$$\begin{aligned} R_{bar} &= \iota \{ \alpha . D + \mu_R - \alpha . \mu_D \} \\ &= \iota \{ \alpha . (D - \mu_D) + \mu_R \} \qquad (2) \end{aligned}$$

where $\mathbf{R_{bar}}$ is the coded range block, μ_R is the mean of Range Block and μ_D is the mean of the domain block. Now we can see in detail the various parameters used in our non-contractive affine transforms.

Contrast scaling (α) is usually smaller than 1.0 to avoid the divergence caused by the iteration in the conventional fractal coding schemes. However we can make the contrast

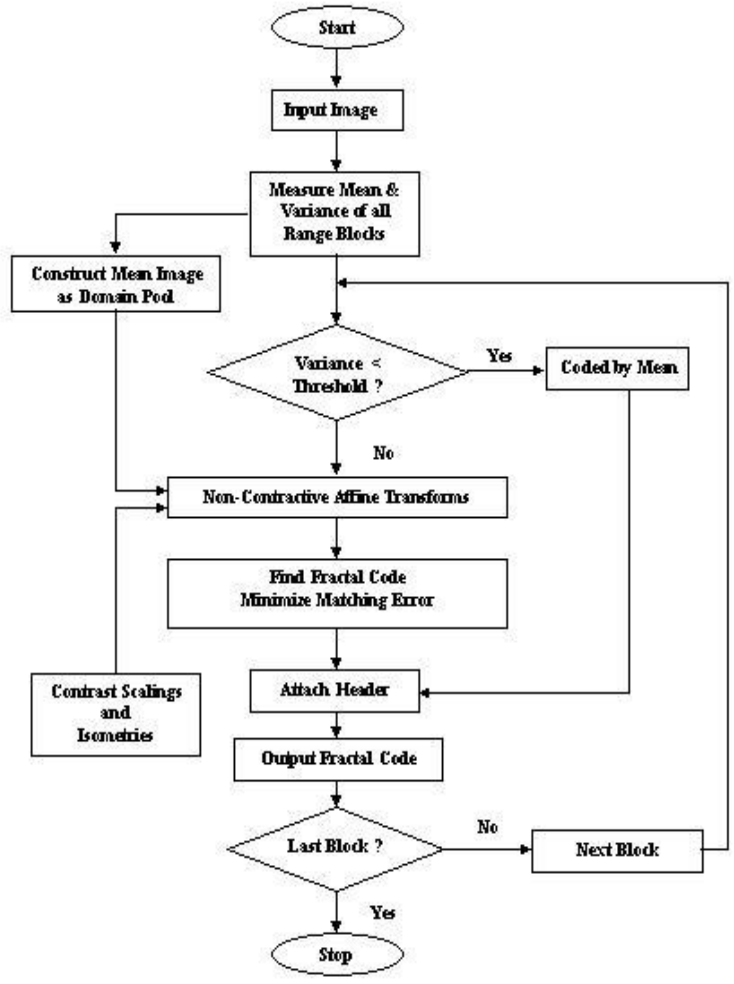

Fig 1. Flowchart of the Encoder for the Iteration-free Scheme

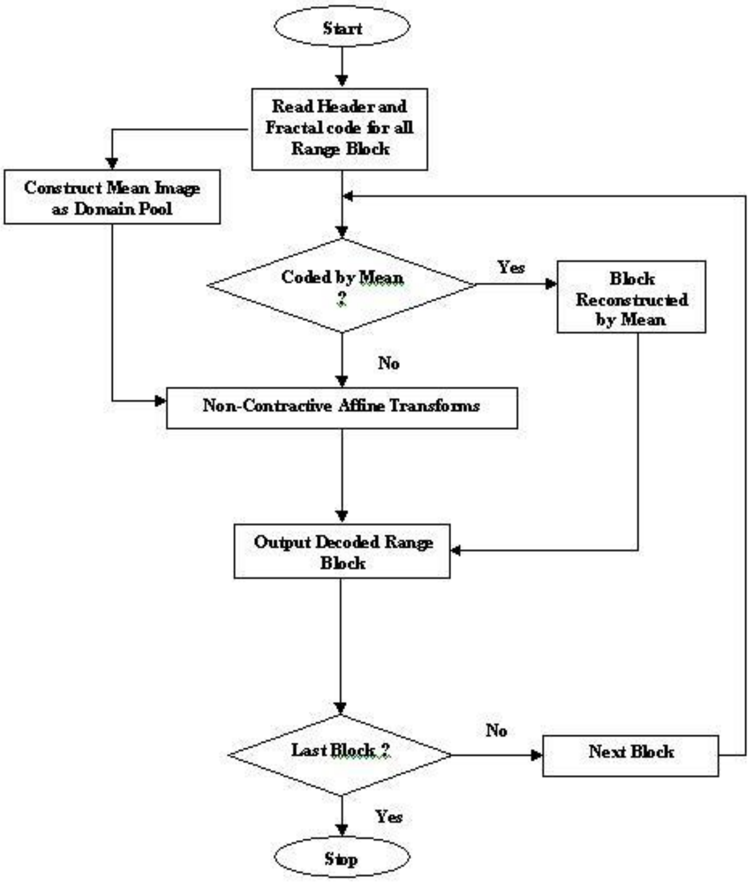

Fig 2. Flowchart of the Decoder for the Iteration-free Scheme

scaling greater than 1.0 to achieve the minimum distortion between the range block and the transformed domain block. Therefore we use an extended range for the contrast scaling. In our design, the contrast scaling is determined by testing all the possible values in the following set $\{ n / 4, \ n = 1, 2, 3, \ldots, 8 \}$ to find the best one that minimizes the distortion. We thus need three bits to denote this contrast scaling.

Isometries (ι) are nothing but the transformation that simply shuffles the pixels with in a block, in a deterministic way. The following are the eight sets of isometries that are used in the iteration-free fractal coding scheme.

1. Identity

$$(\iota_0 \, \mu)_{i,j} = \mu_{i,j} \tag{3}$$

2. Orthogonal Reflection about mid vertical axis ($j = (B - 1) / 2$) of block:

$$(\iota_1 \, \mu)_{i,j} = \mu_{i, B-1-j} \tag{4}$$

3. Orthogonal Reflection about mid horizontal axis ($i = (B - 1) / 2$) of block:

$$(\iota_2 \, \mu)_{i,j} = \mu_{B-1-i, j} \tag{5}$$

4. Orthogonal Reflection about first diagonal ($j = i$) of block:

$$(\iota_3 \, \mu)_{i,j} = \mu_{j, i} \tag{6}$$

5. Orthogonal Reflection about second diagonal ($i + j = (B - 1)$) of block:

$$(\iota_4 \, \mu)_{i,j} = \mu_{B-1-j, B-1-i} \tag{7}$$

6. Rotation around center of the block, through $+ 90^0$

$$(\iota_5 \, \mu)_{i,j} = \mu_{j, B-1-i} \tag{8}$$

7. Rotation around center of the block, through + 180^0

$(\iota_6\ \mu)_{i,j} = \quad \mu_{B-1-i\ ,\ B-1-j}$ (9)

8. Rotation around center of the block, through - 90^0

$(\iota_7\ \mu)_{i,j} = \quad \mu_{B-1-i\ ,\ j}$ (10)

In addition to the above, some other more complex, shuffling transformation can be constructed. But there is no necessity to go for it since the performance of our algorithm is very satisfactory with the available eight isometries itself and the increase in the number of isometries hardly have any performance improvement but only slightly affects the compression ratio.

After testing all combinations of parameters in equation 2, the fractal code is determined when the coded range block $\mathbf{R}_{\mathbf{bar}}$ has the minimum distortion from the original range block **R**. The distortion between the original and the coded block is represented by the mean-squared-error(MSE) measurement defined as

$$MSE\{R,R_{bar}\} = (1 / B^2) \sum_{0 \le i,j < B} (r_{i,j} - r_{bar\ i,j})^2 \qquad (11)$$

Where $r_{bar\ i,\ j}$ denotes the (i, j)th pixel in the coded range block. We finally attach a header for the range block to denote its coding status (either coded by mean or affine transforms). Therefore the decoder can correctly reconstruct each coded range block according to the header.

2.2 Domain Pool Design

In order to obtain an efficient domain pool in which the redundancies between the domain blocks is reduced, we propose a new algorithm known as Block Averaging Method to generate a domain block. Here the coding performance will improve compared to the conventional fractal coding scheme.

2.2.1 Block Averaging Method

In this method the centroid of four blocks which are adjacent and partly overlapped in the mean image is computed to generate a Domain Block in the Domain Pool. More image blocks can be averaged to reduce more redundancies among them. However it is expected that the correlation between the four neighboring blocks will be higher than that between more neigbouring blocks. Therefore we use only four neigbouring blocks in this method.

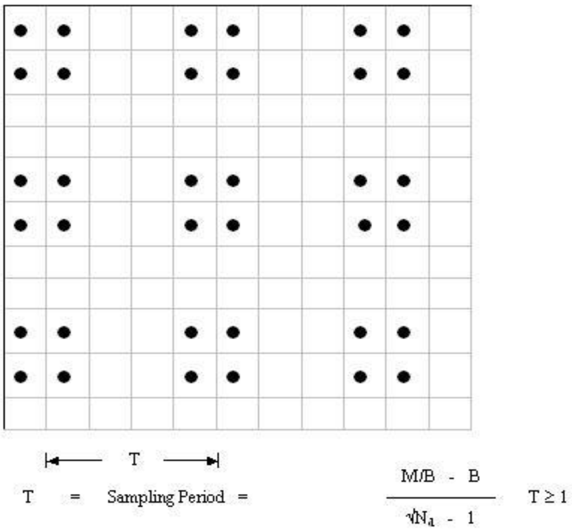

Fig 3. Some sets of Image Blocks used to generate Domain Blocks in Block Averaging Method

Figure 3 shows some set of four neighboring blocks with a four pixel sampling period in the mean image. In the figure, each black point denotes the top left corner of an image block and we use it to represent a B x B image block. Their relative positions are shown in figure 4.

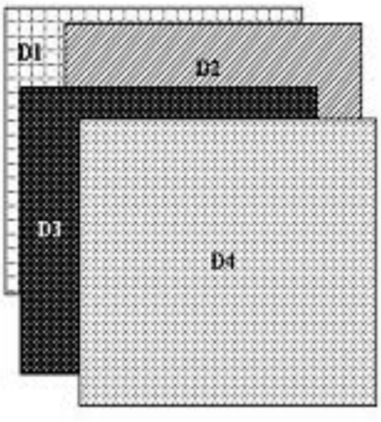

Fig 4. Relative position for four adjacent and partly overlaping image blocks

The pixel $\bar{d}_{i,j}$ in the average block $\bar{D}$ can be calculated by

$$\bar{d}_{i,j} = (d1_{i,j} + d2_{i,j} + d3_{i,j} + d4_{i,j}) / 4 \qquad (12)$$

where $d1_{i,j}$ to $d4_{i,j}$ represent the (i, j)th pixel in the image blocks D1 to D4. Therefore the calculated pixel relates to the information of four adjacent pixels in the original image blocks. The averaged block replaces the original four adjacent image blocks and the redundancy in the four adjacent image block is thus reduced. With this averaged blocks, the constructed domain pool is more efficient than that consist of the domain blocks directly selecting from the mean image.

The domain blocks are uniformly selected from the averaged blocks with a sampling period (T) in the mean image. Let the number of domain block in the domain pool be N_D. The sampling period in both the horizontal and the vertical directions can be calculated by

$$T = \frac{M / B - B}{\sqrt{N_D} - 1} \qquad T \geq 1 \qquad (13)$$

2.3 Decoder

Figure 2 shows the flow chart of the proposed iteration-free decoding scheme. We first receive the entire fractal code and determine whether or not the range block is coded by mean by having a look at its header. The mean image is reconstructed using the mean information in the fractal codes. Note that this mean image is the same as that present in the encoder since both are constructed using the same block means. Therefore the domain blocks generated in both the coding and the decoding phase is by the Block Averaging Method. If the block is coded by mean, the value of pixels in the decoded blocks is equal to the mean value. Otherwise, we perform the contractive affine transforms to reconstruct the Range Blocks. The decoding process ends when all the Range Blocks are reconstructed.

At this point, no iteration is required and thus no convergence criterion and divergence problem for the decoded image to be concerned with. Since only the fixed mean image is reconstructed from the received fractal code, the required memory size in the proposed iteration-free scheme is much smaller than that of the conventional fractal image decoder. On the other hand, having no iteration means that the Range Block can be decoded in parallel. The architectural complexity of the proposed iteration-free decoder is obviously lower than that of the conventional fractal schemes that requires iteration.

2.4 Hybrid Model

Fixed-Size Range Block has given optimal performance in terms of bit rate and PSNR. But it is evident theoretically that as the size of the Range Block increases the bit rate and PSNR is affected. More over the coding time for the fixed block size Range Blocks is considerably higher specially when we try to keep the window size as 4x4. To have a tradeoff between the impacts of Block Size on the coding performance it is proposed to have a Multi-Level Range Block. In this the Image is partitioned in to a B x B Range Blocks. In the coding phase if the variance of the Range Block is less than the threshold value E^{th}, the range block will be coded by mean. Otherwise the block will be sub-partitioned in to 4 range blocks of size $B/2^n$ x $B/2^n$ where n = 1,2,3… (until $B/2^n > 2$). The Range Block is sub-partitioned multilevel until the above condition.

3. RESULTS

The proposed iteration-free fractal coding scheme provides better performance when compared to that of the conventional fractal coding scheme. In the computer simulation it is found that as the size of the Domain Pool increases, the performance gets improved significantly. When the Image is partitioned in to 8x8 Range Blocks, high Compression Ratio is achieved. But it is not possible to maintain the quality of the decoded image as the image will be reconstructed from a small number of domain blocks. But if the Image is partitioned in to 4 x 4 Range Blocks, significantly good Compression Ratio as well as good image quality is maintained which is quiet obvious from the values of PSNR. The most worrying factor in case of 4 x 4 is the coding time which is brought down considerably with the Hybrid Modal with out sacrificing the PSNR or the Compression Ratio, which infact is slightly improved when compared to that of the 4 x 4 Range Blocks.

Given below in fig 5, Table I and fig 6 is the comparison of Compression Ratio, Bit Rate, PSNR and Coding Time for various window sizes including the Hybrid Model of Multilevel Range Block of size of 8x8-4x4 for CT – Abdomen Image of size 512 x 512. Also the fig 7, Table II and fig 8 shows the comparison of Compression Ratio, Bit Rate, PSNR and Coding Time for various window sizes including the Hybrid Model of split window size of 8x8-4x4 for MR- Head Image of size 256 x 256. It is clear from the below graphs that in Hybrid Model with Multilevel Range Block, the coding time is greatly reduced however able to achieve good Compression Ratio without much drop in PSNR.

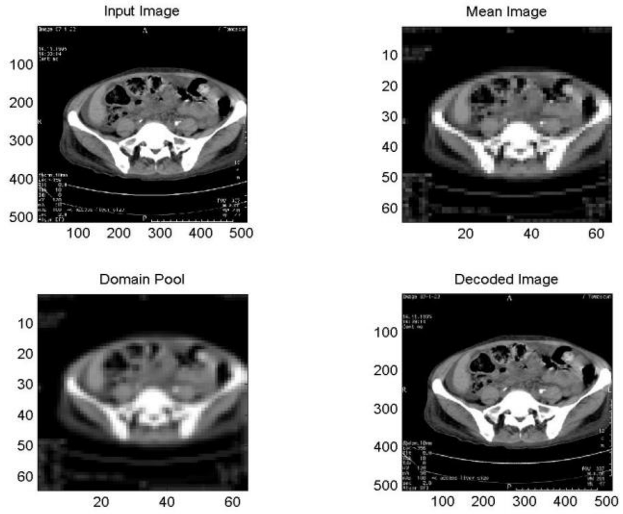

Fig 5. Image showing Input, Mean, Domain Pool and Decoded Image of CT- Abdomen

Range Size	Compression Ratio	Bit Rate (bits/pixel)	PSNR (dB)	Coding Time (hrs)
4 x 4	8.3587	0.95709	36.4383	7.16
8 x 8	33.7706	0.23689	38.4396	0.16
4 x 4 - 8 x 8	10.6616	0.75035	41.733	1.79

Table I. Compression Ratio, Bit Rate, PSNR and Coding time for various window sizes for CT- Abdomen

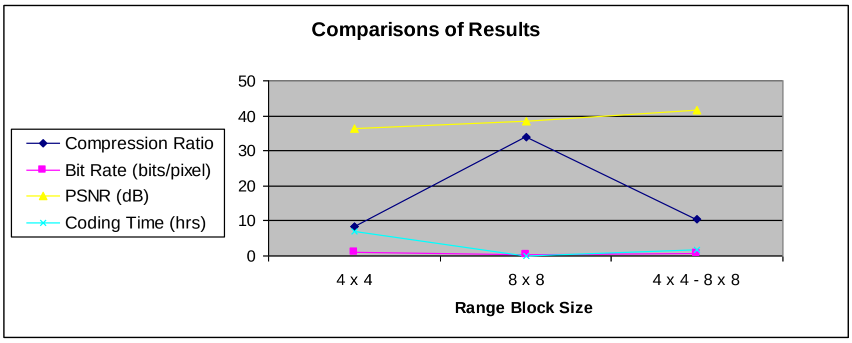

Fig 6. Compression Ratio, Bit Rate, PSNR and Coding time for various window sizes for CT- Abdomen

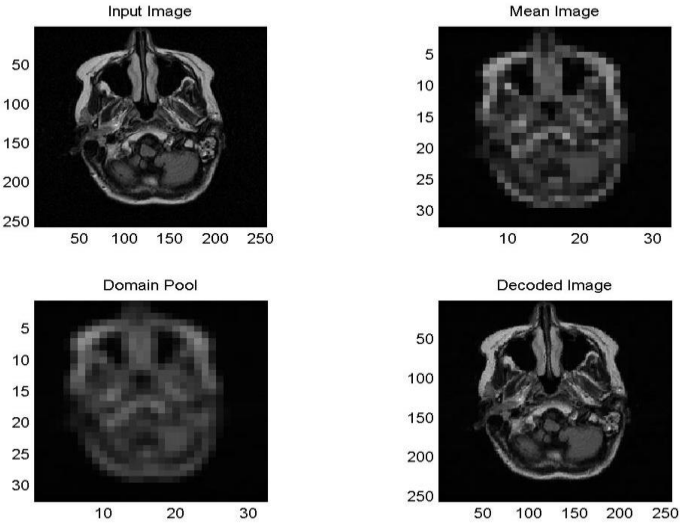

Fig 7. Image showing Input, Mean, Domain Pool and Decoded Image of MR - Head

Range Size	Compression Ratio	Bit Rate (bits/pixel)	PSNR (dB)	Coding Time (mins)
8 x 8	37.4598	0.21356	42.3261	0.9
4 x 4	8.589	0.93143	44.3357	27.76
4 x 4 - 8 x 8	11.4378	0.69943	48.3723	6.7

Table II. Compression Ratio, Bit Rate, PSNR and Coding time for various window sizes for MR - Head

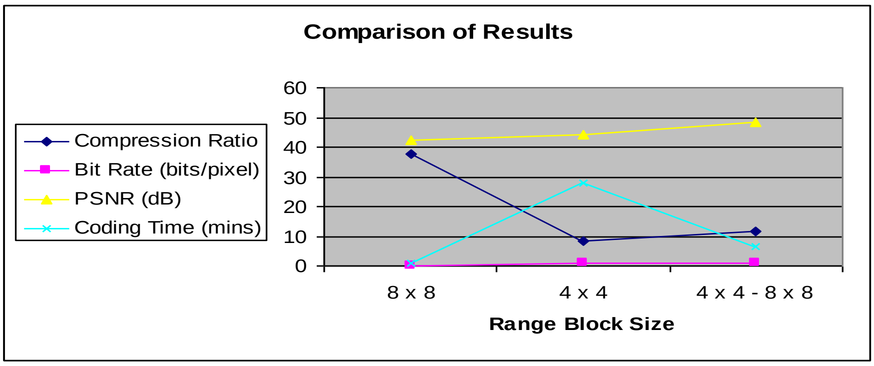

Fig 8. Compression Ratio, Bit Rate, PSNR and Coding time for various window sizes for MR – Head

4. CONCLUSION

In this paper a novel method namely the Iteration-Free Fractal Image Coding for Medical Images using the Block Averaging Method of Designing the Domain Pool is proposed. By this the redundencies between the adjacent blocks in the Domain Pool is reduced thereby making the coding process efficient. In addition with the Hybrid Model, we were able to reduce the coding time without much loss in PSNR and at the same time able to achieve good Compression Ratio. In future we will investigate the behaviour of Deferring Range/Domain Comparison (DRDC) based coding process, which intends to further reduce the coding time.

REFERENCES

[1] ***Aranud E. Jacquin*** (1992) ‘Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformation’, IEEE Transaction on image processing vol.1, no.1, pp. 18-29.

[2] ***Barthel K. U. and Voye T.*** (1994), ‘Adaptive Fractal Image Coding in the Frequency Domain’, Proceedings International workshop Image Processing, pp 33-38.

[3] ***Davoine F, Svensson J. and Chassery J. M.*** (1995), ‘A Mixed Triangular and Quadrilateral partition for fractal image coding’, Proceedings IEEE International Conference Image Processing, pp 284-287.

[4] ***Fisher Y, Jacobs E. W. and Boss R. D.*** (1992), ‘Fractal Image Compression using Iterated Transforms’, Image and Text Compression.

[5] ***Hamzaoui R., Muller M. and Saupe D.*** (1996), ‘VQ Enhanced Fractal Image Compression’, Proceedings IEEE International Conference Image Processing, pp 153-156.

[6] ***Hsuan T. Chang and Chung J. Kuo*** (2000) ‘Iteration-Free Fractal Image Coding based on Efficient Domain Pool Design’, IEEE Transaction on Image Processing, vol.9, no.3, pp. 329-339.

[7] ***Kim K. and Park R. H.*** (1994), ‘Image Coding Based on Fractal Approximation and Vector Quantization’, Proceedings IEEE International Conference Image Processing, pp 132-136.

[8] ***Kwon Kim*** (1996), ‘Still Image Coding based on Vector Quantization and Fractal Approximation’, IEEE Transaction on Image Processing, vol.5, no.4, pp. 587-597.

[9] ***Novak M.*** (1993), ‘Attractor Coding of Images’, Proceedings on International Picture Coding Symposium, p. 15.6.