Comparative Assessment of Fractal Image Compression with and without Neural Networks for Medical Images

Dr. Rohini Nagapadma¹ and S. Padmashree ²
¹ The National Institute of engineering/ECE department, Mysore, India
Email: rohini_nagapadma@yahoo.co.in
² GSSS institute of Engineering and technology for women/ECE department, Mysore, India
Email: spshree40@gmail.com

Abstract—The Demand for bulk storages of Medical Images in today’s growing world has lead to many advanced Image compression algorithms. One such algorithms used for compressing images is Fractal image Compression (FIC). Fractal Image compression is found effective for medical images in preserving the quality of the image with high Compression Ratio. Fractal Encoding involves partitioning the images into Range Blocks and Domain Blocks and each Range Block is mapped onto the Domain Blocks by using contractive transforms called the Affine Transforms. One of the drawbacks of FIC is that it has long encoding time and short decoding time. The encoding time can be minimised by reducing the time taken to search suitable domain blocks for each Range Blocks. This paper is focused in finding a technique to search a suitable domain block through feature extraction method and neural network. To speed up the encoding time an expert system has been trained using Hopfield neural network. In this paper, A comparative analysis of FIC algorithm with neural network and without neural network is being Performed using a set of Magnetic Resonance (MR) images of the Brain, ultrasound Image of Abdomen based on encoding time, Decoding Time, Peak signal To Noise Ratio (PSNR) and Compression Ratio. The Results are simulated using graphical Mat lab and the results of the comparison showed that the Performance of FIC algorithm with neural networks using this proposed technique has improved considerably with regard to encoding time, preserving the quality of the image.

Index Terms—Fractal Image Compression (FIC), Iterated Function systems, Range Blocks, Hopfield Neural network, Neural Networks, Feature Vectors.

I. INTRODUCTION

Benoit. B. Mandelbrot, a mathematician first developed Fractal theory in Mathematical geometry which was further developed by M Barnsley who introduced the fundamental principle of fractal image compression in the year 1988 [1]. Fractal image compression is also called fractal image encoding as the compressed image is represented by contractive transforms and mathematical functions which are required for reconstruction of the image [2]. Transforms ensure that the distance between any two points on the transformed image will be less than the distance between the same points on the original image. These transforms are composed by the union of number of affine mappings on the entire image known as the...
Iterated Function Systems (IFS) [2-4]. The fractal image compression based on IFS was complicated and was still not practical. Arnaud Jacquin settled the problem with the partition of IFS in 1990. Jacquin partitioned the image into sub images called the Range Blocks and PIFS were applied on the sub images rather than the whole image. Temporary images used to form the range blocks are called the domain blocks. The whole process of fractal image encoding involves Partitioning the image to form the Range blocks and domain blocks [4], detection of domain blocks [5-6] and Proper application of affine transforms. Further the choice of domain blocks depends on the type of partition scheme used. The domain pool in fractal encoding is similar to the codebook in vector quantization (VQ) which is referred to as virtual codebook or domain codebook. The set of transformations selected is applied on each Range block to map on to the domain blocks As the encoding time is larger than the decoding time fractal coding is asymmetric in nature.

II. FUNDAMENTALS OF FRACTAL IMAGE COMPRESSION

This section gives a brief description on the general aspects of fractal image compression and the algorithm for fractal encoding and decoding algorithm is discussed. Fractal Image Compression is based on fractal theory of self similar affine transforms [6].

A. Self Similar and self affine Transforms

In this section, theory involved in Fractal Image Compression is being discussed. It is basically based on fractal theory of self- affine transformations and self-similar transformations [6]. A self- affine transformation w: defined by $\mathbb{R}^n$ to $\mathbb{R}^n$ transformation is of the form

$$W(x) = T(x) + b,$$

where $T$ is a linear transformation on $\mathbb{R}^n$ and $b \in \mathbb{R}^n$ is a vector. If there exists a mapping $w$ of the form $D$ to $D$ and $D$ is a subset of , $\mathbb{R}^n$ then $D$ is a contraction of $w$ such that $d(w(x), w(y)) \leq \rho d(x, y)$ for $x, y \in D$ and for a metric $d$ on $\mathbb{R}^n$. The real number $\rho$ is called the contractivity of $w$. If $d(w(x), w(y)) = \rho d(x, y)$ then $w$ is called a similarity. A family of $\{w_1, w_2, w_3, \ldots, w_n\}$ of contractions is known as Local Iterated function scheme (LIFS). Once the LIFS is formed the encoding of the image can be obtained so that the attractor is similar to the original image. Barnsley’s collage theorem tells how well the attractor can be approximated to the original image. This theorem says that if a transformation is contractive, then when applied repeatedly starting with any initial point, then the point converges to a unique fixed point. This simple looking theorem tells us how we can expect a collection of transformations to define an image.

B. Open Issues of Fractal image compression

The Issues pertaining to fractal Image compression are

- How to partition Image into Range Blocks and domain Blocks (Partitioning schemes)
- How to search a suitable domain block for a give Range Block - speed up technique
- How to improve encoding time - speed up technique
- How to optimize domain pool selection - speed up technique
- How to use neural networks for image compression - speed up technique

With reference to the above issues to fractal image compression, the current implemented work provides a speed up technique to improve the encoding time of the fractal image compression. This paper focuses in finding a technique to search a suitable domain block through feature extraction method and neural network. To speed up the encoding time an expert system has been trained using Hopfield neural network. The flow chart of the implemented technique is shown in figure 1

C. Implemented Fractal Encoding and decoding Algorithm without Neural network

- Input Image is loaded into the buffer
- The Image is partitioned using quad tree partitioning technique [4][5]. The Image is partitioned into non overlapping square blocks called the Range blocks
- The Initial size of the domain blocks is chosen to be twice the size of the range block
- The domain blocks is down sampled to the size of range blocks and the eight possible affine transformations is computed for each Range block
- The domain block chosen, shall resemble the range block with respect to some metric and accordingly encoding parameters are computed The decoding time generally depends on the number of Iterations and here it takes only few iterations ranging from 4-8 to reach the fixed point.
The reconstruction process of the original image consists of the applications of the transformations described in the fractal code book iteratively to some initial image until the encoded image is retrieved back.

III. IMPLEMENTED FRACTAL ALGORITHM USING HOP FIELD NEURAL NETWORKS WITH FEATURE EXTRACTION AND CLASSIFICATION METHOD

In this section implementation of Fractal image Compression with Neural networks [7] is discussed. There are mainly four basic steps involved in Fractal image Compression. First step is to partition the Image into Range and domain blocks. There are many partitioning schemes [8] available but the partitioning scheme suitable for medical images used in this technique is quad tree Partitioning technique [9]. The Second step is to select a suitable domain pool from the set of available domain blocks to speed up the encoding process. Thirdly a suitable transformation function is to be chosen and applied on the domain blocks so as to map the domain blocks to the Range blocks. Here, in this implemented technique Affine transforms is being used. As Fractal Encoding process takes a longer time and hence to overcome this problem, speed up techniques using feature extraction methods and classification methods are being employed in this proposed technique. One such technique which is being implemented in this paper is a combined technique of feature extraction methods and classification method based on hop field neural Network method.

In suitable domain search, compatibility between the range blocks and domain blocks is sort. One way is to compare the images as a whole and the other is to extract a few numbers of features that characterize the domain and range images then the comparison of range blocks and domain blocks is made based on these features rather than on individual pixels. In this way the complexity of the problem is reduced, which results in fast coding process.

D. Feature Extraction

In feature extraction method the features extracted from the image are Mean, standard deviation, skewness and standard Deviation [10] from all the image blocks. The mean value of image gives the measure of average gray level of image (m), standard deviation (v) defines the dispersion of its gray level from mean. Skewness (s) describes existence of symmetry/asymmetry from the normal distribution in image. Kurtosis (k_u) characterizes the relative peakedness or flatness of a distribution compared to the normal distribution. Positive kurtosis indicates a relatively peaked distribution where as negative kurtosis indicates a relatively flat distribution of gray levels in the image These features can be explained mathematically using the following equations based on the histogram of the image.
\[
m = \sum_{k=1}^{k} k p_k
\]

\[
v = \sqrt{\sum_{k=1}^{k} (k - m)^2 p_k}
\]

\[
s = \frac{1}{v^3} \sum_{k=1}^{k} (k - m)^3 p_k
\]

\[
k u = \frac{1}{v^4} \sum_{k=1}^{k} (k - m)^4 p_k - 3
\]

Where \( P_k \) is given by

\[
p_k = \frac{h_k}{\sum_{k=1}^{k} h_k}
\]

In this technique an intermediate image called as average image (\( \bar{A} \)) which is equal to the average of all the range blocks (\( r \)) is chosen, then the features are extracted from range images (\( r \)), domain images (\( d \)) and average image and a feature vector corresponding to every image is formed. Further operations of desired task are performed on these feature vectors.

Range feature vector \( F_r = [m_r, v_r, s_r, ku_r] \)

Domain feature vector \( F_d = [m_d, v_d, s_d, ku_d] \)

Average feature vector \( F_{\bar{A}} = [m_{\bar{A}}, v_{\bar{A}}, s_{\bar{A}}, ku_{\bar{A}}] \)

E. Classification method

Domain feature vectors (\( F_d \)) are compared with average feature vector (\( F_{\bar{A}} \)), difference between these vectors is computed and stored. This set of error is called as domain code-book. The Range feature vector corresponding to the \( i^{th} \) block [\( R_i \)] is compared with average feature vector (\( F_{\bar{A}} \)) and distance between them is calculated. This error value is used as search key and nearest value search is applied in domain code-book [10] and assigned to it for encoding. This search process is repeated for each range Block. In this way the suitable domain block corresponding to every range block is searched.

The suitable domain blocks selected through the above feature extraction method is used to train the Expert system whose indices are , selected domain blocks and Range blocks. The Expert system trained in this technique is Hop field neural networks [11][12][13]. Hopfield NN [HPNN] can minimize the energy function. In HFNN, all connective weights are calculated initially from the system data without training. The neurons are initiated by the initial value then the network goes through the series of iterations until it reaches a final output that represents a minimum of an energy function. In this Hopfield neural network algorithm the weights from input-hidden-output layer are updated iteratively during the learning phase. The features of HPNN are

- These are single layered recurrent networks
- All the neurons in the network are fed back from all other neurons in the network
- The states of neuron are either +1 or -1 instead of (1 and 0) in order to work correctly.
- No of the input nodes should always be equal to no of output nodes

The figure 2 shows a Hopfield network with four nodes

F. Implemented Algorithm using Hopfield neural Network

Fractal Encoding and Decoding process with neural networks is discussed in this section

Hopfield neural network

Hopfield networks are constructed from artificial neurons (see Fig. 2). These artificial neurons have N inputs. With each input i there is a weight \( w_i \) associated. They also have an output. The state of the output is maintained, until the neuron is updated. Updating the neuron entails the following operations:

- The value of each input, \( x_i \) is determined and the weighted sum of all inputs, \( \sum w_i x_i \) is calculated.
Figure 2. Hopfield network with four nodes

- The output state of the neuron is set to +1 if the weighted input sum is larger or equal to 0. It is set to -1 if the weighted input sum is smaller than 0.
- A neuron retains its output state until it is updated again.

Fractal Encoding Process

In general, in fractal image encoding the input image is divided into number of range blocks of size RxR and number of domain blocks of size D x D. The domain size is taken twice as the range block. Then each range block is compared with all the domain blocks to find the best matched domain block based on features extracted. In fractal image encoding, searching the best matched domain block is a time taken process. In order to reduce search space and encoding time an expert system using hopfield neural network is designed.

In our proposed algorithm, initially an input MRI image of size (m x n) is divided into range blocks of size RxR and stored into a vector. Subsequently, the index of each range block is given as input to the expert system one by one and the expert system recognizes the range block immediately. Since the expert system is already trained, it yields reduced search space. Instead of searching all the domain blocks in an image, the search is done only in the resulted domain block set and also it reduces the search space. Thus the encoding time is reduced automatically.

Fractal Decoding Process

The reverse process of encoding is done in decoding. At the decoding phase the transformation parameters are recursively applied to an initial image with mean value, which will then converge to the fractal image after fewer than ten iterations. Initially, a new matrix is formed with size of image (MXN) and the mean value of the image is placed in it. Subsequently, the input MRI image is partitioned into fixed blocks of size (RxR) are called range blocks RB. Then the block process operation is executed to reduce the newly formed image by averaging the intensities of four neighbouring pixels. The resulted vector of encoding phase is taken into consideration and a single block is extracted from the reduced image first. The corresponding best matched domain block index is retrieved from the vector and the pixel values of the reduced domain block are then placed in the location in the range determined by the orientation information. Executing all the range blocks constitutes a single iteration. After one to ten iterations, the decompressed image will be constructed.

Algorithm for training the expert system

The steps involved in training process are as follows.

- The input MRI image [14] of size (MXN) is partitioned into a fixed block size, for instance, (Rx R). The resulting blocks are called range blocks and converted into a vector \( R_B \). Where \( R_B = [r_1, r_2, r_3, ..., r_4] \); where \( 0 < k \leq |R_B| \)
- The block process operation is executed. During the block process operation the input image is reduced by averaging the intensities of four neighbouring pixels. Then the reduced image is partitioned into fixed blocks of size (D x D) and is put into a vector. Subsequently, eight isometric transformed matrices for each reduced block is found out and appended in to the same vector continuously. The resulting blocks are called Domain blocks \( D_B \). \( D_B = [d_1, d_2, ..., d_n] \); where \( 0 < l \leq |D_B| \); where \( |D_B| = 2^k |R_B| \)
- For each range block of the input image a best matched domain block is found by executing the following procedure repeatedly. It is repeated for \( k \) times, where \( k \) is number of range blocks in the input image.

- A vector of size \( |R_B| \) is created and an initial value of the vector \( a^3 \) allotted and \( DB_{bestk} = a^3 \); where \( a = m/8 \times n/8 \); where \( 0 < k \leq |R_B| \)
A single range block is extracted and converted into a vector (\(V_{RB}\)). Subsequently, the mean value (\(V_{RB}\)) of the vector (\(V_{RB}\)) is calculated.

\[
\bar{V}_{RB} = \frac{\sum_{i=1}^{p} V_{RB}^i}{p}; \text{ Where } p = |V_{RB}|
\]

A single reduced domain block is extracted and converted into a vector (\(V_{DB}\)). Consequently, the Mean value (\(\bar{V}_{DB}\)) of the vector (\(V_{DB}\)) is calculated.

\[
\bar{V}_{DB} \leftarrow DB_j; \text{ where } 0 < j < |V_{DB}|; \quad \frac{\sum_{i=1}^{p} V_{DB}^i}{p}; \text{ Where } q = |V_{DB}|
\]

The scaling parameter \(a\) and offset value \(b\) is computed using the following formula.

\[
a = \frac{(\bar{V}_{RB} - V_{RB})^T \cdot V_{DB}^y - V_{DB}^y \cdot \bar{V}_{RB}}{(\bar{V}_{DB} - V_{DB})^T \cdot V_{DB}^y - V_{DB}^y \cdot \bar{V}_{DB}}; \text{ where } 0 < x \leq |V_{RB}|
\]

\[
b = (\bar{V}_{RB} - a)
\]

\(C\) is initialized as the best matched domain block for the current range block first. Then \(c\) value is calculated by using the following formula

\[
C \leftarrow (V_{RB}^x, V_{DB}^y, b)^T \cdot (V_{RB}^x, V_{DB}^y, b)
\]

The \(c\) value is Compared against the element of the vector \(DB_{best(i)}\). If the calculated \(c\) is less than the current element \(DB_{best(i)}\) vector and the calculated scaling parameter \(a\) is less than one then the current index \(j\) of the domain block, scaling parameter \(a\) and the offset value \(b\) are placed in the vectors \(DB_{best(i)}, DB_{best(k)}, DB_{best(k)}\) using the following procedure, otherwise aforesaid process is repeated with next domain block.

\[
DB_{best} \ll j
DB_{best(i)} \ll a
DB_{best(k)} \ll b
DB_{best(k)} \ll c
\]

Then the steps 3 to 7 are repeated to find the best matched domain block for the next range block. The process will be repeated for large set of data until the expert system has been trained with lot of images. Thus Domain-range matches are performed for those domains that belong to a class similar to the range. The feature computation serves to identify those domains belonging to the class of sub images whose feature vectors are within the feature tolerance value of the feature vector belonging to the range cell. During encoding, it assigns the range cell to a class and domain range comparisons are performed only against the domains of the same class as the range cell.

IV. RESULTS AND CONCLUSIONS

Twenty images out of which ten images of Magnetic Resonance Images [14][15] of brain (T1 and T2 weighted), and 10 ultrasound image of abdomen of size 256x256 with 8 bit gray scale were collected from JSS Hospital and Vikram Hospital Mysore. Fractal image compression using quad-tree Partitioning Iterated function systems (with and without NN) was applied on the MR images. The threshold was varied from 0.1 to 0.4. These images were trained using Hopfield neural network and the minimum Range Block sizes was taken to be 4. Encoding and decoding time were computed for the images and a comparison of the, encoding time and decoding time with and without neural network were carried out and tabulated in Table I.

A comparative study on the Performance measures like PSNR, CR [16][17] with and without neural network is tabulated in Table II. The results were simulated using Matlab on core2duo processor with 4GB RAM.

Figure 3, Figure 4, Figure 5, Figure 6, Figure 7 are the results obtained using the proposed Algorithm applied on MR Image of the Brain and Ultrasound sound image of abdomen.

Figure 8 shows a comparative bar chart of the results obtained by implementing the above FIC algorithm using Hopfield Neural network and without neural network. It is understood from Table 1 that the encoding time on an average using Hopfield Neural network is 259 sec and Decoding time is 35 sec. The Encoding time and decoding time without Neural networks is on an average 345 Sec and 35 Sec.
V. CONCLUSIONS

- A fast image compression method has been proposed, which uses different membership values for comparison of ranges with domains of the same image. The main disadvantage of the fractal compression is the encoding complexity. A fractal image compression algorithm using Hopfield Neural Network implemented on Medical images has reduced the search time for domain blocks and hence the encoding time has decreased considerably.
- The expert system employed during the encoding process, to speed up the encoding without severely affecting the image quality. Hopfield neural network is used to train the expert system.
<table>
<thead>
<tr>
<th>Figures</th>
<th>Threshold</th>
<th>With NN Enc(s)</th>
<th>Without NN Enc(s)</th>
<th>With NN Dec(s)</th>
<th>Without NN Dec(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 3</td>
<td>0.1</td>
<td>268.16</td>
<td>345.67</td>
<td>35.80</td>
<td>35.16</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>253.10</td>
<td>347.70</td>
<td>35.49</td>
<td>35.36</td>
</tr>
<tr>
<td></td>
<td>0.27</td>
<td>256.58</td>
<td>344.29</td>
<td>35.41</td>
<td>35.47</td>
</tr>
<tr>
<td>Figure 4</td>
<td>0.1</td>
<td>254.64</td>
<td>346.30</td>
<td>35.53</td>
<td>35.10</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>245.45</td>
<td>345.45</td>
<td>35.50</td>
<td>35.25</td>
</tr>
<tr>
<td></td>
<td>0.27</td>
<td>243.30</td>
<td>352.86</td>
<td>35.49</td>
<td>35.71</td>
</tr>
<tr>
<td>Figure 5</td>
<td>0.1</td>
<td>267.24</td>
<td>351.45</td>
<td>36.53</td>
<td>35.20</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>255.34</td>
<td>349.68</td>
<td>35.54</td>
<td>35.45</td>
</tr>
<tr>
<td></td>
<td>0.27</td>
<td>253.46</td>
<td>352.78</td>
<td>35.65</td>
<td>35.38</td>
</tr>
<tr>
<td>Figure 6</td>
<td>0.1</td>
<td>258.96</td>
<td>351.95</td>
<td>36.63</td>
<td>35.52</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>265.93</td>
<td>333.85</td>
<td>35.96</td>
<td>35.69</td>
</tr>
<tr>
<td></td>
<td>0.27</td>
<td>249.80</td>
<td>344.02</td>
<td>35.77</td>
<td>35.76</td>
</tr>
<tr>
<td>Figure 7</td>
<td>0.1</td>
<td>258.96</td>
<td>351.95</td>
<td>36.63</td>
<td>35.52</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>265.93</td>
<td>333.85</td>
<td>35.96</td>
<td>35.69</td>
</tr>
<tr>
<td></td>
<td>0.27</td>
<td>249.80</td>
<td>344.02</td>
<td>35.77</td>
<td>35.76</td>
</tr>
</tbody>
</table>

The PSNR decreases with increase in threshold value with Neural networks while the PSNR increases with increase in threshold value without Neural network.

CR decreases with increase in threshold values for both with and without neural network.

REFERENCES

Figure 8. Comparative Bar Chart


152