Fractal Image Compression Using Neighborhood Region Search Method

I. MR. RAJESH BULBULE, 2. MRS. R. S. BALPANDE, 3. DR. A. S. KHOBragade
1. Department of Electronics & Telecommunication, 2. Department of Electronics Engineering 3. Department of Electronics Engineering
1. Assistant Professor (RSCOE, Pune), 2. Assistant Professor (VCCE, Nagpur), 3. Professor (RGCOE, Nagpur)

Abstract: The main inspirations behind the image compression are in its speed of transmission and storage of data. This paper proposed that the speed of fractal encoder is increased by the Neighborhood region search method i.e. to decrease the encoding time. Fractal image compression is capable for both practical as well as theoretical work. Here co-ordinate system is used to find two lowest discrete cosine transformation (DCT) coefficients of different image blocks. In this paper, Image blocks with similar edge shapes will be concentrated in specific regions in a co-ordinate system. The main aim of this paper is that to reduce the number of computations and to keep the reconstructed image quality better. Furthermore, this given proposed method is embedded with edge property of block which gives rise in more speed rate.

IndexTerms - FIC (Fractal image compression, PIFS (Partition iterated function system), DCT (Discrete cosine transform), Frequency domain neighborhood region search method

I. INTRODUCTION

Fractal image compression is a lossy compression technique which was first proposed in 1985 by Barnsley and Demko operating from IFS. Fractal image compression was first introduced practically in 1992 by Jacquin.[1,2] In this paper new method Neighborhood region search method is used to speed up the fractal encoder. Using this method compression ratio is increase and reconstruction of image is also improved. Basically Fractal image compression is structure based method i.e. group of pixels. The paper presents how to reduce the encoding time and number of computations using the Neighborhood region search method. High computational complexity of fractal image encoding significantly controls its application. Basically Fractal image compression is structure based method. Structure means a group of pixels. Paper presents show to reduce the encoding time using Neighborhood region search method. DCT method is used to find the coefficients of an image. Firstly, Image is divided into range and domain block. Size of range block is smaller than domain block, i.e. size of domain block is double as that of range block. This paper presents section I is Introduction, Section II is Fractal Image Encoding, Section III is Neighborhood region search method, Section IV is Mathematical analysis and Section V is Experimental results.

II. FRAC TAL IMAGE COMPRESSION

The Basic idea is coming from PIFS (Partition Iterated Function System). Consider original gray level image 'f' of size (m x m). Let R be range pool defined as set of all non-overlapping blocks of size (n x n) of the image f. Using (m/n)^2 we can find the number of range pool. On other side domain blocks are overlapping blocks which are twice time greater than that of range block. Suppose D is the domain pool defined as set of all possible blocks of size (2n x 2n) of the image f. Using (m-2n+1)^2 we can find some domain block. For each range block v from R, fractal affine transformation is constructing by searching all of domain block i the D to find most similar block and the parameters indicating the fractal affine transformation will form fractal compression codec of v. To execute the similarity between the range and domain block, the size of domain block must be sub-sampled to (8 x 8) such that its size is same as v.

TABLE I. AFFINE TRANSFORMATIONS

<table>
<thead>
<tr>
<th>T0</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0</td>
<td>0 1</td>
<td>-1 0</td>
<td>0 -1</td>
</tr>
<tr>
<td>0 1</td>
<td>1 0</td>
<td>0 -1</td>
<td>1 0</td>
</tr>
</tbody>
</table>

The fractal affine transformation can allow the eight transformations of the domain block u in the Diheral. The eight transformations Tk : k = 0 to 7 can be expressed by matrices which is as shown in Table.1 in which the origin of u is assumed to be at the center of the block. By eight transformations, eight blocks are to be generated and which are denoted by uk : k = 0 to 7 respectively.

The table shows 8 isometric transformations also called as Affinetransformation. Here matrices T0, T1, T2 and T3 are relate to Horizontal positions, whereas matrices T4, T5, T6 and T7 are relate to Vertical positions respectively. T0 picks the origin block u. T1 and T2 are the flip of u with respect to horizontal and vertical lines, whereas T3 is the flip of u with respect to both horizontal as well as vertical lines. T4, T5, T6, T7 are the transformations which flip the u0,u1,u2,u3 along the main diagonal line y=x respectively.

III. NEIGHBORHOOD REGION SEARCH METHOD

Proposed method shows Neighborhood region Search Method to speed up the fractal encoder and which improves the quality of reconstructed images. Instead of a block matching, we are only interested in edge. First a two dimensional coordinate system of frequency domain is built. Here all the range and domain block are mapped into the coordinate system. Then we find the lowest
vertical F(1,0) and lowest horizontal F(0,1) DCT coefficients. We know the two dimensional DCT Of an image \( f(i, j) \) of size \((N \times N)\) defined by

\[
F(m, n) = \frac{2}{N} c_m c_n \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) \cos \left( \frac{(i+1)mn}{2N} \right) \cos \left( \frac{(j+1)mn}{2N} \right)
\]

Where \( m, n = 0, 1, 2, \ldots, N-1 \) and

\[
c_k = \begin{cases} 
1 & \text{if } k = 0 \\ 
\frac{1}{\sqrt{2}} & \text{otherwise}
\end{cases}
\]

For example consider \( N=8 \) then

\[
F(1,0) = \sqrt{2} \sum_{i=0}^{7} \sum_{j=0}^{7} f(i, j) \cos (0i)
\]

\[
F(0,1) = \sqrt{2} \sum_{i=0}^{7} \sum_{j=0}^{7} f(i, j) \cos (0j)
\]

Where F(1,0) and F(0,1) are lowest vertical and lowest horizontal DCT coefficients. By using this method somehow number of computations were reductions which explained in Experimental results and also complexity will get reduces.

A. System block diagram

By observing the output of different transformations DCT is best as compare with Wavelet, Hadamard transforms because the maximum amount of information is concentrated towards the origin. Hence DCT is the best method for Image compression. To decide the class of Image block, first we have to calculate F(1,0) and F(0,1) from equation (1) Image block Image blocks divided into five classes: Smooth class Horizontal class Vertical class Diagonal class Sub diagonal class

B. Flow Chart

Initially we have to divide the whole image into range and domain block by applying PIFS function on the image. We apply DCT on these blocks to find the coefficients F(1,0) and F(0,1). Then we use affine transformations T0 to T7 which are explained in introduction.

In this method, these transformations are divided into two categories i.e. T0 to T3 and another is T4 to T7. Here coordinate system is used to find two lowest discrete cosine transformation coefficients of different image blocks. In this, Image blocks with
similar edge shapes will be concentrated in specific regions in a coordinate system. Therefore, by searching similar edge, the purpose of speedup can be reached by limiting the search space. [4]

IV. MATHEMATICAL ANALYSIS

After decoding the image we get reconstructed image. Now we have to calculate different factors like PSNR, SNR, Compression ratio, MSE and No. of computations. As we know, MSE is inversely proportional to PSNR. To increase the value of PSNR we should have to decrease the value of MSE. Experimental results for PSNR are as shown in Table

\[
\text{MSE} = \frac{1}{n \times n} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (u(i,j) - v(i,j))^2
\]

\[
\text{PSNR} = 20 \times \log_{10} \left( \frac{n \times n}{\text{MSE}} \right)
\]

\[
\text{Compression ratio} = \left[ \frac{L_i - L_o}{L_i} \right]
\]

Where \( L_i \) and \( L_o \) are the length of Input image and Length of output image.

We have 8 isometric transformations. In this proposed method, these transformations are divided into two parts, i.e. Horizontal and vertical parts because of this, searching space is reduced. In this, the value of \( k \) is restricting 0 to 3 and 4 to 7 respectively. Here \( f_0, f_1, f_2, \) and \( f_3 \) are obtained by performing \( T_0, T_1, T_2 \) and \( T_3 \) transformations and gives following relations

\[
F_1(1.0) = F(1.0), F_2(1.0) = -F(1.0)
\]

\[
F_1(0.1) = -F(0.1), F_2(0.1) = F(0.1)
\]

\[
F_3(1.0) = -F(1.0)
\]

\[
F_3(0.1) = -F(0.1)
\]

On other side \( f_4, f_5, f_6 \) and \( f_7 \) are obtained by performing \( T_4, T_5, T_6 \) and \( T_7 \) transformations in Table 1

\[
F_4(1.0) = F(0.1), F_5(1.0) = F(0.1)
\]

\[
F_4(0.1) = F(1.0), F_5(0.1) = -F(1.0).
\]

The main aim is that to reduce the number of computations which will automatically reduce encoding time, which is reduced by using the proposed method.

We know that the relationship between PSNR and MSE is inversely proportional. The main task is that, we have to increase the PSNR. Here by using proposed method PSNR is increased as compared with other methods. Also using the proposed method, the number of computations is also reduced, encoding time required to encode the image is decreased.
V. EXPERIMENTAL RESULTS

Experimental results show the performance of the proposed method on different images which are as shown in following figure. Following figure (a) indicates the various input images of size 256 X 256 and Figure (b) indicates reconstructed images respectively. Table II shows the performance of the proposed method with PSNR, SNR, MSE, Compression ratio and Number of computations. Table III shows the matrix getting after applying affine transformations and matching between horizontal and vertical coefficients on Fractal4 image.

Comparing with Duh’s algorithm and full search algorithm, PSNR and quality of reconstructed image is increases which are as shown in following figures.

![Original Images](image1.png) ![Reconstructed Images](image2.png)

**Fig. 3 (a) is original images with size (b) is reconstructed images with PSNR**

<table>
<thead>
<tr>
<th>Image</th>
<th>PSNR</th>
<th>SNR</th>
<th>MSE</th>
<th>Compression Ratio</th>
<th>Number of Computations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cameraman</td>
<td>65.1337 dB</td>
<td>40.5994</td>
<td>0.0199</td>
<td>71.3554%</td>
<td>482184</td>
</tr>
<tr>
<td>Fractal1</td>
<td>55.8717 dB</td>
<td>14.9438</td>
<td>0.1682</td>
<td>75.3913%</td>
<td>335565</td>
</tr>
<tr>
<td>Fractal2</td>
<td>56.3742 dB</td>
<td>15.6472</td>
<td>0.1499</td>
<td>70.1252%</td>
<td>514056</td>
</tr>
<tr>
<td>Fractal3</td>
<td>63.0569 dB</td>
<td>39.7249</td>
<td>0.0322</td>
<td>71.5755%</td>
<td>414516</td>
</tr>
<tr>
<td>Fractal4</td>
<td>64.8780 dB</td>
<td>30.6368</td>
<td>0.0211</td>
<td>74.0143%</td>
<td>394267</td>
</tr>
</tbody>
</table>

**TABLE II.** PERFORMANCE OF PROPOSED METHOD ON DIFFERENT IMAGES

<table>
<thead>
<tr>
<th>Val (:,:,1)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-65.0625</td>
<td>-62.0625</td>
<td>-65.0625</td>
<td>54.9375</td>
<td></td>
</tr>
<tr>
<td>-65.0625</td>
<td>-65.0625</td>
<td>-65.0625</td>
<td>45.9375</td>
<td></td>
</tr>
<tr>
<td>-45.0625</td>
<td>-46.0625</td>
<td>-40.0625</td>
<td>56.9375</td>
<td></td>
</tr>
<tr>
<td>80.9375</td>
<td>76.9375</td>
<td>69.9375</td>
<td>132.9375</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Val (:,:,5)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-65.0625</td>
<td>-62.0625</td>
<td>-65.0625</td>
<td>54.9375</td>
<td></td>
</tr>
<tr>
<td>-45.0625</td>
<td>-46.0625</td>
<td>-40.0625</td>
<td>56.9375</td>
<td></td>
</tr>
<tr>
<td>-65.0625</td>
<td>-65.0625</td>
<td>-65.0625</td>
<td>45.9375</td>
<td></td>
</tr>
<tr>
<td>-65.0625</td>
<td>-62.0625</td>
<td>-65.0625</td>
<td>54.9375</td>
<td></td>
</tr>
</tbody>
</table>

**TABLE III.** AFTER APPLYING AFFINE TRANSFORMATIONS ON IMAGE Fractal4 GIVES (4 X 4) MATRIX

**Conclusion**

In this paper, we use Edge based neighborhood region search method to reduce the number of computations for fractal image compression. An Embedding the edge property of the block into the search process speed up rate can be increased further. We construct a coordinate system using two DCT coefficients F(1,0) and F(0,1) i.e. F(1,0) related to lowest vertical coefficients whereas F(0,1) related to lowest horizontal coefficients. In this paper, eight affine transformations are used, i.e. k=0 to k=7. The value of k is restricted by using classification scheme based on edge type classifier k=0 to k=3 and k=4 to k=7. Because of this restriction of k, searching space is reduced, i.e. which will avoid unnecessary MSE computations.
In encoding process search spaces are limited to reduce the encoding time. The experimental result shows that, at the encoding speed, proposed method requires less time as to compare with other methods. Comparison to Duh’s classification Number of computations are reduced by using the proposed method. Also compare with full search algorithm, Duh’s classification methods, PSNR of the reconstructed image is also get increased by using proposed method and MSE is getting reduced. By experimental results we can also see that the quality of the reconstructed image is also maintained and searching space is also getting reduced. Using this method PSNR is also increasing as compared with other methods, quality of retrieved image is also improved and compression ratio is also increasing.

As the number of computations decreases, the time required to encode the image is automatically decreasing. Comparing with the above methods, PSNR of the image is enhanced by 27.5dB by the proposed method.

REFERENCES