An Improved Algorithm of Fractal Image Compression

Anupam Garg
Bhai Gurdas Institute of
Engineering & Technology,
Sangrur

ABSTRACT
The need for compression is to minimize the storage space and reduction of transmission cost. When a digital image is transmitted through a communication channel, the cost of the transmission depends on the size of the data. The only way currently to improve on these resource requirements is to compress images such that they can be transmitted quicker and then decompressed by the receiver. There are many applications requiring image compression such as multimedia, internet, satellite imaging, remote sensing, preservation of art work, etc. Numerous methods for image compression have been presented in the literature survey but there is always a scope for improvement. In current work the fractal image compression has been employed as an efficient method in image compression. A novel compression encoding technique using hard threshold has been proposed based on fractal image compression and the results are compared with the other state of art image compression methods. The proposed method reduces the Encoding time significantly while some what compromising with the quality of the image. The initial experiments show that the proposed approach could achieve smaller encoding time and higher compression ratio on images. The proposed algorithm exhibits promising results from quantitatively and qualitatively points of view.

Keywords
Compression, Contractive transform, Fractal, Iterative Function System, Thresholding.

1. INTRODUCTION
1.1 Image Compression
Compression is the process of reducing the size of a file or of a media such as high-tech graphical images etc, by encoding its data information more efficiently. By doing this, there is a reduction in the number of bits and bytes used to store the information. Therefore, a smaller file or image size is generated in order to achieve a faster transmission of electronic files or digital images and a smaller space required for downloading.

Compression is done by using compression algorithms that rearrange and reorganize data information so that it can be stored economically.

By encoding information, data can be stored using fewer bits. This is done by using a compression/decompression program that alters the structure of the data temporarily for transporting, reformatting, archiving, saving, etc.

Compression reduces information by using different and more efficient ways of representing the information. Methods may include simply removing space characters, using a single character to identify a string of repeated characters or substituting smaller bit sequences for recurring characters. Some compression algorithms delete information altogether to achieve a smaller size.

Depending on the algorithm used files can be greatly reduced from its original size.

1.2 Lossy vs Lossless Compression
Depending on the detail present, compression can be categorized in two broad ways:

Lossy Compression: Its aim is to obtain the best possible fidelity for a given bit rate or minimizing the bit rate to achieve a given fidelity measure. It reduces a file by permanently eliminating certain information especially redundant information. When the file is uncompressed, only a part of the original information is only present, although the user may not notice it. Used for images, video or sound where a certain amount of information loss will not be detected by most users and the loss of quality is affordable.

Lossless Compression: In this data is compressed and can be reconstituted without loss of detail or information. This is referred to as bit- preserving or reversible compression systems. To achieve this, algorithms create reference points for things such as textual patterns, store them in a catalogue and send the along the smaller encoded file. When uncompressed, the file is regenerated by using those documented reference points to re-substitute the original information [2] [7]. It is a form of compression in which data files are split up into different chunks and reorganized to optimize them. This sort of compression very rarely saves much space, but it is ideal for transporting enormous files by breaking them into easier-to-handle pieces. Lossless compression is used when every bit of data is needed in the end product, often when transmitting a file to a designer. a lossless compression allows the designer to be sure that any data they may want to alter will be there, letting them create a final product before compressing the file further using a lossy compression. Lossless compression is ideal for documents containing text and numerical data where any loss of textual information can not be tolerated.

The advantage of lossy methods over lossless methods is that in some cases a lossy method can produce a much smaller compressed file than any lossless method, while still meeting the requirements of the application. If an image is compressed, it needs to be uncompressed before it can be viewed. Some processing of data may be possible in encoded form. Lossless compression involves some form of entropy encoding and is based in information theoretic techniques whereas lossy compression use source encoding techniques that may involve transform encoding, differential encoding or vector quantization.

2. FRACTAL IMAGE COMPRESSION TECHNIQUE

Compression is a method that allows images to be stored on computers using much less memory than the original image. Fractal Image Compression is a method to encode the image in a way that would require less storage space by using the self-similar nature of an image [2].

Michael Barnsley was the first person to use the idea of fractals in image compression. It has been claimed that fractal coding reach compression ratios upto 10000:1 which is a impressive rate of compression. He suggested that by using repeated iterations of affine transformation of the plane, one could reproduce a fractal like image by storing the image as a collection of transformations rather than a collection of pixels. Fractals are complicated looking images that arise from simple algorithms. These images are generated by iterative execution of simple algorithms at different scales. This property is called self-similarity which is used to re-construct the image. Fractal Image Compression is a type of lossy image compression.

Fractal Compression: It is the technique that relies on the fact that in certain images, parts of the image resemble other parts of the same image. Fractal algorithms convert these parts or more precise geometric shapes into mathematical data called fractal codes which are used to recreate the encoded image. Once an image has been converted into fractal code, its relationship to a specific resolution has been lost, it becomes resolution independent. The image can be recreated to fill any screen size without the introduction of image artifacts or loss of sharpness that occurs in pixel based compression schemes.

Fractal image compression works like a photocopier machine which takes an input image, reduces the image into half and produces three copies of the original image as the output [6]. The way the input image is transformed determines the final result when running the copy machine in feedback loop.

There must be a constraint on the transformations that they must be contractive i.e. given transformation applied to any two points in the input image must bring them closer in the copy.

A transformation W is said to be contractive if for any two points P1, P2, the distance is

D (w (P1), w (P2)) < sd (P1, P2)

for some s<1, d = distance

If a transformation is contractive then when applied repeatedly starting with any initial point, we converge to a unique fixed point. If X is a complete metric space and W: X→ X is contractive, then W has a unique fixed point |W|. This tells how to expect the collection of transformation to define an image.

This condition is quite logical because if points in the copy were spread out the final image would have to be of infinite size. Except this condition, transformation can have any form.

It maps a plane to itself. General form of affine transformations can be of the form:

$$W_i \begin{vmatrix} x \\ y \\ z \end{vmatrix} = \begin{vmatrix} a b 0 \\ c d 0 \\ 0 0 s \end{vmatrix} \begin{vmatrix} x \\ y \\ z \end{vmatrix} + \begin{vmatrix} e \\ f \\ o \end{vmatrix} \quad \text{or}$$

$$W_i \begin{vmatrix} x \\ y \end{vmatrix} = \begin{vmatrix} a_i b_i \\ c_i d_i \end{vmatrix} \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} e_i \\ f_i \end{vmatrix}$$

(1)

Where x, y and z are coordinates and a, b, c, d form linear part which determines rotation, skew and scaling, e and f are the translation distances in the x and y directions resp., s and o are contrast and brightness adjustments for the transformation [4].

Fractal Image Compression also known as fractal image encoding because compressed images are represented by contractive transforms. These transforms are composed of a collection of a number of affine mappings on the entire image known as IFS (Iterated Function System).

3. PROPOSED ALGORITHM

The work focuses on developing a new algorithm based on fractal image compression. The motivation behind using fractal image compression as it has been claimed that fractal coding may reach compression ratios up to 10000: 1 which is an impressive rate of compression but it suffers from problems such as expensive encoding time. So, fractal image compression is improved in order to obtain the impressive compression ratio and to reduce the encoding time. The fractal image compression also allows the images to be stored in the lesser computer memory. The proposed algorithm is shown in Fig. 1.

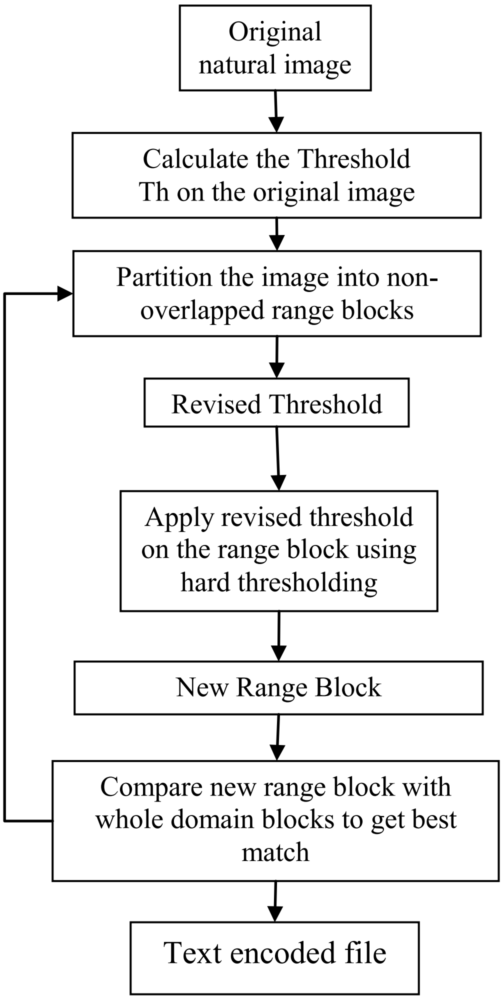


Fig.1 Diagrammatic depiction of Proposed Algorithm

Steps of the proposed algorithm is as follows:

StepI: Consider the input image i.e. .bmp'.

Step II: Compute the threshold on the input image, th.

Step III: Choose the size of the range blocks and domain blocks such that

$$domain\ block size = range\ block\ size * 2 \quad (2)$$

Partition the input image into non-overlapped blocks of ranges of square sizes covering the whole image [6].

Step IV: Revise the above defined threshold using the equations:

$$\mathbf{th = th * 0.1} \quad (3)$$

$$\mathbf{th_R = th / m} \quad (4)$$

where m is the mean of the values of the range block

The use of mean in fractal image compression decreases the encoding time [8].

Step V: Perform Hard Thresholding on each range block in order to obtain the new range block.

Hard Threshold is used as follows:

$$range(r,c) = \begin{cases} range(r,c) & range(r,c) >= th_R \\ 0 & range(r,c) < th_R \end{cases}$$

(5)

where range(r,c) is the pixel value of the range block.

Step VI: Partition the image into non-overlapping domain blocks and rescale the domain blocks tothe size of the range blocks.

Step VII: Construct the possible affine transformation of each domain block and compare each range block with the whole domain blocks to find the best match.

Step VIII: Save the location of the domain, best transformation, contrast and offset coefficients in the text file which is known as **Encoded text file,** which is used for transmission and the image is compressed.

Step IX: Continue to do the same for the rest of the range blocks until it reaches the last range block.

4. EXPERIMENT AND RESULTS

The proposed algorithm has been tested qualitatively as well as quantitatively on gray scale images. The performance of the proposed algorithm has been conducted on some test images.

In this paper, there is use of fractal image encoding with thresholding and decoding for compression purpose. The fractal image encoding with thresholding based compression reduces the encoding time as well as provides higher compression ratio [1] [9]. Hard thresholding is used in the proposed method.

The fractal encoding algorithm with threshold uses self-similarity sets of the image along with thresholding which speeds up the encoding process because after getting the suitable value, it will stop finding the matches for range and domain blocks. Matching the suitable domain block with range block step is computationally very expensive, so the proposed technique tries to fasten this process.

Firstly we applied the encoding algorithm of the proposed algorithm whose result is a text file. This text file acts as the input to the decoder which contains the information related to the range size in the encoded form. In this paper, the range size used is 4, 8 and 16. Then the decoding algorithm is applied in order to decompress the image. The output shown in Table I and II is the original image, decompressed image and error image for each range block size specified above.

TABLE I: Results of Proposed Algorithm for range block size 4

Method	**Range block size= 4**
Pure Fractal Encodin g	
Propose d Method	

TABLE II: Results of Proposed Algorithm for range block size 8 & 16

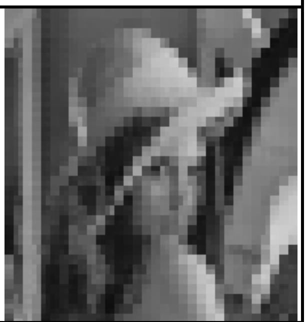
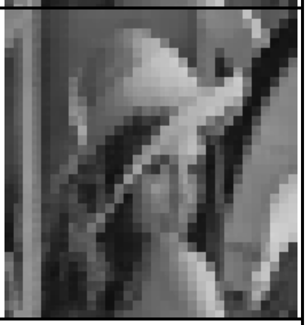
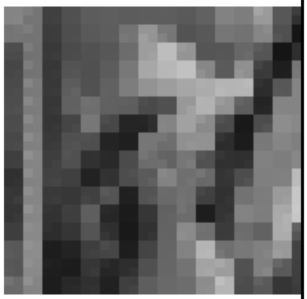
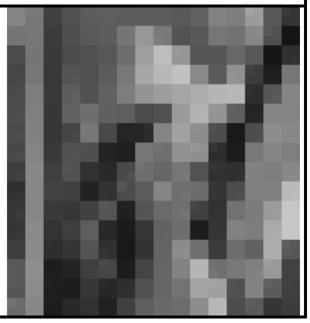
Method	**Range block size= 8**
Pure Fractal Encoding	
Proposed Method	
Method	**Range Block Size=16**
Pure Fractal Encoding	
Proposed Method	

TABLE III: Comparison for Encoding Time Quality Metric (in seconds)

Method	**Encoding Time for Range block size = 4**	**Encoding Time for Range block size = 8**	**Encoding Time for Range block size = 16**
Pure Fractal Encoding	1.5839e+003	139.660	24.8600
Proposed Method	**1.5031e+003**	**111.6400**	**13.4370**

Compression Ratio for Pure Fractal Image Compression	**Compression Ratio for Proposed Compression Method**
98.38	98.38

TABLE IV: Comparison for Encoding Time Quality Metric (in seconds)

4. CONCLUSION

In table III, we observed that the encoding time of the proposed method is lesser than pure fractal image compression. Also, with the increase in range block size the encoding time decreases. The proposed method maintains the image compression ration as that of pure fractal image compression as shown in table IV.

The combination of multiple methods comes from a wide investigation into research papers. A New method for image compression has been developed which uses fractal image compression encoding algorithm with thresholding and then fractal decoding algorithm to get decompressed image.

After taking fractal encoding algorithm, a new developed thresholding method is applied to the whole image and then revised threshold is applied to the obtained range blocks with hard thresholding function to get the new range blocks. Decompressed image is obtained after applying fractal decoding algorithm. The proposed method is tested against pure fractal encoding algorithm.

The fractal based image compression has the disadvantage of having long encoding time with some compromise with PSNR. The proposed method improves the encoding time and provides higher compression ratio.

5. REFERENCES

[1] Al-lahan, M. and El Emary, I. M. M. (2007), "*Comparative Study between Various Algorithms of Data Compression Techniques*", International Journal of Computer Science and Network Security, Vol. 7, Issue 4, pp. 281-291.

[2] Baligar, V. P., Patnaik, L. M. and Nagabhushana, G. R. (2006), "*Low complexity and high fidelity image compression using fixed threshold method*" , International Journal of Information Sciences, Vol. 176, Issue 6, pp. 664-675.

[3] Galabov, M. (2003), "*Fractal Image Compression*", International Conference on Computer Science & Technologies-CompSys Tech 2003, pp. 320-326.

[4] IMAGE PROCESSING http://en.kioskea.net/contents/video/traitimg.php3

[5] Kamali, K. (2005), "*Fractal Video Compression*", University of Southern Queensland.

[6] Sindhu, M. and Rajkamal, R. (2009), "*Images and its compression techniques- A Review*", International Journal of Recent Trends in Engineering, Vol. 2, Issue 4, pp. 71-75.

[7] Wi, Y., Huang, M. and Wen, Y. (2003), "*Fractal Image Compression With Variance and Mean*", International Conference on Multimedia & Expo, Vol.1, pp. 353-356.

[8] Wohlberg, B. and de Jager, G. (1994) *"On the reduction of fractal image compression encoding time"*, IEEE South African Symposium on Communications and Signal Processing (COMSIG '94), pp. 158–161.

[9] Xi, L. and Zhang, L. (2007), *"A Study of Fractal Image Compression Based on an Improved Genetic Algorithm"*, International Journal of Nonlinear Science, Vol.3, Issue 2, pp. 116-124.