Analysis of F-W Image Compression with and without threshold Value

Gurdeep singh [1], **Pooja sharma** [2]
Student, Dept. Of ECE, GGGI,Dinarpur,Ambala Haryana,India[1],

Assistant professor,Dept.of ECE, GGGI,Dinarpur,Ambala Haryana,India[2],

ABSTRACT

Fractal image compression is a lossy image compression technique to achieve high level of compression while preserving the quality of the decompressed image close to that of the original image. The method relies on the fact that in a certain images, parts of the image resemble other parts of the same image (self-similarity). Wavelet has multi- frequency characteristics, and there is self similarity among the sub images decomposed by wavelet.

In this paper we show the implementations of wavelet based fractal image compression technique in which we have use the threshold value for reducing the redundancy of domain block and search block. We applied this technique on the numerous images in order to investigate the compression ratio and corresponding quality of the images using peak signal to noise ratio (PSNR) at different iteration levels. We also analyze the reconstructed images at different level of iterations with threshold and without threshold value. We also analyze the result of reconstructed images with and without threshold at different iterations using graph.

***Keywords*:** *fractal image coding; wavelet; MSE; Compression Ratio; Threshold value.*

I. INTRODUCTION

In 1988 M. Barnsley and Jacquin introduced the FRACTAL image compression techniques are the product of the study of iterated function systems (IFS). For recent years, the application of fractal image coding has become more and more popular. These techniques involve an approach to compression quite different from standard transform coder-based methods. Transform coders model images in a very simple fashion, namely, as vectors drawn from a wide-sense stationary random process. They store images as quantized transform coefficients. Fractal block coders, as described by Jacquin, assume that “image redundancy can be efficiently exploited through *self-transformability* on a blockwise basis” [1]. They store images as contraction maps of which the images are approximate fixed points. Images are decoded by iterating these maps to their fixed points.

Fractal coding is based on fractal geometry, it has a character of big compression ratio and a fast decoding speed, but it cannot be used for real time processing. It is its blocks searching and matching that makes its long time. As wavelet can get good space frequency multi resolution, the energy mainly concentrated in low frequency sub images, and the images with same directions but different resolutions have self similarity, which is consistent with fractal’s nature properties. Recently, much reaching work has focused on fractal coding by using wavelet. It is just at the beginning, but some research results improved this method is practical. The combination of wavelet and fractal is firstly proposed by Pentland and Horowitz. They wanted to find the redundancy of sub images decomposed after wavelet. Later, Rinaldo and Calvagno proposed a new method. First, decompose a image by wavelet, and then code the sub image with minimum resolution, and predict the other sub images. Finally, we’ll finish the compression. Jin Li introduced a new method. They firstly computed the bytes of fractal predicting, and only predicted when economization. But the methods above are all time consumption, and the reconstructed images are not always good. In this paper we proposed a new blocks searching method based on wavelet based fractal technique in which we use the minimum threshold value for reducing the redundancy more and more. And then analyses the result of numerous images. Firstly, we transform the image by wavelet, then divide it into blocks. Before matching, we first reduce the amount of domain blocks and the range blocks to lessen the block pools, then following the contractive mapping transformation.

II. FRACTAL COMPRESSION

A. *FRACTAL*

Fractal is a structure that is made up of similar forms and patterns that occur in many different sizes. The term fractal was first used by Benoit Mandelbrot to describe repeating patterns that he observed occurring in many different structures. These patterns appeared very similar in form at any size although with rotation, sale or flipping. Mandelbrot also is covered that these fractals could be described in mathematical terms. In fractal theory, the formula needed to create a part of the structure can be used to build the entire structure.

B. PURE-FRACTAL IMAGE COMPRESSION ALGORITHM

In Pure-fractal image compression algorithm, an image is regularly segmented into two-dimensional array of $B\times B$ range blocks, specified by R_{ij} . i and j identify the position of the block in the image. For each range block, a $2B\times 2B$ domain block, specified by D_{ij}, is considered in which the transformed D_{ij} matches R_{ij}. If the number of domain blocks in row and column directions are specified by m and n. The domain block pool, $D_{i,j}$ ={i= 1:m , j= 1:n } is generated by sliding a $2B\times 2B$ window within the original image, skipping δ pixels from left to right, top to bottom. The affine transformation mapping domain block into the corresponding range block is $\tau = T \circ S$. Here, S is an average operator which is given by Equation 1.

$$S(\mu_{k,l})= \mu_{2k,2l}+ \mu_{2k+1,2l}+ \mu_{2k,2l+1}+ \mu_{2k+1,2l+1} \quad (1)$$

Where, k and l specify the number of each cell in a 2- by-2 block ($\mu_{k,l}$). And T is given by Equation 2.

$$T(\mu_{k,l})= s.\ \mu_{k,l} + g \quad (2)$$

where s is known as a scale factor ($0\leq s<1$) and g is translation. Range blocks are searched among the domain pool to minimize the following distortion:

$$\min |R_i\text{-}s\check{D}_j\text{-}g| = \min|(R_i\text{-}\check{r})\text{-}s(\check{D}_j\text{-}đ)| \quad (3)$$

where, $g = \check{r} - s\,đ$; and $\check{r}$ and đ are the means of $D_{i,j}$ and $R_{i,j}$,and $D_{i,j}$ is the down sampled version of the domain block. The s and g are calculated using Equations 4 and 5.

$$S=\frac{N^2\sum_{m,n}(Di,j)m,n.(Rk,l)m,n-(\sum_{m,n}(Di,j)m,n)(\sum_{m,n}(Rk,l)m,n)}{N^2\sum_{m,n}((Di.j)m,n)^2-(\sum_{m,n}(Di,j)m,n)^2} \quad (4)$$

$$g = \frac{(\sum_{m,n}(Di,j)\,m,n)^2-(\sum_{m,n}(Rk,l)\,m,n)^2}{N^2\sum_{m,n}((Di,j)m,n)^2-(\sum_{m,n}(Di,j)m,n)^2} \quad (5)$$

Number of the blocks in horizontal and vertical axes, respectively, D i j, is the domain block in the *(i,j)* coordinate, and R_{lk} , is the range block in the *(l,k)* coordinate. The transformed domain block (i.e., the best approximation for the current range block) is assigned to that range block. The coordinates of the domain block along with its scale and offset are saved into a file called Fractal Code Book (FCB) as the compressed parameters. The FCB is saved as the compressed version of the original image. The decompression process is based on an iterative simple algorithm. It is started with a random initial image and usually after eight iterations [10-11, 4], the decoded image is obtained. We have tested the decompression process with different number of iterations as well. There was however no challenge in this part. We also ran the decompression process with different initial images to reach a better quality image but the outcome was with a negligible difference.

III. WAVELET COMPRESSION

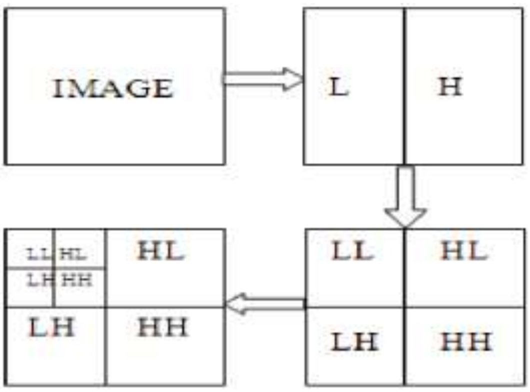


Fig. 1: Two levels Wavelet Decomposition applied on an image

Wavelet Theory deals with both discrete and continuous cases. Continuous wavelet transform (CWT) is used in the analysis of sinusoidal time varying signals. CWT is difficult to implement and the information that has been picked up may overlap and results in redundancy. If the scales and translations are based on the power of two, DWT is used in the analysis. It is more efficient and has the advantage of extracting non overlapping information about the signal. 2-D transform can be obtained by performing two 1-D transform. Signal is passed through low pass and high pass filters L & H, then decimated by a factor of 2, consisting 1 level transform, thus splitting the image into four sub-bands referred as LL, HL, LH & HH (Approximation, Horizontal Detail, Vertical Detail, and Diagonal Detail respectively). Further decomposition is achieved by acting upon four sub-bands. The inverse transform is obtain
ed by up sampling all the four sub bands by a factor of 2 and then using reconstruction
filter. Higher scales correspond to more stretched wavelet. [6, 7].

IV. WAVELET-FRACTAL IMAGE COMPRESSION ALGORITHM

The motivation for Wavelet-fractal image compression stems from the existence of self-similarities in the multi-resolution wavelet domain. Fractal image compression in the wavelet domain can be considered as the prediction of a set of wavelet coefficients in the higher frequency sub bands from those in the lower frequency sub bands. Unlike Pure-fractal estimation, an additive constant is not required in wavelet domain fractal estimation, as the wavelet tree does not have a constant offset. Down sampling of domain tree, matches the size of a domain tree with that of a range tree. The scale factor is then multiplied with each wavelet coefficient of domain tree to reach its correspondence in range tree. The authors of [1] answered the question "why fractal block coders work" comprehensively referring the fundamental limitations of the Pure-fractal compression algorithms [7]. Let D_l denote the domain tree, which has its coarsest coefficients in decomposition level *l*, and let *Rl-1* denote the range tree, which has its coarsest coefficients in decomposition level *l-1*. The contractive transformation (*T*) from domain tree *Dl* to range tree *Rl-1,* is given by T(Di)= α x S.Di where S denotes sub sampling and α is the scaling factor.

Let x= (x1, x2, x3, x4 ,...... xn) be the ordered set of coefficients of a range tree and y= (y1, y2 ,y3 ,y4 ,......, yn) the ordered set of coefficients of a down sampled domain tree. Then, the mean squared error is given by Equation 5.

MSE= $\|R_{l-1} - T(D_l)\|^2 = \sum_{i=1}^{n} (xi - \alpha \text{ x yi})^2$.....(7)

And the optimal α is obtained by Equation 8.

$\alpha = (\sum_{i=1}^{n} xi * yi) / \sum_{i=1}^{n} yi^2$........................(8)

We should search in the domain tree to find the best matching domain block tree for a given range block tree. The encoded parameters are the position of the domain tree and the scaling factor. It should not be left unmentioned that in this algorithm; the rotation and flipping have not been implemented. To increase the accuracy of scale factors, new scheme of Wavelet fractal compression is introduced. In this approach, α in contrast to the previous method which had to be calculated for each block tree individually, is computed for each level separately, hence the more α s and the better quality achieved.

V. METHODOLGY FOR IMPROVING FRACTAL WAVELET COMPRESSION TECHNIQUE

A. Principles of improving

Energy of an image after wavelet transformed mainly concentrated in the low frequency sub image. According to the human vision mechanism, the main vision of people is sensitive to the low frequency information, but not sensitive to the high part. So we take lossless compression to the low frequency information. Previous fractal compression directly divided the original image into range blocks and domain blocks, then affine transform the range blocks and matches with domain blocks. Finally compressing and coding. However, we choose to reduce the redundancy among domain and range blocks before matching, because there are many similar blocks in the block pools. After this, less domain blocks will be left, and less time will be consumed. Here we use Mean Square Error (MSE) to judge the degree of similarity among domain and range blocks. The domain blocks' algorithm is described as follows:

- Assume threshold value= δ and assume all the flags=0.
- Set minimum value of δ, and compute the Error, i.e. MSE of Domain Block.
- Sequence Error from small to large.
- Compare Error and δ, if Ei is less than δ then delete that block, then set flag=1, And save that in affine transformation.
- If Error is not less than δ then consider the MSE with minimum value as a best match. Then repeat the previous step.
- Further for finding out the best threshold value, first of all we set the minimum threshold value, then we vary that value in some range for getting the optimum result.

After simple screened, the representative blocks will be left, the redundancies of the pool have been removed. In the same way, the redundancy of range blocks can be removed. As range blocks are much more important than former, the initial threshold should be smaller.

Divide a 256x256 image into 8×8 sub image blocks. If we set the step size to 8, there will be 32×32=1024 blocks. There will be more similar blocks after averaging 4 neighbor pixels, which makes this algorithm more practical and rational [5].

B. Transform the image with wavelet

First of all, decompose the image with 3scale wavelet, and then process the low frequency and high frequency data separately as below.

C. Processing of low frequency data

Low frequency sub image occupies more than 85% of the whole sub images' energy. It has a large amount of data, a big self similarity and it also contains much important information. We choose to code the low frequency sub image with lossless predictive coding. Concrete steps are as follows: Transform a image with 3scale wavelet, we'll get large low frequency coefficients, and they are very close. Then difference the coefficients of low frequency part, and code the results with Huff man coding, generating the low frequency compression data. To combine the results of low frequency and high frequency part, we will get the coding result of original image.

D. Processing of high frequency data

Here we choose a new method. Search and match the domain blocks and range blocks whose redundancy has been reduced. Record the position of each block when reducing the redundancy, then following the fractal coding.

VI. EXPERIMENTAL RESULTS

In this we have shown the experiments of several images with and without threshold. Below I have given the experimental result of image of sachin and mahi without and with threshold value. In which I have shown the variation of Peak signal to noise ratio and decoding time at different iterations. Here by doing this we can conclude the best decoding time at optimum peak signal to noise ratio.

A. Using Fractal Image Compression Technique in Wavelet Domain without Threshold for image of Gurdeep

In this domain we achieve the compression ratio of 85.1513 and encoding time is 109.3430 seconds

Below we have shown decoded images at different iterations 1, 3, 5, 7.

Below we have shown the table in which we show that how the PSNR increases by increasing iteration number and we can find the minimum decoding time at optimum PSNR.Table1 showing PSNR, Decoding Time for different Iterations for image of Gurdeep

ORIGINAL IMAGE OF GURDEEP

(a) (b)

(c) (d)

Fig3. Decoded Images at different Iterations (1, 3, 5, 7)

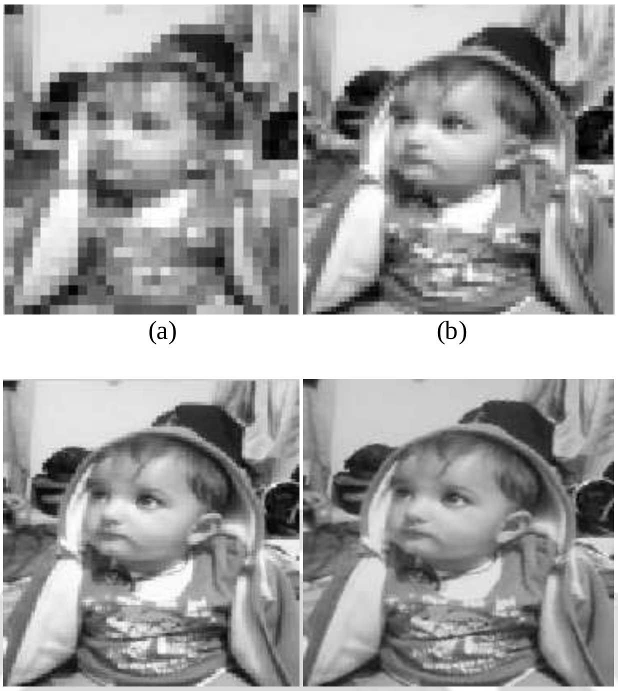
TABLE 1 for gurdeep

Iteration No.	PSNR	Decoding Time (Seconds)
1	22.6199	1.9220
2	26.2271	3.3280
3	31.1831	4.6570
4	33.9630	6.0000
5	34.7396	7.3750
6	34.7396	8.7820
7	34.7396	10.1410
8	34.7396	12.7450

B. Using Fractal Image Compression Technique in Wavelet Domain without Threshold for image of Mahi

In this domain we achieve the compression ratio of 85.1513 and encoding time is 110.2030 seconds
Below we have shown the original image and the decoded images of MAHI at different iterations 1, 3, 5, 7.

Fig4. Original Image of Mahi



(a) (b)

(c) (d)

Fig5. Decoded Images at different Iterations (1, 3, 5, 7)

Table showing PSNR, Decoding Time for different Iterations for image of Mahi

TABLE 2 for mahi

Iteration No.	PSNR (dB)	Decoding Time (Seconds)
1	20.0523	1.8440
2	23.7026	3.3440
3	27.9967	4.7030
4	30.6742	6.0470
5	31.3298	7.4060
6	31.3298	8.7340
7	31.3298	10.0940

C. PSNR and Decoding Time for Different Images without threshold

In this table we have shown the comparison of experimental result of two images in which we have shown the PSNR, decoding time and encoding time.

Table 3 for different images

Images	PSNR	Best Decoding Time (in seconds)	Encoding Time (in seconds)
Gurdeep	34.7396	7.3750	109.3430
Mahi	31.3298	7.4060	110.2030

D. Using Fractal Image Compression Technique in Wavelet Domain with Threshold for Image of Gurdeep

In this domain we achieve the compression ratio of 85.1513 and encoding time is 7.1880seconds

Below we have shown the original image and the decoded images of MAHI at different iterations 1, 3, 5, 7.

(a)

(b)

(c)

(d)

Fig6. Decoded Images at different Iterations (1, 3, 5, 7)

TABLE 4 for image of gurdeep

Iteration No.	PSNR (dB)	Decoding Time (Seconds)
1	22.6199	2.4220
2	26.2271	4
3	27.5891	5.5000
4	27.8941	6.9850
5	27.9297	8.5470
6	27.9297	10.0780
7	27.9297	13.8910
8	27.9297	13.9090

E. Using Fractal Image Compression Technique in Wavelet Domain with Threshold for Image of Mahi

In this domain we achieve the compression ratio of 85.1513 and encoding time is 7.8590 seconds
Below we have shown the original image and the decoded images of MAHI at different iterations 1, 3, 5, 7.

(a) (b)

(c) (d)

Fig7. Decoded Images at different Iterations (1, 3, 5, 7)

Table showing PSNR, Decoding Time for different Iterations for image of Mahi with threshold value

TABLE 5 for image of mahi

Iteration No.	PSNR (dB)	Decoding Time (Seconds)
1	20.0523	1.9370
2	23.7026	3.3130
3	24.7575	5.2030
4	24.9036	6.0470
5	24.9158	7.3440
6	24.9158	8.8440
7	24.9158	10.1880
8	24.9158	11.1880

F. PSNR and Decoding Time for Different Images with threshold

In this table we have shown the comparison of experimental result of image in which we have shown the PSNR, decoding time and encoding time.

Table6 for MAHI

Images	PSNR	Best Decoding Time (in seconds)	Encoding Time (in seconds)
Mahi	24.9158	7.3440	7.8590

G. Graph between PSNR and Iterations with and without threshold for Mahi

This graph showing the relation between PSNR and iteration with threshold and without threshold.
In this it is shown that if we use the threshold PSNR value becomes constant on low value and on low number of iteration. And if we do not use threshold value then in that case PSNR value becomes constant on higher value.

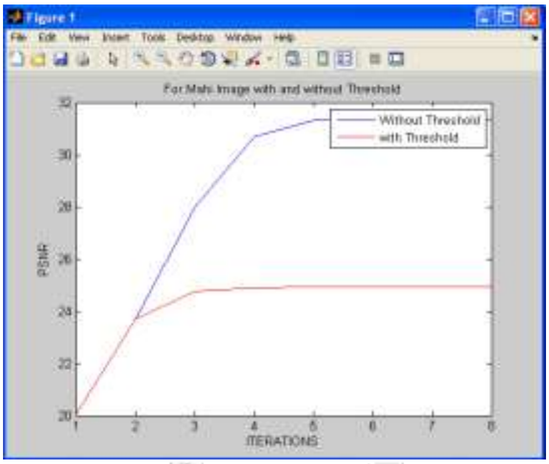


Figure8. *Graph between PSNR and Iterations*

VII. Comparison of our proposed technique with JPEG compression technique

Here we have shown the comparison of wavelet based compression technique and JPEG compression technique. In JPEG compression technique we get very low compression ratio while in case wavelet based fractal technique we achieve very high compression ratio. And in case of Peak signal to noise ratio, JPEG has little edge on this, but it does not affect very much. Because PSNR in case of fractal is nearby of JPEG technique. Initially JPEG also has very low encoding and decoding time but in case of wavelet based fractal technique we also achieve very low encoding time by using threshold value by reducing the redundancy in matching of domain block and range block.

Table7 for mahi and gurdeep

Images	PSNR For JPEG	PSNR For Wavelet Based Fractal	Compression Ratio For JPEG	Compression Ratio Wavelet Based Fractal
Mahi	34.978	24.9158	63.635	85.1563
Guedeep	33.574	23.0327	66.556	85.1563

VIII. CONCLUSION

In this paper, we evaluated wavelet based fractal image compression technique with and without threshold value. And shows various threshold values to get the Encoding/Decoding process as faster as possible, that's exactly the point where it fallen below the JPEG standard. In this paper we also analyze PSNR values for numerous images at various iteration levels.

REFERENCES

[1] G. Davis, "A wavelet-based analysis of fractal image compression," IEEE Trans. Image Process., vol. 7, pp. 141-154, 1998.
[2] M. Polvere and M. Nappi, "Speed-Up in Fractal Image Coding: Comparison of Methods", IEEE transactions on image processing, vol. 9, no. 6, p. 1002-1009, 2000
[3] Zeng Wenqu, Wen Youwei, Sun Wei, "fractal wavelet and image compression", Dongbei University Press, 2002.10.
[4] Kenneth R.Castleman, "Digital Image Processing", Qinghua University Presss, 2003.11
[5] Jiang Haijun, "Study on Fractal Image Coding", CNKI:CDMD:2.2004.129298
[6] Lotfi A. A., Hazrati M. M., Sharei M., Saeb Azhang, *" CDF(2,2) Wavelet Lossy Image Compression on Primitive FPGA"*, IEEE, pp. 445-448, 2005
[7] Kharate G.K., Ghatol A. A. and Rege P. P., *" Image Compression Using Wavelet Packet Tree,"* ICGST-GVIP, Vol. 5, No. 7, pp. 37-40, 2005.
[8] Jose Oliver, Member, IEEE, and Manuel P. Malumbres, Member IEEE, "Low-Complexity Multiresolution Image Compression Using Wavelet Lower Trees", IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 16, NO. 11, NOVEMBER 2006.
[9] M. Wang, Z. Huang and C. H. Lai, "Matching search in fractal video compression and its parallel implementation in distributed computing environments," Appl. Math. Model., vol. 30, pp. 677-687, 2006.
[10] Lu Jingyi, Wang Xiufang, Wang Dongmei, "Fractal Image Coding

Algorithm of Design and Realization in Wavelet Domain", Proceedings of 2006 Chinese Control and Decision Conference.

[11] Mohammad. R. N. Avanaki, Hamid Ahmadinejad, Reza Ebrahimpour, "Evaluation of Pure-Fractal and Wavelet-Fractal Compression Techniques" ICGST-GVIP Journal, ISSN: 1687-398X, Volume 9, Issue 4, August 2009

[12] Hasan F. Ates, Member, IEEE, and Michael T. Orchard, Fellow, IEEE, "Spherical Coding Algorithm for Wavelet Image Compression" IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 18, NO. 5, MAY 2009