

# A Fast Encoding Method for Fractal Image Compression

Suryabhan Singh Hada  
Department of Mathematical Sciences  
Indian Institute of Technology (BHU)  
Varanasi 221005, India

## ABSTRACT

The present work is concerned with a review of various techniques of fractal image compression. Firstly variance-ordered domain block search algorithm and then Randomized domain block search algorithm is discussed. An analysis is performed on both of them and on the basis of the analysis a new algorithm is proposed. After that a detailed comparative analysis is performed by implementing the algorithms in Matlab. For our comparison we compare the PSNR of the reconstructed image.

## Keywords

Fractal image fast encoding, Variance based search, Random search.

## 1. INTRODUCTION

Fractal image compression is a lossy compression method for digital images. It is based on the fact that one part of an image resembles with another. So, one part of an image can be generated with the other. The fractal image compression is done using Iterative function system (IFS)[1].

This self-similarity provides very high compression ratio, but Fractal image takes huge time to encode the image. Hence number of techniques is applied to overcome this problem. In section 2, two such methods of image compression technique are discussed. First Variance-Ordered domain block search method [2] is discussed and then second method compression based on Randomization is discussed[3]. In the section 3, an alternative algorithm is proposed and an analysis is also performed to show the efficiency of the proposed algorithm. For this algorithms are implemented in Matlab and results are analyzed in order to compare the quality of image and the time taken in compression process in the algorithms.

## 2. FRACTAL IMAGE COMPRESSION

The image is divided into non-overlapping blocks of size  $n \times n$ . Each block called range block ( $R_k$ ). The original image is also divided in  $2n \times 2n$  non-overlapping blocks called domain blocks ( $D_k$ ). For a  $128 \times 128$  image with block size of  $4 \times 4$ , there are 1024 range blocks and 14641 domain blocks. For a range block  $R_k$ , the closest match is searched in the domain pool. Each domain blocks is go through under 8 affine rotations, compressed to the size of range block. Once the best match is found it is assigned to the range block and stored into the codebook. But this full search is very time consuming process which makes the encoding process very slow.

To overcome this various techniques is applied to make this full search a partial search. Two of them are-

### 2.1 Variance-Ordered domain block search method

The range block approximation from domain block by affined transformation in Jacque algorithm [1] can be written as:

$$\widehat{R}_k = I_m(\alpha_n \times D_k + \Delta g)$$

where  $I_m$ ,  $\alpha_n$  and  $\Delta g$  are isometric transformation, scale factor of grey level, and shift factor of intensity respectively, and  $D_k$  is spatially contracted domain block. The luminance shift  $\Delta g$  is computed so that the average grey levels of the range block and scaled domain block are the same, that is

$$\Delta g = E(R_k) - \alpha_n \cdot E(D_k)$$

The best estimate,  $R_k$  can be obtained by minimizing the squared Euclidean (SED) defined as

$$d_E^2(R_k, \widehat{R}_k) = \sum_{0 \leq i, j \leq B} (R_k(i, j) - \widehat{R}_k(i, j))^2$$

where the block size of the range block is assumed to be  $B \times B$  pixels.

Sang-Moon Lee [2] proposed an algorithm in which instead of using Squared Euclidean distance (SED), Squared Variance difference (SVD) is used.

In searching for the closest domain block to a given range block in the SED sense, its correlation with the (SVD) is used, which is defined as

$$d_V^2(R_k, \widehat{R}_k) = (V_{R_k} - V_{\widehat{R}_k})^2$$

Where,

$$V_{R_k}^2 = \sum_{0 \leq i, j \leq B} (R_k(i, j) - E(R_k))^2$$

$$E(R_k) = \frac{1}{B^2} \sum_{0 \leq i, j \leq B} R_k(i, j)$$

An important observation in real image is that the greater the variance difference between two image blocks, the greater the SED becomes. This motivated the use of variance as a feature of the image block when comparing image blocks in the SED sense. That is to say, the similarity in the complexity of image blocks, SVD, is a nearly sufficient statistic for the SED. The relation between SVD and SED for any image block can be given by the inequality

$$d_e^2 \geq d_v^2$$

Therefore, if the search for the minimum SED domain block starts with the minimum SVD domain block and proceeds to the next nearest ones in the SVD sense, there is a high probability of arriving at the minimum SED domain block in only a few searching steps.

Let  $d_{min}^2$  be a known current minimum Euclidean distance of  $R_k$  represented a certain domain block. For any domain block  $D_k$ , so for  $\widehat{R}_k$ , if the inequality

$$(V_{R_k} - V_{\widehat{R}_k})^2 \geq d_{min}^2$$

is satisfied,  $D_k$  will not be the closest domain block to  $R_k$  and it is unnecessary to compute  $d_E^2(R_k, \widehat{R}_k)$ . This fact facilitated the early termination of our search process without

considering the possible existence of closer domain block in the remaining domain pool ( for this purpose, all of the domain blocks in the domain pool are sorted in descending order of variance).

## 2.2 Fractal image compression using randomization

The algorithm introduced by S.K. Ghosh[3], is similar in nature to Full search with a difference that instead of calculating SED for every domain block one by one we randomly choose domain blocks here. In this proposed algorithm the domain blocks are searched randomly for a finite number of trials, instead of an exhaustive search of domains. In this approach, for each range block R, the random search is performed for a finite number of times, say t. The implementation of the randomized approach can be summarized as follows:

- For each range block search domain blocks randomly for a fixed number of times.
- Whenever a domain block is found within QBT(qualifying block threshold) i.e. the minimum threshold error , break and select next range block.
- If no domain block is found within the QBT after the random trials, the domain with minimum error is selected.

## 3. PROPOSED ALGORITHM

Unlike Randomized algorithm search instead of searching the domain block randomly in the domain pool search process starts around the domain block with minimum SVD .This search terminated after a limit, which also overcome the problem of getting the condition-

$$(V_{R_k} - V_{D_{compress}})^2 \geq d_{min}^2$$

satisfied after large number of domain blocks searching.

Search process starts from the domain block having minimum SVD then search continues in upward and downward directions. If the termination condition

$$(V_{R_k} - V_{D_{compress}})^2 \geq d_{min}^2$$

is satisfied before the threshold the search process terminated otherwise the search process is terminated as threshold value is reached.

Therefore, for a 128×128 image, there will be maximum of threshold×1024 blocks which need to be traversed. Hence the maximum number of domain blocks processed throughout the process will be  $\frac{threshold}{14641}$  times than the blocks traversed in full search, which decreases the encoding time significantly with a very less decrease in the quality of the image considering huge drop in the encoding time.The encoding is given below:

For 128×128 image with 4×4 size range block the encoding process will be-

Step-1: Divide the whole image in to 1024 non-overlapping range blocks and 14641 overlapping domain blocks.

Step-2: Sort all the domain blocks in ascending order of their variances.

Step-3: Now move through the sorted domain blocks pool, to find the minimum SVD, let the selected domain block is  $U_{th}$  block in the pool.

Step-4: Now calculate the SED between given range block and the selected domain block and this SED is called  $d_{min}$

Step-5: Update the code book.

Step-6: Now, search in upward and downward direction of the  $U_{th}$  domain block, i.e.  $(U - 1)_{th}$  and  $(U + 1)_{th}$  blocks and then  $(U - 2)_{th}$  and  $(U + 2)_{th}$  blocks and so on.

Step-7: If SED becomes smaller then  $d_{min}$ , then assign that SED to  $d_{min}$  and update the code book accordingly.

Step-8: If for any domain block  $D_k$ , so for  $D_{compress}$  , if the inequality –

$$(V_{R_k} - V_{D_{compress}})^2 \geq d_{min}^2$$

is satisfied or threshold value is reached terminate search in that direction.

Step-9: Repeat step 2-7 for all the Range blocks.

## 3.1 Result

On implementing the proposed algorithm in Matlab on different images we get following results –

**Table 1. Experimental result of proposed algorithm on various images with various threshold values**

Image	Threshold	Encoding time (in seconds)	PSNR (in db)
Mandrill	200	2100	+33.54631
	100	1300	+33.268961
	50	723	+33.036294
	10	233	+32.23950
Lena	200	2312	+36.401087
	100	1507	+35.92127
	50	852	+35.549589
	10	252	+34.256054
Pepper	200	2345	+36.25454
	100	1504	+35.90595
	50	839	+35.321031
	10	271	+34.073559
Boat	200	2431	+34.86321
	100	1624	+34.60667
	50	830	+34.21689
	10	325	+33.31420
Marble	200	2433	+37.320102
	100	1674	+36.972205
	50	772	+36.524772
	10	239	+35.498871

Hence from the table it is very clear that there is not much change in the PSNR as we take only few blocks around the domain block with minimum SVD difference .Although there is a decrease in the quality but considering significant change in the encoding time this change is not much.

If only first 200 are selected blocks then the change in PSNR compare to the Variance based method is 0.968921 but the difference in encoding time is 19689 which is very huge. And when only 100 blocks are selected the change in PSNR not that much but encoding time again dropped significantly but if

move further the quality of the reconstructed image is decreases very much.

Hence if only first 100 domain blocks are chosen then a good quality of reconstructed image with significantly drop in the encoding time can be obtained.

#### 4. CONCLUSION

On implementing the algorithms in Matlab by considering the Mandrill's image of size 128x128 as the test image following results came:

**Table 2. Comparison between full search, Variance based algorithm and randomized algorithm**

Method's Name	Encoding time (in secs)	PSNR (in db)
Full search	113423	+34.515231
Variance ordered	21789	+34.515231
Randomized	8775	+34.076570
Proposed (200 threshold)	2100	+33.54631
Proposed (100 threshold)	1300	+33.268961
Proposed (50 threshold)	723	+33.036294
Proposed (10 threshold)	233	+32.23950

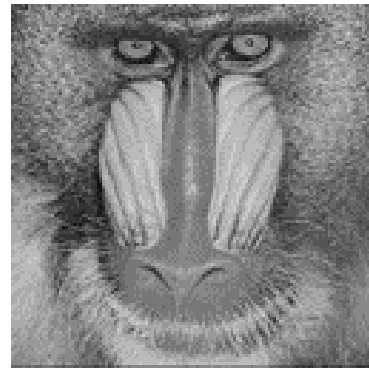
In Variance-ordered total number of blocks visited is 1,697,437 where the total number of blocks visited in Full search is 14,992,384. Hence only 11.322% blocks are traversed and 88.678% blocks are rejected. This improves the running time of the algorithm. But the quality of reconstructed image is unchanged.

In Randomized algorithm the threshold value used is 700. The total domain blocks visited are 716800 which 4.87% of the total domain blocks visited in full search.

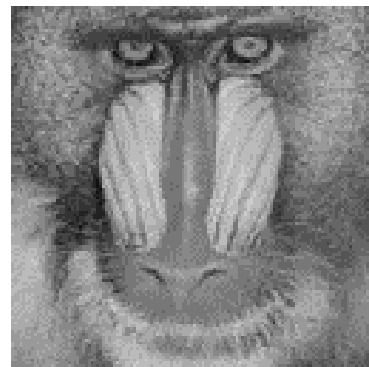
Although in comparison to the Variance-ordered search algorithm Randomized algorithm takes less time as the number of blocks traversed in Randomized algorithm are less but the quality of reconstructed image is less than Variance-ordered search algorithm.

In proposed algorithm, the domain blocks which are having similar variance as of the range block are searched. This results a significant drop in the encoding time, while the decrease in the quality is not that much considering the drop in the encoding time. This decrease is comparable to the randomized algorithm but the drop in encoding time is significant as compared to the other algorithms.

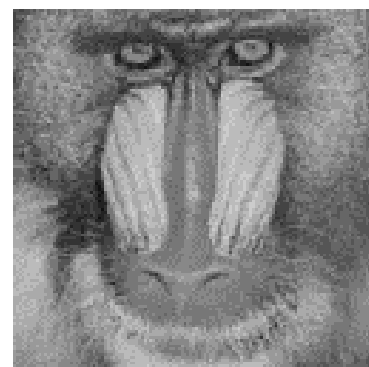
Furthermore, the tabulated values verify that there is no significant difference in the PSNR and RMS values when we decrease the threshold value from 200 to 100. However, there is a very significant decrease in the encoding time with the decrease of the threshold value. Hence it can be concluded that if only first 100 values around the domain block are chosen for minimum SVD, then a suitable domain block for a particular range block can be with a significant decrease in encoding time. This proves the efficiency of the present algorithm and it is believed to be a great advantage of the proposed algorithm.



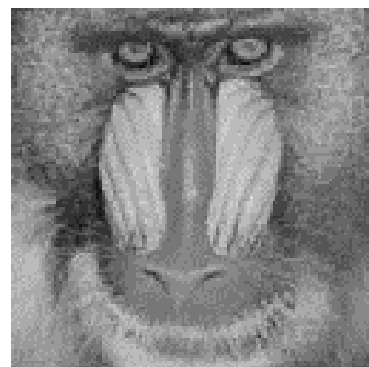
**Fig 1: Original image**



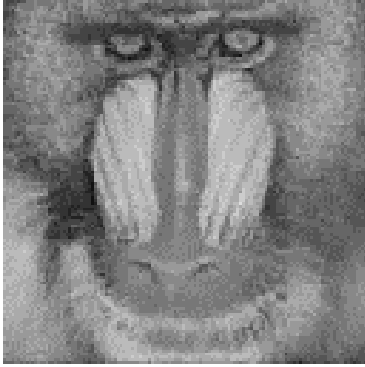
**Fig 2: Threshold value 200**



**Fig 3: Threshold value 100**



**Fig 4: Threshold value 50**



**Fig 5: Threshold value 10**

## **5. ACKNOWLEDGMENTS**

The author thankfully acknowledges the valuable suggestions by Santwana Mukhopadhyay, Department of Mathematical Sciences, IIT (BHU), Varanasi, India to carry out the present work.

## **6. REFERENCES**

- [1] Jacquin A, Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformation, IEEE Trans. Image Process. 1992,IP\_1, pp 18-30
- [2] Sang-Moon Lee, “**A Fast Variance-Ordered Domain Block Search Algorithm for Fractal Encoding**” IEEE 1999
- [3] S.K. Ghosh, Jayanta Mukherjee, P.P. Das, “**Fractal Image Compression: a randomized approach**” IEEE, March 2004
- [4] Alex Candela, Rahmi Hezar, “**An Introduction to Fractal Image Compression**” Literature Number: BPRA065 Texas Instruments Europe October 1997
- [5] Yuval Fisher, “**Fractal Image Compression**” SIGGRAPH '92 Course Notes