Design of Faster Fractal Image Compression using Intensity Mismatch Threshold

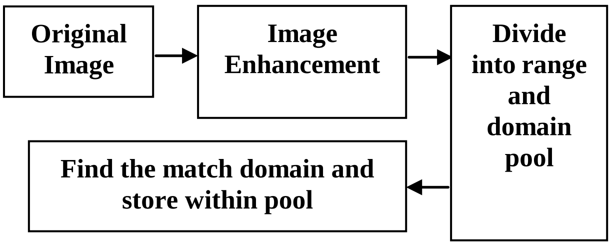
Hasanujjaman, Arnab Banerjee, U. Biswas, M K Naskar

Abstract*: Within the domain of image processing the applications of the fractal image compression through the affine transform is rarely implemented. The major drawback it consists is its time consumptions in simulations as well as in hardware implementation. The reason behind this is its comparatively matching the range blocks with the varied number of domain blocks. Additionally, the compression ration obtained from this sets of techniques are much more that the other similar working techniques. In order to overcome, this drawback, here a concept of intensity mismatch searching techniques is intruded within this particular paper. While in the mean time, it only concluded in much reduce in time of searching for similar domain block form a pool. And also this technique does not affect within the compression ratio and PSNR measurement form the retrieved image. Further benefitted in hardware applications where the matters for time consumptions is a major factor.

***Keywords* : fractal image compression, time consumption, intensity mismatch, range block and domain block.**

I. INTRODUCTION

In today's world, the usage for cameras for image capturing turns into a vital component. Due to which a high increment upon storage data is primarily concerned under the database structure. Moreover, a varied number of impacts are intercepted within the main coursed regarding storages of data in small allocated data sets. For that reason the use of fractal image compression techniques is used. The steps initiated by the fractal image compression[1-3] are based upon the concept of "Portioned Iterated Function System" or in short PIFS [4-6]. That basically works upon estimating the actual requirements of blocks that are needed to be present within the compressed image. Therefore, much more less number of storage data will be allocated with the purpose of storing the image. In order to do so, the technique used in fractal image compression [7] is to divide the image into two large blocks, which consists of, the main image or the range blocks and in other the domain blocks. Then the further process of checking for similar weight age of the rage blocks will processed proceed towards matching with the domain blocks within the pool, so that the coordinates of the main block will be noted down for further information content of data. Therefore the recovery of the overall sets of blocks and the original image can be easily processed while needed. The overall process initiated through the fractal image compression technique is described as:-



Therefore, lastly a minimum data is initiated for the storage of image information within the hardest. At the same time, while it is required to receive the details within image, then the decompression technique is initiated. That helps in obtaining the contents of data that are stored within the original image with an effect of lossless or lossy contents in data. Here in next section the in-depth working steps for fractal image compression is discussed, and then the further proposed work is discussed in third section. Additionally, the obtained result with the conclusion and discussion is also held in the fourth and fifth section.

II. FRACTAL IMAGE COMPRESSION

According to Fisher and Yuval, the outcomes of fractal image compression is in higher compression rations, but the major drawbacks for is the compression speed initiated for the overall operations. Additionally, the actual processing of finding the matches between the Domain pool and range blocks are mainly consists in the linear way. due to the learner process of processed the matching facilities with the domain blocks in different geometrical concepts, consumed the time as well as cause the major drawbacks of the method. That overall method of finding the match between domain and range blocks are processed into different geometrical rotation orders; in order to achieve more compression rations [2]. However, the major drawbacks of attaining operated speed along with consumption of time are remained negligible changes. Among the geometrical matching facilities, the processes of tilting, rotating and reflection of the range blocks are considered to check over the preferable domain blocks. That is further described as the isometric matching operations within FIC.
The range pixels will then be approximated by,

$$R(i,j)=S x I_d(i,j)+O.$$

The more is the domain pool, the more possibilities are there for a good quality of matching to occur. In order to expand the domain pool, the domains are used to create it's 8 isometrics namely,

I. Identity:

$$I_d\,(i,j)= I_d\,(i,j)$$

Revised Manuscript Received on October 15, 2019.

Hasanujjaman*, Assistant Professor Dept. of ECE, Kalyani Govt. Engineering College , Kalyani , West Bengal Email:
Arnab Banerjee , Dept. of ECE, Kalyani Govt. Engineering College , Kalyani , West Bengal.
Prof. U Biswas, University of Kalyani , West Bengal
Prof. M K Naskar , Jadavpur University , Kolkata

II. Reflection about mid vertical axis:

I_d (i,nd-j+1)= I_d (i,j). [where nd is the pixel dimension of domain]

III. Reflection about mid horizontal axis:

I_d (nd-i+1,j)= I_d (i,j)

IV. Reflection about first diagonal:

I_d (j,i)= I_d (i,j)

V. Reflection about second diagonal:

I_d (nd-j+1,nd-i+1)= I_d (i,j)

VI. 90° rotation:

I_d (j,nd-i+1)= I_d (i,j)

VII. 180° rotation:

I_d (nd-i+1,nd-j+1)= I_d (i,j)

VIII. 270° rotation:

I_d (nd-j+1,i)= I_d (i,j)

The domain with the minimum MSE will get selected to best approximate the range .The modified domain pixels are obtained after evaluating the contrast scale factor(s) and the brightness offset factor(0).

$$\text{MSE} = \sum (s d_i + o - r_i)^2$$

$$s = \frac{[n \sum d_i r_i - \sum d_i \sum r_i]}{[n \sum d_i^2 - (\sum d_i)^2]}$$

$$o = \frac{[\sum d_i - s \sum r_i]}{n}$$

When FIC is conducted for a 256X256 grey scale image using range of dimension 4X4 and domain of dimension 8X8 then, a full search encoding will take up more than 30 minutes in a software approach using MATLAB. Thus in order to make the compression realisable in real time various techniques can be taken up to constrict the domain set to be compared to a particular range. One technique is to consider those domains which are in close vicinity of the range block. Other techniques like stopping the comparison of domain blocks for a range block after a domain is obtained which produces a MSE less than some pre-fixed value or threshold. Such a technique was used by [3]. in their hardware implementation on an ASIC. The notion of parallelism is utilised the great extent in hardware approaches to make calculations faster. Techniques like setting up a classifier vector to segment both range and domain into different classes or groups is utilised by Samavi et al. So, only domains belonging to the same class as that of the range will only get considered. While [4] used Bit width optimisation to make faster calculations by sacrificing the precision of calculation to some certain extent.

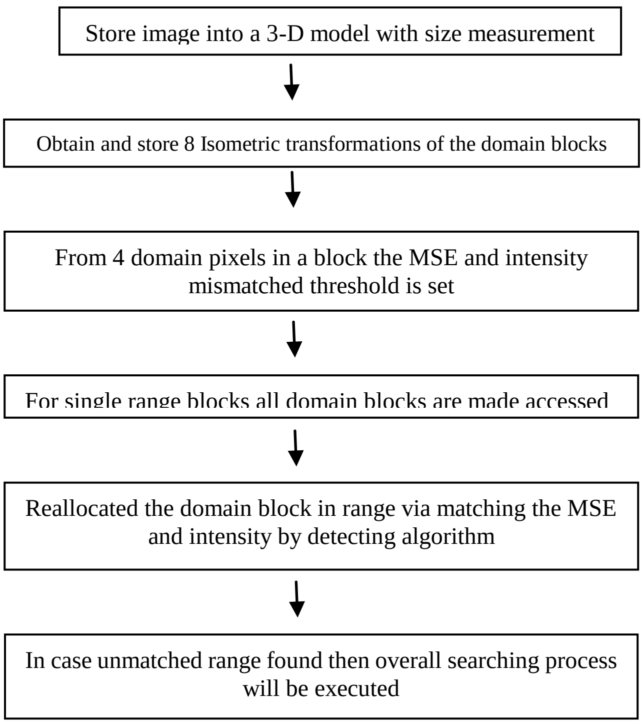
III. RPROPOSED WORK

In this paper a metric is utilised which takes into consideration the average intensities of the range and the domain blocks. For a good approximation the average intensity of the domain should be close to the average intensity of the range. More the mismatch between their average intensities, more error will get added if such a domain is selected to approximate the range block. So, a intensity mismatch parameter is created which is the absolute value between the average intensity difference between range and domain.

$$\text{Intensity Mismatch Value} = \frac{\frac{\text{sum of all pixels values of range}}{\text{total number of range pixels}}}{} - \frac{\frac{\text{sum of all pixels values of domain}}{\text{total number of ddomain pixels}}}{}$$

If for a certain domain the intensity mismatch value is less than the pre-fixed intensity mismatch threshold then, for only such domains we will proceed to evaluate the contrast scale factor (s) , the brightness offset factor (o) and finally the mean square error (MSE).Since most of the available domains would not produce an intensity mismatch value less than the intensity mismatch threshold value, thus a lot of time is saved. Also since, the calculation of s, o and MSE is responsible for occupying most of the encoding time, it is wanted that only some selected domains be allowed to go through the process. The intensity mismatch threshold was varied from 50 down to 0.5.Setting a threshold as high as 50 means that not only domains with average intensity value quite close to that of the range's value are considered for the time consuming evaluation of s, o and MSE but domains with drastic difference in average intensity value from that of the range's will also get entertained. This is bound to increase the encoding time but, since a large domain pool is available for selection the quality of the reconstructed image will be better than if we set a intensity mismatch threshold of 0.5.To further decrement the encoding time, the proposed techniques by [3] is considered. of stopping the range-domain matching process once such a domain is obtained which gives a MSE less than some pre-fixed threshold. Although this will slightly degrade the quality of reconstructed image but, improvement in compression time is more advantageous.

For compression



The obtained comp matrix is the compressed image. In order to make the compression ratio higher, 7 decimal digits to represent the contrast scale factor (s) and 5 decimal digits to represent the brightness offset factor (o).

For the s, we observed that the absolute value of s mostly lies between 0.1 and 1.5 but sometimes it becomes a 2-digit number. Certain number from decimals numbers are sorted for further interrogates processes. Similarly o is generally a 2 or 3 digit number but, sometimes it may be a 4 digit number. Similarly the decimals numbers are also sorted for further easy achievement. Along with the selected domain number we concatenate all these digits to form a single number and store them in a transmit matrix. The data in this matrix can easily be converted into bits and then transmitted. For a 256X256 image it can be observed that compression ratio of 2.37 and 9.48 is obtained for 4 x4 and 8x8 range blocks.

For decompose

Concatenated values are extracted and stored in different matrixes

↓

Take a sample image and evaluates the range pixels in it

↓

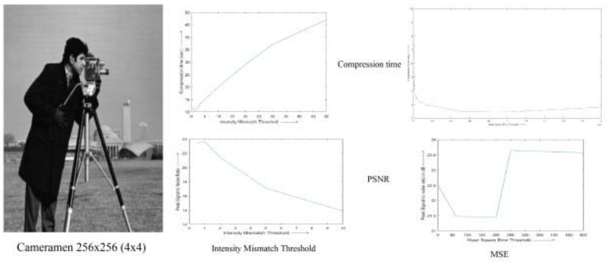
Use the domain blocks for recovered the matrix

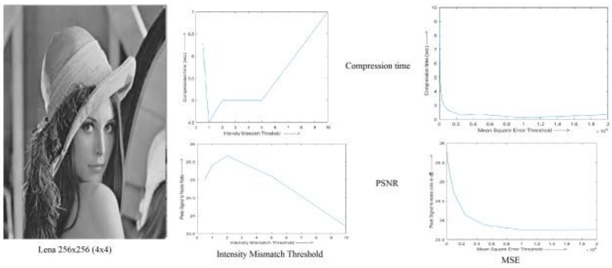
↓

After 10 iterations same original image will be achieved

IV. RESULT AND DISCUSSION

- For images of dimension around 128X128, if PSNR is to be maximised (better quality) range blocks of dimension (2X2) is used with intensity mismatch threshold of 1. If compression time is to be minimised range blocks of dimension (4X4) is selected with intensity threshold above 10.
- For images of dimension around 256X256, if PSNR is to be maximised, range blocks of dimension (4X4) is utilised with an intensity mismatch threshold of 10 and above. If compression time is to be minimised range blocks of dimension (4X4) is selected with intensity threshold above 1.
- For images of dimension around 400 X 400 , if PSNR is to be maximised, range blocks of dimension (4X4) is utilised with an intensity mismatch threshold of 1 . If compression time is to be minimised range blocks of dimension (8X8) is selected with intensity threshold above 10.
- For images of dimension in and above 512X512, range blocks of dimension (8X8) is used with intensity mismatch threshold of 1.
- For range blocks of (2X2) dimension, mean square error threshold of 0 is utilised.
- A mean square threshold of 250 will keep the PSNR as well as the compression time to an acceptable limit.
- For images of dimension around (256X256), if compression ratio is to be maximised with finite degradation of PSNR, range blocks of dimension (8X8) is utilised with intensity mismatch threshold greater than 10 and mean square error threshold of 0. This will lead to a finite increment in compression time.
- Range blocks of dimension (8X8) leads to a compression ratio lying between 9.5 to 10.3, while (4X4) range block scan best give us a compression ratio of 2.4. For a 256X256 image normally 4X4 range blocks are used cause using8X8 range block degrades quality of reconstructed image. For Cameraman 256X256 image using 8X8 range blocks with intensity mismatch threshold of 1 can best give PSNR of 20.42 with 2.48 sec compression time with mse threshold of 0.For Lena 256X256 image using 8X8 range blocks with intensity mismatch threshold of 50 can give PSNR of 23.1 with 10.47 sec compression time with mse threshold of 0.





Comparison between proposed and contemporary Techniques:

	Proposed		Vidya(8)	Samavi(9)	Jackson(10)
Platform	Matlab	Matlab	FPGA	FPGA	FPGA
Result based on	Simulation	Simulation	Simulation	Simulation	Hardware
Image size	256X256	256X256	128 x 128	256 x 256	512 x 512
Search type	Full Search	Full Search	Full Search	Partial	Searchless
Range block size	4 X 4	8 X 8	4x4	8 x 8	Flexible
PSNR of Lena image (dB)	25.41	23.1	28.1	25.3	33.1
Compression ratio	2.46	9.77	4	26	12.8
Coding time	2.41s	10.47s	14.1s	0.8 ms	8.4ms

V. CONCLUSION

This proposed algorithm assures to get PSNR above 25 in most of the sample images and compression time of less than 10 seconds in most simulations. It is flexible towards the selection of dimensions of the range blocks. For 512X512 image range block of dimension 8X8 is used extensively while for 256X256 images the dimension of range block is a matter of preference.

If compression ratio is given higher preference before PSNR then range blocks of 8X8 gives PSNR around 23dB and compression ratio above 9.5. If PSNR is given higher importance before compression ratio then range blocks of 4X4 gives PSNR greater than 25dB and compression ratio around2.5.The computationally heavy processing steps in FIC deters the implementation of this technique for real time applications. Efforts are being made to reduce the processing time so that, the high compression ratio offered by FIC can be exploited.

REFERENCES

1. Michael F. Barnsley , Arnaud Jacquin , Francois Malassenet , Laurie Reuter , Alan D. Sloan, Harnessing chaos for image synthesis, Proceedings of the 15th annual conference on Computer graphics and interactive techniques, p.131-140, June 1988
2. Jacquin A., Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding, Doctoral Thesis, Georgia Institute of Technology, 1989.
3. Barnsley M, Sloan A (1990) Methods and apparatus for image compression by iterated function systems. U.S. patent no. 4,941,193.
4. Iterated Systems Inc. changed its name to MediaBin Inc. Inc. Archived 2008-01-31 at the Wayback Machine in 2001 and in turn was bought out by Interwoven, Inc. in 2003)
5. Arnaud E. Jacquin. Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations. IEEE Transactions on Image Processing, 1(1), 1992.
6. Jacquin A., Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding, Doctoral Thesis, Georgia Institute of Technology, 1989.
7. R.D. Boss, E.W. Jacobs, “Fractals-Based Image Compression,” NOSC Technical Report 1315, Sept. 1898. Naval Ocean Systems Centre, San Diego CA 92152-5000.
8. D. Vidya, R. Parthasarathy, T. Bina, N. Swaroopa, Architecture for fractal image compression, J. Syst. Archit. 46 (14) (20 0 0) 1275–1291, doi: 10.1016/ S1383-7621(0 0)0 0 018-7
9. S. Samavi, M. Habibi, S. Shirani, N. Rowshanbin, Real time fractal image coder based on characteristic vector matching, Image Vision Comput. 28 (11) (2010) 1557–1568, doi: 10.1016/j.imavis.2010.03.011
10. D.J. Jackson, H. Ren, X. Wu, K.G. Ricks, A Hardware architecture for real-time image compression using a searchless fractal image coding method, J. Real- Time Image Process. 1 (3) (2007) 225–237, doi: 10.1007/s11554- 007- 0024- 2 .

AUTHORS PROFILE

Mr. Hasanujjaman currently Assistant Professor of ECE dept of Kalyani Government Engineering College. He has more than 13 years experience.

Mr. Arnab Banerjee PG student of ECE dept. of Kalyani Government Engineering College.

Prof.(Dr.) U Biswas received his B.E, M.E and PhD degrees in Computer Science and Engineering from Jadavpur University, India in 1993, 2001 and 2008 respectively. He served as a faculty member in NIT, Durgapur, India in the department of Computer Science and Engineering from 1994 to 2001. Currently, he is a Professor in the department of Computer.

Prof.(Dr.) M .K NASKAR received his B.Tech (Hons.) and M.Tech from E&ECE Dept., IIT Kharagpur in 1987 and 1989 respectively and Ph.D. from Jadavpur University in 2006. He served as a faculty member in NIT, Jamshedpur (then RIT Jamshedpur) and NIT, Durgapur (REC Durgapur) from 1991-1996 and 1996-1999 respectively. Currently he is a Professor in the Department of Electronics and Telecommunications Engineering, Jadavpur University, Kolkata, India