A New Approach to Fractal Image Compression Using DBSCAN

Jaseela C C and Ajay James
Dept. of Computer Science & Engineering, Govt. Engineering College, Trichur, Thrissur, India
Email: cp.jesi@gmail.com, ajayjames80@gmail.com

***Abstract*—Fractal image compression is a popular lossy image compression technique. Fractal encoding, a mathematical process to encode image as a set of mathematical data that describing fractal properties of the image. Fractal image compression works based on the fact that all natural, and most artificial, objects contain similar, repeating patterns called fractals. But problem is the encoding of fractal image compression takes lot of time and it is computationally expensive. A large number of sequential searches are required to find matching of portions of the image. In this paper we introduce a new algorithm to fractal image compression by using density based spatial clustering of applications with noise (DBSCAN). The modification is applied to decrease encoding time by reducing the sequential searches through the whole image to its neighbors. This method compress and decompress the color images quickly.**

***Index Terms*—fractal, DBSCAN, fractal image compression, RGB, clustering, algorithm**

I. INTRODUCTION

The need for mass information storage and fast communication links grows day by day. Storing images in less memory leads to a direct reduction in storage cost and faster data transmissions. Fractal compression is a popular lossy image compression technique using fractals [1]. There are two sorts of image compression [2], lossy image compression and lossless image compression. In lossless image compression, it is possible to go back to the original data from the compressed data and there is no loss of information. In most cases it doesn't matter if the image is changed a little without causing any noticeable difference [3]. Lossy image compression works by removing redundancy and by creating an approximation to the original with high compression ratio. Fractal image compression works based on the fact that all natural, and most artificial, objects contain similar, repeating patterns called fractals. Fractal encoding, a mathematical process to encode image as a set of mathematical data that describing fractal properties of the image [4]. A fractal is a rough geometric shape that is made up of similar forms and patterns [5].

Fractal encoding is highly used to convert bitmap images to fractal codes. Fractal decoding is just the reverse, in which a set of fractal codes are converted to a bitmap [6]. The encoding process is extremely computationally intensive. Millions or billions of iterations are required to find the fractal patterns in an image. Depending upon the resolution and contents of the input bitmap data, and output quality, compression time, and file size parameters selected, compressing a single image could take anywhere from a few seconds to a few hours (or more) on even a very fast computer [7]. Decoding a fractal image is a much simpler process. The hard work was performed finding all the fractals during the encoding process. All the decoding process needs to do is to interpret the fractal codes and translate them into a bitmap image.

The remainder of the paper is organized as follows. Section II introduces fractal image compression. Section III describes our proposed method in detail. Section IV presents the experimental results. Conclusion is given in Section V.

II. FRACTAL IMAGE COMPRESSION

As a product of the study of iterated function systems (IFS), Barnsley and Jacquin introduced Fractal image compression techniques. 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 block wise basis" [8]. 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 encoding works based on the fact that all objects have information in the form of similar, repeating patterns called an attractor [9]. The encoding process consists of finding the larger blocks, called domain blocks, of the image corresponding to the small image blocks, called range blocks through some operations.

Fractal compression is highly suited for use in image databases and CD-ROM applications. In fractal image compression, dividing the image into non overlapping BXB blocks called range and divide image into overlapping 2B x 2B blocks called domain [10]. Then for each range block, search for the domain block that most closely resembles the range block. Transformations such as scaling, translation, rotating, sharing, scaling etc and adjustment of brightness/contrast are used on the domain block in order to get the best match. The color, brightness, contrast adjustments and the translation done on the domain blocks to match the position of their associated

Manuscript received July 15, 2013; revised September 5, 2013.

doi: 10.12720/ijoee.2.1.18-22

range blocks are stored in the compressed file. This information is called fractal codes. No pixels are stored at all, only mathematical functions. Domain block is used to describe a range block by checking root mean square error between the corresponding domain and range block and domain block which gives the minimum root mean square value among all domain blocks is selected. So the encoding process is time consuming.

III. Proposed Method

The DBSCAN (Density based spatial clustering of applications with noise) algorithm can identify clusters in large spatial data sets by looking at the local density of database elements [11] and [12]. Two global parameters for DBSCAN algorithms are Eps, maximum radius of the neighborhood and MinPts, minimum number of points in an Eps - neighborhood of that point [13]. The main idea of DBSCAN is that, for each object of a cluster the neighborhood of a given radius; Eps must contain at least a minimum number of points, MinPts [14].

We revised the DBSCAN in this new approach. Instead of checking in circle, we used square blocks. In our compression, divide the image into non overlapping B x B blocks called range blocks. We can change the value of B, block size. Then pixels in the color image is grouped into different clusters [15]. Here cluster of maximum radius, Eps is changed to 3B x 3B block called domain blocks. Here we did not use minimum points, MinPts. Instead we compare each range block with the center of the corresponding domain block. Total 9 range blocks are considered in the process of forming a cluster.

Here we use a threshold value, t. The difference between the corresponding R component, G component and B component of each range block and the center of the domain block is compared with this t. If this difference value is below t, a matching is found. This procedure is separately done for the three components in the RGB color scheme. The decompression is just the reverse of compression. But the time taken to decompress the image is very less.

A. *Encoding Algorithm*

- Store the height, width and block size, B of the image into a file.
- Divide the image into non overlapping range blocks of size B x B.
- Cluster the range blocks in the neighborhood into groups, domain blocks.
- For each domain block d,

a. Save the average value of the pixels of the center block(C) of the domain block.

b. For each range block, r in d,

i. If r is center block of d, then avoid it and move to next range block.

ii. Compare the average value of pixels of r with average value of pixels of C separately for red, green and blue values.

iii. If the difference is below a threshold, t, then save a bit 1; which shows blocks are matching for the particular component R, G or B, else save the corresponding average value by saving a bit 0.

B. *Decoding Algorithm*

- Read the height, width and block size from the file.
- Draw an image with default RGB with corresponding height and width.
- Divide this image into non overlapping range blocks of size B x B.
- Cluster the range blocks in the neighborhood into groups, domain blocks.
- For each domain block d,

a. Color the center block of the domain with corresponding RGB average values cr, cg, cb respectively saved in the file.

b. For each range block, r in d,

i. If r is center block of d, then avoid it and move to next range block.

ii. Read a bit from the file. If it is 1, then color the block r with red values of the center block. Else read the next value from the file and color the block r with the read values.

iii. Repeat step ii for green and blue component of r.

IV. Implementation

Images require compression in order to reduce disk space needed for storage and time needed for transmission. The lossless compression methods of still images can shorten the file only to a very limited degree. Fractal image compression is an emerging compression method for compressing images especially for self similar images. But fractal coding is always applied to grey level images. The most straight forward method to encode a color image by gray-level fractal image coding algorithm is to split the RGB color image into three Channels, red, green and blue, and compress them separately by treating each color component as a single gray-scale image, the so called three-component Separated Fractal Coding. In place of going for three independent planes, in this, one plane image is composed from the three planes of RGB color image using trichromatic coefficients. This one plane image is then compressed by proposed modified fractal coding.

This new method of fractal image compression is implemented using Java programming language with the help of net-beans IDE. During encoding read an image using jfilechooser method. Then create a new file with extension .fru. We can select the location of file and block size dynamically. Then save the height, width and block size into a file. Do the image segmentation using the block size which is read during encoding process. Then do the clustering process on the segments of the image. After that save the non redundant information about the image into file in order to reproduce the image. Saving the values in bit by bit manner instead of byte in order to reduce the size of file further. Only 1 bit is needed to indicate matching, bit 1 for indicating matching and bit 0 for indicating non matching.

Color images can have pixel depth equal 4, 8 or 16 bits; full color can be achieved with 24 or 32 bits. In case of an image in RGB color model, for saving the average value of a block 24 bits are used, 8 for average value of red components of block, 8 for average value of green components of block and 8 for average value of blue components of block. Actually 24 bits are needed to represent a pixel of an image in RGB colour model. Here comparison is done for each component of the block separately. So if any component is matching, a bit 1 is saved. If a block is approximately similar to centre of its domain block, then only 3 bits are needed to represent the corresponding block. So size of file will decrease significantly.

V. Experiments and Results

Several images are compressed with this method in order to do analysis. After the compression, images are coded to files of reduced size. Then from the compressed file, image is reproduced during decompression. The experiment was performed using Java programming language with the help of net-beans IDE. Fig. 1, Fig. 3, Fig. 5, Fig. 7, Fig. 9, Fig. 11 and Fig. 13 are some example images we considered to complete our experiment. The actual height and width are different. The images in Fig. 2, Fig. 4, Fig. 6, Fig. 8, Fig. 10, Fig. 12 and Fig. 14 are images after decompression corresponding to the images in Fig. 1, Fig. 3, Fig. 5, Fig. 7, Fig. 9, Fig. 11 and Fig. 13 respectively.

From experiments it is clear that the proposed method gives high compression ratio especially for self similar images. The size of the image is reduced during compression and it is approximately about 80% less than the original image. The size is further reduced during decompression and it is approximately about 90% less than the original image.

From the example figures, an analysis table Table I is formed. The table shows the original size of the images, size of the file which is produced during the compression of the image, the size of the reproduced image during the decompression, percentage less in size of the compressed file and percentage less in size of the reproduced image during decompression. The images Fig. 1, Fig. 3, Fig. 5, Fig. 7, Fig. 9, Fig. 11 and Fig. 13 are considered for this analysis.

Figure 1. Original image.

Figure 2. After decompression.

Figure 3. Original image.

Figure 4. After decompression.

Figure 5. Original image.

Figure 6. After decompression.

Figure 7. Original image.

Figure 8. After decompression.

Figure 9. Original image

Figure 10. After decompression.

Figure 11. Original image.

Figure 12. After decompression.

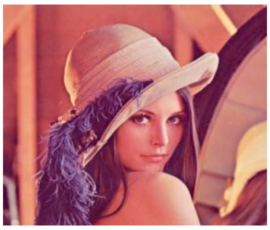

Figure 13. Original image.

Figure 14. After decompression.

TABLE I. ANALYSIS

Image	Original Size	Compressed file size	Decompressed image size	% in reduction of compressed file	% in reduction of size of decompressed image
Fig.1	759 KB	125 KB	68.9 KB	83.5	91
Fig.3	1.38 MB	373 KB	230 KB	73.6	83.7
Fig.5	826 KB	120 KB	73.3 KB	85.4	91
Fig.7	757 KB	131 KB	51.9 KB	82.6	93
Fig.9	1.02 MB	226 KB	128 KB	78.4	87.7
Fig.11	1.02 MB	251 KB	69.2 KB	76	93.4
Fig.13	67.5 KB	49.1 KB	28.8 KB	27.26	57.33

VI. APPLICATIONS

Fractals have more and more applications in science. The main reason is that they very often describe the real world better than traditional mathematics and physics. Fractals have applications in Astronomy, Computer science, Fluid mechanics, Telecommunications, Surface physics, etc. Galaxy correlations are fractal and not homogeneous up to the limits of the available catalogues. The result is that galaxy structures are highly irregular and self-similar. Fractals are used to model soil erosion and to analyze seismic patterns as well.

The study of turbulence in flows is very adapted to fractals. Turbulent flows are chaotic and very difficult to model correctly. A fractal representation of them helps engineers and physicists to better understand complex flows. Flames can also be simulated. Porous media have a very complex geometry and are well represented by fractal. This is actually used in petroleum science. A new application is fractal-shaped antennae that reduce greatly the size and the weight of the antennas. Fractals are used to describe the roughness of surfaces. A rough surface is characterized by a combination of two different fractals. Biosensor interactions can be studied by using fractals.

VII. CONCLUSION

The method we proposed is better than the current methods of fractal image compression. The execution time of the compression algorithm is reduced significantly compared to the traditional fractal image compression. In internet image database is very large. So storing images in less space is a challenge. Image compression provides a potential cost savings associated with sending less data over switched telephone network where cost of call is really usually based upon its duration. It not only reduces storage requirements but also overall execution time.

The proposed method compress images in block by block basis, instead of checking in pixel by pixel. This method compress the color images and decode them quickly by considering their RGB values separately. This method is very useful for storing images in image database with less storage space and retrieve them quickly. Also the proposed method provide high compression ratio especially for self similar images. JPEG image compression is not work with discontinuity in image. But the proposed method is applicable to images with discontinuity. In this, image is coded into a file of very small size which can be used as the encrypted form of image and it is possible to send this encrypted file into any where securely. Then the receiving person can decompress the encrypted file easily through this method and reproduce the image.

In this, regular partition is used during image segmentation. The work can be extended to irregular partitioning of image in future. The fractal image compression using DBSCAN can be extended to do compressing videos. Video compression is the future work of the proposed method.

REFERENCES

[1] Y. C. Yang and R. D. Peng, "Fast fractal coding based on dividing of image," in *Proc. 2010 International Conference on Computational Intelligence and Software Engineering* (CiSE), 2010, pp. 1–4.

[2] Q. Wang, D. Q. Liang, and Sh. Bi, "Fast fractal image encoding based on correlation information feature," presented at 2010 3rd International Congress on Image and Signal Processing (CISP), Yantai, China, 16-18 Oct. 2010.

[3] Y. Iano, F. S. da Silva, and A. L. M. Cruz, "A fast and efficient hybrid fractal-wavelet image coder," *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 98, 2006.

[4] Y. Zhao, J. Hu, D. X. Chi, and M. Li, "A novel fractal image coding based on basis block dictionary," in *Proc. EEEE '09. International Conference on E-Learning, E-Business, Enterprise Information Systems and E-Government*, 2009, pp. 165–168.

[5] S. Zhu and J. Chen, "Research on fractal image coding methods," in *Proc. 2012 International Conference on Computer Science and Information Processing*, pp. 1039–1043.

[6] E. Zhao and D. Liu, "Fractal image compression methods: A review," presented at 2005 International Conference on Information Technology and Applications, Sydney, 4th-7th, July 2005.

[7] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD'96*, Portland, OR, pp. 226-231.

[8] M. Verma, M. Srivastava, N. Chack, A. K. Diswar, and N. Gupta, "A comparative study of various clustering algorithms in data mining," *International Journal of Engineering Research and Applications*, vol. 2, no. 3, pp. 1379-1384, May-June 2012.

[9] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed, Pearson Education Asia, 2005.

[10] J. Gao, A. Song, W. Liu, and Y. Chen, "Lossless image compression coding method based on static dictionary," *Journal of Computer Applications*, vol. 31, no. 6, pp. 1578, 2012.

[11] A. Jacquin, “Fractal image coding based on a theory of iterated contractive image transformations,” in *Proc. SPIE Conf. Visual Communications and Image Processing*, 1990, pp. 227–239.

[12] A. Ram, S. Jalal, A. S. Jalal, and M. Kumar, “A density based algorithm for discovering density varied clusters in large spatial databases,” *International Journal of Computer Applications*, vol. 3, no. 6, pp. 1-4, June 2010.

[13] K. Mumtaz, *et. al*., “An analysis on density based clustering of multi dimensional spatial data,” *Indian Journal of Computer Science and Engineering*, vol. 1, no. 1, pp. 8-12, 2011.

[14] J. H. Peter and A. Antonysamy, “An optimised density based clustering algorithm,” *International Journal of Computer Applications*, vol. 6, no. 9, pp. 20-*25,* September 2010.

[15] M. Salarian, E. Nadernejad, and H. MiarNaimi, “A new modified fast fractal image compression algorithm,” *Imaging Science Journal*, vol. 61, no. 2, pp. 219-231, 2013.

Jaseela C C is a mtech second year student in computer science at Government Engineering College, Trichur, Kerala, India. She received first class btech degree in information technology from Calicut University Institute of Engineering and Technology, Calicut, Kerala, India during 2007-2011. Her research interests are in image processing and in data mining. She published a paper in national conference, Horizontal Aggregations and Integrating K Means Algorithms in SQL.

Ajay James is presently working as Assistant Professor in the department of Computer Science and Engineering at Government Engineering College, Trichur, Kerala, India. He has been teaching for UG and PG courses for the past 8 years. He has completed his Master of Technology in Computer Science and Engineering from Pondicherry University in 2008. He has published papers in International Journals and also has attended many conferences. He is a life member of Indian Society for Technical Education (ISTE) and International Association of Computer Science and Information Technology (IACSIT).