Volume 2, Issue 2, February 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Fractal image compression- a review

Hitashi
Research Scholar
Indo Global College
Chandigarh

Gaganpreet Kaur
Assistant Professor
Sri Guru Granth Sahib World University
Fatehgarh Sahib

Sugandha Sharma
Assistant Professor
Indo Global College
Chandigarh

Abstract—**Fractal image compression is a comparatively recent technique based on the representation of an image by a contractive transform, on the space of images, for which the fixed point is close to the original image. This broad principle encompasses a very wide variety of coding schemes, many of which have been explored in the rapidly growing body of published research. While certain theoretical aspects of this representation are well established, relatively little attention has been given to the construction of a coherent underlying image model that would justify its use. Most purely fractal-based schemes are not competitive with the current state of the art, but hybrid schemes incorporating fractal compression and alternative techniques have achieved considerably greater success. This review represents a survey of the most significant advances, both practical and theoretical in original fractal coding scheme. In this paper, we review the basic principles of the construction of fractal objects with iterated function systems (IFS).**

Keywords—**Fractals, image coding, iterated function system.**

1. INTRODUCTION

Data compression has become an important issue forinformation storage and transmission. This is especially true for databases consisting of a large number of detailed computer images [3], [4], [5]. Recently, a large quantity of methods has appeared in the literature for achieving high compression ratios for compressed image storage, and among them, the fractal approach become a feasible and promising compression technique. The field of image coding (or compression) deals with efficient ways of representing images for transmission and storage purposes.The basic objective of video coding is to compress the data rate by removing redundant information. There are two major categories of coding schemes (i.e. sourcecoding and entropy coding). Multimedia data requires considerable storage capacity and transmission bandwidth. The data are in the form of graphics, audio, video and image. These types of data have to be compressed during the transmission process. Large amount of data can't be stored if there is low storage capacity present. The compression offers a means to reduce the cost of storage and increase the speed of transmission. Image compression is used to minimize the size in bytes of a graphics file without degrading the quality of the image. There are two types of image compression is present. They are lossy and lossless [1]. In lossless compression, the reconstructed image after compression is numerically identical to the original image. In lossy compression scheme, the reconstructed image contains degradation relative to the original. Lossy technique causes image quality degradation in each compression or decompression step. In general, lossy techniques provide for greater compression ratios than lossless techniques i.e. Lossless compression gives good quality of compressed images, but yields only less compression where as the lossy compression techniques [3] lead to loss of data with higher compression ratio. The approaches for lossless image compression include variable-length encoding, Adaptive dictionary algorithms such as LZW, bit-plane coding, lossless predictive coding, etc. The approaches for lossy compression include lossy predictive coding and transform coding. Transform coding, which applies a Fourier-related transform such as DCT and Wavelet Transform such as DWT are the most commonly used approach [3]. Over the past few years, a variety of powerful and sophisticated Fractal image compression schemes for image compression have been developed and implemented. The iteration function system provides a better quality in the images.

Source coding deals with source material and yields results which are lossy (i.e. picture quality isdegraded). Entropy coding achieves compression by using the statistical properties of the signals and is, in theory, lossless. Numerous video compression techniques have been proposed in the last two decades and new ones are being developed every day. For acceptable image quality, these techniques can only achieve moderate reduction in the source data not exceeding 25 and 200 times with still and continuous images, respectively (for example by usingan adaptive discrete cosine transform(ADCT) coding schemes). Unfortunately,this is proved to be not sufficient to cope with the increasing demand in the use of transmission channels and storage media.Therefore, there is a continuous need for further reduction in image data in order to benefit from the fast development in modern communication technology in the most efficient way. This assumption is backed up by noticing its inclusion into end user products such as Microsoft's Encarta or as a Netscape plug-in by Iterated Systems Inc. [2].

Fractal image compression exploits the natural affine redundancy present in typical images to achieve high compression ratios in a lossy compression format. The main idea of the method consists in finding a construction rule that produces a fractal image, approximating to the original one. Fractal imagecoding has its roots in the mathematical theory of iterated function systems (IFS) developed by Barnsley [l], [2], whilst the first fully automated algorithm was developed by Jacquin [6].Fractal image coding consists of finding a set of transformations that produces a fractal image which approximates the original image. Redundancy reduction is achieved by describing the original image through smaller copies or parts of the image.

Iterated functions systems (IFS) theory, closely related to fractal geometry, has recently found an interesting applicationin image compression. Barnsley [7] and Jacquin [8] pioneered the field, followed by numerous contributions [9], [10].

The approach consists of expressing an image as the attractor of a contractive functions system, which can be retrieved simply by iterating the set of functions starting from any initial arbitrary image. The form of redundancy exploited is named piece-wise self-transformability. This term refers to a property that each segment of an image can be properly expressed as a simple transformation of another part of higher resolution. IFS-based still-image compression techniques can pretend to have very good performance at high compression ratios (about 70–80).

The major problem with fractal-based coding techniques is that of complexity at the encoding stage.However, the complexity of the decoder remains reasonable when compared to the encoding.

Fractal-based techniques produce outstanding results in terms of compression inimages, retaining a high degree of self-similarity. Another interesting feature of fractal-based techniques is their ability to produce a good-quality rendered image for an arbitrary scaling factor.

Fractal image compression is time consuming in the encoding process. The time is essentially spent on the search for the best-match block in a large domain pool.

In this paper, we review the basic principles of the construction of fractal objects with iterated function systems (IFS), then we explain how such a technique has been adopted by Jacquin**[8]**for the coding (compression) of digital images.

2. ITERATED FUNCTION SYSTEMS

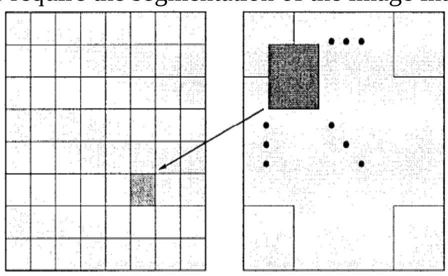
The primary tool used in describing images with iterated function systems is the affine transformation. This transformation is used to express relations between different parts of an image. Affine transformations can be described as combinations of rotations, scalings and translations of coordinate axes in n-dimensional space [9]. For example, in two dimensions a point (x, y) on the image can be represented by (xn, yn) under affine transformation.

This transformation can be described as follows: The parameters a, b, c and d perform a rotation, and their magnitudes result in the scaling. For the whole system to work properly; the scaling must always result in shrinkage of the distances between points; otherwise repeated iterations will result in the function blowing up to infinity. The parameters e and f cause a linear translation of the point being operated upon. If this transformation is applied to a geometric shape, the shape will be translated to a new location and there rotated and shrunk to a new, smaller size. In order to map a source image onto a desired target image using iterated function systems, more than one transformation is often required and each transformation, i, must have an associated probability, 9, determining its relative importance with respect to the other transformations. The random iteration algorithm given by Barnsley [9] can be used to decode an IFS code in order to reconstruct the original image. This algorithm is given in the following pseudo code:

(1) Set x=O and y=O,
(2) Select transformation wi depending on its probability $\mathbf{p_i}$
(3) apply transformation wi to the point (x, y) to obtain (xn, yn),
(4) set x=xn, y=yn and plot (x, y),
(5) go to step (2) and repeat as many times as required.

Fractal has the following properties:

1. It has a fine structure, i.e., details on arbitrarily small scales.
2. It is too irregular to be described in a traditional geometrical language, both locally and globally. It usually has some form of self-similarity, perhaps approximate or statistical.
4. Its fractal dimension (Hausdorff dimension) is usually higher than its Euclidean dimension.
5. In most cases of interest, a fractal is defined in a very simple way, perhaps, recursively. Most fractal compression algorithms require the segmentation of the image into blocks



Partition Scheme (Range Block) **Virtual codebook (Domain Block)**

Figure 1 Block mappings in a PIFS representation.

3. SELF-SIMILARITY PROPERTY

To encode an image according to self-similarity property. Each block to be encoded must search in a large pool to find the best match [12], [13], [14]. For the standard full search method, the encoding process is time-consuming because a large amount of computations of similarity measure are required.

A typical image of a face does not contain the type of self-similarity like the lena in Figure 2. The image does contain other type of self-similarity. Figure 3 shows regions of Lena identical, and a portion of the reflections of the hat in the mirror is similar to the original. These distinctions form the kind of self similarity shown in Figure 2 rather than having the image formed by whole copies of the original (under appropriate affine transformations), here the image will be formed by copies of properly transformed parts of the original [15], [16]. These transformed parts do not fit together, in general, to form an exact copy of the original image, and so it must allow some error in our representation of an image as a set of transformations. Figure 3. shows a reconstructed image regions of Lena identical, and a portion of the reflection of the hat in the mirror is similar to the original [17].

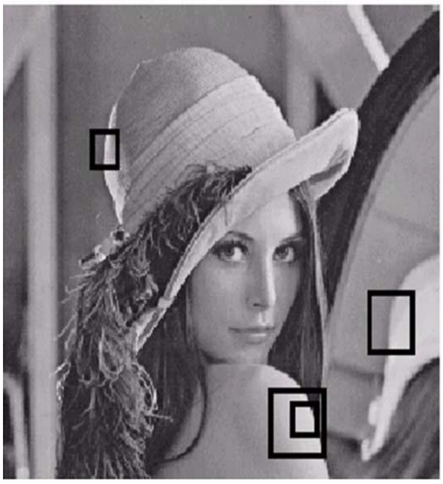


Figure 2 Displaying Self-Similarity Different Location

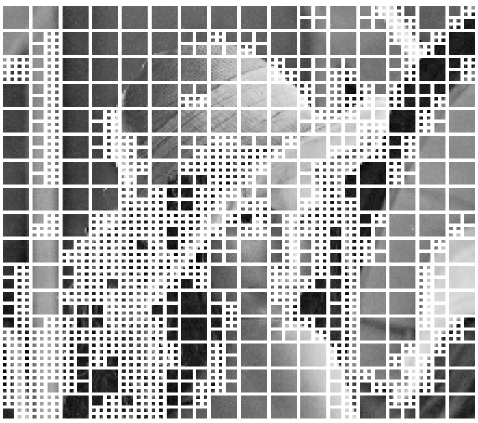


Fig 3 Reconstructed image displaying different fractals

4. FRACTAL IMAGE COMPRESSION

A. Using IFS fractals for Fractal Image Compression

The IFS compression algorithm starts with some target image *T* which lies in a subset $S \subset \mathbb{R}^2$. The target image *T* is rendered on a computer graphics monitor. In order to begin fractal image compression, an affine transformation,

$$w_1(\mathbf{x}) = w_1\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \qquad (1)$$

is introduced with coefficients that produce a new image, $w_1(T)$ with dimensions smaller than that of *T*. This ensures a contraction mapping.

The user adjusts the coefficients *a*, *b*, *c*, *d*, *e*, *f* in order to shrink, translate, rotate, and shear the new image, $w_1(T)$ on the screen so that it lies over a part of *T*. Once $w_1(T)$ is in place, it is fixed, the coefficients are recorded, and a new affine transformation $w_2(x)$ is introduced along with its sub-copy of T, $w_2(T)$. The same process is carried out with this new image as was done with $w_1(T)$. Whenever possible, overlaps between $w_1(T)$ and $w_2(T)$ should be avoided. Overlaps only complicate the situation, although there exist compression methods, such as wavelets, which confront this issue. In this manner, a set of affine transformations $w_1, w_2, w_3 \ldots, w_n$ is obtained such that

$$\tilde{T} = \bigcup_{n=1}^{N} w_n(T) \qquad (2)$$

where *N* is as small as possible.

The Collage Theorem assures us that the attractor A of this IFS will be visually close to T. Moreover, if $\tilde{T} = T$, the $A = T$. As desired, A provides an image which is visually close to T and is resolution independent using a finite string of ones and zeros. By adjusting the parameters in the transformations we can continuously control the attractor of the IFS. This is what is done in fractal image compression.

B. The fractal transform theory

Fractal transform theory is the theory of local IFS. Although local IFS do complicate the theory of fractal image compression, in practice it simplifies the process.

A global transformation on a space X is a transformation, which is defined on all points in X; whereas, a local transformation is one whose domain is a subset of the space X and the transformation need not act on all points in X. Rather than allowing an IFS to act upon only on the whole domain, it is convenient to allow an IFS to act upon domains that are subsets of the space. This type of IFS is called a local IFS.

The idea of fractal image compression, as briefly mentioned above, is to find subspaces (or sub-images) of the original image space, which can be regenerated using an ifs. Where possible, if one IFS can be used in place of several IFS's which reproduce similar sub-images, it is more efficient in terms of storage space to use that one IFS. It is more likely that an image will require more than one IFS to reproduce a compressed image, which closely resembles the original.

Definition 14 let (X, d) be a compact metric space. Let r be a nonempty subset of X. Let $w : R \to X$ and let s be a real number with $0 \le s < 1$. If

$$d(w(x), w(y)) \le (s)(d(x, y)) \quad \forall x, y \in R, \qquad (3)$$

Then w is called a local contraction mapping on (X, d). The number s is a contractivity factor for w [2].

C. Algorithm for Fractal Image Compression

1. Input a binary image, call it M.
2. Cover M with square range blocks. The total set of range blocks must cover M, without overlapping.
3. Introduce the domain blocks D; they must intersect with M. The sides of the domain blocks are twice the sides of the range blocks.
4. Define a collection of local contractive affine transformations mapping domain block D to the range block R_i.
5. For each range block, choose a corresponding domain block and symmetry so that the domain block looks most like the part of the image in the range block.
6. Write out the compressed data in the form of a local IFS code.
7. Apply a lossless data compression algorithm to obtain a compressed IFS code.

Conclusion

The field of fractal compression is relatively new, as is the study of fractals, and as such there is no standardized approach to this technique. The main concept in this compression scheme is to use Iterated Function Systems **(IFS)** to reproduce images. An important property offractals is that they exhibit self-similarity. By partitioning an image into blocks, typically 8x8 or 16x16 pixels, it becomes possible to map small portions of an image to larger portions. In addition, the smaller portions are reproduced by use of affine transformations. These transformations effectively map squares to parallelograms through translation, scaling, skewing, rotation, etc. In this way an image can be stored as a collection of affine transformations that can be used to reproduce a near copy of the original image.
The process is iterative in that detail is added after each pass through the function set. The process is computationally intensive but can yield much improved compression ratios. Fractal compression area is great. It should be possible to take advantage of the large compression ratios achieved from fractal compression and produce a trade-off of compression ratios for information loss to achieve a lossless result. This could be achieved through a post comparison of afractally compressed file and its original data. By then using a traditional compression scheme, encoding of the differences could be implemented in such a way that a lossless representation of the original data can be reproduced. Another approach is to let the user select sections of the image to be compressed as lossless.

References

1. Gonzalez, R. and Eugene, R. Digital image processing, 466, . 2008.
2. Thyagarajan, S.K. *Still image and video compression with matlab*, 97-100, 2007.
3. Kaouri, A. H. . *Fractal coding of still images*. Queen's university of Belfast, UK. 2002.
4. Hu, L., Chen, Q. and qing, Z. An image compression method based on fractal theory. The 8th international conference on computer supported cooperative work in design proceedings, 546 – 550. 2003.
5. Clarke r. J., "*digital compression of still images and video*", London, academic press, 2nd printing, 1996.
6. Jacquin**a.** E., *"image coding based on a fractal theory of iterated contractive image transformations*", ieee trans. Image processing, 1(1), 18-30, 1992.
7. M. F. Barnsley, fractals everywhere. San Diego, ca: academic, 1988.
8. E. Jacquin, "*image coding based on a fractal theory of processing*, vol. 1, pp. 18–30, jan. 1992.
9. Y. Fischer, *"a discussion of fractal image compression*," in chaos and fractals, h. O. Peitgen, h. Jurgens, and d. Saupe, eds. New York: springer-verlag, 1992.
10. G. E. Oien, "l2-*optimal attractor image coding with fast decoder convergence*," PhD. Dissertation, institute for teleteknikk, universities i Trondheim, Trondheim, Norway, 1993.
11. M. F. Barnsley and a. D. Sloan, "*a better way to compress images,"* byte, vol. 13, no. 1, pp. 215– 233, 1988.
12. M. Ghazel, g. H. Freeman, and e. R. Vrscay,"*fractal image denoising,"* ieee trans. Image process., vol. 12, no. 12, pp. 1560–1578, dec.2003.
13. R. Pan and s. J. Reeves, "*efficient huber-markov edge-preserving image restoration*," ieee trans. Image process., vol. 15, no. 12, pp. 3728–3735,dec. 2006.
14. Z. Wang, d. Zhang, and y. L. Yu, *"hybrid image coding based on partial fractal mapping*," signal process. image commun., vol. 15, pp.767–779, 2000.
15. R. Distasi, m. Nappi, and d. Riccio, *"a range/domain approximation error-based approach for fractal image compression*," ieee trans. Image process., vol. 15, no. 1, pp. 89–97, jan. 2006.
16. T. K. Truong, j. H. Jeng, i. S. Reed, p. C. Lee, and a. Q. Li, *"a fast encoding algorithm for fractal image compression using the dct inner product,"* ieee trans. Image process., vol. 9, no. 4, pp. 529–535, apr. 2000.
17. Yancong, Y. and Ruidong, P. *Fast Fractal Coding Based on Dividing of Image*, 2010.
18. Chaurasia, V. and Somkuwar, *A. Speed up Technique for Fractal Image Compression*, 2009.
19. Zhao, E. and Liu, D. *Fractal Image Compression Methods: A Review*. International Conference on Information Technology and Applications. 2005.