

Quad Compression Technique Using Parallelization Approach

R.Arulvelan¹, G.Thangaraj², K.Ragunathkumar³, M.Muthuraja⁴Research Scholar, Dept. of Computer Science and Engineering, Pawai College of Technology, Tamilnadu, India¹Research Scholar, Dept. of Computer Science and Engineering, Pawai College of Technology, Tamilnadu, India²Research Scholar, Dept. of Computer Science and Engineering, Pawai College of Technology, Tamilnadu, India³Asst. Professor, Dept. of CSE, Pawai College of Technology, Tamilnadu, India⁴

ABSTRACT: Fractal compression is a lossy compression method for digital images, based on fractals. The method is best suited for textures and natural images, relying on the fact that parts of an image often resemble other parts of the same image. Fractal algorithms convert these parts into mathematical data called "fractal codes" which are used to recreate the encoded image. With fractal compression, encoding is extremely computationally expensive because of the search used to find the self-similarities. Decoding however is quite fast. While this asymmetry has so far made it impractical for real time applications, when video is archived for distribution from disk storage or file downloads fractal compression becomes more competitive. In this project, the basic theory of image fractal compression is briefly introduced; some typical fractal compression approaches are discussed. Based on this discussion, according to the problem of long encoding time which exists in the typical fractal compression approaches, a fast image fractal compression approach is proposed based on the combination of adaptive quad tree compression approach and graph-based image segmentation algorithm. The project includes RGB to Grayscale conversion, image encoding, decoding, time elapsed details during encoding and decoding options. The improved approach can validly accelerate the fractal coding speed on the premise of guaranteeing the decoded image's quality.

KEYWORD: Data Compression, Image Classification, Data Compression Algorithm, Classification Accuracy.

I. INTRODUCTION

Data compression is a ubiquitous feature of modern society, occurring in many forms and in many settings. The variety of compression techniques is also very diverse, ranging from lossless compression of quantitative data through to lossy compression of diffuse data such as audio, image and movie files. The advantage of lossy compression is that excellent compression ratios may be obtained by ignoring aspects of the data that are unimportant. For example, if the human eye is incapable of distinguishing between two images then it is reasonable to assume that any differences between them are not significant. It follows therefore that, by their nature, lossy compression algorithms tend to be designed with a particular application in mind; image data being a particularly common example.

By far the most popular lossy compression algorithm for photographic types of image is the JPEG algorithm. This is widely used in all areas of image storage and processing, from digital cameras through to photographic archives and libraries. For most purposes the JPEG algorithm is highly satisfactory in that it is fast and efficient, allowing a choice between moderate compression ratio with small loss of quality, through to large compression ratio with a more noticeable loss of quality.

Other image compression techniques are available however. One such approach is based upon fractal image compression (FIC) [9, 22-32]. This technique seeks to exploit affine redundancy that is present in typical images in order to achieve high compression ratios, generally maintaining good image quality with resolution independence. The main drawback of FIC however is that there is a very high computational cost associated with the encoding phase. This is no doubt the main reason that the approach has remained less popular than faster alternatives such as JPEG. Nevertheless, there are a number of reasons why FIC should not be ignored. Firstly, although the encoding is

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

expensive, the decoding is fast and straightforward, and allows smooth images to be recovered at all levels of resolution. Furthermore, the compression ratios achieved can be very high and so if data density or data transmission rates are of more importance than real-time compression then FIC may be advantageous. Finally, as computer hardware continues to increase in speed and decrease in cost, resources that once seemed prohibitive, for encoding large quantities of data, have now become widely accessible.

Problem Definition

Lossy methods are especially suitable for natural images such as photographs in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences may be called visually lossless. Fractal compression is a lossy compression method for digital images, based on fractals. The method is best suited for textures and natural images, relying on the fact that parts of an image often resemble other parts of the same image. Fractal algorithms convert these parts into mathematical data called "fractal codes" which are used to recreate the encoded image.

With fractal compression, encoding is extremely computationally expensive because of the search used to find the self-similarities. Decoding however is quite fast. While this asymmetry has so far made it impractical for real time applications, when video is archived for distribution from disk storage or file downloads fractal compression becomes more competitive.

EXISTING WORK

Basic fractal image compression approaches don't consider the image's content, and divide images into regular square image block. When image block has a larger size, it can affect the recovered image's visual effect, and reduce (Signal to Noise Ratio) SNR. But when image block has a smaller size, it can increase D blocks' amount, thus increase the encoding time and reduce compression ratio. Inspired by this, it is very natural to think that use some statistic information of image block (such as gray average, variance, fractal dimension) to speed the matching process to reduce time complexity.

Drawbacks of Existing System

- Fractal image compression approaches don't consider the image's content.
- Divides images into regular square image block only.
- Increases the encoding time and reduce compression ratio.
- They only consider some statistic information of image block, and neglect the image block's semantics and the semantic relationship between block and block.
- The image block from the image's background has little probability of matching the image block from image's main content
- Once a R block can't find the best-match D block in its own search space, it can't expand its search to his adjacent search space, therefore, the decoding image has poor quality.

PROPOSED SYSTEM

A graph-based image segmentation algorithm is proposed, and this algorithm overcomes the shortcomings of traditional image segmentation algorithm, i.e., excessive segmentation over image.

At the same time, this algorithm can adjust its segmentation scale according to different images, thus, it can achieve better content-based image segmentation. In the fractal encoding process, most blocks which come from different image's contents have different textures, so blocks come from different image contents can't be the best-match block for each other in most cases, but for the blocks come from the same image content, they can easily match each other well.

Advantages of Proposed System

- Once R block can't find his best-match D block; it can expand its search to its adjacent class, so the decoded image has a higher quality.
- This algorithm can adjust its segmentation scale according to different images, thus, it can achieve better content-based image segmentation.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

- Most blocks which come from different image's contents have different textures, so blocks come from different image contents can't be the best-match block for each other in most cases, but for the blocks come from the same image content, they can easily match each other well.

System Architecture

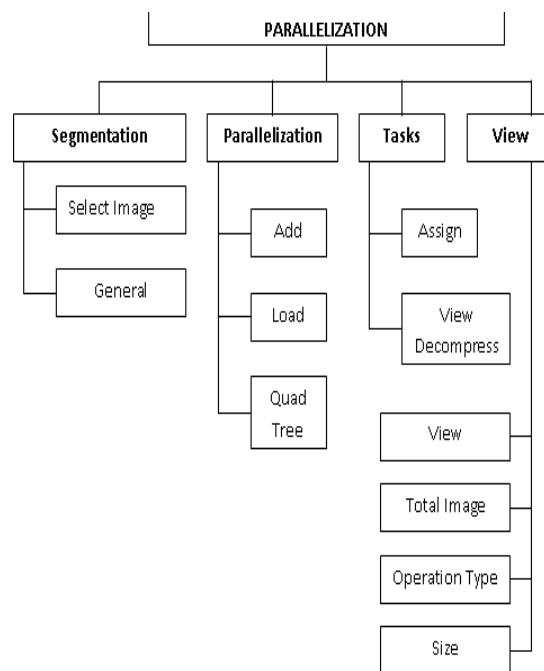


Fig 1.1 System Architecture

Overview

Image compression may be lossy or lossless. Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossy methods are especially suitable for natural images such as photographs in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate.

The lossy compression that produces imperceptible differences may be called visually lossless. Fractal compression is a lossy compression method for digital images, based on fractals. The method is best suited for textures and natural images, relying on the fact that parts of an image often resemble other parts of the same image.

II. RELATED WORKS

The basis for Fractal image compression is the construction of an Iterated Function System (IFS) that approximates the original image

A large number of different strategies have been considered for parallelization of the encoding stage of the fractal image compression algorithms outlined in the preceding section. The interest in using parallel computer architecture stems from the very high computational cost associated with fractal-based compression strategies. As has been explained above, for an $I \times I$ pixel image, the cost of compression is $O(I^4)$ and so the goal of a parallel implementation should be speed up the compression substantially. The decompression algorithm is considerably faster to

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

execute and so parallelization of this is of less importance: although this has received a small amount of attention in the literature.

There are a number of papers that have attempted to categorize and summarize the available approaches to parallel implementation of fractal compression. Perhaps the most noteworthy of these are [45, 49, 53], which each provide a description of a variety of parallelization techniques which are grouped together in different classes by these authors. In this review, our aim is not only to update these previous summaries, but also to present the available choices in the context of current computer architectures, which have developed significantly since the work of [49] for example. In addition, although heavily influenced by [45, 49, 53] our categorization is slightly different to any one of these. Finally, we will provide some thoughts on possible alternative parallel variants that are likely to be suited to modern parallel computing environments.

2.1 Classification by granularity

A very simple view of parallel algorithms typically classifies them as being either “fine grained” or “coarse grained”. Interestingly however, for parallel fractal image compression, a range of algorithms have been developed over time that have far more than just two different levels of granularity. At one extreme [52] proposes using n^2 processors for an $n \times n$ pixel image, with each processor working with a single pixel of the image. This is extremely fine-grained parallelism, where expressions such as the sums appearing in equations (6) or (7) are computed in parallel. At the other end of scale, the image can be partitioned amongst the available processors for each processor to then simultaneously apply the sequential fractal compression algorithm on its own sub-image without any communication [45]. This embarrassingly parallel algorithm is extremely coarse grained but suffers from the drawback that the quality of the compressed image will be inferior to that obtained using the sequential algorithm. This is because each processor is only working with a subset of the complete domain pool and so will not generally get as good a match with its range blocks as will be obtained using the sequential algorithm. For the remainder of this review we will only consider parallel algorithms which are able to reproduce the compressed images that are obtained using accepted sequential algorithms

2.2 Classification by load-balancing algorithm

In this section of the paper we describe the available parallel algorithms in terms of the techniques that they use to ensure that the parallel load is equally distributed across the available processing elements. These are described in terms of three general classes of algorithm however, as we will explain, these classes are not disjoint. We begin with a discussion of pipeline algorithms then discuss static load balancing based upon a partition of the range pool, and finish with a discussion of dynamically partitioning the range pool. The systolic architecture approach of [44] has already been introduced above. In this approach each range block is compared with a different domain block at any given step. Once the comparison step is completed the domain blocks are shifted to the next processor in the pipeline, for another comparison step. When all domain blocks have passed through the pipeline (which is actually a ring, so as to save any start-up overhead) the comparison step is complete for all range blocks. A similar technique is also adopted in [43]. Another pipeline approach is described in [48], which focuses on the design of a parallel image processing architecture. In this case the pipeline is used to enhance the performance of the arithmetic in calculating the terms in the expressions (6) and (7) using fine-grained parallelism.

2.3 Modifications to the standard compression algorithms

So far we have only discussed the parallel implementation of “standard” fractal compression algorithms. Typically these require each range block to be compared against the image of all domain blocks in the domain pool. If a satisfactory match is not found an adaptive quad-tree data structure will also be used. There are however a number of techniques that have been developed in an attempt to improve the speed of the sequential fractal compression algorithms without significantly reducing the quality of the compressed image. These complexity reduction schemes do have an effect on the quality of the compressed image however they are designed with the aim of minimizing this effect. A number of such techniques are described in. Current interest in this topic appears to be as active as ever, e.g.. A number of authors have attempted to combine such complexity reduction schemes with their parallel algorithms, as discussed in for example. Typically these schemes are based upon the use of classification techniques such as that outlined in section 2, where the elements of the range and domain pools are assigned to different classes and range

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

elements of a given class are only compared against the images of domain blocks from the same class. For example, in, this block classification approach is used to create 72 different classes based 11 upon the mean and variance values of the grey values in each block. Many other discrete feature methods exist however and can be incorporated within parallel algorithm.

III. CONCLUSION

It is believed that almost all the system objectives that have been planned at the commencement of the software development have been met with and the implementation process of the project is completed. A trial run of the system has been made and is giving good results the procedures for processing is simple and regular order. By analyzing the sequential version of fractal coding algorithm, parallelization of Fisher algorithm is researched and implemented based on parallel programming model. Experimental results show that the proposed parallel algorithm for fractal coding is feasible and improvement in the encoding speed is extremely obvious. The application eliminates the difficulties in the existing system. It is developed in a user-friendly manner. The application is very fast and any transaction can be viewed or retaken at any level. Error messages are given at each level of input of individual stages. This software is very particular in reducing the difficulty in compressing the image. The application works well for given tasks in network environment. Any node with .Net framework installed can execute the application.

REFERENCES

- [1] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes", *Proceedings of the Institute of Radio Engineers*, 40 (9): 1098-1101, 1952.
- [2] J. Ziv, A. Lempel, "Universal Algorithm for Sequential Data Compression", *IEEE Transactions on Information Theory*, 23 (3): 337-343, 1977.
- [3] Video codec for audio visual services of Px64 Kbits/s. CCITT Recommendation H. 261, Aug. 1990.
- [4] Digital Compression and Coding of Continuous-Tone Still Images Part 1, Requirements and Guidelines. ISO/IEC JTC1 Committee Draft 10918-1.
- [5] Digital Compression and Coding of Continuous-Tone Still Images Part 2, Compliance Testing. ISO/IEC JTC1 Committee Draft 10918-2.
- [6] M. L. Liou, "Overview of the Px64 Kbits/s Video Coding Standard", *Communications of the ACM*, Vol.34, No.4, 1991.
- [7] Gregory K. Wallace, "The JPEG Still Picture Compression Standard", *Communications of the ACM*, Vol.34, No.4, 1991.
- [8] S. G. Mallat, "Multifrequency Channel Decomposition of Images and Wavelet Models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.37, No.12, 1989.
- [9] A. E. Jacquin, "Fractal Image Coding: A Review", *Proceedings of the IEEE*, Vol.81, No.10, 1993.
- [10] Y. Fisher, "Fractal Image Compression: Theory and Application". New York: Springer-Verlag New York, Inc., 1995.
- [11] B. Wohlberg and G. d. Jager, "A Review of the Fractal Image Coding Literature", *IEEE Transaction on Image Processing*, vol. Vol. 8, 1999.
- [12] J. DONGARRA, D. WALKER, E. LUSK, et al, "Special Issue - MPI - A Message-Passing Interface Standard", *International Journal of Supercomputer Applications and High Performance Computing*, 8 (3-4): 165-&, 1994.
- [13] M. F. Barnsley, "Fractal Everywhere", Academic Press Inc., 2nd Edition, 1993.
- [14] Zhao Erjun, Liu Dan, "Fractal Image Compression Methods: A Review", *The Third International Conference on Information, Technology and Applications, ICITA2005(Sydney)*, pp.756-759, 2005.
- [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, 2006.