© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 07 Issue: 01 | January – March 2016

INTEGRATION OF QUAD TREE ALGORITHM & CONVENTIONAL DCT BASED FRACTAL IMAGE COMPRESSION FOR HIGHER COMPRESSION RATIO & PSNR WITH REDUCED COMRESSION ERROR

[1]Abhishek Sharma,Babu Banarasi Das University, Lucknow,India,

[2]Prof. Shubha Mishra, Deptt. of CSE., Babu Banarasi Das University,Lucknow,India,

abhishek.sharma697@gmail.com, iamshubha@gmail.com

Abstract: We have made image/picture compression using efficient fuzzy logic in this research. We have used quadtree algorithm for this purpose. We opt for fuzzy logic based method as fuzzy logic is considered strong tool to handle vagueness. When images are vague in terms of pixel values fuzzy logic is considered appropriate logic for its analysis. In proposed technique one domain block is considered for every range block & searched only for matched contrast scaling. So outcomes fractal code does not contain coordinates of matched domain block. *Quadtree algorithm* may be here applied in such case & size of range block may be minimized as small as 2x2 pixels. proposed research deals with *integration of quad tree algorithm with conventional DCT based fractal image compressio*n in order to produce higher compression ratio PSNR with less compression error.

[1] FRACTAL IMAGE COMPRESSION

Fractal compression is known as lossy compression method for digital images that is based on fractals. method is best suited for textures & natural images, relying on fact that parts of an image/picture often resemble o r parts of same image.

Fractal algorithms generally convert such parts into ma matical data known as fractal codes that are used to recreate encoded image. Fractal image/picture representation may be described ma matically as an iterated function system. Binary image/picture has been represented. Here image/picture may be thought of as a subset of $\mathbb{R}^2$. An IFS is a set of contraction mappings $f_1,...,f_N$,

$$f_i : \mathbb{R}^2 \to \mathbb{R}^2.$$

IFS describes a two-dimensional set S as fixed point of Hutchinson operator according to se mapping functions.

$$H(A) = \bigcup_{i=1}^{N} f_i(A), \quad A \subset \mathbb{R}^2.$$

H is an operator mapping sets to sets, & S is unique set satisfying $H(S) = S$. idea is to construct IFS such that this set S is input binary image. Set S may be recovered from IFS by fixed point iteration: for any nonempty compact initial set A_0, iteration $A_{k+1} = H(A_k)$ converges to S.

set S is self-similar because $H(S) = S$ implies that S would be union of mapped copies of itself:

$$S = f_1(S) \cup f_2(S) \cup \cdots \cup f_N(S)$$

So we see IFS is a fractal representation of S.

Extension to grayscale

IFS representation may be extended to a grayscale image/picture by considering image's graph as a subset of $\mathbb{R}^3$. For a grayscale image/picture $u(x,y)$, consider set $S = \{(x,y,u(x,y))\}$. n similar to binary case, S is described by an IFS using group of contraction mappings of $f_1,...,f_N$, but in $\mathbb{R}^3$,

$$f_i : \mathbb{R}^3 \to \mathbb{R}^3.$$

Encoding

Challenging problem of ongoing research in fractal image/picture representation is how to choose $f_1,...,f_N$ such that its fixed point approximates input image, & how to do this

efficiently. A simple approach for doing so is following:

1. Partition image/picture domain into blocks R_i of size $s \times s$.
2. For every R_i, search image/picture to find a block D_i of size $2s \times 2s$ that is very similar to R_i.
3. Select mapping functions such that $H(D_i) = R_i$ for every i.

It is important to find a similar block so that IFS accurately represents input image/picture, so a enough number of candidate blocks for D_i must be considered in second step,. A large search considering many blocks is computationally costly on o r hand,. This bottleneck of searching for similar blocks is why fractal encoding is much slower than for example DCT & wavelet based image/picture representations.

Features

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

[2] EVOLUTION OF FRACTAL COMPRESSION

Michael Barnsley developed fractal compression in 1987, & was granted several patents on technology. Mostly considered practical fractal compression algorithm was developed by Barnsley & Alan Sloan. Barnsley's graduate student Arnaud Jacquin implemented first automatic algorithm in software in 1992. All methods are based on fractal transform using iterated function systems. Michael Barnsley & Alan Sloan formed Iterated Systems Inc. in 1987 that was granted over twenty additional patents that were related to fractal compression.

Major breakthrough for Iterated Systems Inc. was automatic fractal transform process that eliminated requirement for human intervention during compression as was case in early experimentation with fractal compression technology. Iterated Systems Inc. received a US$2.1 million government grant to develop a prototype digital image/picture storage & decompression chip using fractal transform image/picture compression technology in 1992. Fractal image/picture compression has been used in a number of commercial applications: onOne Software, made under license from Iterated Systems Inc., Genuine Fractals five that is Photoshop plugin capable to save files in compressed Fractal Image Format. To date most successful use of still fractal image/picture compression is by Microsoft in its Encarta multimedia encyclopedia.

Iterated Systems Inc. supplied a shareware encoder, a Netscape plug-in decoder, a stand-alone decoder & a development package for use under Windows. As wavelet-based methods of image/picture compression improved & were more easily licensed by commercial software vendors adoption of Fractal image/picture Format failed to evolve. Redistribution of decompressor DLL that was provided by ColorBox III SDK was checked by restrictive per disk licensing regimes to proprietary software vendors & by a discretionary scheme that entailed promotion of Iterated Systems products for certain classes of o r users.

During 1990s Iterated Systems Inc. & its partners expended considerable resources to bring fractal compression to video. While compression results were found promising, computer hardware at that time have less processing power for fractal video compression to be practical beyond a few select usages. About 15 hours were needed to compress a single minute of video.

ClearVideo that is also known as RealVideo (Fractal) & SoftVideo were old fractal video compression products. ClearFusion was considered a freely distributed streaming video plugin for web browsers. SoftVideo had been licensed to Spectrum Holobyte for use in its CD-ROM games including Falcon Gold in 1994 & Star Trek: Next Generation A Final Unity.

In 1996, Iterated Systems Inc. announced an alliance with Mitsubishi Corporation to market ClearVideo to ir Japanese customers. original ClearVideo 1.2 decoder driver is supported by Microsoft in Windows Media Player although encoder is not supported.

Two firms, Total Multimedia Inc. & Dimension, both claim to own or have exclusive licence to Iterated's video technology, but nei r has yet released a

working product. technology basis appears to be Dimension's U.S. patents 8639053 & 8351509, that have been considerably analyzed. In summary, it is a simple quadtree block-copying system with neither bandwidth efficiency nor PSNR quality of previous DCT-based codecs. Old research papers have been published during past few years discussing possible solutions to get better fractal algorithms & encoding hardware.

[3] QUADTREE

A **quadtree** is a tree data structure in that every internal node has exactly four children. Quadtrees are used to partition a 2D space by recursively subdividing it in four quadrants/regions. regions may be square or rectangular, or may have arbitrary shapes. Such data structure was called a quadtree by Raphael Finkel & J.L. Bentley in 1974. A similar partitioning is called as a *Q-tree*. All forms of quadtrees share some features:

1. Decompose space into adaptable cells
2. Every cell has a maximum capacity. As maximum capacity is reached, bucket splits
3. Tree directory follows spatial decomposition of quadtree.

Quadtrees may be classified according to type of data. They represent, including areas, points, lines & curves. Quadtrees may also be classified by whether shape of tree is independent of order data is processed. Some common types of quadtrees are region quadtree represents a partition of space in two dimensions by decomposing region into four equal quadrants, subquadrants, & so on with every leaf node consisting data corresponding to a specific subregion. every node in tree either has exactly four children, or has no leaf node. Region quadtree is a type of trie. A region quadtree with a depth of n may be used to represent an image/picture comprises $2^n \times 2^n$ pixels, where every pixel value is 0 or 1. root node represents entire image/picture region. If pixels in any region are not entirely 0s or 1s, it is subdivided. In such application, every leaf node represents a block of pixels that are all 0s or all 1s.

Point quadtree

Point quadtree is considered as an adaptation of a binary tree used to represent two-dimensional point data. It shares features of all quadtrees but is a true tree as center of a subdivision is always on a point. Tree shape depends on order in that data is processed. It is efficient in comparing two-dimensional, ordered data points, usually operating in O(log n) time.

Edge quadtree

Edge quadtrees are specifically used to store lines ra r than points. Curves are approximated by subdividing cells to a very fine resolution. This may result in extremely unbalanced trees that may defeat purpose of indexing.

Polygonal map quadtree

polygonal map quadtree (or PM Quadtree) is a variation of quadtree that is used to store collections of polygons that could be degenerate. There are three main classes of PMQuadtrees, that vary depending on what information y store within every black node. PM3 quadtrees may store any amount of non-intersecting edges & at most one point. PM2 quadtrees are same as PM3 quadtrees except that all edges must share same end point. PM1 quadtrees are considered similar to PM2, but black nodes could contain a point & its edges or just a set of edges that share a point, but you cannot have a point & a set of edges that do not contain point.

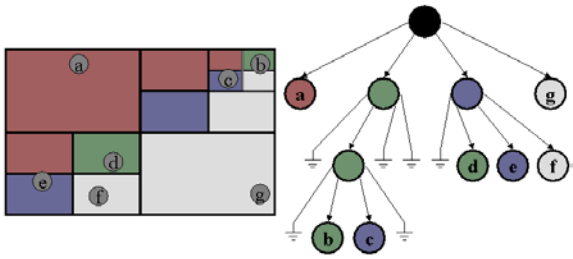

Fig 1 Quadtree example

[4] DCT

DCT is an orthogonal transform, Discrete Cosine Transform attempts to decorrelate image/picture data. After decorrelation every transform coefficient could be encoded independently without losing compression efficiency.

DCT separates images in parts of various frequencies where less significant frequencies are discarded through

quantization important frequencies are used to retrieve image/picture during decompression. Compared to other input dependent transforms, DCT has many benefits:

a) It has been implemented in single integrated circuit.
b) It has capability to pack most in-formation in fewest coefficients.
c) It reduces block like appearance called blocking artifact that results when boundaries between sub-images become visible.

Discrete cosine transform helps divide image/picture into parts of differing importance. DCT is similar to discrete Fourier transformation that transforms an image/picture from spatial domain to frequency domain.

Original image/picture is usually divided into 8 x 8 blocks. Pixel values of a black white image/picture range from 0-255 where 0 corresponds to a pure black 255 correspond to a pure white. But DCT is designed to work on pixel values ranging from -128 to 127. refore every block is modified to work in range. n working from left to right, top to bottom & DCT has been applied to every block.

- Than every blocks elements are compressed through Quantization means breaking by some fixed 8X8 matrix called QMatrix rounding to nearest integer value.
- After Quantization, all of quantized coefficients are ordered into zigzag sequence.
- Compressed image/picture is reconstructed through re-verse process. Inverse DCT is used for decompression.

[5] HYBRID IMAGE COMPRESSION

Hybrid image/picture Compression using Quadtree DCT has been used that will performs discrete cosine transformation on discrete wavelet transformed coefficients. Compression techniques are useful to compressed one image/picture for high value of peak signal to noise ratio compression ratio this method known as called hybrid compression technique.

[6]OBJECTIVE OF RESEARCH

Main objective is to design a more efficient compression system by integrating quad tree algorithm conventional DCT based fractal image/picture compression, suitable for processing, storage transmission, as well as providing acceptable computational complication suitable for practical implementation. Basic rule of compression is to reduce numbers of bits needed to represent an image. In a computer an image/picture is usually represented as an array of numbers, integers to be more specific, that is called a digital image. Number of bits required to represent information in an image/picture can be reduced by removing redundancy present in it.

[7] PROPOSED METHODOLOGY

Proposed work deals with integration of quad tree algorithm with conventional DCT based fractal image/picture compression in order to produce higher compression ratio PSNR with less compression error. Main purpose is to develop a compression system that would be suitable for processing, storage transmission. Transformation is a lossless step in that image/picture is transformed from grayscale values in special domain to coefficients in some o r domain. No loss of information occurs in transformation step.

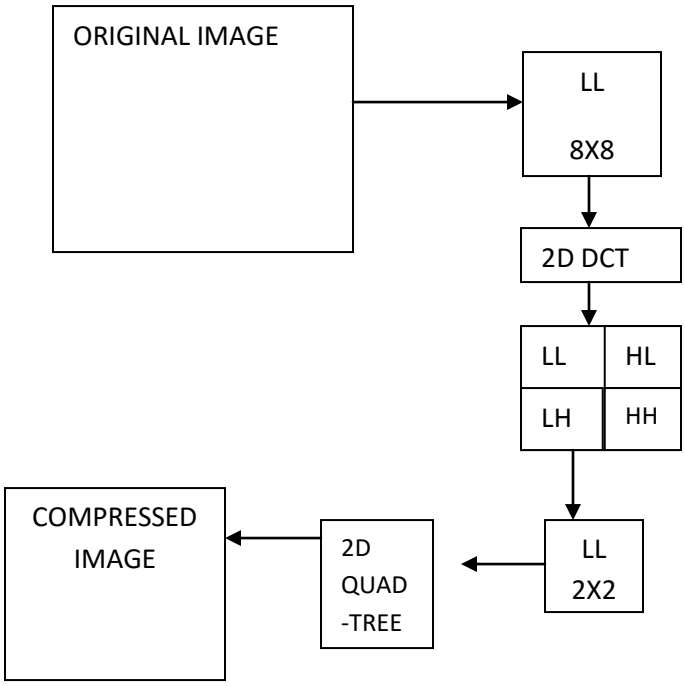

Fig 2

[8] SCOPE OF RESEARCH

Fuzzy logic technique used for calculation of coefficients could be modified to improve its accuracy. Fuzzy based image/picture compression technique provides higher compression ratio as compare to normal hybrid image/picture compression. Quality of decoded image/picture can be improved while compression ratio can be maintained. Benefit of using fuzzy based technique is that, Fuzzification of an image/pictureleads to reduction in contrast brightness of input image/picture to be compressed. Benefit of this reduction in contrast brightness is that, this reduction leads to increase pixel redundancy hence help to raise compression ratio & peak signal to noise ratio during image/picture compression.

Reference

1. A. E. Jacquin, “Image coding based on a fractal ory of iterated contractive image/picture transformations,” IEEE Trans. Image Process.,vol. 1, no. 1, pp. 18–30, Jan. 1992.
2. A. E. Jacquin, “Fractal image coding: A review,” Proc. IEEE, vol. 81, no. 10, pp. 1451–1465, Oct. 1993.
3. H. Hartenstein, M. Ruhl, & D. Saupe, “Region-based fractal image compression,” IEEE Trans. Image Process., vol. 9, no. 7, pp. 1171–1184, Jul. 2000.
4. Z. Wang & A. C. Bovik, “A universal image quality index,” IEEE Signal Process. Lett., vol. 9, no. 3, pp. 81–84, Mar. 2002.
5. T. K. Truong, C. M. Kung, J. H. Jeng, & M. L. Hsieh, “Fast fractal image compression using spatial correlation,” Chaos Solitons Fractals, vol. 22, no. 5, pp. 1071–1076, 2004.
6. R. Distasi, M. Nappi, & D. Riccio, “A range/domainapproximation error-based approach for fractal image compression,”IEEE Trans. Image Process., vol. 15, no. 1, pp. 89–97, Jan. 2006.
7. M. Ghazel, G. H. Freeman, & E. R. Vrscay, “Fractal-wavelet image denoising revisited,” IEEE Trans. Image Process., vol. 15, no.9, pp. 2669–2675, Sep. 2006.
8. S. G. Lian, “Image au ntication based on fractal features,” Fractals, vol. 16, no. 4, pp. 287–297, 2008.
9. Y. Zhou, C. Zhang, & Z. Zhang, “An efficient fractal image coding algorithm using unified feature & DCT,” Chaos Solutions Fractals, vol. 39, no. 4, pp. 1823–1830, 2009.
10. X. Tang & C. Qu, “Facial image recognition based on fractal image encoding,” Bell Labs Tech. J., vol. 15, no. 1, pp. 209–214, 2010.
11. H. N. Chen, K. L. Chung, & J. E. Hung, “Novel fractal image encoding algorithm using normalized one-norm & kick-out condition,” Image Vis. Comput., vol. 28, no. 3, pp. 518–525, 2010.
12. J. Wang, Y. Liu, P. Wei, Z. Tian, Y. Li, & N. Zheng, “Fractal image coding using SSIM,” in Proc. 18th ICIP, Sep. 2011, pp. 245–248.
13. W. R. Schwartz & H. Pedrini, “Improved fractal image compression based on robust feature descriptors,” Int. J. Image Graph., vol. 11, no. 4, pp. 571-587, 2011.
14. K. T. Lin & S. L. Yeh, “Encrypting image by assembling fractalimage addition method & binary encoding method,” Opt. Commun.,vol. 285, no. 9, pp. 2335–2342, 2012.