

FRACTAL IMAGE COMPRESSION USING QUANTUM ALGORITHM

T Janani* and M Bharathi*

Department of Electronics and Communication Engineering,
Kumaraguru College of Technology, Coimbatore, India - 641049.
Email: bharathi.m.ece@kct.ac.in, janani1993@gmail.com

ABSTRACT

Fractal image compression (FIC) is an image coding technology based on the local similarity of image structure. FIC offers high compression ratio without degrading quality of retrieved images, which makes FIC, a widely approved technology. However, despite of the linearity of the decoding phase, the coding process is much more time consuming, because of search involved in finding local self-similarities in an image. Algorithms like Quad tree Partitioning Huffman Coding (QPHC) and DCT based FIC (DCT-FIC) have been developed to reduce the computational complexity in the coding phase. The proposed method, FIC through quantum representation exploit enhanced computational power and huge storage capacity, which makes it significantly faster than any classical algorithm solving the same problem. For this reason, an attempt is made to apply QA to reduce the computational complexity of FIC.

Keywords: Fractal Image Compression, Local self-similarities, Quantum Algorithm, Iterated Function System.

INTRODUCTION

Image compression is an essential technology which deals with minimizing number of bits used to represent a digital image and maintaining good quality of the retrieved image. Data compression has become a vital concern for information transmission and storage. Large amount of data cannot be stored if there is a low storage capacity, hence the compression plays a very vital role in transmission and storage. Compressing an image significantly differs from compressing raw binary data. There are various general purpose compression algorithms, which can be used to compress images, but the result is less than optimal. Because the images have certain statistical properties which can be exploited by encoders specifically designed for them and also, some of the finer details in the image will be sacrificed for the purpose of saving a little more bandwidth or storage space. Therefore, the objective of image compression is to reduce irrelevance and redundancy of the image data in order to store or transmit data in an efficient form.

FRACTAL IMAGE COMPRESSION

Self-similarity concept is the basis and premise of Fractal image compression [1]. Fractal Image compression (FIC) is the compression techniques used in the spatial domain which encodes the image in such a way that it reduces the storage space by using self-similar portion of the same image. A general image has copies of parts of itself rather than the whole self. For example, the image Lena in Fig. 1 has sample regions in the white squares. These sample regions are similar at different scales: a portion of her shoulder overlaps a region that is almost identical, and a portion of the reflection of the hat in the mirror is similar to a part of her hat [17].



Fig. 1. Lena Image with Self-similarities at different scale

For conventional fractal compression schemes [7], an image is partitioned into domain blocks and range blocks, the self-similarities exploiting, between these two kinds of blocks in the spatial domain are computationally expensive, usually hundreds of seconds is used in encoding an image, which restricts the application of fractal image compression.

The process of fractal image coding is finding the appropriate domain block for each range block using Iterated Function System (IFS) mapping [8]. In IFS mapping, the coefficient will represent a data corresponding to the block of the compressed image. Thus a digitized image can be stored as a collection of IFS transformation parameters and is easily regenerated or decoded for use or display. The storage of the IFS transformation coefficients relatively results in high compression ratios and better reconstruction fidelity. Fig. 2 illustrates the storage of IFS transformation coefficients along with fractal structure [1,2].

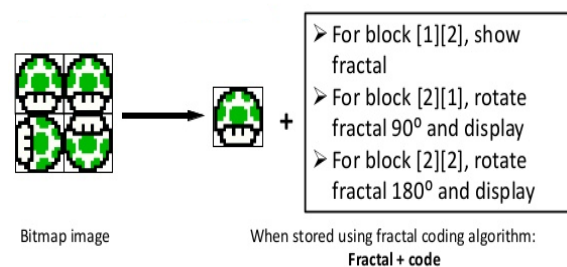


Fig. 2. Fractal Image and Storage of IFS

Transformation coefficients with Fractal Structure

The decision making levels in Fractal compression system are, Image Partitioning, Domain Block Transformation and Suitable Domain Search. The following section addresses the different techniques that can be adopted in each decision level.

A. Image Partitioning

In all the Fractal compression system, the first and foremost decision is to choose the type of image partition for the range block and domain block formation. A wide variety of partition has been investigated [17]. Of which, Fixed size square blocks are the simplest of all possible partitions. Since, they are easy to implement, but its performance decreases for images

with varying “activity” levels of different range blocks [8]. The solution of this problem is to use some adaptive scheme for block size so that large blocks are assigned to a low detail region and small blocks for the significant detail region.

Quad-tree partition is an adaptive partition scheme which divides an image into a variable size range block. First, a square image is split into square blocks of equal sizes, and then each block is checked whether the specified criteria of homogeneity are met [5]. If a block meets the specified criteria, it is not divided any further, if the block does not meet the criteria, then the block is split into further four blocks and again the test is applied to the blocks split. Two approaches that can be used are Fixed-size partitions and Quad-tree partitions given in Fig. 3 respectively.

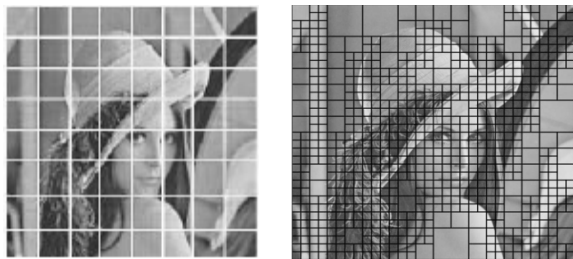


Fig. 3. Fixed-size partitions and Quad-tree partitions respectively

B. Domain Block transformation

The next step is to apply transforms on domain blocks to form range blocks and determine the convergence properties of decoding. All the transforms applied for this purpose should be contractive in nature [9,10]. Each transform can skew, stretch, rotate, scale and translate any domain image. The pixel values of the contracted image blocks on the range block are the average values taken from the four neighboring pixels of the domain block. T_i is a transformation applied to process image blocks. These transforms generally do not modify pixel values; they simply shuffle pixels within a range block in a deterministic way. They are also called as isometrics. The generally used isometric operators are an orthogonal reflection about a desired axis. These transforms are also performed some gray scale operations. Above explained transformation is universally accepted for fractals. Extensions to the transform could be possible by using multiple fixed blocks, i.e., fixed blocks with a constant gradient in the horizontal and vertical directions respectively. By including blocks with quadratic form and also by adding cubic blocks, further extensions are possible. Second order transformation provides best results in a rate distortion sense [7].

C. Suitable Domain Search

After selection of suitable partitioning, domain-pool and transformation, the next step of fractal encoding process is search of suitable candidate from all available domain blocks to encode any particular range block [12,13]. This step of fractal image compression is computationally expensive, because it requires a large number of domain range comparisons [11]. The

attempts to improve encoding speed are addressed as speedup techniques.

METHODOLOGY

A. DCT based Fractal Compression

To improve the fractal encoding speed, the algorithm proposes a new block classification method based on the edge characteristic of an image block. The essence of this method is that if the domain block has the same edge characteristic to the range block then they are similar in fractal meanings. By restricting the exploiting range of domain block, this method can not only fasten the fractal encoding speed, but also guarantees the quality of the decoded image [16]. In DCT coefficients, lower frequency coefficients represent the main energy of an image, while the higher frequency coefficients represent the edge information. Therefore, if two image blocks are similar besides some detailed information, then their DCT lower frequency coefficients are approximately equal. So it is sufficient to use lower coefficient for evaluating the similarity degree between two image blocks [9]. The general steps involved in DCT fractal compression are:

- Read the binary image.
- Convert it into a gray level image.
- Partition the converted image into non-overlapping small square blocks called as range blocks.
- Introduce overlapping large square blocks called as domain blocks. The size of domain block is double the size of range block.
- For every range block (R_i) find the matching domain block (D_j) which closely resembles range block with respect to some metric and accordingly parameters are computed.
- Write the compressed data in the form of local IFS code.
- Apply data compression algorithm to obtain a Compressed IFS code.

The best matching D block for R block is determined by evaluating the MSE (Mean Square Error) between R_i and each D_j . The minimum MSE means the best match. The MSE is determined by

$$MSE = \frac{\sum_{k=1}^N [r_k - s_i \cdot d'_k + o_i]^2}{N}, (N = B \times B)$$

Where, S_i and O_i are the contrast and brightness factor of the i^{th} range block respectively.

Finally, the four parameters of the best match D block constructs the fractal code, they are position of the block, isometric transformation number, contrast factor and brightness factor.

B. Quantum Based Fractal Compression

Three key steps in quantum based Fractal Compression are, partition and transformation, quantum representation of classical images, and search optimal fractal code with QA, which is explained as a flow diagram in Fig. 4.

- Partition and Transformation: Firstly, a given input image with size $M \times N$ is partitioned into two kinds of

square blocks: one is the non-overlapping range blocks with size $B \times B$, and the other one is the overlapping domain blocks with size $2B \times 2B$. The size of the range pool can be easily calculated by dividing $M \times N$ by $B \times B$, and the size of the domain pool should be $M - 2B + 1 \times N - 2B + 1$. Subsequently, all domain blocks are contracted into the same size with range blocks by a spatial contraction, such as averaging four pixels to one pixel, etc. After that, to improve the quality of retrieved images, isometric operations are applied to all domain blocks to octuple the number of domain blocks.

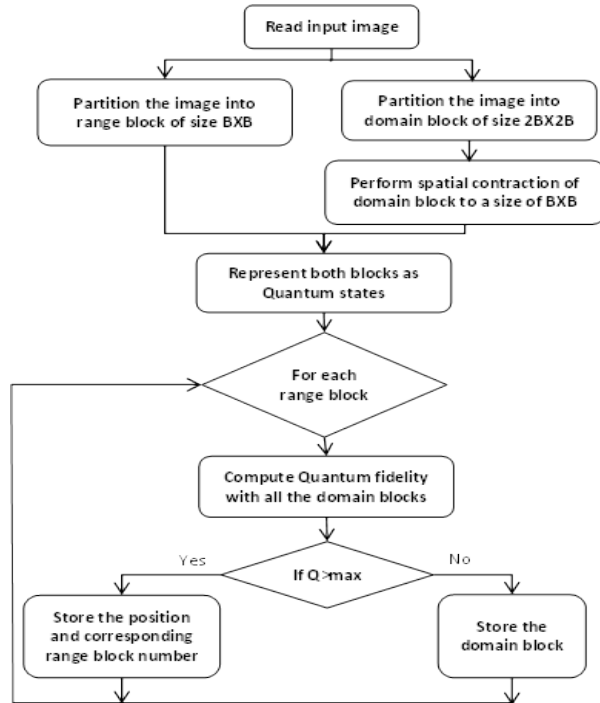


Fig. 4. Flow chart for Quantum Algorithm based Fractal Image Compression

- **Quantum Representation of Classical Image:** To make the search of QA effective in FIC, we represent both domain block and range block as quantum states. A flexible method to represent classical images as normalized quantum states is proposed in [16]. It is performed by capturing information about colours and their corresponding positions in the image. For example, to represent a gray-scale image of 8-bit depth, we need 256 angles θ_i to encode its gray levels, where $i = 1, 2, \dots, 2^8 - 1$. The quantum state $|I\rangle$ which is represented from an original gray-scale image I is

$$|I\rangle = \cos\theta_i |0\rangle + \sin\theta_i |1\rangle$$

Here, $\theta_i \in [0, \frac{\pi}{2}]$. In this scenario, let's assume that $f(m,n) \in [0,1]$ is the normalized grayscale value of a pixel in domain block (or range block), where (m, n) denotes the coordinates of the pixel. Then θ_i can be achieved by

$$\theta_i = \frac{\pi}{2} \cdot f(m,n)$$

The quantum representation of the domain block D by

$$|D\rangle = \cos f(m,n) \cdot \frac{\pi}{2} |0\rangle + \sin f(m,n) \cdot \frac{\pi}{2} |1\rangle$$

The quantum representation of the range block R by

$$|R\rangle = \cos f(m,n) \cdot \frac{\pi}{2} |0\rangle + \sin f(m,n) \cdot \frac{\pi}{2} |1\rangle$$

- **Quantum Algorithm:** Based on the above preparations, the best matching domain block for every range block can be searched by QA. In the quantum scenario, the proximity between two states is measured from the quantum fidelity. In the Quantum based FIC, the best matching domain block for every range block is determined by maximizing their quantum fidelity, i.e.

$$\max_{i \in \{1, \dots, N\}} [\text{Tr}(\sqrt{\rho_D \rho_R \sqrt{\rho_D}})]^2$$

Where $\text{Tr}(\cdot)$ denotes matrix trace, ρ_D and ρ_R are density matrices of quantum states $|D\rangle$ and $|R\rangle$, respectively.

Compression result is achieved by recording parameters of the search results, optimal affine scalar parameters, serial number of the best matching domain block, and serial number of the isometric operations.

SIMULATION RESULTS

In this section, the performance of proposed method in comparison with other fractal image coding methods like QPHC [21], DCT-FIC is discussed. All the methods are programmed using MATLAB on Intel(R) Core i5 2.5 GHz PC. The execution time in each method is measured in seconds. The decoded image quality is measured by peak signal-to-noise ratio (PSNR) defined by:

$$PSNR = 20 \log \frac{\text{Max}_i}{\sqrt{MSE}}$$

Here Max_i is the maximum pixel value of the image.

$$MSE = (1/N) \sum_{i=1}^N (f_i - \hat{f}_i)^2$$

The coding methods are applied on several types of images: natural images, textures, satellite images, benchmark images shown in Fig. 5, such that the performance of proposed algorithm can be verified for various applications. These benchmark images are the standard image generally used for the image processing applications. The results of the meticulous simulation for all images and are presented in this section.

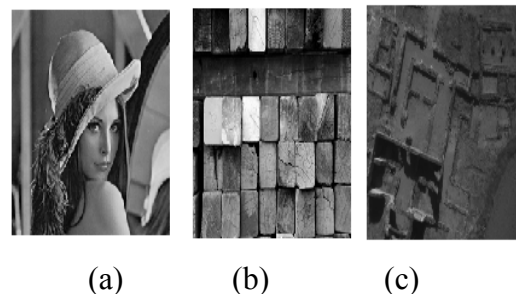


Fig. 5. Input image sets: Lena, Texture and Satellite Image

For testing, select benchmark 8 bit level gray-scale Lena image with 256×256 pixels. The encoding time taken in QPHC algorithm is 76.8s, which is relatively longer than other algorithms. However, the encoding speed is significantly improved than standard fractal coding scheme. Although the decoded image has no relationship with the selected initial image, the

decoding time is variant with different initial image. With the size of R block and D block are 16×16 , 32×32 respectively and the step of adjacent D block is 8, DCT algorithm takes only 7.69s to encode the image. On compression ratio aspect, each R block needs 2×8 bits to markup the position of the best match D block, and 7 bits for brightness factor o , 5bits for contrast factor, and 3bits for the isometric transformation No.. There are 32×32 R blocks, so only $1024 \times (16+7+5+3)/8=3968$ Bytes is needed to store the compressed image. The size of original image is 65536 bytes, so the compression ratio is 14.6, which is less than QPHC. Whereas on, PSNR aspect, it can reach 34.32dB which is relatively higher than QPHC.

The proposed QA achieves almost same performance as that of DCT, while at the same time reduces the time taken for compression to 1.2s shown in Fig. 7. Fig. 5 (a) is the original Lena image, Figure 6a, b and c are compressed images of Lena, based on QPHC, DCT and Quantum with PSNR as 26.66, 34.32 and 30.36 respectively.

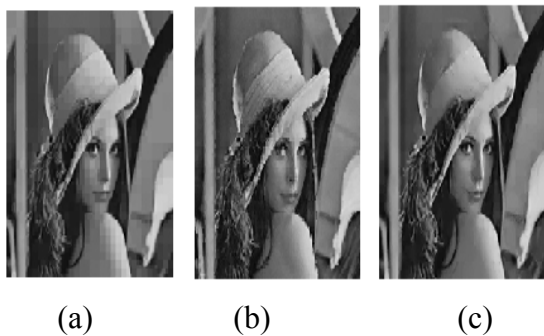


Fig. 6. Compressed Lena Image obtained from QPHC, DCT-FIC and Quantum Algorithm

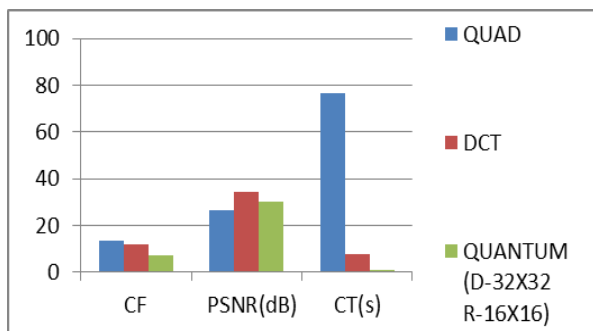


Fig. 7. Comparison of Performance Metrics for Lena Image

Fig. 8(a), (b), (c) shows compressed images of Texture, based on QPHC, DCT and Quantum with PSNR as 24.23, 32.14 and 30.81 respectively.

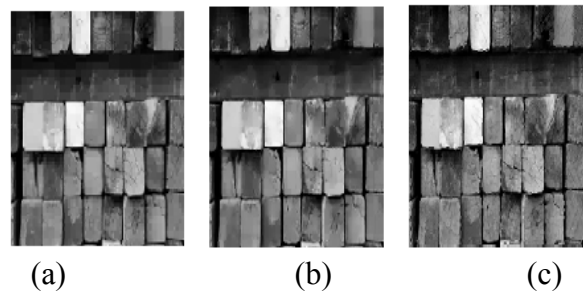


Fig. 8. Compressed Texture Image obtained from QPHC, DCT-FIC and Quantum Algorithm

Figure 10a, b and c shows compressed images of Satellite, based on QPHC, DCT and Quantum with PSNR as 29.21, 33.99 and 38.03 respectively.

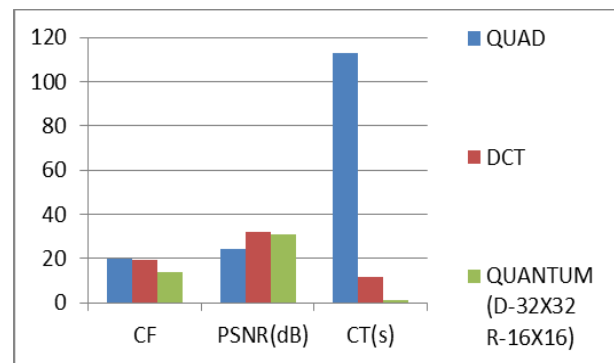


Fig. 9. Comparison of Performance Metrics for Texture Image

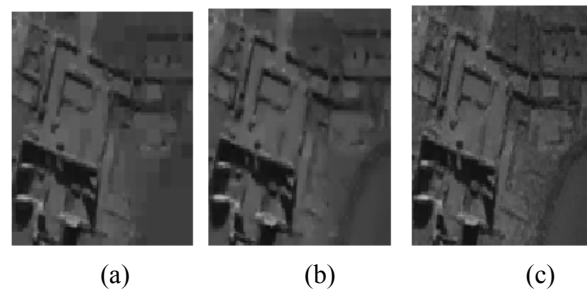


Fig. 10. Compressed Satellite Image obtained from QPHC, DCT-FIC and Quantum Algorithm

Fig. 7, Fig. 9 and Fig. 11 depicts the comparison of performance metrics for Lena, Texture and Satellite image respectively and it is observed that, in all three algorithms Compression ratio obtained for fractal geometry based image is higher than that of Lena image.

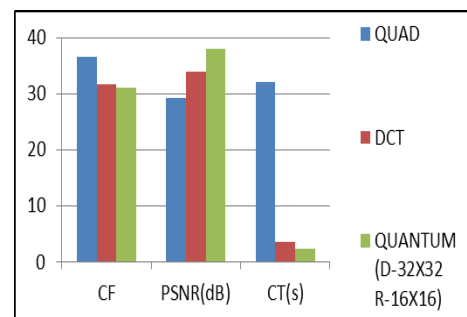


Fig. 11: Comparison of Performance Metrics for Satellite Image

The proposed algorithm outperforms the other algorithm in terms of Compression time. The time taken for encoding in QA is 1.2s, 1.17s, 2.34s for Lena, Texture and Satellite image respectively.

CONCLUSION

Fractal Image Compression has been carried out using Quadtree Partitioning with Huffman Coding (QPHC) algorithm, DCT based Fractal Image Compression Algorithm (DCT-FIC) and Quantum Algorithm (QA). Based on the comprehensive simulation results presented for different images it can be seen that the DCT-FIC and QA algorithm outperforms QPHC algorithm. Especially, QA performs better for the images that consist of detailed view and structural similarities. Hence, it can be implemented for compressing natural, texture and satellite images. In spite of more advancement [7], computational and time requirements of encoding part remained as a drawback in the standard algorithms such as DCT. Besides these drawbacks, DCT technique consistently provides more compression ratio, resolution independency and a better reconstruction quality. It is also able to reduce the false contouring effect and artifacts for images. Therefore the drawback in DCT-FIC is overcome in QA by obtaining 50% reduced compression time than DCT.

In order to improve Compression ratio in QA, the size of range and domain block is reduced further. Smaller size of the block indicates a larger compressed file, because of more fractal codes. This further results in larger encoding times, so as a future work, Grover's database search algorithm (QSA) can be implemented to achieve faster encoding times. Grover's Search Algorithm, a method for manipulating quantum bits that result in a successful search with certainty in only $O(\sqrt{N})$ attempts.

REFERENCES

- [1] Barnsley, M.F. and A. E. Jacquin, Application of recurrent iterated function systems to images. Proc. SPIE **1001**: 122–131 (1988).
- [2] Jacquin, A.E., Image coding based on a fractal theory of iterated contractive image transformations. IEEE Trans. Image Process **1**(1):18-30 (1992).
- [3] Kaouri, A. H., Fractal coding of still images. Queen's university of Belfast, UK (2002).
- [4] Hu, L., Q. Chen and Z. Qing, An image compression method based on fractal theory. The 8th Int. Conf. on computer supported cooperative work in design Proc. Pp. 546–550 (2003).
- [5] Veenadevi, S. V. and A.G. Ananth, Fractal image compression using Quadtree decomposition and Huffman coding. Signal and Image Processing International Journal **3**(2): 207-212 (2012).
- [6] Gonzalez, R. and R.Eugene, Digital image processing (2008).
- [7] Yancong, Y. and P. Ruidong, Fast Fractal Coding Based on Dividing of Image (2010).
- [8] Distasi, R., M. Nappi and D. Riccio, Range/domain approximation error-based approach for fractal image compression. IEEE trans. Image Processing **15**(1): 89–97 (2006).
- [9] Truong, T.K., 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 **9**(4): 529–535 (2000).
- [10] Chaudhari, R.E. and S. B. Dhok, Wavelet transformed based fast fractal image compression. Proc. Int. Conf. Circuits, Systems, Communication and Information Technology Applications (CSCITA) Pp. 65-69 (2014).
- [11] Lu, J., Z. Ye and Y. Zou, Huber fractal image coding based on a fitting plane. IEEE Trans. Image Process **22**(1): 134–145 (2013).
- [12] Jianji Wang and Nanning Zheng, A Novel Fractal Image Compression Scheme with Block Classification and Sorting Based on Pearson's Correlation Coefficient. IEEE Trans. Image Process **22**(9): 3690-3702 (2013).
- [13] Kamel Belloulata and Janusz Konrad, Fractal Image Compression with Region-Based Functionality. IEEE Trans. Image Process **11**(4): 351-362 (2002).
- [14] Michele Nappi, Riccardo Distasi and Daniel Riccio, A Range/Domain Approximation Error-Based Approach for Fractal Image Compression. IEEE Trans. Image Process **15**(1): 89-97 (2006).
- [15] Saupe, D., Accelerating fractal image compression by multidimensional nearest neighbor search. IEEE Data Comp. Conf. Proc. Pp.222–231 (1995).
- [16] Fu, C. and Z. Zhu, A DCT-based fractal image compression method. In Proc. Int. Workshop Chaos-Fractals Theories and Applications (IWCFTA) Pp. 439–443 (2009).
- [17] Hitashi Gaganpreet Kaur and Sugandha Sharma, Fractal image compression- a review. Int. Journal of Advanced Research in Computer Science and Software Engg. **2**(2): (2012).