DCT with Quad tree and Huffman Coding for Color Images

Sandhya Kadam
Research Scholar, Faculty of Engg.
Pacific Academy of Higher Education and Research,
Udaipur, India

Vijay Rathod
St. Xavier's Institute of Technology
Mumbai, India

ABSTRACT

Many methods are available for compression of an image file. Images are usually in the form of matrices and an uncompressed image uses a huge number of bytes for storage. Its applications in various fields are quality control, remote sensing, imaging science etc. The image compression methods which are popular on the transform based coding methods like Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and fractals. However, these methods have drawbacks like low compression ratio and high encoding time. The proposed hybrid technique combines DCT and fractal quad tree decomposition with Huffman encoding of fixed threshold value for color images. The results for the proposed method are displayed and compared for performance parameters as compression ratio, encoding time, decoding time and PSNR.

General Terms

Discrete Cosine Transform, Fractal coding, Huffman Coding

Keywords

DCT, Fractal Quad tree, Huffman coding, Fractal Image Compression, Hybrid methodology

1. INTRODUCTION

Image compression can be lossy or lossless compression. Lossless compression produces the exact replica of the original image which is used to compress binary data like executables, documents etc. In lossy compression techniques, some of the high-quality details in the image can be sacrificed for saving a little more bandwidth or storage space and images need not be reproduced exactly as long as an error between the original image and the compressed one is tolerable. So there is an approach to implement an efficient algorithm for lossy image compression. Images are usually in the form of matrices. An uncompressed image uses huge number of bytes for storage. Image compression is used to reduce the size of the bitmap without compromising on the quality of an image. This enables to store a large number of images in the given amount of storage space. Compressing an image will reduce the overall time taken to transfer or to receive images from the internet.

Here, different transform coding techniques as DCT and fractal are combined to compress an image.

2. LITERATURE REVIEW

Transform coding is an efficient coding scheme based on utilization of inters pixel correlation. The simple block diagram of the transform based image compression method is shown in fig. 1. Transform is applied to input image and transformed image is quantized. After quantization, entropy coding is applied to compress the input image.

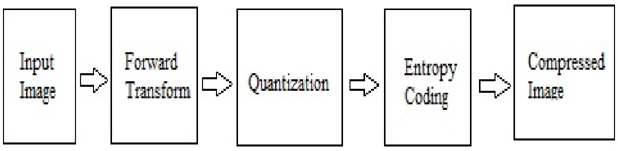

Fig 1: Transform based Image Compression

Fractal Image Compression [1] was first proposed by Barnsley in 1988. It was later improved by A.E Jaquin in 1992[2]. Fractal Image Compression takes large encoding time. So different fast Fractal compression techniques are implemented [3]. A hybrid image compression scheme in which DCT and fractal image compression technique can be used. Low Complexity Scalable DCT Image Compression Method[4] which gives good performance at low complexity which is same as non scalable baseline JPEG. Also no additional quantization or entropy coding is required. An improved partitioned Iterated Function System[5] [6]is introduced which converts RGB color space into YUV color space where input is color image and chromatic bands are down sampled by 2. Progressive structure is proposed [7] and it takes quad tree depth into consideration for fractal image coding. They also proved with simulation results that image quality at different data rates is better than without considering quad tree level. For further improvement, Fractal with DCT and Huffman coding is proposed.

3. DCT BASED FRACTAL IMAGE COMPRESSION ALGORITHM

To improve the fractal encoding speed, this 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 and 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 guarantee the quality of the decoded image. 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 [2].

3.1 Image Partitioning

Let, I be a gray-scale image to be encoded. The partition I into a set of $B\times B$ pixels range blocks, which are non-overlapping and overlay the whole image. The image I is also partitioned into a set of $2B\times 2B$ pixels domain blocks, which can be overlapping and need not overlay the whole image. The D block partition process can be done by sliding a $2B\times 2B$ window from left to right, top to bottom with horizontal step δ_h and vertical step δ_V. Here $\delta_h = \delta_V$=B.

3.2 Best Match Exploiting

After *R* blocks and *D* blocks are constructed, the next step is exploiting best match for each *R* block. First, each *D* block is compressed in spatial domain to reduced block *D*', which has the same size to *R* block. The compression method is reducing four adjacent pixels to one pixel, whose gray-scale is the average of the four pixels. The matching process is selecting a block *Ri*, then finding a *Dj* block with the same class as the block *Ri*, 8 isometric transformations is done for each *Dj* block. The best matching *D* block for *R* block is determined by evaluating the MSE (Mean Square Error) between *Ri* and each *Dj* with 8 isometric transformations. The minimum MSE means the best match. The MSE is determined. 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.

4. PROPOSED SYSTEM

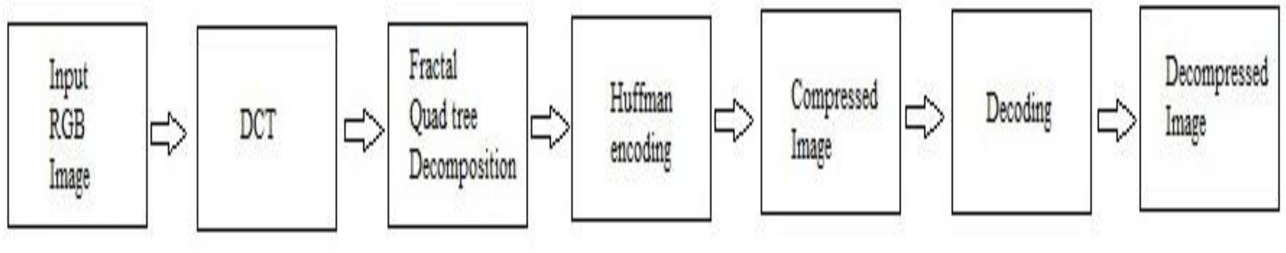

Fig 2: Proposed System Flow

The system flow is shown in fig. 2 with different stages.

The input image is the entry point of the Image compression System. Here an image is inputted in the system which should be a standard image of size 256x256, 512x512, 1024x1024 and it should be colored image.

4.1 Discrete Cosine Transformation

To find DCT, each image is divided into 8x8 pixel grids and each image's pixel grid is encoded with its own special DCT matrix. For image compression, we need to combine the 8x8 pixel grids which have their own DCT's and frequencies. Each one is weighted based on some value which is a number which represents the contribution of each of these 8x8 pixel grids to the final image. The higher the contribution of any pixel grid, the higher the value. The actual contribution of the pixel grid is then measured by multiplying a coefficient value to the pixel grid. DCT coefficients are rounded off by the quantizer. Then, lossy nature of the image is reproduced and large compression ratios are allowed. The compression technique uses a variable length code on these coefficients and then writes the compressed data stream to an output file (*.jpg). The algorithm takes the inverse transforms and displays the image after recovering the quantized DCT coefficients from the compressed data stream for decompression.

4.2 Fractal Quad tree Decomposition

The Fractal quad tree is basically recursive algorithm. It is based on entropy of an image, spatial redundancy and threshold. Fractal basically divides the image into different fractal and calculates the entropy if the fractals matched with each other then only they get counted at the time of encoding else the remaining fractal which didn't found match get discarded that's why fractal is lossy image compression technique, hence to overcome from this problem we have merged fractal with quad tree decomposition. In quad tree decomposition, each image get divided into 4 quadrants which are known as domain. Then, further domain gets divided into 4 quadrants which are known as ranges and each range get compared with domains to find its match. we find match with the help of entropy. Each range block finds its match if it failed to find its match, then that range block gets divided into 4 quadrants until it finds match. This procedure gets repeated until all range block gets its match.

Spatial Redundancy: Elements which are duplicated within the image like pixels, fractal quad tree elements are considered.

Threshold: The main aim of threshold is to replace the pixels with black which are having low intensity and with white which are having high intensity.

4.2.1 Advantages of fractal Quad tree Decomposition

a) Reduces spatial Redundancy

b) Requires less storage space as it eliminates redundant information

c) Fast decompression time

d) Advance detail interpolation

e) High compression ratio

f) Provides multi resolution property

g) Image can be decoded at higher or lower resolution than the original image

h) Reduces time for encoding

4.3 Huffman Coding

Huffman coding is an efficient entropy coding for lossless image compression. Huffman code is a an optimal prefix code used for lossless data compression. David A. Huffman developed an algorithm to find such a code This algorithm is a form of statistical coding and works well for text and fax transmission. It is a lossless data compression algorithm. The basic idea behind this algorithm is to assign variable length codes to input characters. The length of code assigned is dependent on corresponding frequencies of characters. The charater which occur most frequently has assigned smaller code. The character which occurs least frequently is assigned larger code. The input character gets some variable length code and this code is called as prefix code. There are two main parts in huffman coding one is to take the input character and build a huffman tree from that and second is to assign codes to those characters by traversing the huffman tree.

5. WORKING OF PROPOSED SYSTEM

The proposed system combines Discrete Cosine transform with Quad tree fractal with Huffman coding.

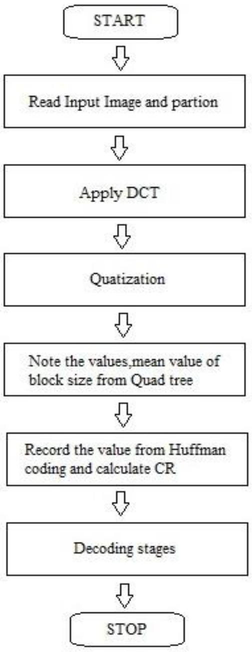
START
Read Input Image and partion
Apply DCT
Quatization
Note the values,mean value of block size from Quad tree
Record the value from Huffman coding and calculate CR
Decoding stages
STOP

Fig 3: Flow Chart of the Proposed Method

The algorithm consists of following steps

a) Read input image.

b) Divide the given image into blocks of 8X8.

c) Apply DCT to individual blocks of 8X8

d) Apply Fractal Quad tree decomposition to DCT transformed image which is divided using quad tree decomposition where the threshold value used is 0.2. 2 is the minimum dimension used and 64 is the maximum dimension used. Note down the x and y co-ordinate’s value, the mean value and the block size from the quad tree decomposition.Record the fractal coding information for completing the encoding.

e) Apply Huffman coding. At input we take an array of unique character along with their occurrences of frequency and the output of this huffman tree.For each unique character create a leaf node and for all leaf nodes build a min heap. min heap is basically a priority queue. The frequency field 's value is used to compare 2 nodes in minimum heap. At first, the least frequent character is at the root. From the minimum heap extract 2 nodes with minimum frequency. A new internal node is to be created with equal frequency to the sum of the 2 nodes frequencies. First extracted node should be at the left child and other extracted one should be at its right child. This node should be added to minimum heaps. The above steps need to be repeated until heap has only one node. The one which is left is the root node. Once tree is complete, it is traversed from the root node and maintains an auxiliary array. When moved to the left child, assign 0 to the array and while moved to the right child, assign 1. When a leaf is encountered, the array will be printed.

f) Calculate compression ratio. It is defined as ratio between size of original image and the compressed image. Compression ratio determines efficiency of an algorithm. Compression ratio= size of original image/size of compressed image

g) Huffman Decoding.

h) Decompressed image.

i) Calculate PSNR which gives quality measurement ratio between original and compressed image. Value of PSNR should be high.

6. HARDWARE AND SOFTWARE REQUIREMENTS

- Processor- Pentium 4, RAM- 512 MB or more, Hard Disk- 16 GB
- Matlab R2016b, O S – Windows /XP or higher/Linux/Mac

7. RESULTS

Images of different sizes are used from standard test images. Here, the result is displayed for the proposed system for size of 256X256 image of girl.tiff . The Girl images are displayed after each stage of implementation as original RGB image, Y-component, DCT compressed, Quad tree decomposition and decompressed image. The different parameters as encoding time, compression ratio, decoding time and PSNR are also calculated for different images. The 256X256 size test images as Girl.tiff, House.tiff, Jelly bean.tiff, Tree.tiff are used as input to proposed system. These images are from SIPI image database.

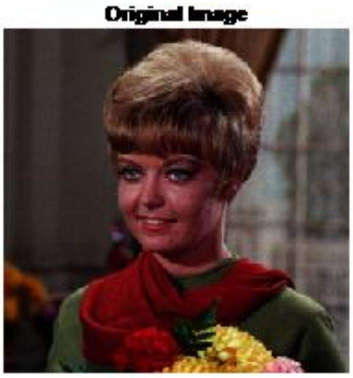

Fig 4: Original image of Girl.tiff

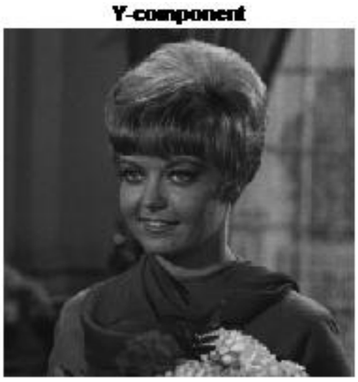

Fig 5: Y- component of Girl image

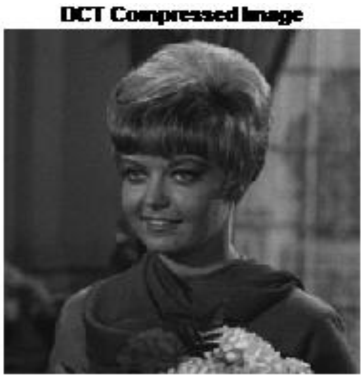

Fig 6: DCT compressed Girl image

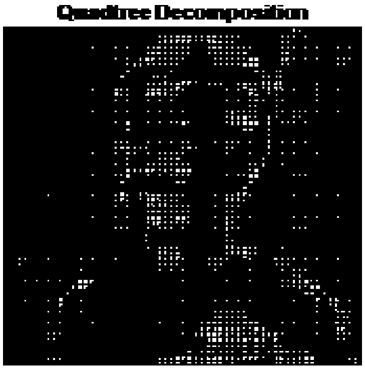

Fig 7: Quad Tree Decomposition of Girl Image

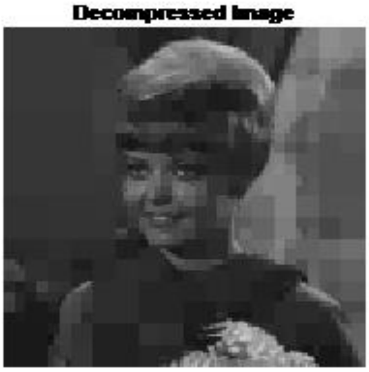

Fig 8: Decompressed Girl image

The results are given in Table 1. for the proposed system and in Table 2 for Fractal Quad tree with Huffman coding. After comparison, we have seen that compression ratio for the proposed system is less as compared with the Fractal Quad tree with Huffman method. Also PSNR value is less and encoding time required is more for the proposed method as compared with Fractal Quad tree with Huffman coding.

Table1: The performance parameters for proposed system for different color images

Color Images	Encoding time(s)	CR	PSNR	Decoding time(s)
Girl.tiff	3.11	8.26	10.74	48.72
House.tiff	3.50	6.01	5.24	67
Jelly bean.tiff	2.53	9.58	3.56	32.5
Tree.tiff	4.58	2.78	5.31	146.3

Table 2: The performance parameters for Quad tree fractal method with Huffman coding.

Color Images	Encoding time(s)	CR	PSNR	Decoding time(s)
Girl.tiff	0.99	20.14	28.19	7.26
House.tiff	1.21	14.97	27.66	11.19
Jelly bean.tiff	0.94	22.88	29.88	6.45
Tree.tiff	1.38	8.57	24.50	19.69

The proposed system has less compression ratio as compared to the Quad tree fractal with Huffman coding and reconstructed image is of less quality.

8. CONCLUSION

In this paper, different algorithms like discrete cosine transformation (which is lossy) with fractal image compression(lossy) using quad tree decomposition with Huffman encoding (lossless) are murged for achieving better image compression. The fractal Quad tree with Huffman coding gives higher compression ratio and better quality reconstructed images. Here, DCT and fractal quad tree with Huffman is successfully implemented and compared with Quad tree with Huffman coding. The results show that the proposed system has achieved less compression and requires more decoding time. In future, we can extend this project for video compression. 3D DCT can also be applied to 2-D images to improve the compression ratio. Instead of DCT, wavelet approach can also be applied to get higher compression ratio and better quality of reconstructed image.

9. REFERENCES

[1] M. Barnsley, “Fractals Everywhere”, New York, Academic 1988.

[2] A. E. Jaquin, “Image Coding based on a fractal theory of iterated contractive image transformation”, IEEE Transaction on Image Processing, 1992.

[3] X. Y. Wang, D. D. Zhang, “Discrete Wavelet Transform based simple range classification strategies for fractal image coding”, Non Linear Dy. 75, 3, pp.439-448, 2014.

[4] Rene J. Richard P, Christian H, “ Low complexity Scalable DCT Image Compression, IEEE, 2000.

[5] Dr. Loay E George, Nevart A. Minas, “ Speeding up Fractal Image Compression using DCT Descriptors”, Journal of Information and Computing Science, vol.6, No. 4, 2011, pp 287-294.

[6] Dr. Eman A, Dr. Loay E. George, “Study of Fractal Color Image Compression using YUV Component”, IEEE 36th International Conference on Computer Software and Applications, 2012.

[7] Ching Hung Yuen, Oi Yan Lui, Kwok Wo Kong, “Hybrid fractal image coding with quad tree based progressive structure”, J. Vis. Communication and Image Representation, 24(2013), 1328-1341.

[8] Shiwangi, Sanjay Kumar, Analysis of Image Compression Algorithm Using DCT, DFT and DWT Transforms. International Journal of Advanced Research in Computer Science and Software Engineering Vol. 6, Issue 7, July 2016.

[9] Prabhakar. Telagarapu, V.Jagan Naveen,"Image Compression Using DCT and Wavelet Transformations "International Journal of Signal Processing, Image Processing and Pattern Recognition ,Vol. 4, No. 3, September2011.

[10] A. A. Shaikh, P. P. Gadekar, Huffman Coding Technique for Image Compression."International Journal of Advanced Computer Technology" Vol.4, Issue 4 , April-2015.

[11] R.Praisline Jasmi, B.Perumal, M.Pallikonda Rajasekaran, 'Comparison Of Image Compression Techniques Using Huffman Coding, Dwt And Fractal Algorithm' International Conference on Computer Communication and Informatics,Coimbatore, India, Jan. 08 – 10, 2015.

[12] Padmavati.S, Dr. Vaibhar MesharamDCT Combined With Fractal Quad tree Decomposition and Huffman Coding for Image Compression' International Conference on Condition Assessment Techniques in Electrical Systems.

[13] Rachit Patel, Virendra K, Vaibhav T, Vishal A,'A fast and Improved Image Compression technique using Huffman coding', IEEE, International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET),Pages: 2283 – 2286, 2016.

[14] Surekha R Gondkar, Girish V Attimarad, B. Chandrasekhar,'Compression of 2D image using 3D DCT'

[15] Kulkarni S, Naik A, Nagori N, A comparison of real valued transforms for image compression, Int. Jour. Eng. Sci 4(1), 17, 2008.

[16] Annadurai S, Sundaresan M, Wavelet based color image compression, Proceedings of the International Conference on Advances in Computing, Communication and Control, USA, pp. 391-396, 2009.