A Hybrid Image Compression Scheme using DCT and Fractal Image Compression

Chandan Singh Rawat[1] and Sukadev Meher[2]
[1]Department of Electronics and Communication Engineering, National Institute of Technology, India
[2]Department of Electronics and Computer Engineering, National Institute of Technology, India

Abstract: *Digital images are often used in several domains. Large amount of data is necessary to represent the digital images so the transmission and storage of such images are time-consuming and infeasible. Hence the information in the images is compressed by extracting only the visible elements. Normally the image compression technique can reduce the storage and transmission costs. During image compression, the size of a graphics file is reduced in bytes without disturbing the quality of the image beyond an acceptable level. Several methods such as Discrete Cosine Transform (DCT), DWT, etc. are used for compressing the images. But, these methods contain some blocking artifacts. In order to overcome this difficulty and to compress the image efficiently, a combination of DCT and fractal image compression techniques is proposed. DCT is employed to compress the color image while the fractal image compression is employed to evade the repetitive compressions of analogous blocks. Analogous blocks are found by using the Euclidean distance measure. Here, the given image is encoded by means of Huffman encoding technique. The implementation result shows the effectiveness of the proposed scheme in compressing the color image. Also, a comparative analysis is performed to prove that our system is competent to compress the images in terms of Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM) and Universal Image Quality Index (UIQI) measurements.*

Keywords: *Image compression, DCT, fractal image compression, quantization, zig-zag scanning, huffman coding.*

Received May 23, 2011; accepted December 29, 2011; published online August 5, 2012

1. Introduction

Digital image processing is a quickly developing field with widespread use in the domain of mobile technology. An increasing number of products [18] like cellular phones, laptop computers and cameras used in surveillance [14] transmit and receive videos and images by means of portable wireless devices. The demand for efficient techniques that can store and transmit visual information has been increased by the increasing use of color images in the continuous growth of multimedia application [5]. Because of this demand, image compression has become a crucial factor and the requirement for efficient algorithms that can yield large compression ratio with low loss has increased [17]. Consequently, downloading image files from the internet can be an extremely time consuming task. In multimedia communication, a major portion of the communication bandwidth is occupied by image data because they are the major portion of the multimedia data [19]. Hence, formation of efficient techniques for image compression has become reasonably important [4]. Image compression is a method that decreases the quantity of digital information required to store visual images electronically.

Image compression reduces the size in bytes of a graphics file without deteriorating the quality of an image beyond tolerable limits. It also, decreases the time taken to send images via the Internet or downloaded from web pages. Image compression is performed by decreasing repetition between adjacent pixels and maintaining features like edges and contours of the original image, image compression [8]. Transformation, quantization and encoding are the three basic steps employed in each one of the several still image compressing methods [14]. Lossless and lossy image compression, are the two types into which the diverse image compression techniques available today can be classified. Lossless compression techniques allow exact reconstruction of the original, but the achievable compression ratios are only of the order approximately 2:1 [22]. A widely used form of lossy image compression process is the JPEG. The extensively used JPEG process based on Discrete Cosine Transform (DCT) is a form of lossy image compression [24]. Operates by splitting images into differing frequency parts.

Since, part of compression really happens during a step called quantization, where the less important frequencies are removed, it is called “lossy” [7]. A series of finitely several data points are expressed by a kind of Image Transform called DCT in terms of a sum of cosine functions oscillating at diverse frequencies [20]. Disintegrating the images into segments is the fundamental operating principle of DCT [12]. A better signal approximation with fewer transform coefficients are provided by DCT which is a real value unlike Discrete Fourier Transform [2]. In several practical image compression systems the invertible linear transform called 2D DCT is extensively used because of its compression performance and computational efficiency [31]. Data (image pixels) is converted into sets of frequencies, by DCT. The frequencies in the set are arranged in ascending order of significance. On the basis of tolerable resolution loss, the least meaningful frequencies can be discarded [26].

Quantizing the image's DCT coefficients and entropy coding the quantized coefficients are the two techniques on which DCT-based image compression depends in order to decrease the data necessary for representing the image [1]. In addition the functioning of these algorithms is estimated for diverse sizes of transform block, number of coefficients transmitted from each block and number of bits used to signify each pixel coefficient [10]. The image obtained after converting the representation of the colors in the original image from RGB to YCbCr by means of standard JPEG encoding is divided into 8×8 blocks and the blocks are transformed from the spatial to the frequency domain by means of the DCT transform. Later, all the DCT coefficients are divided by their matching constant in a standard quantization table and rounded down to the nearest integer.

From the lowest (upper left corner) to the highest (lower right corner) frequencies 64 DCT coefficients are constructed in each block [29]. All DCT pixels are decoded by using a constant number of bits. But, the importance (the ratio between an upper left corner pixel and the one in the right bottom corner) is not the same for all the pixels in a DCT [8, 11]. In DCT, 8×8 or 16×16 or bigger blocks are formed by segmenting the images. The problem with this block results as these blocks become visible when the image is reduced to higher compression ratios [3]. This is known as the blocking effect [9]. Computation time is another problem with the DCT. When used on grey scale images, a better performance is exhibited by the Sub Band-DCT (SB-DCT) [13, 27]. DCT (relative to the DFT) has the benefits of real-valued; better energy compaction (requires only a few coefficients to signify much of the signal energy), experimentally proven good performance and almost uncorrelated coefficients [25].

So, Huffman coding when combined with technique of reducing the image redundancies using DCT helps in compressing the image data to a very good extent. In our proposed method, the given color image can be compressed using DCT and to avoid compression on similar blocks of the image using Fractal image compression. The self similarity of the image blocks can be calculated using euclidean distance. Then effectively encode the image using Huffman encoding. The rest of the paper is organized as follows: Section 2 describes some of the recent related works. Section 3 briefs about the fractal image compression process. The proposed methodology is described in section 4 with needed formulas and diagrams. Experimental results and analysis of the proposed methodology are discussed in section 5. Finally, concluding remarks are provided in section 6.

2. Related Works

Numerous researches have been proposed by researchers for the color image compression process. In this section, a brief review of some important contributions from the existing literature is presented.

Chung *et al.* [8], has presented a spatial as well as DCT based hybrid gray image representation approach. In the first phase, the decomposed bin tree of the input gray image has been represented using an S-tree Spatial Data Structure (SDS), according to the bin tree decomposition principle under the specified error. Homogeneous leaves and the non-homogeneous leaves are the two types into which the constructed S-tree SDS leaves have been partitioned. One rectangular or square homogeneous sub image with smooth or in other words low frequency content has been represented using the homogenous leaf whereas, one non-homogeneous sub image with non-smooth or in other words high frequency content has been represented using a non-homogeneous leaf. The memory requirement has been reduced in the second phase by encoding each non-homogeneous leaf by the DCT-based coding scheme.

Thota and Devireddy [31], have tried to implement the basic JPEG compression by utilizing only fundamental Matlab functions. In those areas where data loss cannot influence image clarity, the lossy compression technique has been used. For image compression they have used JPEG which is a Discrete Cosine Transform based still frame compression standard sufficient for most compression application.

Pandian and Anitha [23], have presented a transform domain based technique for color image compression. Vector Quantization (VQ) technique has been used for compression of images and Kohonen's Self Organizing Feature Maps (SOFM) have been used in the design of the codebook in VQ. Special features of SOFM for generic codebook generation that permit to create the codebook only once have been exploited by their work. Image quality measures like PSNR, structural content, image fidelity and mean structural similarity index of the images on HSV color space have been used to observe the performance of decompression.

Krikor *et al.* [17], have presented a technique for image encryption has considered certain chosen higher frequencies of DCT coefficients as the characteristic values, encrypt them according to a pseudorandom bit sequence and shuffles the resulting encrypted blocks. The computational requirements of huge volumes of images have been decreased by the recent selective encryption approach.

Khalil [15], have described and implemented a RUN-Length coder that has been made simple and more effective. Their proposed algorithm has worked on quantized coefficients of the DCT where several concurrent tokens existed. The new approach has been proved to attain competitive performance by experimental results.

Kharate and Patil [16], have proposed that the compression ratio as well as the quality has been considerably improved by appropriate selection of the mother based on the nature of images. The technique they have proposed has been based on Threshold Entropy with enhanced run-length encoding based wavelet packet best tree. As complete tree has not been decomposed, their method has minimized the time complexity of wavelet packets decomposition. Sub-bands that include significant information based on threshold entropy have been chosen by their algorithm.

Meng and Zong [21], have used DCT, VQ coding and a new proposed method that combines DCT and wavelet transform in the implementation of their proposed color image compression algorithm. Their new proposed algorithm has been proved to achieve high compression ratio and efficient by experiment results.

Vijaya-Prakash and Gurumurthy [32], have proposed a technique to enhance the data compression technique. A new DCT and Quantization (DCTQ) architecture have been designed in their work for performing image compression. Compression of image data could be achieved by employing the DCT which is a kind of image transform. Later, compression has been achieved by performing quantization of the DCT data coefficients.

Singh and Kumar [28], have discussed the pros and cons of diverse transform-based image compression models in their detailed literature survey which they have presented. There are so many recent researches are presented here. From the literature survey, we observed that image compression based on DCT, will produce blocking artifacts and when we use pure fractal image compression scheme, the encoding time and complexity is too high. So, in order to overcome these issues, we proposed an efficient Hybrid image compression scheme based on DCT and Fractals. This will reduce the blocking artifacts as well as the encoding time and complexity of the compression scheme.

3. Basic Concept of Fractal Image Compression

The mathematical process called Fractal encoding is utilized to encode a given image into a set of mathematical data that illustrates the fractal properties of the image. Fractal encoding is based on the fact that all objects consists of information in the form of related, repeating patterns called an attractor. An image is converted into fractal code mostly by fractal encoding [34]. The huge number of iterations required to determine the fractal patterns in an image makes the encoding process to have extreme computation. Either Iterated Function Systems (IFS) or by Partitioned Iterated Function Systems (PIFS) are used to achieve Fractal Image Compression (FIC). Subsequently each IFS is coded as a contractive transformation [30] with coefficients. The collage theorem, which yields a bound on the distance between the image to be encoded and the fixed point of IFS is used by the encoding process as the basis for finding an IFS whose fixed point is near the given image [6].

The term affine transformation refers to such a transformation. Therefore, an appropriate transformation which ensures that the fixed point of that transformation is adjacent to the original image may be created [33]. This transformation consists of the association of several affine mappings on the whole image [12].

4. A Hybrid Technique for Image Compression using DCT and Fractal Image Compression Method

Compressing the image effectively using DCT and fractal image compression is the primary intension of our research. Initially the image will be segmented into 8×8 non-overlapping blocks. DCT will be employed to every block of the image. Then the DCT coefficients of each block will be quantized. The zero coefficients will be then prevented by scanning the block values in a zig-zag manner, zig-zag scanning improves the compression efficiency in addition, to extracting the nonzero coefficients. Here, the process of fractal image compression will be used. Then the image will be encoded using Huffman encoding. The entire process of the proposed method is described in the following.

Let I be a color image of dimension $M\times N$. Generally, the image will be in the RGB color space. Here compression can be performed both in the RGB and YcBcR color spaces. So, for YcBcR color image, the following equations are used to convert the image from RGB to YcBcR.

$$Y = (Y_{ran} / RGB_{ran})(0.299R + 0.587G + 0.114B) + 64 \quad (1)$$

$$Cb = (Cb_{ran} / RGB_{ran})(-0.1687R - 0.3313G + 0.5B) + 512 \quad (2)$$

$$Cr = (Cr_{ran} / RGB_{ran})(0.5R - 0.4187G - 0.0813B) + 512 \quad (3)$$

Since the Y_{ran} output range is 876 (940-64), the *Cbran/Crran* output range is 896 (960-64) and the RGB_{ran} input Range is 1023 (1023-0). Divide the image *I in* to number of $N\times N$ non overlapping blocks. This can be represented as:

$$I = \{Ib_1, Ib_2, \ldots, Ib_{Nb}\} \quad (4)$$

where *Nb* represents the total number of blocks in the image.

4.1. Color Image Compression using DCT

Image processing extensively use DCT, particularly for compression. While still image compression and compression of individual video frames are performed by some applications of two-dimensional DCT, compression of video streams is the most common application of multidimensional DCT. We use DCT to compress the image in our proposed method. This is performed by employing DCT to each non overlapping block of the image as described in the following equation.

$$D(i,j)=\frac{1}{\sqrt{2N}}C(i)C(j)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}I(x,y)\cos\left[\frac{2(x+1)i\pi}{2N}\right]\cos\left[\frac{(2y+1)i\pi}{2N}\right] \quad (5)$$

$$C(u)=\begin{cases}\frac{1}{\sqrt{2}} & \text{if } u=0\\ 1 & \text{if } u>0\end{cases} \quad (6)$$

here $I(x,y)$ represents the $(x,y)^{th}$ element of the image represented by I. N is the size of the block that the DCT is done on. The equation calculates one entry $(i,j)^{th}$ of the transformed image from the pixel values of the original image matrix. If the coefficients are specified to an infinite precision, then the DCT presented in above equations is ortho-normal and perfectly reconstructing. Generally the coefficients of a DCT are linearly quantized by dividing by a predetermined quantization step.

4.2. Quantization

Based on the two techniques, quantizing the image's DCT coefficients and entropy coding the quantized coefficients, DCT-based image compression minimizes the data required to represent an image. Quantization process minimizes the number of bits required to represent a quantity by minimizing the number of possible values of the quantity. A range of values are compressed to a single quantum value to achieve quantization. The stream becomes more compressible as the number of discrete symbols in a specified stream is reduced. Transformation is performed by using a quantization matrix in combination with a DCT coefficient matrix.

According to the quantization matrix, the DCT coefficients are normalized by different scales, for high compression [6]. The transformed image matrix is divided by the employed quantization matrix to achieve quantization. Then the values of the resultant matrix are rounded off. The coefficients located near the upper left corner in the resultant matrix have lower frequencies. Human eye is more sensitive to lower frequencies. So, higher frequencies are eliminated and the image is reconstructed by using the lower frequencies. The quantization process can be described in the following equation:

$$Q_{DCT}=round\left(\frac{D(i,j)}{Z(i,j)}\right) \quad (7)$$

here $D(i,j)$ are the *DCT* coefficients of the transformed image and $Z(i,j)$ is the quantization table which is shown in the following Figure1.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Figure 1. Quantization table.

4.3. Zig-zag Scanning

The entire quantized coefficients are rearranged in a zigzag manner, after the DCT coefficients are quantized. Most of the high frequency coefficients (lower right corner) become zeros after quantization. A zig-zag scan of the matrix yielding long strings of zeros is used to exploit the number of zeros. The current coder acts as filter and passes only the string of non-zero coefficients. A list of non-zero tokens for each block proceeded by their count will be obtained at the end of this process, following Figure 2 illustrates this.

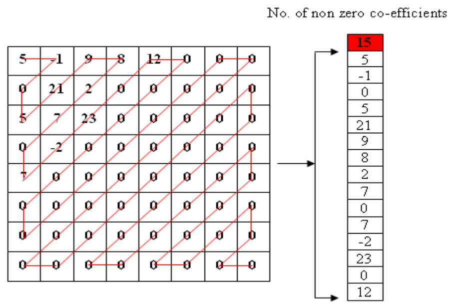


Figure 2. Zig-zag scanning.

4.4. Fractal Image Compression

The property of self-similarity of fractal objects is used by fractal compression and fractal encoding. Some of the blocks obtained by dividing the color image into several 8×8 blocks are similar. So, the concept of fractal image compression is used to prevent performing repetitive compression on the same block. Fractal image compression must be used before encoding the quantized image blocks. Similar blocks in a given input image are identified using fractal image compression i.e., the matched domain blocks for each range block in an image. The Euclidean distance measure is used to calculate the similarity between the images.

Our proposed method, calculates the similarity of the a^{th} block for the fractal image compression by comparing the distance measure of the a^{th} block and its n neighboring blocks. The following equation is used to calculate the distance measure.

$$S_d = \sqrt{\sum (Ib_a - Ib_b)}^2 \tag{8}$$

here Ib_a and Ib_b; where $b=\{1,2,...,n\}$ represent the current block and blocks adjacent to the current block, respectively. This is shown in the following Figure 3.

b1	b2	b2
b4	a	b5
b6	b7	b8

Figure 3. Range blocks and domain blocks.

The flag value is set according to the threshold D_{tsh}, after the distance measure is calculated. The calculated distance S_d is compared with the threshold D_{tsh} as follows:

$$I_F^b = \begin{cases} I_F^b = 1; & if\ S_d < D_{tsh} \\ I_F^b = 0; & otherwise \end{cases} \tag{9}$$

here b and F represent the image block and the flag value of each block of the image, respectively. Both a^{th} and b^{th} blocks are said to be similar if the b^{th} block yields a flag value of 1, when it is compared with the a^{th} block. Otherwise they are said to be dissimilar. This is illustrated in Figure 4.

1	1	0
1	a	0
1	0	0

Figure 4. Flags assigned to each domain blocks.

So, we store the indices by identifying the blocks similar to the a^{th} block. In fractal image compression, a^{th} block is the range block and the analogous similar blocks are domain blocks. Instead of all the similar domain blocks, we use only the range block once the indices of range block and its corresponding domain blocks are accumulated. The time and memory complexity is decreased by this.

4.5. Entropy Encoder

The quantized values are further compressed by a lossless entropy encoder to give a better overall compression. For each quantized value, an output code stream that is smaller than the input stream is produced by it using a model to accurately determine the probabilities. Huffman Encoder and the Arithmetic Encoder are the most commonly used entropy encoders, although simple Run-Length Encoding (RLE) has been proven to be very effective for applications requiring fast execution.

Entropy encoding is a lossless data compression technique. A distinct prefix code is created and assigned by one of the main types of entropy coding to each unique symbol that occurs in the input. Then, by replacing each fixed-length input symbol with the corresponding variable-length prefix codeword compression of data is achieved by these entropy encoders. The shortest codes are used for the most common symbols because the length of each codeword is more or less proportional to the negative logarithm of the probability. Here, Huffman coding is used to compress the image effectively. Huffman coding is a proficient source coding algorithm for source symbols that are not equally probable. In 1952, Huffman suggested a variable length encoding algorithm, based on the source symbol probabilities $p(x_i)$; where $i=1,2,...,L$. The algorithm is optimal if the prefix condition is met, because then the average number of bits needed to represent the source symbols is minimum. The steps present in the Huffman coding algorithm are as follows:

1. Organize the source symbols in increasing order of their probabilities.
2. Bind the bottom two signals together and write the sum of the probabilities of the two symbols on the combined node. Label the two branches with '1'and '0'.
3. Consider this sum of probabilities as a new probability corresponding to a new symbol. Again form a new probability by binding together the two smallest probabilities. The total number of symbols is reduced by one each time two symbols are combined. The two branches of the two low probabilities bound together are always labeled as a '0' and '1'.
4. Continue the procedure until only one probability remains (and it should be '1' if the additions performed are correct). This completes the creation of the Huffman Tree.
5. Follow the branches from the final node back to the symbol to identify the prefix codeword for any symbol. Read out the labels on the branches while the route is traced back. This gives the codeword for the symbol.

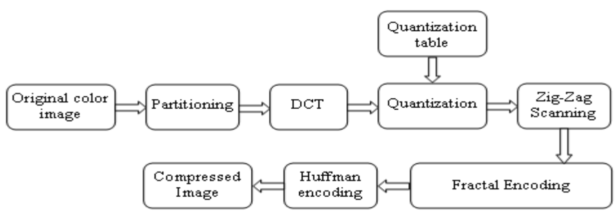


Figure 5. Block diagram of the compression process.

By coding the symbols one at a time, the Huffman's procedure generates the optimal code for a set of symbols and probabilities. Figure 5 illustrates the entire compression process.

4.6. Decompression Process

The decompression process is quite simple. We first decode the fractal parameters and repeatedly transform an arbitrary image by fractals. A DCT domain fractal approximation of the original image is produced by this process. Conversely, we get the difference image by decoding the Huffman code and de-quantizing it. Finally, we get the decompressed image by transforming the addition of the fractal approximation and the difference image by inverse 2D DCT. Figure 6 illustrates the entire decompression process.

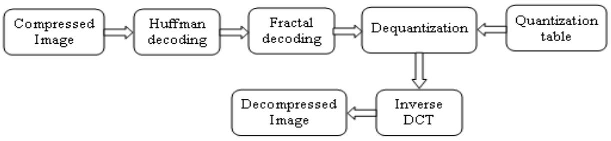
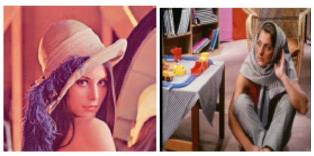


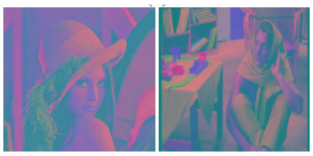
Figure 6. Block diagram of the decompression process.

5. Results and Discussions

In this section, we illustrate the effectiveness of the proposed hybrid coding scheme in image compression by means of the results obtained from the experimentation, proposed method was implemented in Matlab (Matlab 7.10) and the proposed hybrid coding scheme was evaluated using color images. The test images used in the experiments include: Lena, Barbara, Baboon and Peppers. Quality of the reconstructed images was determined by measuring the PSNR, SSIM and UIQI values and the compression efficiency of the proposed hybrid scheme was determined in terms of the compression ratio, sample output obtained from the proposed method as follows:



a) Original images.



b) Corresponding YcBcR images.

c) Decompressed images in YcBcR color space.

Figure 7. The sample output obtained from the color space conversion process.

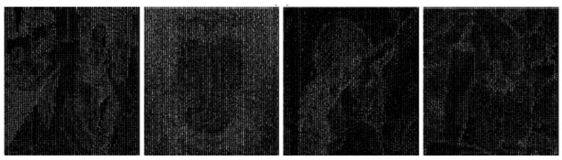
Our proposed method of image compression can be performed in both RGB and YcBcR color spaces, Figure 7 represents the images in RGB color space as well as in YcBcR color space, Figure 8 represents the block partitioning of an image.

a) Original lena and peppers image.

b) Corresponding 8×8 blocks of the original images.

Figure 8. The sample output obtained from the image partitioning process.

a) Original images.

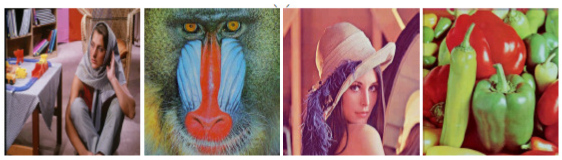


b) Corresponding quantized images.

Figure 9. The sample output obtained from the quantization process.

After applying DCT to each block in the image, quantize the DCT coefficients. Figure 9 shows the quantized images. Finally, the image is encoded by using the huffman encoding scheme. Figure 10 describes the original and the decompressed images.

a) Original images (barbara, baboon, lena, peppers).



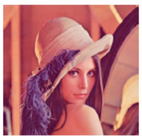
b) Decompressed images using proposed method.

Figure 10. The sample output obtained from the compression process.

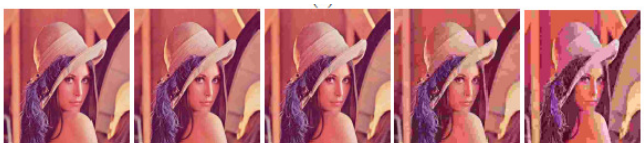
5.1. Comparative Analysis

Our proposed method is compared with the known image compression standard JPEG with particular image qualities. Here the comparison is done only with JPEG and other existing techiques based on DCT. We can't compare the proposed work with JPEG 2000.

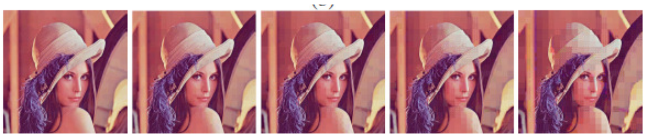
Because it is based on DWT, Figure 11 represents the decompressed images of both JPEG and proposed method with certain image qualities.



a) Original lena image.



b) Decompressed lena images using JPEG with image quality 14,12,10,5, respectively.



c) Decompressed images using proposed method with threshold 3, 5, 7, 9, 15 respectively.

Figure 11. Proposed vs. JPEG.

From the above figure, our proposed method gave better visuality than JPEG, there are some blocking artifacts in the decompressed image of the proposed method, but the blocking artifacts appear only when we increase the threshold value. At lower threshold value the image quality is good when compared to JPEG.

5.1.1. PSNR

The Peak Signal to Noise Ratio (PSNR) is the ratio between a signal's maximum power and the power of the signal's noise. To evaluate the performance of the motion estimation technique in the proposed system, the Peak PSNR based on the Mean Square Error (MSE) is used as a quality measure and its value can be determined using the following equation:

$$PSNR = 10log\left(\frac{(255)^2}{MSE}\right)dB \tag{10}$$

$$MSE = \frac{1}{MN}\sum\left(\hat{f}(x,y) - f(x,y)\right)^2 \tag{11}$$

here MN is the total number of pixels in the image. $\hat{f}(x,y)$ is the decompressed image and $f(x, y)$ is the original image.

5.1.2. UIQI

One of the many techniques to measure the image quality is called the Universal Image Quality Index (UIQI) [6]. The UIQI is designed by modeling any image distortion as a combination of three factors: loss of correlation, luminance distortion, and contrast distortion. The universal image quality index for each block can be calculated as:

$$Q = \frac{4\delta_{xy}\mu_x\mu_y}{(\delta_x{}^2 + \delta_y{}^2)[\mu_x{}^2 + \mu_y{}^2]} \tag{12}$$

here μ_x and μ_y are the mean value of the original and the decompressed images. Then δ_x and δ_y are the standard deviation of the original and decompressed images and δ_{xy} is the covariance. Then the local image quality results are averaged to determine the overall image quality index of the whole image:

$$Q = \frac{1}{P}\sum_{j=1}^{P} Q_j \tag{13}$$

5.1.3. SSIM

The Structural Similarity (SSIM) index is a method for measuring the similarity between two images. The SSIM index is a full reference metric, in other words, the measuring of image quality based on an initial uncompressed or distortion-free image as reference. The similarity measure that compares the local patterns of pixel intensities that have been normalized for luminance and contrast is known as SSIM [33].

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x{}^2 + \mu_y{}^2 + C_1)(\sigma_x{}^2 + \sigma_y{}^2 + C_2)} \tag{14}$$

here μ_x and μ_y are the mean value of the luminance in the original and decompressed image respectively. σ_x and σ_y are the standard deviation of the luminance. C_1 and C_2 are the contrast values of the original and decompressed images. Figures 12, 13 and 14 represent the performance analysis of proposed method with JPEG. The proposed method has higher PSNR, SSIM and UIQI when compared to JPEG. Here we compare the JPEG with quality 14, 12, 10, 5, 3 respectively.

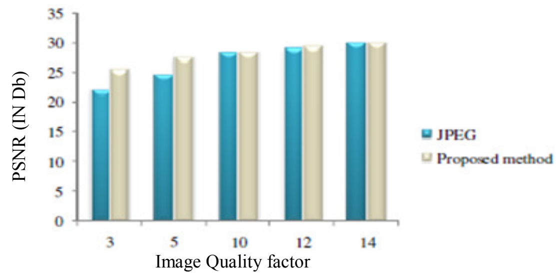


Figure 12. Performance analysis of proposed method and JPEG (PSNR) for lena image.

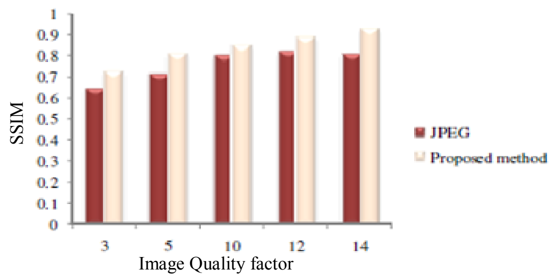


Figure 13. Performance analysis of proposed method and JPEG (SSIM) for lena image.

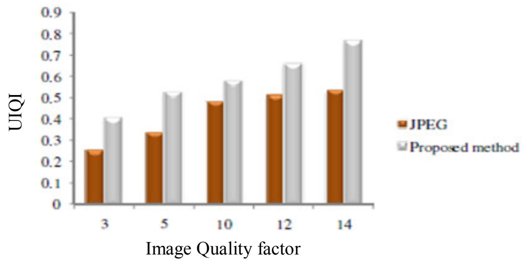


Figure 14. Performance analysis of proposed method and JPEG (UIQI) for lena image.

This comparative analysis can be represented in Table 1. The JPEG images with quality 14, 12, 10, 5, 3 are compared with our proposed method images with threshold value 3, 5, 7, 9, 15 respectively. From the table, we observe that our method has higher PSNR, SSIM and UIQI when compared to the JPEG with particular image quality factors.

Table 1. Comparison of proposed method with JPEG with various image qualities.

Image Quality Factor for JPEG	PSNR (in dB)		SSIM		UIQI	
	JPEG	Proposed Method	JPEG	Proposed Method	JPEG	Proposed Method
3	21.9690	25.5449	0.6444	0.7263	0.2559	0.4056
5	24.6225	27.5195	0.7112	0.8138	0.3409	0.5247
10	28.3131	28.3775	0.8014	0.8500	0.4828	0.5810
12	29.1461	29.4161	0.8204	0.8911	0.5148	0.6579
14	29.8870	30.0605	0.8068	0.9302	0.5350	0.7676

The quality measure for various images such as Lena, Peppers, Barbara and Baboon is represented in Figures 15 and 16.

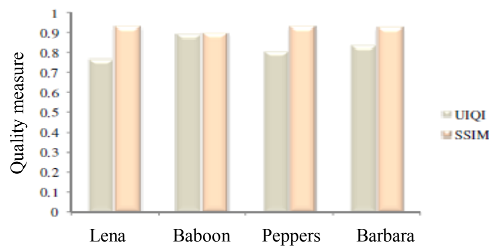


Figure 15. The quality measure for various images (SSIM and UIQI).

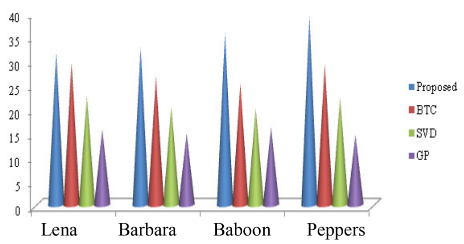


Figure 16. The quality measure for various images (PSNR).

The quality measure of images such as UIQI and SSIM for proposed method is described in Table 2.

Table 2. Quality measure for various images.

Images \ Quality Measure	UIQI	SSIM
Lena	0.7676	0.9302
Baboon	0.8895	0.8944
Peppers	0.8042	0.9301
Barbara	0.8372	0.9276

Following table represents the PSNR comparison of various images using the proposed method and other existing techniques. From Table 3 and Figure 16, we observed that the proposed method has effectively compress the images when comparing with other techniques.

Table 3. Quality measure for various images.

Methods	Images			
	Lena	Barbara	Baboon	Peppers
Proposed Method	31.5739	32.781	35.816	39.2185
Block Truncation Coding	29.6116	26.5894	25.1743	29.2346
Singular Value Decomposition	22.6225	20.4283	20.0996	22.3442
Gaussian Pyramid	15.6656	14.7322	16.1080	14.5351

The above table represents that the proposed method has higher PSNR values than other existing methods such as Block truncation Coding, Singular Value Decomposition and Gaussian Pyramid techniques. The Compression Ratio (CR) of the proposed method for various images is described in Table 4.

Table 4. Compression ratio for various images.

Images	Compression Ratio
Lena	11.1544
Barbara	7.1266
Baboon	4.5253
Peppers	10.8559

From the comparative analysis, we observe that the proposed method will effectively compress the image with high PSNR and image quality.

6. Conclusions

In our proposed technique, the color images were compressed effectively using DCT. Generally, the DCT based compression technique produce some blocking artifacts. Here, the artifacts were removed by utilizing the fractal image compression method. Also, the self similarities between the analogous blocks were found by using the euclidean distance measure. So, this eliminates the continual compression of analogous blocks. From the implementation results, we have proved that the proposed system was efficient in compressing the images. Also, when compared to JPEG with image quality 14,12,10,5,3 respectively, we have concluded that our proposed technique has successfully compressed the images with high PSNR value, SSIM index and the UIQI value.

References

[1] Alfalou A., Elbouz M., Jridi M., and Loussert M., "A New Simultaneous Compression and Encryption Method for Images Suitable to Recognize form by Optical Correlation," *in Proceedings of SPIE*, Berlin, vol. 7486, pp. 1-8, 2009.

[2] Al-Gindy A., Al-Ahmad H., Qahwaji R., and Tawfik A., "A Novel Blind Image Watermarking Technique for Colour RGB Images in the DCT Domain using Green Channel," *in Proceedings of Mosharaka International Conference on Communications, Computers and Applications*, Amman, pp. 26-31, 2008.

[3] Al-Gindy A., Al-Ahmad H., and Qahwaji R., Tawfik A., "Watermarking of Colour Images in the DCT Domain using Y Channel," *in Proceedings of International Conference on Computer Systems and Applications*, Rabat, pp. 1025-1028, 2009.

[4] Ames G., "Image Compression," *International Journal of Advancements in Computing Technology*, vol. 1, no. 2, pp. 205-220, 2002.

[5] Annadurai S. and Sundaresan M., "Wavelet Based Color Image Compression using Vector Quantization and Morphology," *in Proceedings of the International Conference on Advances in Computing, Communication and Control*, USA, pp. 391-396, 2009.

[6] Bovik Z. and Alan C., "A Universal Image Quality Index," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81-84, 2002.

[7] Cabeen K. and Gent P., "Image Compression and the Discrete Cosine Transform," *Mathematica Journal*, vol. 4, no. 1, pp. 81-88, 2010.

[8] Chung K., Liu Y., and Yan W., "A Hybrid Gray Image Representation using Spatial- and DCT-Based Approach with Application to Moment Computation," *Journal of Visual Communication and Image Representation Archive*, vol. 17, no. 6, pp. 1209-1226, 2006.

[9] Dorfler P. and Mitra S., "Compression of Color Images without Visible Blocking Artifacts using a Modified DCT Computation Scheme," *in Proceedings of International Conference on Image Processing and its Applications*, Manchester, vol. 1, pp. 169-173, 2002.

[10] Fisher Y., "Fractal Image Compression," *SIGGRAPH Course Notes*, vol. 43, no. 6, pp. 657- 662, 1992.

[11] Gershikov E., Lavi-Burlak E., and Porat M., "Correlation-Based Approach to Color Image Compression," *Journal in Image Communication*, vol. 22, no. 9, pp. 719-733, 2007.

[12] Gray R. and Neuhoff D., "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325-2383, 1998.

[13] Jacquin A., "A Novel Fractal Block-Coding Technique for Digital Images," *in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Albuquerque, vol. 4, pp. 2225-2228, 1990.

[14] Kesavan H., "Choosing a DCT Quantization Matrix for JPEG Encoding," available at: http://scien.stanford.edu/pages/labsite/1997/ee392c/demos/kesavan/, last visited 2010.

[15] Khalil M., "Image Compression using New Entropy Coder," *International Journal of Computer Theory and Engineering*, vol. 2, no. 1, pp. 39-42, 2010.

[16] Kharate G. and Patil V., "Color Image Compression Based on Wavelet Packet Best Tree," *International Journal of Computer Science Issues*, vol. 7, no. 2, pp. 31-35, 2010.

[17] Krikor L., Baba S., Arif T., and Shaaban Z., "Image Encryption using DCT and Stream Cipher," *European Journal of Scientific Research*, vol. 32, no. 1, pp. 48-58, 2009.

[18] Kulkarni S., Naik A., and Nagori N., "A Comparison of Real Valued Transforms for Image Compression," *International Journal of Engineering and Natural Sciences*, vol. 4, no. 1, pp. 17, 2008.

[19] Loussert A., Alfalou A., El-Sawda R., and Alkholidi A., "Enhanced System for Image's Compression and Encryption by Addition of Biometric Characteristics," *International Journal of Software Engineering and its Applications*, vol. 2, no. 2, pp. 111-118, 2008.

[20] Martucea S., "Symmetric Convolution and the Discrete Sine and Cosine Transform," *IEEE Transaction on Signal Processing*, vol. 42, no. 5, pp. 1038-1051, 1994.

[21] Meng M. and Zong M., "A New Zerotree Structure for Color Image Compression based on DWT and VQ," *in Proceedings of IEEE International Conference on Information Management and Engineering*, Chengdu, pp. 339-342, 2010.

[22] Palanisamy G. and Samukutti A., "Medical Image Compression using a Novel Embedded Set Partitioning Significant and Zero Block Coding," *International Arab Journal of Information Technology*, vol. 5, no. 2, pp. 132-139, 2008.

[23] Pandian S. and Anitha J., "A Neural Network Approach for Color Image Compression in Transform Domain," *International Journal of Recent Trends in Engineering*, vol. 2, no. 2, pp. 152-154, 2009.

[24] Radhakrishnan S. and Subramaniam J., "Novel Image Compression using Multi-Wavelets with SPECK Algorithm," *International Arab Journal of Information Technology*, vol. 5, no. 1, pp. 45-51, 2008.

[25] Raj N. and Venkateswarlu T., "A Novel Approach to Medical Image Compression using Sequential 3D DCT," *in Proceedings of International Conference on Computational Intelligence and Multimedia Applications*, Sivakasi, vol. 3, pp. 146-152, 2007.

[26] Salomon D., *Data Compression*, Springer Verlag, Berlin, 2002.

[27] Saxena V. and Gupta J., "Collusion Attack Resistant Watermarking Scheme for Colored Images using DCT," *IAENG International Journal of Computer Science*, vol. 34, no. 2, pp. 1-7, 2007.
[28] Singh S. and Kumar S., "Mathematical Transforms and Image Compression: A Review," *Maejo University International Journal of Science and Technology*, vol. 4, no. 2, pp. 235-249, 2010.
[29] Sousa C., Cavalcante A., Guilhon D., and Barros A., "Image Compression by Redundancy Reduction," *in Proceedings of the 7th International Conference on Independent Component Analysis and Signal Separation*, Berlin, vol. 4666, pp. 422-429, 2007.
[30] Thakur N. and Kakde O., "Color Image Compression with Modified Fractal Coding on Spiral Architecture," *Journal of Multimedia*, vol. 2, no. 4, pp. 55-66, 2007.
[31] Thota N. and Devireddy S., "Image Compression using DCT," *Geogian Electronic Scientific Journal*, vol. 3, no. 17, pp. 35-42, 2008.
[32] Vijaya-Prakash A. and Gurumurthy K., "A Novel VLSI Architecture for Digital Image Compression using Discrete Cosine Transform and Quantization," *International Journal of Computer Science and Network Security*, vol. 10 no. 9, pp. 175-182, 2010.
[33] Wang Z., Bovik A., Sheikh H., and Simoncelli E., "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 1-14, 2004.
[34] Wohlberg B. and Jager G., "A Review of the Fractal Image Coding Literature," *IEEE Transaction on Image Processing*, vol. 8, no. 12, pp. 1716-1729, 1999.

Chandan Singh Rawat received his BE degree from the Department of Electronics and communication Engineering Nagpur University, and his ME degree from Mumbai University. Currently, he is pursuing PhD at National Institute of Technology Rourkela, India. He is presently working as an assisted professor in Chembur. His area of interest is image processing.

Sukadev Meher received his B.Tech degree in information technology, in 1984, Ms of Technology M.Tech in 1995, and his PhD in 2005. Currently, he is working an associate professor in electronics and computer engineering, National Institute of Technology, India. His field of interest includes circuit and systems, optical communication, digital signal processing.