ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

Adoption of Global Structure Transformation in lossy image compression based on Curvelet and Cosine transforms

M. Mahasree[1], D.J. Ashpin Pabi[2], P. Aruna[3], N. Puviarasan[4]

PG Scholar, Dept. of Computer Science & Engineering, Annamalai University, Tamil Nadu, India [1]

Research Scholar, Dept. of Computer Science & Engineering, Annamalai University, Tamil Nadu, India [2]

Professor, Dept. of Computer Science & Engineering, Annamalai University, Tamil Nadu, India [3]

Associate Professor, Dept. of Computer and Information Science, Annamalai University, Tamil Nadu, India[4]

ABSTRACT: Recent studies on curvelets reveal the significance of applying curvelet transform for compressing images. The predominance of curvelets over other transforms is due to their sparse representation of edges. In this paper, an optimal approach for lossy image compression is proposed based on hybrid of curvelet and cosine transformations. Initial step is to obtain the curvelet coefficients using Fast Discrete Curvelet Transform (FDCT). This is followed by the application of Discrete Cosine Transform (DCT) for improving the quality as well as compression factors. Further, to enhance the quality of compression, Global Structure Transformation (GST) is adopted at the encoding stage. Application of GST in this curvelet-based image compression is a novel idea for improving compression ratio. Finally, arithmetic encoding is used to reduce coding redundancy by assigning shorter codeword to source symbols. To select an efficient GST, the performance of three methods, namely, move-to-front, distance coding and inversion frequencies are evaluated. The criteria used for this evaluation are compression ratio, encoding and decoding speeds. Experimental results show the effectiveness of proposed compression method at various quantization levels.

KEYWORDS: Fast Discrete Curvelet Transform; Discrete Cosine Transform; Global Structure Transformation; arithmetic encoding; codeword

I. Introduction

Compression is one of the active areas in the field of Digital Image Processing. Image Compression emphasizes the importance of squeezing digital images to reduce storage and transmission cost. Basically, it is done either in lossless or lossy manner. When the recovered image from the compressed file is same as the original image, then it is called lossless compression. In contradiction, lossy compression gives recovered image which is nearly same as the original image with minimal variation in its intensity values. Many techniques have been proposed so far in lossy and lossless methods [1]. In lossy compression, effective results are obtained when the image is handled in the frequency domain. So, transformations like Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) are utilized to obtain frequency coefficients.

DCT is widely adopted for its simplicity and energy compaction. But it suffers undesirable blocking artifacts at lower bit rates. Wavelets are preferred over DCT to overcome this limitation. DWT possess multi-resolution properties, i.e. it transforms an input image into sub-bands with different energy coefficients in each sub-band. Proper encoding of wavelet coefficients gives higher performance in terms of peak-signal-to-noise ratio and compression ratio. Unlike DCT, the input image need not be partitioned into blocks since DWT provides a multi-level decomposition. Despite these advantages, the main flaw in DWT is that it requires more coefficients to represent edge discontinuities along

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

curves. Curvelets is a better choice than wavelets because it uses fewer coefficients for edge discontinuities. Being an extension of wavelets, it provides all advantages of wavelets. It gives better compression with less distortion in images.

Global Structure Transformations (GST) was originally developed for the betterment of lossless compression techniques. The GST is the second stage in Burrows-Wheeler Compression Algorithm (BWCA) which is used for transferring local structure into a globally useful structure for compression. In this work, a novel idea of utilizing the GST method before entropy coding is proposed for improving compression ratio. This scheme helps in reducing the total number of coefficients required for reconstruction.

The rest of the paper is structured as follows. Section 2 gives brief study of lossy image compression methods based on curvelet transformation. In section 3 the detailed description of the proposed system is introduced. Section 4 presents the experimental results. The final section summarizes the proposed work and draws the conclusion.

II. Related work

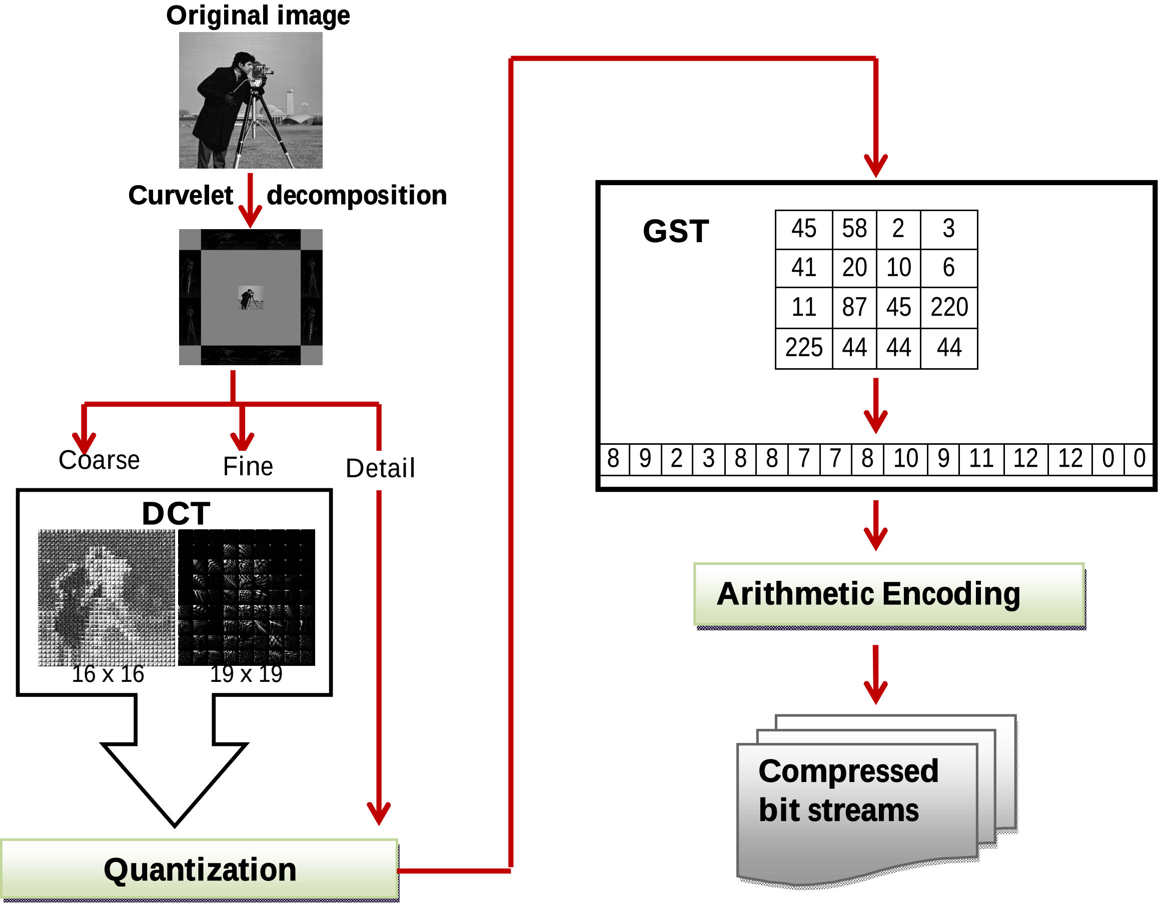
In this section, the literature related to the proposed scheme is presented. These researches give motivation for improving the curvelet based compression further.

Iqbal et al. [2] proposed an image compression technique based on curvelets and SPIHT (Set Partitioning in Hierarchical Trees) encoding method. They used thresholding and quantization on curvelet coefficients. These coefficients are sent into sorting pass followed by refinement pass of the SPIHT encoding scheme. A different encoding scheme based on embedded bit stream is proposed by Manikandan et al. [3]. The results are compared with wavelets to show the performance of their proposed methodology. Majumdar [4] utilized a sparse multivariate tool known as Sparse Principal Component Analysis (SPCA). The column dimensions of curvelet matrices are reduced using sparse PCA. This proposed technique is compared with JPEG2000, which is based on wavelets and the results are satisfactory with curvelets. Yang et al. [5] have used support vector machine (SVM) to improve compression performance. SVM regression is a learning based decision making system. The highest sub-band coefficients are discarded in this process. Singh and Murthy [6] have proposed the compression using Back-Propagation Neural Network and curvelet transform. To achieve compression, the hidden layer of BPNN is designed with less number of neurons than other two layers (input and output layers). The quantization procedure followed here is vector quantization. Experiments are done with well-known grayscale test images.

III. Proposed algorithm

In this paper, the proposed compression technique follows curvelet and DCT transformations, the coefficients are applied to GST stage which is followed by arithmetic encoding. Experiments are conducted on standard gray scale images of size 512×512. The implementation was done using MATLAB. Several images are collected from different databases. Dataset contains about 20 images from various categories. Fig.1 shows some of the sample images from dataset. To describe the working of proposed technique, cameraman image is used for illustration throughout this work. Fig.2 shows the block diagram of the proposed compression method.

Fig.1. Sample images from dataset



45	58	2	3
41	20	10	6
11	87	45	220
225	44	44	44

Fig.2. Block diagram of the encoder of the proposed compression method

1. *Curvelets:*

Lines and curves are the main building blocks of an image. To represent the regularity of curved edges, wavelets require more coefficients. And so, the curvelets are introduced to overcome the limitations associated with wavelets. Curvelet transforms are multi-scale representations of an image whose basis elements exhibit directional and anisotropic properties [7, 8]. The term anisotropic refers to structures with length greater than width. Due to this property, basis elements of curvelet appear like an elongated needle at finer scales as shown in Fig.3. This property is lagging in wavelets where basis elements are like square and requires more in counts to represent the curve.

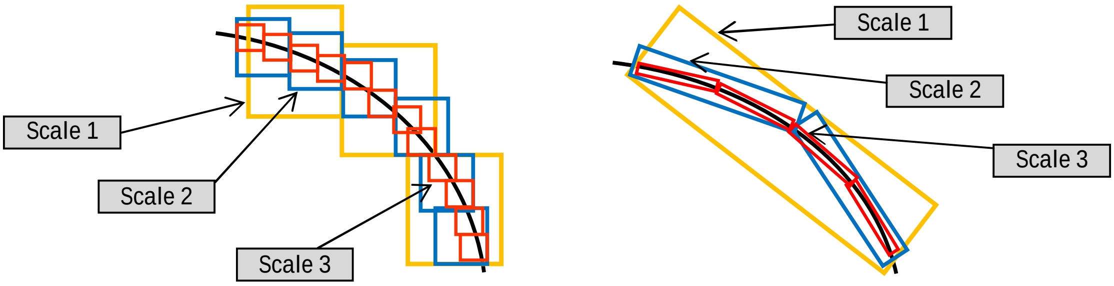


Fig.3. Edge representation in Wavelets (left) and Curvelets (right)

There are three major developments occurred in curvelets. The first generation of curvelets [9] provides only suboptimal results in compression field due to its complicated structure and high computational cost. To overcome the limitations of first generation curvelets, the second generation curvelet transform was presented with some modifications. New tight frames are constructed without the use of Ridgelet transform [10]. This leads to an optimal sparse representation along the edges of the objects. Further, there are two simpler digital versions of second generation curvelet transform are implemented as shown below:

1) FDCT via USFFT
2) FDCT via Frequency wrapping

The first Fast Discrete Curvelet Transform (FDCT) is based on unequally-spaced fast Fourier transforms (USFFT) and the second FDCT is based on the frequency wrapping Fourier samples [11]. These two techniques handle the translation step with different choice of spatial grid. The FDCT using wrapping method is faster than USFFT. So, in this work wrapping method is used.

1.1. FDCT via Frequency wrapping:

The first step in the proposed method is to decompose the input image into curvelet coefficients using wrapping technique. In this work, input values for curvelet decomposition are set as follows: scales=3, angles=8. The algorithm of FDCT via wrapping is:

Step 1. **FFT:** Let the input image be $g[n1, n2]$ with $n1 > 0, n2 < k$ where $n = 2^k$, k belongs to ℕ. Perform 2D FFT (Fast Fourier Transform) on $g[n1, n2]$ to obtain Fourier samples.

$$f[n1, n2] = fft(g[n1, n2]), -k/2 \leq n1, n2 < k/2$$

Step 2. **Digital tiling:** Multiply the window function $\widetilde{U}_j$ with Fourier samples f. In other terms, FDCT via wrapping periodization is done to localize the obtained Fourier samples in window (or rectangular) region.

$$\hat{f}[n1, n2] = f[n1, n2]\,\widetilde{U}_j[n1, n2]$$

Step 3. **Translation:** Wrap (W) this product around a rectangle centered at the origin.

$$f'[n1, n2] = W(\hat{f}[n1, n2])$$

Step 4. **IFFT:** Get the discrete curvelet coefficients by applying inverse 2D FFT to the wrapped objects. Curvelet coefficients are a function of scale (j), space (k) and orientation (l).

$$C(j,l,k) = ifft(f'[n1,n2])$$

After decomposition, the curvelet coefficients are generated in three different levels: coarse, detail and fine. For the input image of size 512x512, the coarse level contains low-frequency coefficients of size 171x171. Fine level of middle-frequency coefficients is of size 512x512. Detail coefficients (high-frequency) are cell arrays of varying sizes according to angles.

It should be noted that four-step algorithm shown above constitute the forward FDCT decomposition. The inverse transform to be applied at decoder is computed by taking the adjoint function of each step then and applying them in reverse order.

2. *Discrete Cosine Transformation*:

The coarse level of curvelets contains low-frequency coefficients with reduced dimension. But their values are higher and difficult to encode. So to further increase the efficiency of curvelets, the idea of applying DCT to the curvelet coefficients is proposed in this paper. This approach reduces the number of high-valued coefficients thereby improving compression ratio. The advantages of both curvelets and DCT transforms are exploited efficiently. DCT is the widely adopted transform in digital image compression systems due to is simplicity and flexibility in representing the images as frequency domain elements [12, 13]. Low-frequency coefficient is at the upper-left corner while high-frequency coefficients are kept increasing along the lower-right part of the DCT matrix. DCT can be applied to an image in sub-blocks of varying sizes to help reducing the distortion error in final image. So, the curvelet levels are divided into sub-blocks. The coarse level is divided as sub-blocks of size 19x19 while the fine level is processed as 16x16 sub-blocks.

The forward and inverse equations of DCT are:

$$D(m,n) = \frac{2}{\sqrt{MN}}\alpha(m)\alpha(n)\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} C(i,j)\,cos\left[\frac{m\pi(2i+1)}{2M}\right]cos\left[\frac{n\pi(2j+1)}{2N}\right] \quad (1)$$

$$C'(i,j) = \frac{2}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \alpha(m)\alpha(n)\,D(m,n)cos\left[\frac{m\pi(2i+1)}{2M}\right]cos\left[\frac{n\pi(2j+1)}{2N}\right] \quad (2)$$

$$\alpha(m),\alpha(n) = \begin{cases} \frac{1}{\sqrt{2}}, & m,n = 0 \\ 1, & otherwise \end{cases} \quad (3)$$

where $C(i,j)$ is the Curvelet processed image, i,j are the discrete frequency variables, MN denotes the dimension of image such as 8×8, 16×16, $D(m,n)$ is the resultant image of forward DCT and $C'(i,j)$ is the resultant image of inverse DCT.

Fig.4 shows a part of curvelet coefficients obtained from decomposition of cameraman image and their corresponding coefficients after applying DCT.

Curvelet coefficients

454.7008	473.9320	466.7442	465.9391
479.6581	511.8777	490.6722	499.2474
442.1809	473.2429	446.6587	465.3727
462.0554	491.9560	461.0627	485.8588

$\xrightarrow{\text{DCT}}$

DCT coefficients

9074.9425	-54.8960	4.4804	73.0325
20.2178	-103.7135	-28.3106	35.2062
-41.5804	-25.9602	164.7436	8.9017
65.6493	24.5760	-51.4033	-63.9898

Fig.4. Transformation of curvelet coefficients after the application of DCT for the cameraman image

3. *Quantization*

Quantization is the process of reducing the number of possible values of a coefficient/symbol, thereby reducing the number of bits needed to represent it. A simple form of quantization is division. The DCT coefficients so far obtained contains decimal values which are quite difficult to encode. So, they are rounded to nearest integer values to implement compact representation. This process causes loss in information. To achieve fine quantization, a fraction is used for dividing the coefficients. Let q denotes the quantization factor and δ is the weight factor. The values for quantization factor used in the experiment are 15, 30, 45 and 60. The formulae for quantization and de-quantization stages are as follows:

$$Q(m,n) = round\left(\frac{D(m,n)}{q + \delta}\right) \tag{4}$$

$$Q'^{(m,n)} = Q(m,n) * (q + \delta) \tag{5}$$

where $q = 2/3$ (assumption based on trial and error), $Q(m,n)$ is the quantized matrix and $Q'(m,n)$ is the dequantized matrix.

4. *Global Structure Transformation:*

The novel idea of utilizing the benefits of GST is proposed in this work. The purpose of using GST in compression is to produce an output which is more compressible by the entropy encoder. As the name suggests, it transforms the local structure of a sequence into a global structure and thus it improves coding redundancy. GST stage is included before the encoder to increase the compression ratio. Being a reversible transform, it is used to transfer large number of quantized coefficients into a compact form without degrading the quality of reconstructed image. There are several variants of GST available. Among them, three widely used techniques are presented and their performances for our proposed method are analysed.

4.1. Move-to-front Transform:

Move-to-front (MTF) transform is the first known GST suggested for BWCA [14, 15]. MTF converts input symbols into a sequence of indices (or positions) i.e. the symbols with large values can be efficiently represented as small-valued indices. The length of MTF output sequence remains the same as that of the input sequence. Let the input sequence be X and output is denoted as Y_{mtf}. The unique symbols obtained from X are stored as symbol list. Let it be denoted as S. The algorithm works as follows:

Algorithm 1: Algorithm of Move-to-front transform

```
for $i = 1$ to $length(X)$ do
    for $j = 1$ to $length(S)$ do   /*find position of current symbol in symbol list*/
        if $X[i] == S[j]$
        $Y_{mtf}[i] = j$
        end
    end
    move $S[j]$ to the front (first position) of $S$
end
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

Table 1. MTF transform for Cameraman image

Input Sequence (X)	MTF sequence (Y_{mtf})	Symbol list (S)
219,7,94,5,78,7,38,3,5	6	3,5,7,38,78,94,**219**
219,**7**,94,5,78,7,38,3,5	6,3	219,3,5,**7**,38,78,94
219,7,**94**,5,78,7,38,3,5	6,3,6	7,219,3,5,38,78,**94**
219,7,94,**5**,78,7,38,3,5	6,3,6,4	94,7,219,3,**5**,38,78
219,7,94,5,**78**,7,38,3,5	6,3,6,4,6	5,94,7,219,3,38,**78**
219,7,94,5,78,**7**,38,3,5	6,3,6,4,6,3	78,5,94,**7**,219,3,38
219,7,94,5,78,7,**38**,3,5	6,3,6,4,6,3,6	7,78,5,94,219,3,**38**
219,7,94,5,78,7,38,**3**,5	6,3,6,4,6,3,6,6	38,7,78,5,94,219,**3**
219,7,94,5,78,7,38,3,**5**	**6,3,6,4,6,3,6,6,4**	3,38,7,78,**5**,94,219

Table 1 shows the working of MTF transform in the proposed method for a set of values (coarse level at 153rd row and columns from 39 to 47) from cameraman image. At first row, the current symbol '219' is searched in Symbol list and found at 6th position. So, 6 is coded in MTF sequence and '219' is moved to front of S. Similarly, all other symbols are processed in this way. The inverse of the above steps are followed at decoder to recover the original string i.e. input sequence is recovered from MTF sequence.

4.2. Distance coding transform:

Distance coding (DC) transform is a GST in which the distance between a symbol and its next occurrence are calculated [16, 17]. DC transform produces two sequences as the output. One is the 'info list' and the other is DC output sequence. Let us denote 'info list' as P and the input sequence as X. The current position and next position of current symbol is denoted as $start$ and $next$ respectively. Assume length of X is len, total number of elements in X as N, total number of unique symbols in X as M, list of those symbols as S, sorted list of first known positions (indices) of those corresponding symbols in X as K, output sequence be Y_{dc}. The steps of DC are as follows:

Algorithm 2 : Algorithm of Distance Coding transform

```
declare $i = 1, k = 2, c = 1$
while $k = 2$ do
 $j = X[i]$
     while $k = 2$ do    /* skip over successive positions*/
            $i = i + 1$
            if $j \geq len$ or $X[i] \cong j$ then break
          end
     end
 if $j \geq len$ then break
 end
 $next = j + 1$
 $start = j - 1$
     while $X[next] \cong j$ do   /*find next occurrence*/
            $next = next + 1$
            if $next > len$ then break
```

DOI: 10.15680/IJIRCCE.2017. 0503060

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

```
            end
        end
    if $next > len$   /* output is zero when no occurrence exists */
        $Y_{dc}[c] = 0$
        $c = c + 1$
    else  /* evaluate distance between current and next positions excluding known positions*/
        $distance = next - start$
        for $n = 2$ $to$ $M$ do
            if $next > K[n]$ then decrement $distance$
            end
        end
        $Y_{dc}[i] = distance$
    end
        update $K$
    end
```

After these steps, the number of elements of input sequence is greatly reduced in DC output. The two output sequences, info list and DC list, are separately encoded in this work.

Table 2. DC transform for the cameraman image

Input Sequence (X)	DC sequence (Y_{dc})
219,7,94,5,78,7,38,3,5	0
219,7,94,5,78,7,38,3,5	0,1
219,7,94,5,78,7,38,3,5	0,1,0
219,7,94,5,78,7,38,3,5	0,1,0,1
219,7,94,5,78,7,38,3,5	0,1,0,1,0
219,7,94,5,78,7,38,3,5	0,1,0,1,0,0
219,7,94,5,78,7,38,3,5	0,1,0,1,0,0,0
219,7,94,5,78,7,38,3,5	0,1,0,1,0,0,0
219,7,94,5,78,7,38,3,5	**0,1,0,1,0,0,0**

Table 2 shows the steps of DC transform with the same set of values for cameraman image. Consider the first row, where the current symbol is '219'. The second occurrence of '219' in not found in X. So, 0 is coded as its corresponding value in DC output. Moving on to second row, the next symbol to be searched is '7'. The next occurrence of '7' is at the 5th position in X i.e. (5-1=4) '4' is the distance between them. Moreover, the known positions K between those symbols should be excluded. And so, subtract 3 (number of underlined positions) from 4. The resultant value 1 is appended to Y_{dc}. Consider the 8th row where current symbol '3' is in 8th position. Obviously it checks for any successive symbols at 9th (last) position of X and found that there is no such case. It is understood that '3' does not have any occurrences. So, nothing is coded as output. Similarly at last row, the current symbol '5' is the last symbol and of course it cannot contain any next occurrences. Hence, nothing is coded.

When sending DC sequence to receiver, the first known positions of symbols and unique symbol list are also to be sent to successfully decode the original sequence.

4.3. Inversion frequencies:

Inversion Frequencies (IF) is based on the distance between a symbol and its consecutive occurrence as well as the frequency of symbols in a sequence. In IF, the term frequency of any symbol denotes the number of occurrences of that symbol in a given sequence. Similar to DC method, the output consists of two sequences. One is the list of unique symbols named as S . The other is the output sequence Y_{if}. The length of Y_{if} is equal to the length of input sequence. Let us denote the input sequence as X and all indices of current symbol as B. The algorithm is shown in below steps:

Algorithm 3: Algorithm of Inversion Frequencies transform

```
Set $i = 1, t = 1, q = 1$
for $j = 1$ to $length(S)$ do   /* find all indices of current symbol in input sequence */
    for $k = 1$ to $length(X)$ do
        if $S[j] == X[k]$
        $B[i] = k$
        $i = i + 1$
       end
    end
     $Y_{if}[t] = B[q]$ /* first occurrence index of symbol makes its first entry to output and this is referred as current position of that symbol*/
     $t = t + 1\,; q = q + 1$
    while $q \le length(B)$ do  /* find distance between current and new positions */
        $Y^{if}[t] = B[q] - B[q-1]$
        $t = t + 1\,; q = q + 1$
    end
    $Y_{if}[t] = 0$  /*indication of  no more occurrences of current symbol in input sequence and so move to next symbol in symbol list */
    $t = t + 1$
    $q = 1$
    delete all positions $j$ from $X$
end
```

The IF sequence along with the list of unique symbols and their counts are needed to decode successfully. So, this information is sent to the receiver.

Table 3. IF transform for the cameraman image

Symbols (S)	**input sequence** (X)	**IF sequence** (Y_{if})
3	219,7,94,5,78,7,38,**3**,5	8
	219,7,94,5,78,7,38,**~~3~~**,5	**8,0**
5	219,7,94,**5**,78,7,38,~~3~~,5	4
	219,7,94,**~~5~~**,78,7,38,~~3~~,**5**	4,4
	219,7,94,~~5~~,78,7,38,~~3~~,**~~5~~**	**4,4,0**
7	219,**7**,94,~~5~~,78,7,38,~~3~~,~~5~~	2
	219,**~~7~~**,94,~~5~~,78,**7**,38,~~3~~,~~5~~	2,3
	219,~~7~~,94,~~5~~,78,**~~7~~**,38,~~3~~,~~5~~	**2,3,0**

38	219,~~7~~,94,~~5~~,78,~~7~~,**38**,~~3~~,~~5~~	4
	219,~~7~~,94,~~5~~,78,~~7~~,**~~38~~**,~~3~~,~~5~~	**4,0**
78	219,~~7~~,94,~~5~~,**78**,~~7~~,~~38~~,~~3~~,~~5~~	3
	219,~~7~~,94,~~5~~,**~~78~~**,~~7~~,~~38~~,~~3~~,~~5~~	**3,0**
94	219,~~7~~,**94**,~~5~~,~~78~~,~~7~~,~~38~~,~~3~~,~~5~~	2
	219,~~7~~,**~~94~~**,~~5~~,~~78~~,~~7~~,~~38~~,~~3~~,~~5~~	**2,0**
219	**219**,~~7~~,~~94~~,~~5~~,~~78~~,~~7~~,~~38~~,~~3~~,~~5~~	1
	~~219~~,~~7~~,~~94~~,~~5~~,~~78~~,~~7~~,~~38~~,~~3~~,~~5~~	**1,0**
	~~219~~,~~7~~,~~94~~,~~5~~,~~78~~,~~7~~,~~38~~,~~3~~,~~5~~	**8,0,4,4,0,2,3,0,4,0,3,0,2,0,1,0**

Table 3 shows IF output for cameraman image with specified set of values. For each symbol, IF sequence is calculated and their concatenation gives the final code. In row 1, current symbol is '3' and its first occurrence is at 8th position (input). So, output it as 8**.** At second row, the current pointer points to 8th position of '3'. There is no occurrence of '3' thereafter and hence 0 is included to show that searching process for '3' is finished. Consider the 3rd row, the previous positions of symbol '3' are not considered and hence it is given in gray. Now, the first occurrence of '5' is found. In row 4, the distance between current and next positions of '5' are evaluated (distance=4) and appended to the output Y_{if}. This process is continued till all symbols are processed. The IF sequence along with the list of unique symbols and their counts are needed to decode successfully. So, this information is sent to the receiver.

5. *Arithmetic encoding:*

Arithmetic coding is the concept of subdividing the interval [0, 1) iteratively based on the probability distribution of the input data. There is no direct correspondence between the arithmetic code and the source symbols. It is better than Huffman encoding in providing higher compression ratios [18, 19]. For convenience, integer registers are used here to hold the floating-point intervals. The GST output is finally encoded using arithmetic encoding to produce the compressed image.

IV. Experimental Results

In this section, the proposed image compression is tested on 6 images: Barbara, Gold Hill, Airplane, Butterfly, Cameraman and Lena. The performance measures used to evaluate the proposed system are Peak-Signal-to-Noise Ratio, Structural Similarity Index Measure, Compression Ratio, Encoding and Decoding time. The formulae for PSNR, SSIM and CR are as follows:

$$PSNR = 10 * log\left(\frac{255^2}{MSE}\right) \quad (6)$$

$$MSE = \frac{1}{MN}\sum_{i=0}^{M}\sum_{j=0}^{N}(I(i,j) - J(i,j))^2 \quad (7)$$

where *MSE* denotes Mean Square Error, $I(i,j)$ is the original image and $J(i,j)$ is the reconstructed image.

$$SSIM(A,B) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \quad (8)$$

where μ_x, μ_y are average values, σ_x^2, σ_y^2 denotes variance, σ_{xy} is the covariance of x and y, $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ are variables for stabilization, $L = 255$ denotes pixel range, $k_1 = 0.01$, $k_2 = 0.03$ are default constants and x, y are original and reconstructed images respectively.

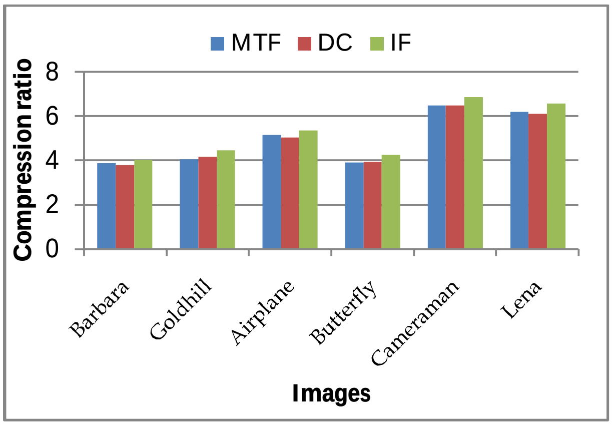
DOI: 10.15680/IJIRCCE.2017. 0503060

$$CR = \frac{Uncompressed\ size}{compressed\ size} \tag{9}$$

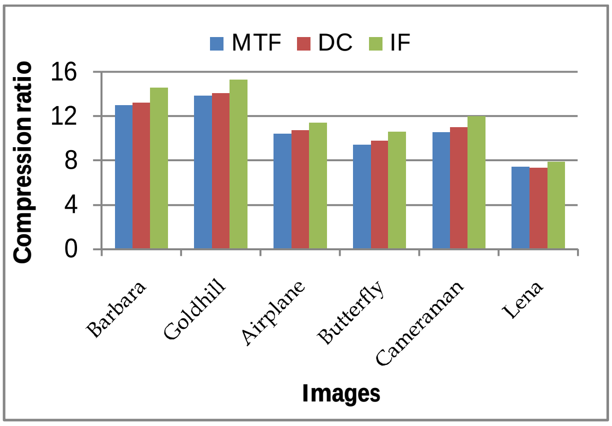
Table 4. Experimental results showing PSNR and SSIM at various quantization factors

Images	***q*=15**		***q* =30**		***q* =45**		***q* =60**	
	PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Barbara	37.0289	0.9489	33.0496	0.9100	30.6545	0.8701	28.916	0.8256
Gold hill	35.7320	0.9172	31.6746	0.8222	29.7532	0.7626	28.5808	0.7238
Airplane	38.0030	0.9409	34.0597	0.8976	31.6413	0.8634	29.8994	0.8325
Butterfly	35.6479	0.9151	31.8944	0.8483	30.1365	0.8065	28.9751	0.7760
Cameraman	39.8987	0.9599	35.1545	0.9133	32.7732	0.8774	31.3279	0.8542
Lena	37.4241	0.9264	34.2764	0.8858	32.3623	0.8541	30.9068	0.8280
Average	**37**	**0.93**	**33**	**0.87**	**31**	**0.83**	**30**	**0.8**

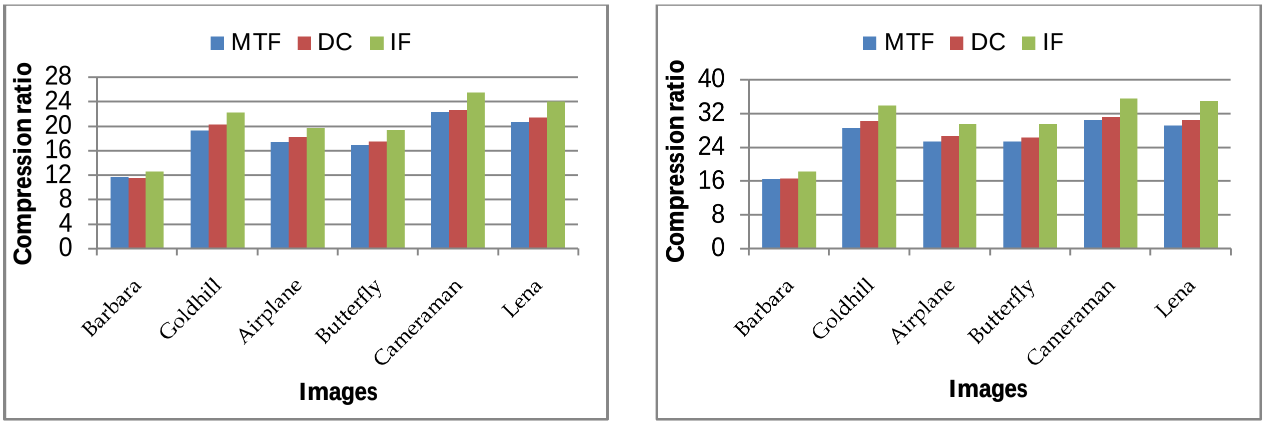
From Table 4 it can be seen that the PSNR values of all the images are higher at lower quantization values. The SSIM of cameraman image shows gradual changes: 0.9599, 0.9133, 0.8774 and 0.8542. This proves that the structural differences between original and reconstructed images are preserved to a greater extent at higher compression rates. On average, the PSNR range obtained for our proposal at given quantization factors falls within the acceptable range (30 to 40 dB) of image quality. One can vary the degree of quantization depending on the requirements.



a) Compression Ratio at *q*=15



b) Compression Ratio at *q*=30



c) Compression Ratio at q_v=45

d) Compression Ratio at q_v=60

Fig.5. Bar charts showing the compression ratio of three GST methods at various quantization levels

Fig.5 shows the comparison of three GST methods in terms of compression ratio at various quantization values. In Fig.5. a) with quantization value 15, three cases can be seen: i) For Butterfly and Cameraman images, the distance coding and move-to-front transforms have nearly equal compression ratios ii) The DC transform outperforms MTF only in Goldhill image iii) MTF is higher than DC in three other images- Barbara, Airplane and Lena. This proves that neither MTF nor DC transforms gives equivalent results when images are of different categories. However in each quantization level, IF transform gives high CR values for all the images than other two transforms.

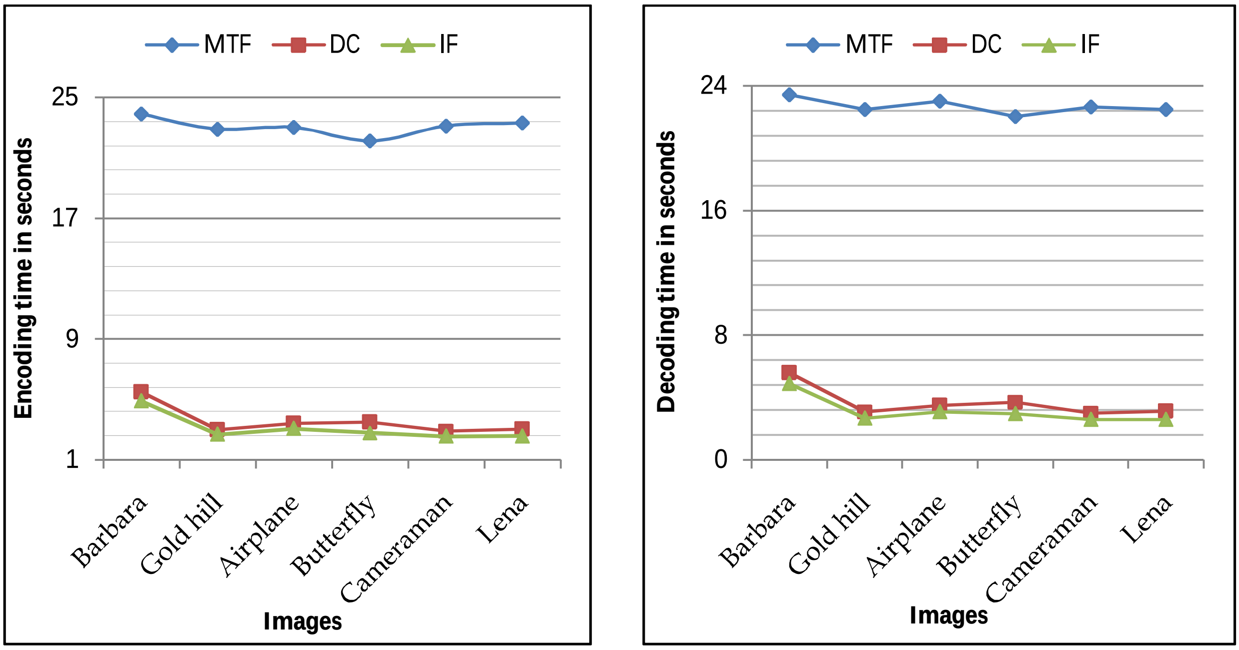


Fig.6. Line Chart showing the comparison of encoding and decoding time for three GST methods

Fig.6 shows the encoding and decoding time of proposed technique at q=60. The comparison reveals that inversion frequencies method is faster than other two methods. Slight variations are found between Distance coding and inversion methods. When all the three measures- CR, encoding and decoding time- are taken into account, IF shows higher performance than MTF and DC. The reconstructed images (q = 15) are shown in Fig.7. The screenshot of the proposed Curvelet based image compression method in GUI is shown in Fig.8.

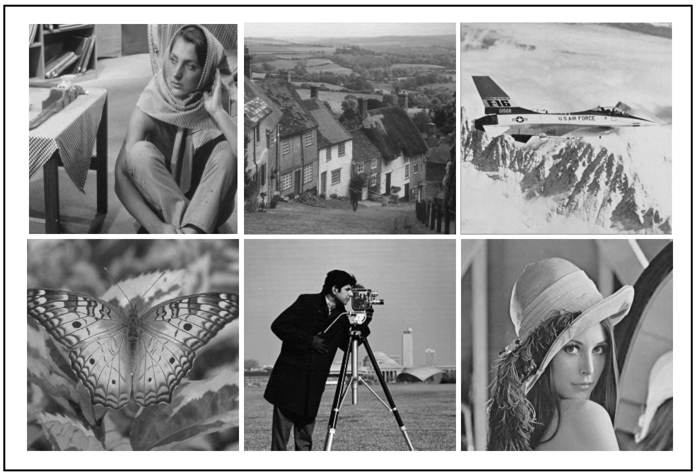
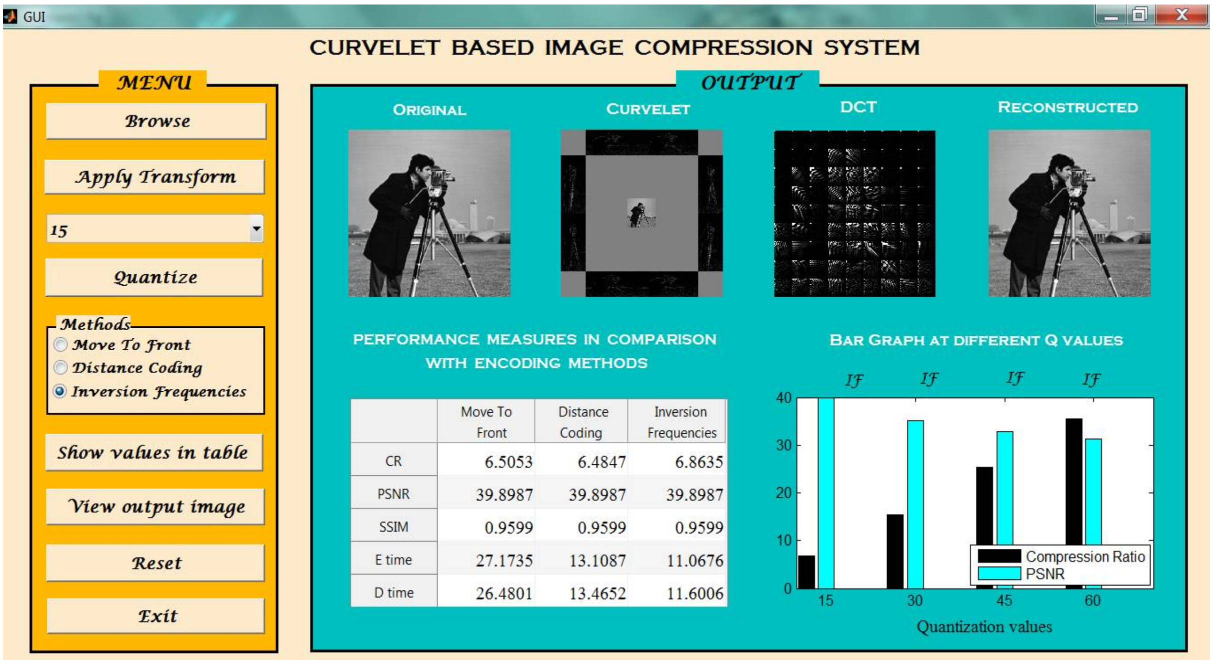


Fig.7. Reconstructed images at quantization level 15



CURVELET BASED IMAGE COMPRESSION SYSTEM

Methods
○ Move To Front
○ Distance Coding
● Inversion Frequencies

PERFORMANCE MEASURES IN COMPARISON WITH ENCODING METHODS

	Move To Front	Distance Coding	Inversion Frequencies
CR	6.5053	6.4847	6.8635
PSNR	39.8987	39.8987	39.8987
SSIM	0.9599	0.9599	0.9599
E time	27.1735	13.1087	11.0676
D time	26.4801	13.4652	11.6006

Fig.8. Screenshot of the proposed compression method in GUI

V. Conclusion

In this paper, a new approach based on Curvelets, DCT and GST is proposed. The novel approach of using DCT on curvelet coefficients shows good PSNR values. The quantization produced coefficients with reduction in total bits. Then, the coefficients are processed in GST to generate highly redundant coefficients. At last, when arithmetic encoding was used, the redundant coefficients are pretty well exploited to produce higher compression. Moreover, the proposed approach works better on high compression ratios thereby reducing distortion error in reconstructed images. Adoption of GST stage gives better compression. Their comparison shows that Inversion Frequencies are faster and gives optimal results. Experimental results proved that the efficiency of IF transform for DCT based curvelet compression is good. Thus the lossless structural transformations when implemented efficiently in lossy compression methods, gives better performances.

REFERENCES

1. Mrinal Kr. Mandal , 'Digital Image Compression Techniques', Multimedia Signals and Systems, Springer, Vol. 716, pp.169-202, 2003.
2. Muhammad Azhar Iqbal, Dr. Muhammad Younus Javed, Usman Qayyum, 'Curvelet–based Image Compression with SPIHT', International Conference on convergence Information Technology, pp. 961-965, 2007.
3. M. Manikandan, A. Saravanan and K. Bhoopathy Bagan, 'Curvelet Transform Based Embedded Lossy Image Compression', IEEE-ICSCN, pp. 274-276, 2007.
4. A. Majumdar, 'Image Compression by sparse PCA coding in curvelet domain', Signal, Image and Video Processing, Springer, Vol. 3, Issue 1, pp 27–34, 2009.
5. Yuancheng Li, Qiu Yang, Runhai Jiao, 'Image compression scheme based on curvelet transform and support vector machine', Expert Systems with Applications, Elsevier, Vol. 37, Issue 4, pp. 3063–3069, 2010.
6. Arun Vikas Singh and K. Srikanta Murthy, 'Neuro-Curvelet Model for Efficient Image Compression Using Vector Quantization', Proceedings of International Conference on VLSI, Communication, Advanced Devices, Signals & Systems and Networking , pp. 179-185, 2013.
7. Jalal M. Fadili, Jean-Luc Starck, 'Curvelets and Ridgelets', Computational Complexity, Springer, pp. 754-773, 2012.
8. Jianwei Ma and Gerlind Plonka , 'A Review of Curvelets and Recent Applications', International Journal of Recent Trends in Engineering, Vol 1, No. 3, 2009.
9. Emmanuel J. Candes and David L. Donoho , 'Curvelets - A Surprisingly Effective Non adaptive Representation For Objects with Edges', Saint-Malo Proceedings, 1999, http://www-stat.stanford.edu/{~emmanuel,~donohog}/.
10. Emmanuel J. Candes and David L. Donoho, 'New Tight Frames of Curvelets and Optimal Representations of Objects with C2 Singularities', Wiley Online Library, 2002.
11. Emmanuel Candes, Laurent Demanet, David Donoho and Lexing Ying, 'Fast Discrete Curvelet Transforms', Multiscale Modeling & Simulation, vol. 5, pp. 861–899, 2006.
12. Wael M. Khedr and Mohammed Abdelrazek , 'Image Compression using DCT upon Various Quantization', International Journal of Computer Applications, Vol. 137, No.1, pp.11-13, 2016.
13. D. J. Ashpin Pabi, M. Mahasree, P. Aruna, N. Puviarasan , 'Image Compression based on DCT and BPSO for MRI and Standard Images', International Journal of Engineering Research and Application, ISSN : 2248-9622, Vol. 6, Issue 10, (Part -5) October 2016, pp. 24-31.
14. C.D.Rawat and Smita Rao., 'Evaluation of Burrows Wheeler Transform based Image Compression Algorithm for Multimedia Applications', International Conference on Advances in Communication and Computing Technologies, 2014.
15. Sarada Prasad Dakua1 and Jyotinder Singh Sahambi , 'Lossless ECG Compression for Event Recorder Based on Burrows-Wheeler Transformation and Move-To-Front Coder', International Journal of Recent Trends in Engineering, Vol. 1, No. 3, May 2009.
16. R. Bastys, 'Fibonacci Coding Within the Burrows-Wheeler Compression Scheme, 2010.
17. Binder E., 'Distance coding algorithm', http://groups.google.com.
18. K. S. Thyagarajan , 'Still image and Video compression With matlab', A John Wiley & Sons, Inc., Publication, 2011.
19. Mark Nelson, Gean-Loup Gailly, 'The Data Compression Book', Second edition.