RESEARCH ARTICLE OPEN ACCESS

Image Compression Using Hybrid Combinations of DCT SVD and RLE

Raghavendra.M.J[1],
[1]Assistant Professor, Department of Telecommunication,
P.E.S. Institute of Technology,
Bangalore, India

Dr.Prasantha .H.S[2] , Dr.S.Sandya[3]
[2, 3] Professor, Department of Electronics and Communication,
Nitte Meenakshi Institute of Technology,
Bangalore, India

--*************************--

Abstract:

Image Compression finds a significant place in the field of research. In this paper we are proposing a scheme for hybrid image compression which uses Discrete Cosine Transform, Singular Value Decomposition and Run length Encoding. Discrete Cosine Transform is applied to the image. Then DC-Coefficient is taken out from Discrete Cosine Transformed Matrix and stored or transmitted separately. The Discrete Cosine Transformed matrix without DC-coefficient is truncated with a threshold value. To this truncated matrix Singular Value Decomposition is applied. The matrices obtained from the Singular Value Decomposition are again truncated with suitable threshold value. Then these matrices are multiplied back. The resultant matrix is again truncated with threshold value. Then this matrix is quantized. The quantized matrix is converted into sparse matrix form. Then sparse matrix elements under goes data type conversion. The column elements of the sparse matrix are run length encoded and then compressed form of the image can be obtained. This compressed form can be stored or transmitted. An effort is also made to compare the number of memory bytes obtained in this method with the three other methods which are discussed.

Keywords-- **DCT-Discrete Cosine Transform, SVD-Singular Value Decomposition, MSE-Mean Squared Error, PSNR-Peak Signal to Noise Ratio, CR-Compression Ratio ,RLE-Run Length Encoding**

--*************************--

I. INTRODUCTION

There exists always demand for Image compression in the field of Multimedia. Image Compression is broadly classified into two types. They are lossless image compression techniques and lossy image compression techniques. It can be learnt that in the lossless image compression techniques the reconstructed image quality is better than the lossy image compression techniques. But when we compare with the compression ratio, lossy compression technique is better than the loss less compression technique. In this paper we are proposing hybrid image compression technique using DCT, SVD and RLE. This is a lossy compression technique.

This paper consists of seven sections. The first section deals with the introduction, the second section deals with literature survey, the third section deals with the methodology, the fourth section deals with implementation, the fifth section deals with the results and discussions , the sixth section deals with the scope for further enhancement and the seventh section deals with the references.

II. LITERATURE SURVEY

There are different contributions to the above discussed problem. Few papers are discussed in this section.

Raghavendra.M.J [1] and others have worked on Image Compression using DCT and SVD to achieve image compression. Prasantha.H.S and others [2] have worked on image compression using SVD. S.Sridhar and others [3] have worked on image compression using different types of wavelets. T.D.Khadatre and others [4] have worked on compression of image using vector quantization and wavelet transform. Athira.M.S and others [5] have worked on image compression using artificial neural networks. Pallavi and others [6] have worked on image compression using Wavelets and Huffman Coding. E.Praveen Kumar and others [7] have worked on image compression using multiwavelet transforms. D.Vishnuvardhan and others [8] have worked on image compression using curvelets.

Birendrakumar Patel and others [9] have worked on image compression using Artificial Neural Networks. Sumegha.Y and others [10] have worked on fractal image compression using Discrete Cosine Transform and Discrete Wavelet Transform. Rowayda A.S [11] worked on SVD for image processing applications. K.R.Rao [12] and others have worked on DCT.

III. METHODOLOGY

In the proposed scheme, discrete cosine transform and singular value decomposition and run length encoding are used to compress the image data.

Discrete Cosine Transform

Discrete cosine transform very useful in image compression. In this it will transform the energy of the signal into lower order frequency coefficients. The formula of 2-dimensional DCT for the input function f(x,y)is as follows.

$$A(u,v) = B(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left\{\frac{(2x+1)u\pi}{2N}\right\}\cos\left\{\frac{(2y+1)v\pi}{2N}\right\} \quad (1)$$

Where u =0, 1, 2...N-1, v=0, 1, 2...N-1, f(x, y) =input function

The inverse 2-dimensional DCT formula is as follows

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} B(u)C(v)A(u,v)\cos\left\{\frac{(2x+1)u\pi}{2N}\right\}\cos\left\{\frac{(2y+1)v\pi}{2N}\right\} \quad (2)$$

Where B (u) $=\sqrt{1/N}$ for u=0, B (u) $=\sqrt{2/N}$ for u=1, 2 ...N-1

Similarly C (v) $=\sqrt{1/N}$ for v=0, C (v) $=\sqrt{2/N}$ for v=1, 2 ...N-1

Singular Value Decomposition

Singular value decomposition takes rectangular matrix as input and transforms it into three matrices “U”, “S” and “V”. If the input rectangular matrix is “X”, then the relationship between “X” and “U”, “S” and “V” are $X=U*S*V^T$, where V^T is the transpose of the “V” matrix. If “X” matrix is of the order m×n, then order of the “U” matrix is of m×m, order of the “S” matrix is m×n and the order of “V”is n×n. The “S” matrix is the important matrix because it has the singular values of the input matrix. The “S” matrix has only principal diagonal elements. The magnitudes of the diagonal elements are placed in decreasing order.

Run Length Encoding

It is a lossless compression technique. In this method number of frequently occurring symbols are counted and it encoded before the symbol. In this way it reduces the transmission bandwidth.

Sparse Matrix

Sparse matrix is one in which majority of the elements are zero. Since majority of the elements are zero, the sparse notation is applied to reduce the transmission bandwidth. In the sparse notation only non-zero element’s row, column and value are stored.

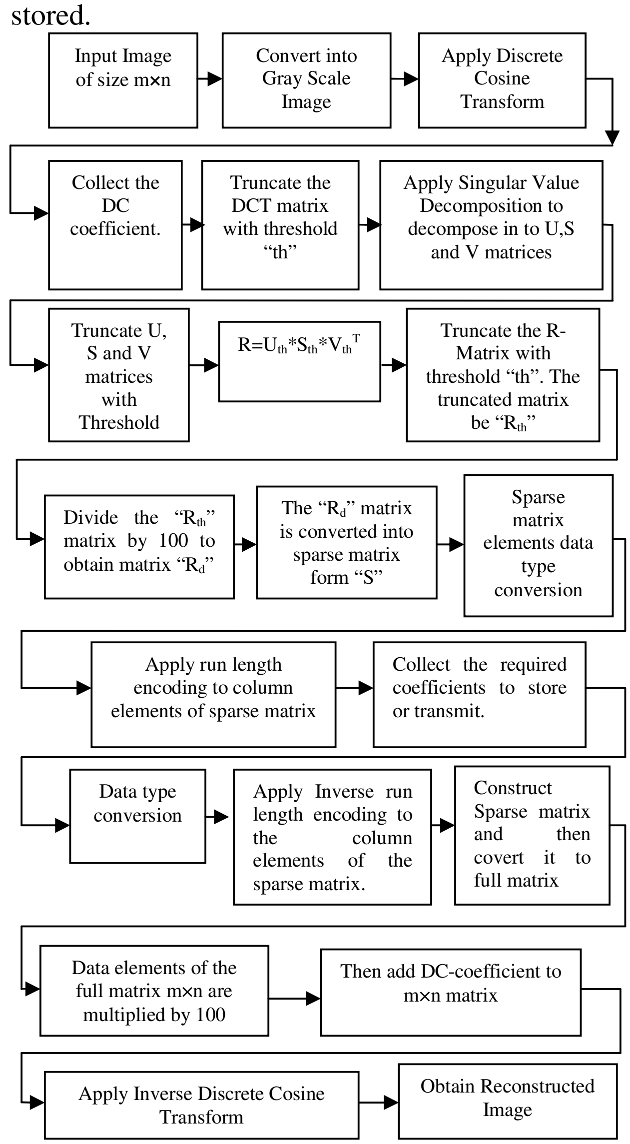


Figure1. Block diagram of the image compression using DCT-SVD-RLE method.

In this paper an effort is made to compress the image using hybrid compression techniques. They are (i) Image Compression using DCT-SVD-RLE method. (ii) Image compression using DCT-SVD method (iii) Image compression using DCT-RLE method, and (iv) Image compression using DCT method.

The figure1 shows the block diagram of the image compression using DCT-SVD-RLE method. In this method an image of size m×n is applied as input. Suppose, if the given image is not in the Gray Scale format, it is converted into Gray Scale format. To this gray scale image, DCT is applied. Let this Discrete Cosine Transformed matrix be "D". In the matrix "D", the DC-coefficient is taken out and stored separately. The DC coefficient is the largest and important coefficient. Therefore it is stored separately. Then in the matrix "D", all those coefficients less than the threshold "th" are neglected. After neglecting the coefficients less than "th" the discrete cosine transformed, truncated matrix "D_{th}" is obtained. To this "D_{th}" matrix, singular value decomposition is applied. This singular value decomposition, decomposes matrix "D_{th}" into matrix "U", matrix "S" and matrix "V". Then, in the "U" matrix all those coefficients less than 0.02 are neglected. Let this truncated matrix be "U_{th}". In the "S" matrix all those coefficients less than 400 are neglected. Let this truncated matrix be "S_{th}". In the "V" matrix all those coefficients less than 0.05 are neglected. Let this truncated matrix be "V_{th}". The threshold value for "U" matrix i,e 0.02, the threshold value for "S" matrix i,e 400 and the threshold value for "V" matrix i,e 0.05 are selected empirically. The experiments are conducted for different values but these values found to be optimum. Then truncated matrices are multiplied such that $R=U_{th}*S_{th}*V_{th}^T$. Then "R" Matrix is again truncated with threshold "th". That means all those coefficients less than "th" are neglected. Let this truncated matrix be "R_{th}". The elements of "R_{th}" matrix are divided by 100 as a quantization. This results in matrix "R_d". The "R_d" matrix contains most of the elements as zero, few coefficients are non-zero elements. Now "R_d" matrix is converted as a sparse matrix. For example if

$$R_d = \begin{bmatrix} 0 & 91 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$ can be represented as

S =

Row	Column	Data Element
1	2	91

Figure 2. Sparse Matrix and its representation.

The element under the column "Row" represents data element's Row, the element under the column "Column" represents the data element's column and the element under the column "Data element" represents element's value.

This sparse notation reduces the memory required to store the data. Also, the sparse notation reduces the number of coefficients to be transmitted after image compression. The elements in the sparse matrix form are represented in double data type which requires 8-bytes for storage in Matlab. Therefore the elements under the column "Row" and "Column" in figure 2 are converted from double data type to int16 data type, where int16 data type requires 2-bytes for storage in Matlab. The elements present under the column "data element" are converted into "int8", because int8 requires 1-byte to store the data in Matlab. Then for the elements present under the column "Column" in Figure.2 are run length encoded. Let the number of elements after run length encoding be "rlecol". Therefore the number of bytes present in the compressed form is

$$dcomp = rowele * 2 + rlecol * 2 + Dataelemnts * 1 + dcf + r + c \quad (3)$$

Where dcomp=number of bytes in the compressed form. rowele =number of elements present in sparse notation under the column "Row". rlecol= number of elements after run length encoding of elements under the column "column" in sparse matrix notation . Dataelemnts=number of elements under the column "Data element" in sparse matrix notation. dcf = 8 bytes to accommodate DC coefficient. r = 2bytes to accommodate the number of rows of the input image and c = 2bytes to accommodate the number of columns of the input image. Eventually "dcomp" is the number of bytes of input image in the compressed form. "rowele" and "rlecol" are multiplied by 2 because they are represented in the data type int16. It is assumed that "dcomp" is the number of bytes transmitted and the same number of coefficients are received. At the receiver, the received elements data types are converted back. The "rowele" , "rlecol" and

"Dataelemnts" are converted into double data type. Then inverse run-length encoding is applied to "rlecol" elements to get back the elements in sparse matrix notation. Then sparse matrix form is converted into full matrix form. Let this matrix be "RR" of size m×n .Then each element of the matrix "RR" are multiplied by 100.Then DC coefficient is added. After this, Inverse Discrete cosine Transform is applied to reconstruct the image. The parameters such as MSE, PSNR and Compression ratio are evaluated.

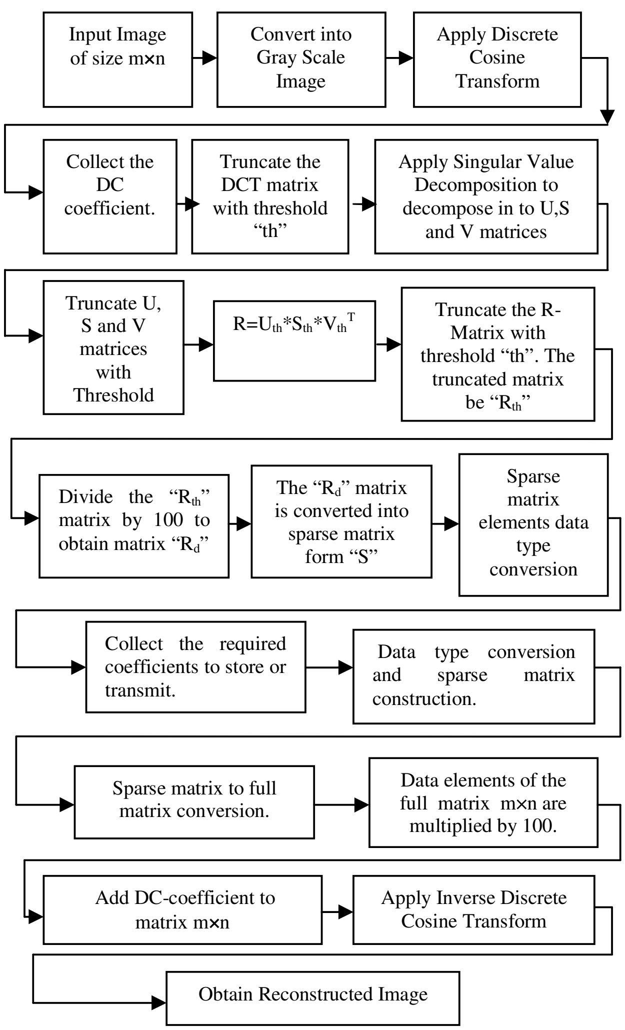


Figure3. Block diagram of the image compression using DCT-SVD method

The above figure shows the block diagram of the image compression using DCT-SVD method. In this method an image of size m×n is applied as input. Suppose, if the given image is not in the Gray Scale format, it is converted into Gray Scale format. To this gray scale image DCT is applied. Let this Discrete Cosine Transformed matrix be "D". In the matrix "D", the DC-coefficient is taken out and stored separately. Then in the matrix "D", all those coefficients less than the threshold "th" are neglected. After neglecting the coefficients less than "th", the discrete cosine transformed, truncated matrix "D_{th}" is obtained. To this "D_{th}" matrix singular value decomposition is applied. This singular value decomposition, decomposes matrix "D_{th}" into matrix "U", matrix "S" and matrix "V". Then, in the "U" matrix all those coefficients less than th_u=0.02 are neglected. Let this truncated matrix be "U_{th}". In the "S" matrix all those coefficients less than th_s = 400 are neglected. Let this truncated matrix be "S_{th}". In the "V" matrix all those coefficients less than th_v = 0.05 are neglected. Let this truncated matrix be "V_{th}". The threshold value for "U" matrix i,e 0.02, the threshold value for "S" matrix i,e 400 and the threshold value for "V" matrix i,e 0.05 are selected empirically. The experiments are conducted for different values but these values found to be optimum. Then truncated matrices are multiplied such that $R=U_{th}*S_{th}*V_{th}^T$. Then "R" Matrix is again truncated with threshold "th". That means all those coefficients less than "th" are neglected. Let this truncated matrix be "R_{th}". The elements of "R_{th}" matrix are divided by 100 as a quantization. This results in matrix "R_d". The "R_d" matrix contains most of the elements as zero, few coefficients are non-zero elements. Now "R_d" matrix is converted as a sparse matrix. Then the elements under the "Row" and the "Column" in figure 2 are converted from double data type to int16 data type. The elements present under the column "data element" are converted into "int8". Therefore the number of bytes present in the compressed form is

$$dcomp = rowele * 2 + colele * 2 + Dataelemnts * 1 + dcf + r + c \quad (4)$$

dcomp=number of bytes in the compressed form. rowele =number of elements present in the sparse notation under the column "Row". colele =number of elements present in the sparse notation under the column "Column". Dataelemnts=number of elements present under the column "Data element" in sparse matrix notation. dcf = 8 bytes to accommodate the DC coefficient. r = 2bytes to accommodate the number of rows of the input image and c = 2bytes to accommodate the number of columns of the input image. "rowele" and "colele" are multiplied by 2 because they are represented in the data type int16. It is assumed that "dcomp" is the number of bytes transmitted and the same number of coefficients are received. At the receiver, the received elements data types are converted back. The "rowele", "colele" and "Dataelemnts" are converted into double data type. Then sparse matrix is constructed. Then sparse matrix form is converted into full matrix form. Let this matrix be "RR" of size m×n. Then each element of the matrix "RR" are multiplied by 100.Then DC coefficient is added. After this, Inverse Discrete cosine Transform is applied to reconstruct the image. The parameters such as MSE, PSNR and Compression ratio are evaluated.

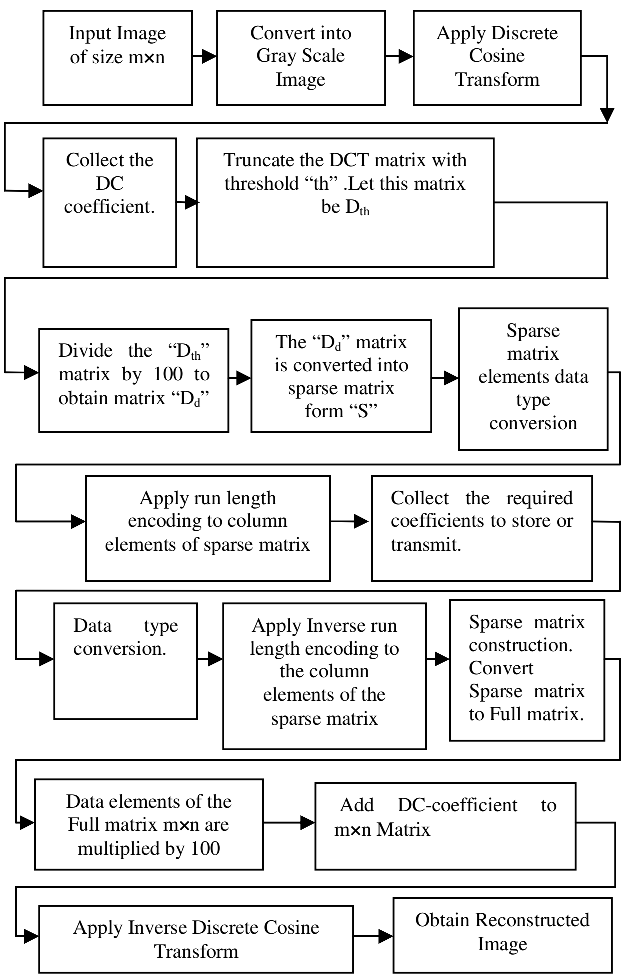


Figure4. Block diagram of the image compression using DCT-RLE method

The above figure shows the block diagram of the image compression using DCT-RLE method. In this method an image of size m×n is applied as input. Suppose, if the given image is not in the Gray Scale format, it is converted into Gray Scale format. To this gray scale image DCT is applied. Let this Discrete Cosine Transformed matrix be "D". In the matrix "D", the DC-coefficient is taken out and stored separately. Then in the matrix "D", all those coefficients less than the threshold "th" are neglected. After neglecting the coefficients less than "th" in the discrete cosine transformed, truncated matrix "D_{th}" is obtained. The elements of "D_{th}" matrix are divided by 100 as a quantization. This results in matrix "D_d". The "D_d" matrix contains most of the elements as zero, few coefficients are non-zero elements. Now "D_d" matrix is converted as a sparse matrix. The elements present under the column "data element" of sparse notation are converted into "int8", because int8 requires 1-byte to store the data in Matlab. The elements present under the column "Row" and "Column" of sparse notation are converted from data type double to int16.Where the data type "int16" requires 2-bytes.Then for the elements present under the column "column" in Figure.2 are run length encoded. Let the number of elements after run length encoding be "rlecol". Therefore the number of bytes present in the compressed form is

$$dcomp = rowele * 2 + rlecol * 2 + Dataelemnts * 1 + dcf + r + c \quad (5)$$

Where dcomp=number of bytes in the compressed form. rowele =number of elements present in sparse notation under the column "Row". rlecol= number of elements after run length encoding of elements under the column "column" in sparse matrix notation . Dataelemnts=number of elements under the column "Data element" in sparse matrix notation.dcf= 8 bytes to accommodate DC coefficient. r = 2bytes to accommodate number of rows of the input image and c = 2 bytes to accommodate number of columns of the input image. Eventually "dcomp" is the number of bytes of input image in the compressed form. "rowele" and "rlecol" are multiplied by 2 because they are represented in the data type int16. It is assumed that "dcomp" is the number of bytes transmitted and it is assumed that same number of coefficients are received. At the receiver, the received elements data types are converted back. The "rowele" , "rlecol" and "Dataelemnts" are converted into double data type. Then inverse run-length encoding is applied to "rlecol" elements to get back the elements in sparse matrix notation. Then sparse matrix form is converted into full matrix form. Let this matrix be "RR" of size m×n. Then each element of the matrix "RR" are multiplied by 100.Then DC coefficient is added. After this, Inverse Discrete cosine Transform is applied to reconstruct the image. The parameters such as MSE, PSNR and Compression ratio are evaluated.

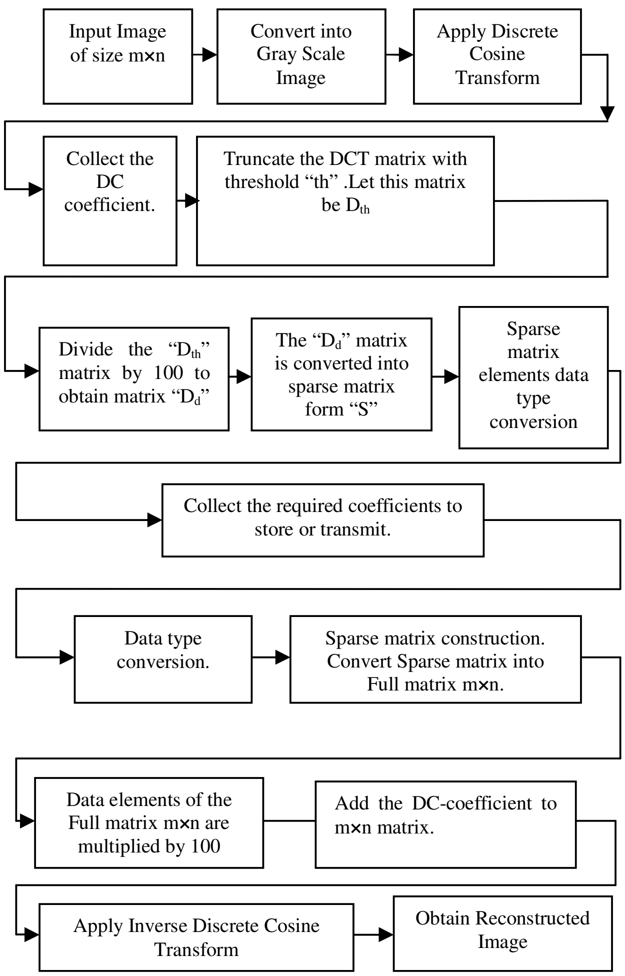


Figure5. Block diagram of the image compression using DCT method

The above figure shows the block diagram of the image compression using DCT method. In this method an image of size m×n is applied as input. Suppose, if the given image is not in the Gray Scale format, it is converted into Gray Scale format. To this gray scale image DCT is applied. Let this Discrete Cosine Transformed matrix be "D". In the matrix "D", the DC-coefficient is taken out and stored separately. Then in the matrix "D", all those coefficients less than the threshold "th" are neglected. After neglecting the coefficients less than "th", discrete cosine transformed, truncated matrix "D_{th}" is obtained. The elements of "D_{th}" matrix are divided by 100 as a quantization. This results in matrix "D_d". The "D_d" matrix contains most of the elements as zero, few coefficients are non-zero elements. Now "D_d" matrix is converted as a sparse matrix. Therefore the elements under the "Row" and the "Column" in sparse matrix notation are converted from double data type to int16 data type. The elements present under column "data element" are converted into "int8". Therefore the number of bytes present in the compressed form is

$$dcomp = rowele * 2 + colele * 2 + Dataelemnts * 1 + dcf + r + c \quad (6)$$

dcomp=number of bytes in the compressed form. rowele =number of elements present in sparse matrix notation under the column "Row". colele =number of elements present in sparse matrix notation under the column "Column". Dataelemnts = number of elements under the column "Data element" in sparse matrix notation. dcf = 8 bytes to accommodate DC coefficient. r = 2bytes to accommodate the number of rows of the input image and c = 2bytes to accommodate the number of columns of the input image. "rowele" and "colele" are multiplied by 2 because they represented in the data type int16. It is assumed that "dcomp" number of bytes are transmitted and the same number of coefficients are received. At the receiver, the received elements data types are converted. The "rowele", "colele" and "Dataelemnts" are converted into double data type. Then sparse matrix is constructed. Then sparse matrix form is converted into full matrix form. Let this matrix be "RR" of size m×n .Then each element of the matrix "RR" are multiplied by 100.Then DC coefficient is added. After this, Inverse Discrete Cosine Transform is applied to reconstruct the image. The parameters such as MSE, PSNR and Compression ratio are evaluated.

IV. IMPLEMENTATION

The experimentation is conducted using Matlab 7.6 on Intel(R) core i3 processor at 2.4 GHz. There are four methods. They are (i) Image Compression using DCT-SVD-RLE method. (ii) Image compression using DCT-SVD method (iii) Image compression using DCT-RLE method, and (iv) Image compression using DCT method.

The algorithm of the Image Compression using DCT-SVD-RLE method is as follows.

(i) Accept an image of size "m" rows and "n" columns.

(ii) If the accepted image is in any form other than the Gray Scale format, convert it into Gray Scale format.

(iii) Apply Discrete Cosine Transform to the Gray Scale format. This results in the matrix "D".

(iv) Collect the DC coefficient of the Discrete Cosine transformed matrix separately.

(v) Apply threshold "th" to the Discrete Cosine Transformed matrix "D".i,e all those coefficients less than "th" in the "D " matrix are neglected. This results in the matrix "D_{th}".

(vi) Apply Singular Value Decomposition to the matrix "D_{th}". This results in three matrices "U", "S" and "V".

(vii) Apply threshold "th_u = 0.02" to the "U" matrix. i,e all those coefficients less than "th_u " are neglected. This results in the matrix "U_{th}". Apply threshold "th_s=400" to the "S" matrix. i,e all those coefficients less than "th_s " are neglected. This results in the matrix "S_{th}". Similarly, apply threshold "th_v =0.05 " to the "V" matrix. i,e all those coefficients less than "th_v" are neglected. This results in the matrix "V_{th}".

(viii) Then multiply the matrices such that $R= U_{th}* S_{th}* V_{th}^T$.

(ix) Then apply threshold "th" to the matrix "R".i,e all those coefficient less than "th" are neglected in this matrix "R". This results in the matrix "R_{th}".

(x) Divide every element of the matrix "R_{th}" by 100.This results in the matrix "R_d".

(xi) Convert Matrix "R_d" from full matrix form to sparse matrix form. Let this sparse matrix be "S"

(xii) In the sparse matrix form "S", convert the data elements under the column "Row" from data type double to the int16, convert the data elements under the column "column" from data type double to the int16 and convert the data elements under the column "data element" from data type double to the int8.

(xiii) After this, apply run length encoding to the data under the column "column" of the sparse matrix. This gives the run length encoded data as "rlecol"

(xiv) Then, the elements of the column "data elements" of the sparse matrix, the elements of the column "row " of the sparse matrix and the run length encoded data "rlecol", DC-coefficient ,number of rows of matrix "R_d" and the number of rows and columns of the matrix "R_d" are transmitted.

(xv) At the receiver, it is assumed that all the coefficients are received. Convert the elements under the column "row" from data type "int16" to double, convert the elements of the run length encoded "rlecol" data from data type "int16" to double and convert the elements under the column "data element" from data type "int8" to double. Then, inverse run length encoding is applied to the elements "rlecol" to obtain the elements under the column "column" of the sparse matrix.

(xvi) Then construct the sparse matrix.

(xvii) Then convert the sparse matrix into full matrix. Let this matrix be "RR".

(xviii) Multiply each element of "RR" by 100.Then add the DC-coefficient to the matrix "RR".

(xix) Apply inverse discrete cosine transform to the matrix "RR" to obtain the reconstructed image.

(xx) Then parameters such as MSE, PSNR and Compression Ratio are evaluated. The mathematical equations for MSE,PSNR and compression ratio are as follows,

$$MSE = \frac{\sum_{i=1}^{i=m}\sum_{j=1}^{j=n}[a(i,j)-b(i,j)]^2}{m\times n} \quad (7)$$

where m=number of rows of the image, n= number of columns of the image, a(i,j)= The element of the original image matrix at the i^{th} row and j^{th} column, b(i,j) is The element of the reconstructed image matrix at the i^{th} row and j^{th} column.

The Peak Signal to Noise Ratio is given by

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \quad (8)$$

Where MSE=Mean Squared Error

The compression ratio is given by

$$CR = \frac{m\times n}{dcomp} \quad (9)$$

Where

$$dcomp = rowele*2 + rlecol*2 + Dataelemnts*1 + dcf + r + c \quad (10)$$

dcomp=number of bytes in the compressed form. rowele =number of elements present in sparse matrix notation under the column "Row". rlecol= number of elements after run length encoding of elements under the column "column" in Sparse matrix notation. Dataelemnts = number of elements present under the column "Data element" in sparse matrix notation.dcf= 8 bytes to accommodate DC coefficient. r= 2bytes to accommodate the number of rows of the input image and c= 2bytes to accommodate the number of columns of the input image. m= number of rows of the input image matrix, n=number of columns of the input image matrix.

(xxi) Then different value of "th" is set and steps from (v) to (xx) are repeated.

The algorithm for the image compression using DCT-SVD method is as follows.

(i) Accept an image of size "m" rows and "n" columns.

(ii) If the accepted image is in any form other than the Gray Scale format convert it into Gray Scale format.

(iii) Apply Discrete Cosine Transform to the Gray Scale format. This results in the matrix "D".

(iv) Collect the DC coefficient of the Discrete Cosine transformed matrix separately.

(v) Apply threshold "th" to the Discrete Cosine Transformed matrix "D". i,e all those coefficients less than "th" in the "D" matrix are neglected. This results in the matrix "D_{th}".

(vi) Apply Singular Value Decomposition to the matrix "D_{th}". This results in three matrices "U", "S" and "V".

(vii) Apply threshold "th_u" = 0.02 to the "U" matrix. i,e all those coefficients less than "th_u " are neglected. This results in the matrix "U_{th}". Apply threshold "th_s=400" to the "S" matrix. i,e all those coefficients less than "th_s " are neglected. This results in the matrix "S_{th}". Similarly, apply threshold "th_v=0.05" to the "V" matrix. i,e all those coefficients less than "th_v " are neglected. This results in the matrix "V_{th}".

(viii) Then multiply the matrices such that R= U_{th}* S_{th}* V_{th}^T.

(ix) Then apply threshold "th" to the matrix "R".i,e all those coefficient less than "th" are neglected in this matrix "R". This results in the matrix "R_{th}".

(x) Divide every element of the matrix "R_{th}" by 100.This results in the matrix "R_d".

(xi) Convert Matrix "R_d" from full matrix form to sparse matrix form. Let this sparse matrix be "S".

(xii) In the sparse matrix form "S", convert the data elements under the column "Row" from data type double to the int16, convert the data elements under the column "column" from data type double to the int16 and convert the data elements under the column "data element" of sparse matrix notation from data type double to the int8.

(xiii) Then, the elements of the column "data elements" of the sparse matrix, the elements of the column "row " of the sparse matrix and the elements of the column "column" of the sparse matrix, DC-coefficient and the number of rows and columns of the Input image matrix are transmitted.

(xiv) At the receiver, it is assumed that all the coefficients are received.

(xv) Convert the elements under the column "row" of the sparse matrix from data type "int16" to double, convert the elements under the column "column" of the sparse matrix from data type "int16" to double and convert the elements under the column "data element" from data type "int8" to double.

(xvi) Then construct the sparse matrix.

(xvii) Then convert the sparse matrix into full matrix. Let this matrix be "RR".

(xviii) Multiply each element of "RR" by 100.Then, add the DC-coefficient to the matrix "RR".

(xix) Apply inverse discrete cosine transform to the matrix "RR" to obtain the reconstructed image.

(xx) Then parameters such as MSE, PSNR and Compression Ratio are evaluated. MSE and PSNR are calculated with formula as explained above, where as compression ratio is calculated by

$$CR = \frac{m \times n}{dcomp} \quad (11)$$

Where

$$dcomp = rowele * 2 + colele * 2 + Dataelemnts * 1 + dcf + r + c \quad (12)$$

dcomp=number of bytes in the compressed form. rowele =number of elements present in sparse matrix notation under the column "Row". colele =number of elements present in sparse matrix notation under the column "Column". Dataelemnts=number of elements present under the column "Data element" in sparse matrix notation. dcf = 8 bytes to accommodate DC coefficient. r = 2bytes to accommodate the number of rows of the input image and c = 2bytes to accommodate the number of columns of the input image. m= number of rows of the input image matrix, n=number of columns of the input image matrix.

(xxi) Then different value of "th" is set and steps from (v) to (xx) are repeated.

The algorithm of the Image compression using DCT-RLE method is as follows.

(i) Accept an image of size "m" rows and "n" columns.

(ii) If the accepted image is in any form other than the Gray Scale format convert it into Gray Scale format.

(iii) Apply Discrete Cosine Transform to the Gray Scale format. This results in the matrix "D".

(iv) Collect the DC coefficient of the Discrete Cosine transformed matrix separately.

(v) Apply threshold "th" to the Discrete Cosine Transformed matrix "D". i,e all those coefficients less than "th" in the "D " matrix are neglected. This results in the matrix "D_{th}".

(vi) Divide every element of the matrix "D_{th}" by 100.This results in the matrix "D_d".

(vii) Convert Matrix "D_d" from full matrix form to sparse matrix form. Let this sparse matrix be "S"

(viii) In the sparse matrix form "S", convert the data elements under the column "Row" from data type double to the int16, convert the data elements under the column "column" from data type double to the int16 and convert the data elements under column "data element" from data type double to the int8.

(ix) After this, apply run length encoding to the data under the column "column" of the sparse matrix. This gives the run length encoded data as "rlecol".

(x) Then, the elements of the column "data elements" of the sparse matrix, the elements of the column "row " of the sparse matrix and the run length encoded data "rlecol", DC-coefficient ,number of rows of Input Image matrix " and the number of columns of the Input Image matrix are transmitted.

At the receiver, it is assumed that all the coefficients are received. Convert the elements under column "row" from data type "int16" to double, convert the element "rlecol" from data type "int16" to double and convert the elements under the column "data element" from data type "int8" to double .First inverse run length encoding is applied to elements "rlecol" to obtain the elements under the column "column" of the sparse matrix.

(xi) Then construct the sparse matrix.

(xii) Then convert the sparse matrix into full matrix. Let this matrix be "RR".

(xiii) Multiply each element of "RR" by 100.Then add the DC-coefficient to the matrix "RR".

(xiv) Apply inverse discrete cosine transform to the matrix "RR" to obtain the reconstructed image.

(xv) Then parameters MSE, PSNR and Compression ratio are evaluated by equation (7), (8) and (9) respectively.

(xvi) Then different value of "th" is set and steps from (v) to (xv) are repeated.

Algorithm of the Image Compression using DCT method is as follows.

(i) Accept an image of size "m" rows and "n" columns.

(ii) If the accepted image is in any form other than the Gray Scale format, convert it into Gray Scale format.

(iii) Apply Discrete Cosine Transform to the Gray Scale format. This results in the matrix "D".

(iv) Collect the DC coefficient of the Discrete Cosine transformed matrix separately.

(v) Apply threshold "th" to the Discrete Cosine Transformed matrix "D".i,e all those coefficients less than "th" in the "D " matrix are neglected. This results in the matrix "D_{th}".

(vi) Divide every element of the matrix "D_{th}" by 100.This results in the matrix "D_d".

(vii) Convert Matrix "D_d" from full matrix form to sparse matrix form. Let this sparse matrix be "S"

(viii) In the sparse matrix form "S", convert the data elements under the column "Row" from data type double to the int16, convert the data elements under the column "column" from data type double to the int16 and convert the data elements under column "data element" from data type double to the int8.

(ix) Then, the elements of the column "data elements" of the sparse matrix, the elements of the column "row " of the sparse matrix and the elements of the column "column" of the sparse matrix, DC-coefficient ,number of rows of Input image matrix and the number of rows and columns of the Input image matrix are transmitted.

(x) At the receiver, it is assumed that all the coefficients are received.

(xi) Convert the elements under the column "row" of the sparse matrix from data type "int16" to double, convert the elements under the column "column" of the sparse matrix from data type "int16" to double and convert the elements under the column "data element" from data type "int8" to double.

(xii) Then construct the sparse matrix.

(xiii) Then convert the sparse matrix into full matrix. Let this matrix be "RR".

(xiv) Multiply each element of "RR" by 100.Then, add the DC-coefficient to the matrix "RR".

(xv) Apply inverse discrete cosine transform to the matrix "RR" to obtain the reconstructed image.

(xvi) Then parameters such as MSE, PSNR and Compression Ratio are evaluated using equations (7), (8) and (11).

(xvii) Then different value of "th" is set and steps from (v) to (xvi) are repeated.

V. RESULTS AND DISCUSSIONS

Experiments are conducted for different set of inputs by considering different resolution and different file formats such as .tiff, .png, .jpg etc**.** A sample of the experimental result is displayed for further discussion and analysis.

The details of the input image and its results are as follows.

Image name: river.jpg
Image size: 425x318

Figure6. Input Image.

Different trails of experimentation in each method are tabulated as follows.

TABLE 1.RESULTS OF THE IMAGE COMPRESSION USING DCT-SVD-RLE METHOD.
th_u=0.02, th_s=400, th_v=0.05

Sl. No.	th	MSE	PSNR (in dB)	Memory (in Bytes)	CR
1	50	187.9812	25.3897	4833	27.9640
2	52	190.5455	25.3308	4532	29.8213
3	54	190.8043	25.3249	4407	30.6671
4	56	195.6602	25.2158	4095	33.0037
5	58	195.5874	25.2174	3948	34.2325

The above table shows the results of the image compression using DCT-SVD-RLE method. "th" is the threshold applied to truncate the Discrete Cosine Transform matrix, "MSE" is the Mean Squared Error. "PSNR" is the Peak Signal to Noise Ratio. "Memory" is the number of bytes of the image in the compressed form. "CR" is the Compression Ratio. " th_u " is the threshold applied for the "U"-Matrix , "th_s" is the threshold applied for the "S"-Matrix , "th_v" is the threshold applied for the "V"-Matrix.

TABLE 2.RESULTS OF THE IMAGE COMPRESSION USING DCT-SVD METHOD.
th_u=0.02, th_s=400,th_v=0.05

Sl. No.	th	MSE	PSNR (in dB)	Memory (in Bytes)	CR
1	50	187.9812	25.3897	7547	17.9078
2	52	190.5455	25.3308	7072	19.1106
3	54	190.8043	25.3249	6857	19.7098
4	56	195.6602	25.2158	6357	21.2600
5	58	195.5874	25.2174	6112	22.1122

The above table shows the results of the image compression using DCT-SVD method. "th" is the threshold applied to truncate the Discrete Cosine Transform matrix, "MSE" is the Mean Squared Error. "PSNR" is the Peak Signal to Noise Ratio. "Memory" is the number of bytes of the image in the compressed form. "CR" is the Compression Ratio. " th_u " is the threshold applied for the "U"-Matrix , "th_s" is the threshold applied for the "S"-Matrix , "th_v" is the threshold applied for the "V"-Matrix.

TABLE 3.RESULTS OF THE IMAGE COMPRESSION USING DCT-RLE METHOD.

Sl. No.	th	MSE	PSNR (in dB)	Memory (in Bytes)	CR
1	50	121.9080	27.2705	8784	15.3859
2	52	122.1413	27.2622	8290	16.3028
3	54	122.7838	27.2394	7846	17.2253
4	56	123.8261	27.2027	7395	18.2759
5	58	125.0919	27.1585	7005	19.2934

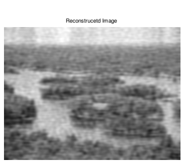
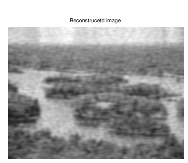
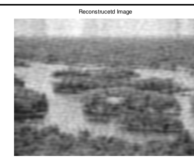
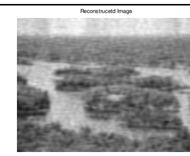
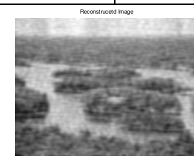
The above table shows the results of the image compression using DCT-RLE method. "th" is the threshold applied to truncate the Discrete Cosine Transform matrix, "MSE" is the Mean Squared Error. "PSNR" is the Peak Signal to Noise Ratio. "Memory" is the number of bytes of the image in the compressed form. "CR" is the Compression Ratio.

TABLE 4.RESULTS OF THE IMAGE COMPRESSION USING DCT METHOD.

Sl. No.	th	MSE	PSNR (in dB)	Memory (in Bytes)	CR
1	50	121.9080	27.2705	13692	9.8707
2	52	122.1413	27.2622	12902	10.4751
3	54	122.7838	27.2394	12182	11.0942
4	56	123.8261	27.2027	11457	11.7963
5	58	125.0919	27.1585	10847	12.4597

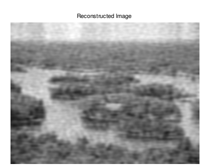
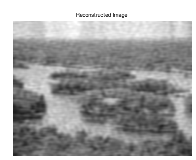
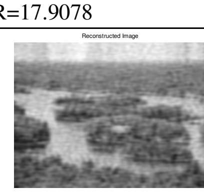
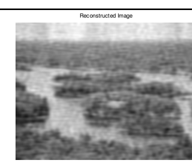
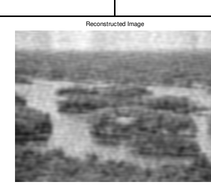
The above table shows the results of the image compression using DCT method. "th" is the threshold applied to truncate the Discrete Cosine Transform matrix, "MSE" is the Mean Squared Error. "PSNR" is the Peak Signal to Noise Ratio. "Memory" is the number of bytes of the image in the compressed form."CR" is the Compression Ratio.

TABLE5 RECONSTRUCTED IMAGES OF THE IMAGE COMPRESSION USING DCT-SVD-RLE METHOD.

 th=50,PSNR=25.3897, CR=27.9640	 th=52,PSNR=25.3308, CR=29.8213
 th=54,PSNR=25.3249, CR=30.6671	 th=56,PSNR=25.2158, CR=33.0037
 th=58,PSNR=25.2174,CR=34.2325	

The above table shows the reconstructed images of the Image Compression using DCT-SVD-RLE method. "th" is the threshold, PSNR= Peak Signal to Noise Ratio and CR=Compression ratio.

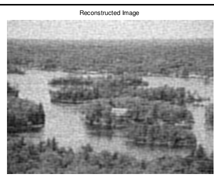
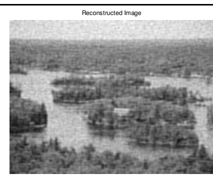
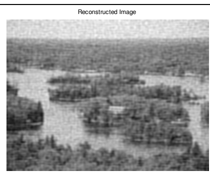
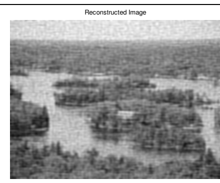
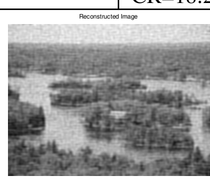
TABLE 6 RECONSTRUCTED IMAGES OF THE IMAGE COMPRESSION USING DCT-SVD METHOD.

 th=50,PSNR=25.3897, CR=17.9078	 th=52,PSNR=25.3308, CR=19.1106
 th=54,PSNR=25.3249, CR=19.7098	 th=56,PSNR=25.2158, CR=21.2600
 th=58,PSNR=25.2174,CR=22.1122	

The above table shows the reconstructed images of the Image Compression using DCT-SVD method.

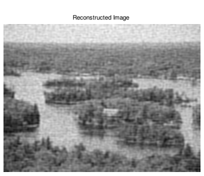
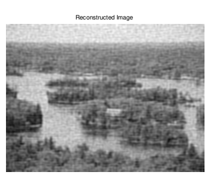
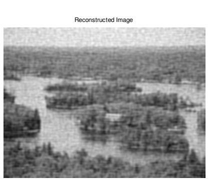
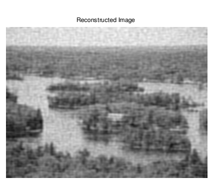
"th" is the threshold, PSNR= Peak Signal to Noise Ratio and CR=Compression ratio.

TABLE7 . RECONSTRUCTED IMAGES OF THE IMAGE COMPRESSION USING DCT- RLE METHOD.

 th=50,PSNR=27.2705, CR=15.3859	 th=52,PSNR=27.2622, CR=16.3028
 th=54,PSNR=27.2394, CR=17.2253	 th=56,PSNR=27.2027, CR=18.2759
 th=58,PSNR=27.1585,CR=19.2934	

The above table Shows the reconstructed images of the Image Compression using DCT-RLE method. "th" is the threshold, PSNR= Peak Signal to Noise Ratio and CR=Compression ratio.

TABLE 8 . RECONSTRUCTED IMAGES OF THE IMAGE COMPRESSION USING DCT METHOD.

 th=50,PSNR=27.2705,CR=9.8707	 th=52,PSNR=27.2622,CR=10.4751
 th=54,PSNR=27.2394,CR=11.0942	 th=56,PSNR=27.2027,CR=11.7963
 th=58,PSNR=27.1585,CR=12.4597	

The above table shows the reconstructed images of the Image Compression using DCT method. "th" is the threshold, PSNR= Peak Signal to Noise Ratio and CR=Compression ratio.

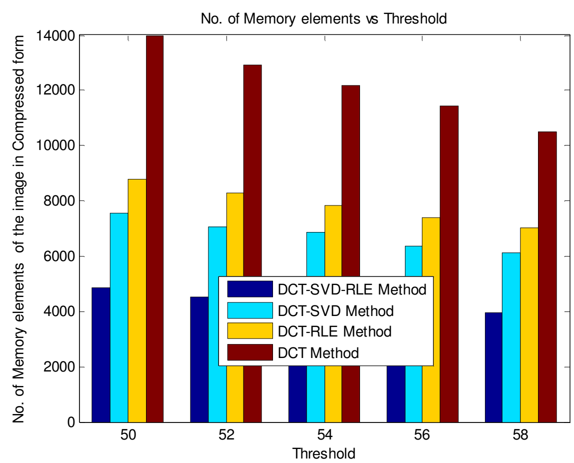


Figure7. Graph of No. Of memory elements Vs Threshold

The above figure shows the graph of No.of memory elements of the image in the compressed form vs Threshold. It can be observed that for the given threshold, the number of memory bytes to be stored in the compressed form or to be transmitted in the DCT-SVD-RLE method is very less compared to other three methods. As the threshold increases number of memory elements to be stored or transmitted decresases.It can be seen that for a threshold 58 the number of memory elements of compressed image is 3948 bytes in DCT-SVD-RLE method, but for the same threshold the number of memory elements of the compressed image are 6112 bytes, 7005 bytes and 10847 bytes in DCT-SVD method, DCT-RLE method and DCT method respectively. Therefore DCT-SVD-RLE method is more efficient.

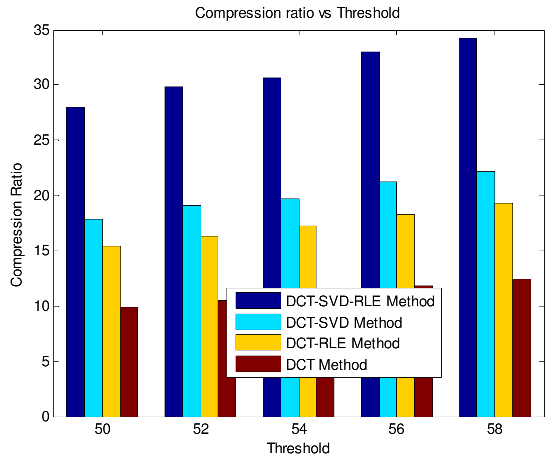


Figure8. Graph of Compression ratio Vs Threshold

The above figure shows the graph of Compression ratio vs Threshold. In the above figure for the given threshold compression ratio in the DCT-SVD-RLE method is more compared to other three methods.As the threshold increases the compression ratio also increases. It can be seen that for a threshold 58 compression ratio is 34.2325 in DCT-SVD-RLE method, but for the same threshold compression ratio are 22.1122, 19.2934 and 12.4597 in DCT-SVD method, DCT-RLE method and DCT method respectively. Here also results shows that DCT-SVD-RLE method is more efficient

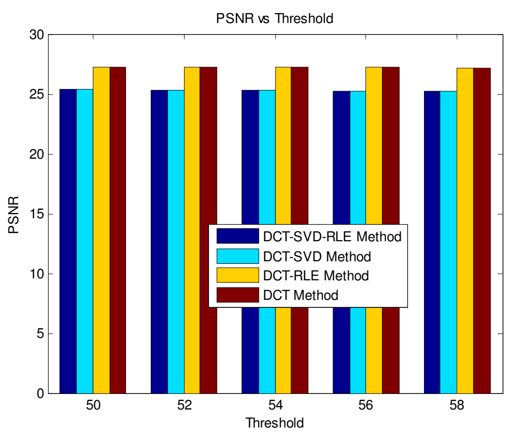


Figure9. Graph of PSNR Vs Threshold

The above figure shows the graph of PSNR vs Threshold. In the above figure it can be seen that as the threshold increase PSNR decreases. PSNR of the DCT-SVD-RLE method is lesser than DCT-RLE method. Since we are able to achieve a good compression Ratio with DCT-SVD-RLE method, the price paid happens to be PSNR.

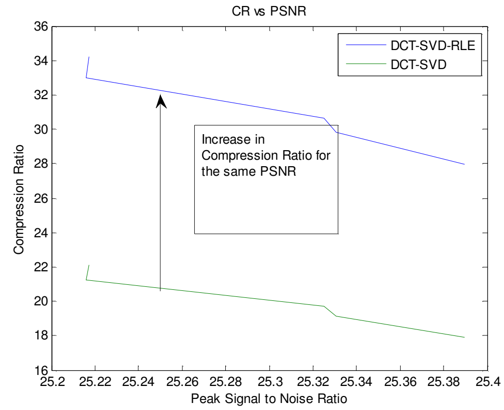


Figure10. Graph of Compression ratio Vs PSNR

In the above figure, graph of Compression ratio Vs PSNR is shown. For the same PSNR, Compression Ratio in DCT-SVD-RLE method is more than the DCT-SVD method.There fore DCT-SVD-RLE method is good.

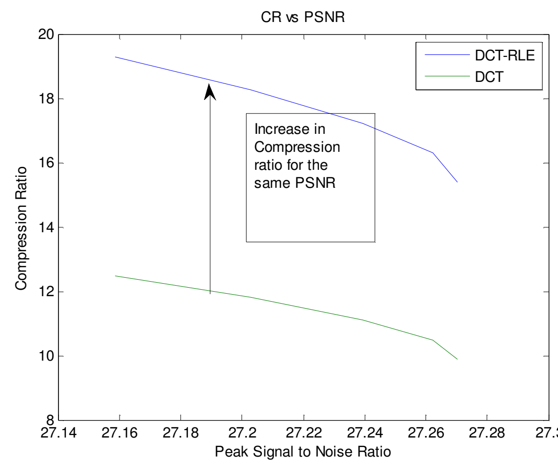


Figure11. Graph of Compression ratio Vs PSNR

In the above figure , graph of Compression ratio vs PSNR is shown.For the same PSNR, Compression Ratio in DCT-RLE method is more than the DCT method.

But the compression ratio of DCT-RLE method is not so good as DCT-SVD-RLE method.

With all this, it can be shown that using DCT-SVD-RLE method good compression ratio can be obtained by loosing PSNR around 2dB.

VI. SCOPE FOR FURTHER ENHANCEMENT

Four methods are compared to achieve a good compression ratio for the fixed threshold. However the compression ratio can also be explored by using other transforms such as wavelet, KLT, Hadamard and slant. The experimentation can be extended by considering different combinations such as DCT-Hadamard, DCT-Wavelets, and DCT-slant.

REFERENCES:

[1] Raghavendra.M.J,Prasantha.H.S and S.Sandya, "DCT SVD Based Hybrid Transform Coding for Image Compression", International Journal of Recent and Innovative Trends in computing and communication. 2015

[2] Prasanth.H.S,Shashidhara.H.L and Balasubramanyamurthy.K.N, "Image compression using SVD" Intenational Conference on Computational Intelligence and Multimedia applications, Vol.3,2007

[3] S.Sridhar,P.Rajeshkumar and K.V.Ramanaiah ," Wavelet Transform Techniques for Image Compression-An evaluation " International Journal of Image,Graphics and Signal Processing,2014

[4] T.D.Khadatre, Mayuri Chaudari, Sushma B, and Yogita Raut, "A combined novel approach for Image Compression using Vector Quantization and wavelet transform" International Journal of Application or Innovation in Engineering & Management.Vol.3,April 2014

[5] Athira.M.S. and V.Kalaichelvi ,"An Intelligent Technique for Image compression " International Journal of Recent Developments in Engineering and technology ,June 2014

[6] Pallavi.M.Sune and Vijay.K.Shandilya, "Image Compression Techniques based on Wavelet and Huffman coding " International Journal of Advanced Research in Computer Science and Software Engineering, April 2013

[7] E.Praveenkumar and M.G.sumithra, "Medical Image Compression using Integer Multi wavelets Transform for Telemedicine Applications " International Journal of Engineering and Computer Science ,May 2013

[8] D.Vishnuvardhan, Sreenivasan.B and I.Suneetha. "Advanced Digital Image compression Technique using curvelet Transform" International Journal of Engineering Research and Applications Vol.3,Issue-4,Aug 2013

[9] Birendrakumar Patel, Suyesh Agrawal, "Image Compression Techniques using artificial neural networks" International Journal of Advanced Research in Computer Engineering & Technology,Vol.2,October 2013

[10] Sumegha yadav, Tarun kumar.R. "Transform Based Hybrid Image Compression Techniques in conjunction with Fractal Image compression scheme" International Journal of Advancements in Rsearch & Technology,Volume 1,Issue 4 Aril 2013.

[11] Rowayda A.Sadek , "SVD Based Image Processing Applications:State of the Art,Contributions and Research challenges", International Journal of Advanced Computer Science and Applications.Vol 3,2012

[12] K.R.Rao ,Ahmed.N,Natarajan.T , "Discrete Cosine Transform*",* IEEE Transaction on Computers,1974

AUTHOR'S PROFILE

Raghavendra.M.J obtained his Bachelor degree from Mysore University and Master Degree from NITK,Suratkal.His reserch interest includes Multimedia and Signal Processing.He is persuing research program in V.T.U.He is currently working as an Assistant professor in the Department of Telecommunication Engineering,PES Institute of Technology, Bangalore.

Dr.Prasantha.H.S received Bachelor degree from Bangalore University, Master Degree from V.T.U, Belgaum, and Ph.D from Anna University, Chennai, in the area of Multimedia and Image Processing. He has 16+ years of teaching and research experience. His research interest includes Multimedia and Signal Processing. He is currently guiding students for their research program in V.T.U and other university. Currently, he is working as a Professor in the department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bangalore.

Dr.S.Sandya obtained her Ph.D from Indian Institute of Science, Bangalore. She has vast experience in the field of industry, research and teaching. Her research interest includes Satellite communications, Wireless Sensor Networks and Embedded Systems. She is currently guiding students for their research program under V.T.U and other university. Currently, she is working as a Professor and Head of Electronics and Communication Engineering department, Nitte Meenakshi Institute of Technology, Bangalore.