An Amelloriation Technique of Imaging Compression on Crossbreeding Wavelets using Cosine Transformation

Monika Tomer[1], Hariom Tyagi[2]

[1]Research Scholar, [2]Associate Professor, CSE Department, R.D Engineering College, Ghaziabad, India

[1]mtomer01@gmail.com, [2]hariom2678@gmail.com

Abstract: Image compression is a science that is concerned with the reduction of the number of bits required to store, transmit and reconstruct the image without any loss of information. Though various transforms are used in the field of image compression. Compression technique are proposing with the help of cosine transformation and wavelets transformation with 'db7','db7' and 'db8'.And calculate the compression ratio using crossbreed architecture and wavelets 'db7','db7' and 'db8'.In this thesis "An Amelloriation technique of imaging compression on crossbreeding wavelets using cosine transformation" proposed algorithm is implemented in MAT LAB 2010.Proposed algorithm has been successfully programmed using MAT LAB and tested.

Keywords: Image Compression, PSNR, Crossbreed wavelets, Wavelets family, MAT LAB.

I. INTRODUCTION

Image compression is the application of Data compression on digital images. The discrete cosine transform (dct) is a technique for converting a signal into elementary frequency components. It is widely used in image compression. Here we develop some simple functions to compute the dct and to compress images. An image compression algorithm was comprehended using Matlab code, and modified to perform better when implemented in hardware description language. Compression is useful because it helps reduce the consumption of expensive resources such as hard disk space or transmission bandwidth. In this paper, we briefly introduce the fundamental theory of image compression in chapter 1, two typical standards - JPEG and JPEG 2000 will be described. Finally, the newly proposed image compression algorithm. Proposed algorithm has been successfully programmed using MAT LAB and tested.

The purpose of this literature review was to explore the research conducted to date on the use of compression in MATLAB. To meet this goal, four ideas were developed to systematically capture data not only from the literature, but also from acknowledged experts in clinical radiology, and related professional administrations, for their opinion as well. In general, the results of these studies show that image types in digital radiology are different based on their mode of generation, as well as their spatial and contrast resolution, determined by their matrix size/pixel size, and bit depth respectively. Furthermore, there are numerous forms of irreversible compression algorithms, and they are not all equal in terms of

performance. Additionally, of the three evaluation methods used to measure observer performance on compressed images, the ROC methodology is most commonly used.

II. PROBLEM STATEMENT

As a user of graphics file conversion applications we have been interested in this field since our interest in computing began. Our own experiences of using images for compression for course-work has led us to think about many questions as to why there are so many algorithms and methods for storing and transferring images in a compressed form. This project has given us the opportunity to find out the answers to our questions. Our knowledge of this field at the start of the project was informal. We knew generally about Softwares available on internet for compressing images without knowing anything specific about the formats, compression techniques and overall structure of the images we were using. As this is a subject we are interested in, working on image compression and decompression and how this can be affected by implementing different methods on images seemed a choice of study which we knew would be both difficult and interesting. The learning curve embarked on has been considerably steeper than previous work we have undertaken. The software component constitutes our first true software development culminating in a final product. Our previous knowledge of the MATLAB did not outfit for the scale of this work, and our skills in other languages were only of a basic level. Through the development we have learnt everything necessary about this language and how they can be applied to creating image compression software. From the theory aspect, we have done much research into the principles of images and its related areas including compression and decompression, image displaying, and some advanced techniques used to enhance compression ratios.

III. IMPLEMENTATION WORK

3.1. Implementation and Algorithm:

In the imaging context, the pixel or block of pixel is considered as a set of symbol. A code, on the other hand, is sequence of symbols or number that is used to represent information. A string of codes is called a code word. The whole data compression process can now we visualized as a mapping of all possible sequence of symbols of a message (of N1 bits), separately or in a file, to a set of codes, separately or infield, using N2

bits. The compression ratio (Cr) is then defined as N1/N2 and the relative redundancy id defined as

$$Rd = 1-1/Cr$$

Now these scenarios emerge. In scenario one, N2=N1, this means that the compression ratio is 1 and relative redundant is 1-1/1=0. This indicates that there is no redundancy in the image. The input message is reproduces exactly.

3.2. Transformation Based Compression:

Transform compression is based on a simple premise: when the signal is passed through the Fourier (or other) transform like DFT, DCT etc. The resulting data values will no longer be equal in their information carrying roles. In particular, the small frequency components of a signal are more important than the high frequency components. Removing 50% of the bits from the high frequency components might remove, say, only 5% of the encoded information.
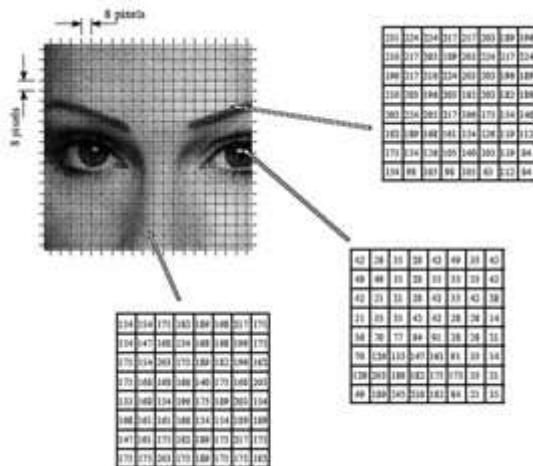

Fig. 3.1. JPEG Transformation Compression Starts by Breaking into 8x8 Groups Each Contain 64 Pixels

As shown in Fig 3.1, JPEG compression starts by breaking the image into 8×8 pixel groups. The full Joint Photographic Experts Group algorithm can accept a wide range of bits per pixel, together with the use of color information. In the above example, each pixel is a single byte, a greyscale value between 0 and 255. These 8 into 8 pixel groups are treated independently for the period of compression. That is, every group is represented by 64 bytes. After transforming and eliminate data, each group is represented by, 2 to 20 bytes. During uncompress ion, the inverse transform is taken of the 2 to 20 bytes to create an approximation of the original 8×8 group. These approximated groups are then fitted together to form the uncompressed image. Why use 8×8 pixel groups instead of, for instance, 16×16? The 8×8 grouping was based on the maximum size that integrated circuit technology could handle at the time the standard was developed. In any event, the

8×8 size works well, and it may be possible to change in the future.

3.3. Discrete Wavelet Transform:

The Discrete Wavelet Transform, which is based on sub-band coding, is found to give up a fast computation of Wavelet Transform. It is easy to execute and reduces the computation time and resources mandatory. The discrete wavelet transform (DWT) uses filter banks for the construction of the multire solution time-frequency plane. The Discrete Wavelet Transform (DWT) analyses the signal at dissimilar frequency bands with different resolutions by decomposing the signal into an approximation and aspect information. The decomposition of the signal into different frequency bands obtained by successive high pass g[n] and low pass h[n] filtering of the time domain signal. The combination of high pass g[n] and low pass filter h[n] comprise a pair of analyzing filters The output of each filter contains half the frequency content, however an equal amount of samples as the input signal. The two outputs together hold the same frequency content as the input signal; however the amount of data is doubled. So down sampling by a factor two, denoted by 2, is applied to the outputs of the filters in the analysis bank. Reconstruction of the original signal is possible using the synthesis filter bank. In synthesis bank the signals are up sampled and passed through the filters g[n] and h[n]. The filters in the synthesis bank are based on the filters in the analysis bank. The right choice of the combination of the analyzing filters and synthesizing filters will provide perfect reconstruction (PR). The Perfect reconstruction is defined by the output which is generally an estimate of the input, being exactly equal to the input applied. Decomposition process can be iterated with successive approximations being decomposed in return, therefore that one signal is broken down into many minor resolution components. The Decomposition can be performed as ones requirement.
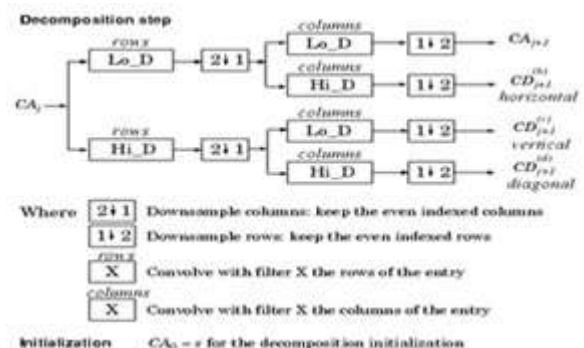

Fig. 3.2. Two Dimensional DWT

3.4 Inverse Discrete Wavelets Transform:

The Inverse Discrete Wavelet Transform (IDWT) block computes the IDWT of the input sub bands. By default, the block accepts a particular sample-based vector or

matrix of concatenated sub bands. The output of IDWT is frame based, and has the similar dimensions as the input [30]. Every column of the output is the inverse discrete wavelet transform of the corresponding input column.
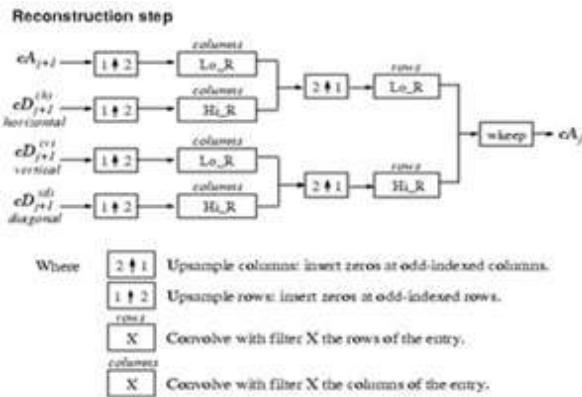
3.5. Algorithm:

3.5.1. Compressed Algorithms:

Step1: Take the input image (e.g 5.bmp).

Step2: using wavelets transformation decomposition at second level (db6.db7, db8)

Step 3: Find out the block 8x8 after step 2and applied cosine transformation for each block.

Step4: apply quantization and sampling process for each block.

Step5: Save the compressed image into 5_com.NIT

3.5.2. Decompressed Algorithms:

Step 1: read the compressed image (5_com.NIT)

Step2: Apply de sampling and de quantization for each block.

Step 3: apply inverse cosine transformation.

Step 4: then combined the block and apply inverse wavelets transformation.

Step 5: Save the decompressed image into 5_dcom.bmp

3.6. Compression with 'db6':

Algorithm for hybrid methodology for compression has been finalized with this architecture wavelets **'db6'** are using for compression aspects the image and calculate the compression ratio with different input image 5.bmp,6.bmp,8.bmp and 9.bmp. and results in term of size are shown in table 1.

Result for Wavelets 'db6':

Table 3.1. Compression Details with 'db6'

S.N.	Name of Image	Dimension	Size Before Compression	After Compression
1	5.bmp	1024x768	2.54mb	100KB

2	6.bmp	1440 x900	3.70MB	118KB
3	8.bmp	800 x600	1.37MB	25.1KB
4	9.bmp	320 x240	225KB	13.8KB

Fig. 3.4. Input Image Before Compression

3.6.1. Compression Ratio with 'db6':

An image 5.bmp is 2.54 MB before compression and 100Kb after compression what are the values of compression ratio and saving parentage.

Compression Ratio = $(2.54 \times 1024)/100 = 26/1$

The interpretation is that 26 pixels of input image are expressed as 1 pixel in the output image.

Saving Percentage = $1-1/26 = 96.15 \%$

The interpretation is that 26 pixels of input image are expressed as 1 pixel in the output image.

Saving Percentage = $1-1/26 = 96.15 \%$

An image 6.bmp is 3.70MB before compression and 118Kb after compression what are the values of compression ratio and saving parentage.

Compression Ratio = $(3.70 \times 1024)/118 = 32.10/1$

The interpretation is that 32 pixels of input image are expressed as 1 pixel in the output image.

Saving Percentage = $1-1/32.10 = 96.88 \%$

An image 8.bmp is 1.37 MB before compression and 25.1 Kb after compression what are the values of compression ratio and saving parentage.

Compression Ratio = $(1.37 \times 1024)/25.1 = 55.89/1$

The interpretation is that 56 pixels of input image are expressed as 1 pixel in the output image.

Saving Percentage = $1-1/55.89 = 98.21 \%$

An image 9.bmp is 225KB before compression and 13.8 Kb after compression what are the values of compression ratio and saving parentage.

Compression Ratio = (225)/13.8 = 16.30/1

The interpretation is that 16.30 pixels of input image are expressed as 1 pixel in the output image.

Saving Percentage = 1-1/16.30 = 93.86 %

Table 3.2. Result of Compression Ratio with 'db6'

Input Image	Compression Ratio
5.bmp	96.15%
6.bmp	96.88%
8.bmp	98.21%
9.bmp	93.86%

Average compression ratio for 'db6' for the given input image: 96.27%.

3.6.2. Decompression with Wavelets 'db6':

Reverser algorithm e for hybrid methodology for decompression has been finalized with this architecture wavelets „db6" are using for compression the image and calculate the compression ratio with different input image 5_COM_DB6.NIT, 6_COM_DB6.NIT, 8_COM_DB6.NIT and 9_COM_DB6.NIT, and results in term of size(in KB or MB) and dimension are shown in table .

Table 3.3. Decompression Details with 'db6'

S.N.	Name of Input Image	Size	Size After Decompression	Dimension	Name of Decompressed Image
1	5_COM_DB6.NIT	100KB	2.26MB	1026x770	5_DCOM_DB6.BMP
2	6_COM_DB6.NIT	118KB	3.72MB	1442x902	6_DCOM_DB6.BMP
3	8_COM_DB6.NIT	25.1KB	1.38MB	802 x602	8_DCOM_DB6.BMP
4	9_COM_DB6.NIT	13.8KB	228KB	322 x242	9_DCOM_DB6.BMP

3.7. Compression with 'db7':

Algorithm for hybrid methodology for compression has been finalized with this architecture.

Wavelets 'db7' are using for compression aspects the image and calculate the compression ratio with different input image.5.bmp, 6.bmp, 8.bmp and 9.bmp. and results in term of size are shown in table.

Table 3.4. Compression Details with 'db7'

S.N.	Name of Image	Dimension	Size Before Compression	After Compression
1	5.bmp	1024x768	2.54mb	116KB
2	6.bmp	1440 x900	3.70MB	116KB
3	8.bmp	800 x600	1.37MB	25.5KB
4	9.bmp	320 x240	225KB	15.7kb

3.7.1 Compression ratio with 'db7':

An image 5.bmp is 2.54 MB before compression and 116Kb after compression what are the values of compression ratio and saving parentage.

Compression Ratio = (2.54×1024)/116 = 22.42/1

The interpretation is that 22.42 pixels of input image are expressed as 1 pixel in the output image.

Saving Percentage = 1-1/22.42 = 95.53 %

An image 6.bmp is 3.70MB before compression and 116Kb after compression what are the values of compression ratio and saving parentage.

Compression Ratio = (3.70×1024)/116 = 32.66/1

The interpretation is that 32.66 pixels of input image are expressed as 1 pixel in the output image.

Saving Percentage = 1-1/32.66 = 96.93 %

An image 8.bmp is 1.37 MB before compression and 25.5 Kb after compression what are the values of compression ratio and saving parentage.

Compression Ratio = (1.37×1024)/25.5 = 55.01/1

The interpretation is that 55 pixels of input image are expressed as 1 pixel in the output image.

Saving Percentage = 1-1/55.01 = 98.18 %

An image 9.bmp is 225KB before compression and 15.7 Kb after compression what are the values of compression ratio and saving parentage.

Compression Ratio = (225)/15.7 = 14.33/1

The interpretation is that 14.33 pixels of input image are expressed as 1 pixel in the output image.

Saving Percentage = 1-1/14.33 = 93.02 %

Result for wavelets 'db7':

Table 3.5. Result of Compression Ratio with 'db7'

Input Image	Compression ratio
5.bmp	95.53%
6.bmp	96.93%
8.bmp	98.18%

Average compression ratio for 'db7' for the given input image: 95.91%

3.7.2. Decompression with 'db7':

Reverser algorithm e for hybrid methodology for decompression has been finalized with this architecture wavelets „db6" are using for compression the image and calculate the compression ratio with different input image 5_COM_DB776.NIT, 6_COM_DB7.NIT, 8_COM_DB7.NIT and 9_COM_DB7.NIT, and results in term of size (in KB or MB) and dimension are shown in table.

Table 3.6. Decompression Details with 'db7'

S.N.	Name of Input Image	Size	Size After Decompression	Dimension	Name of Decompressed Image
1	5_COM_DB7.NIT	100KB	2.25MB	1024x768	5_DCOM_DB7.BMP
2	6_COM_DB7.NIT	118KB	3.70MB	1440x900	6_DCOM_DB7.BMP
3	8_COM_DB7.NIT	25.1KB	1.37MB	800x600	8_DCOM_DB7.BMP
4	9_COM_DB7.NIT	13.8KB	225KB	320x240	9_DCOM_DB7.BMP

3.8. Compression with 'db8':

Algorithm for hybrid methodology for compression has been finalized with this architecture wavelets **'db8'** are using for compression aspects the image and calculate the compression ratio with different input image 5.bmp,6.bmp,8.bmp and 9.bmp.

Table 3.7. Compression Details with 'db8'

S.N.	Name of Image	Dimension	Size Before Compression	After Compression
1	5.bmp	1024x768	2.54mb	116KB
2	6.bmp	1440 x900	3.70MB	126KB
3	8.bmp	800 x600	1.37MB	24.9KB
4	9.bmp	320 x240	225KB	14.9KB

3.8.1. Compression Ratio with 'db8':

An image 5.bmp is 2.54 MB before compression and 116Kb after compression what are the values of compression ratio and saving parentage.

Compression Ratio = $(2.54 \times 1024)/116 = 22.42/1$

The interpretation is that 22.42 pixels of input image are expressed as 1 pixel in the output image.

Saving Percentage = $1-1/22.42 = 95.53$ %

An image 6.bmp is 3.70MB before compression and 126Kb after compression what are the values of compression ratio and saving parentage.

Compression Ratio = $(3.70 \times 1024)/126 = 30.06/1$

The interpretation is that 32.66 pixels of input image are expressed as 1 pixel in the output image.

Saving Percentage = $1-1/30.06 = 96.67$ %

An image 8.bmp is 1.37 MB before compression and 24.9 Kb after compression what are the values of compression ratio and saving parentage.

Compression Ratio = $(1.37 \times 1024)/24.9 = 56.34/1$

The interpretations is that 55 Pixels of input image are expressed as 1 Pixel in the output image.

Saving Percentage = $1-1/56.34 = 98.22$%

An image 9.bmp is 225KB before compression and 14.9 Kb after compression what are the values of compression ratio and saving parentage.

Compression Ratio = $(225)/14.9 = 15.10/1$

The interpretation is that 14.33 pixels of input image are expressed as 1 pixel in the output image.

Saving Percentage = $1-1/15.10 = 93.37$ %

Result for wavelets 'db8':

Table 3.8. Result of Compression Ratio with 'db8'

Input Image	Compression ratio
5.bmp	95.53%
6.bmp	96.67%
8.bmp	98.22%
9.bmp	93.37%

Average compression ratio for 'db8' for the given input image: 95.54%

3.8.2. Decompression with 'db8':

Reverser algorithm e for hybrid methodology for decompression has been finalized with this „db6" are using for compression the image and calculate the compression ratio with different input image 5_COM_DB776.NIT, 6_COM_DB7.NIT, 8_COM_DB7.NIT and 9_COM_DB7.NIT, and results in term of size(in KB or MB) and dimension are shown in table.

Table 3.9. Decompression Details with 'db8'

S.N	Name of Input Image	Size	Size After Decompression	Dimension	Name of Decompressed Image

1	5_COM_DB8.NIT	116KB	2.26MB	1026x770	5_DCOM_DB8.BMP
2	6_COM_DB8.NIT	126KB	3.72MB	1442x902	6_DCOM_DB8.BMP
3	8_COM_DB8.NIT	24.9KB	1.38MB	802x602	8_DCOM_DB8.BMP
4	9_COM_DB8.NIT	14.9KB	228KB	322x242	9_DCOM_DB8.BMP

3.9 Comparison Between Wavelets and Compression Ratio:

Table 3.10. Comparison with Different Wavelets

Wavelets	Compression Ratio
Db6	96.27
Db7	95.91
Db8	95.54

Result: db6 wavelets has maximum compression ratio.

There are different approaches to image quality evaluation and they are based on objective and subjective parameters. The quality of a compressed image is evaluated by analyzing the difference between the original and the compressed one.

One of the mainly extensively used parameters for the evaluation of image quality is the MSE. The list of Image Quality measures implemented in this package include.

3.10. Functional Parameters:

3.10.1. Structural Content (SC):

The structural content measure used to compare two images in a number of small image patches the images have in common. The patches to be compared are selected using 2D continuous wavelet which acts as a low level corner detector. The large value of structural content SC means that image is poor quality.

$$SC = \sum_{j=1}^{M} \sum_{k=1}^{N} x_j k^2 \Big/ \sum_{j=1}^{M} \sum_{k=1}^{N} x_j^i k^2$$

3.10.2. Mean Square Error (MSE):

Mean square error is a measure of image quality index. The large value of mean square means that image is a poor quality. MSE means „Mean Square Error". It is represent the classical error estimate given by the equation:

Where M and N are the image dimensions. It is a widely used criterion but is often not enough representatives. JND and Hosaka Plots may be used as an alternative to MSE.

$$MSE = \frac{1}{MN} \sum_{j=1}^{M} \sum_{k=1}^{N} \left(x_{j,k} - x'_{j,k} \right)^2$$

3.10.3. Peak Signal to Noise Ratio (PSNR in dB):

The PSNR can give an approximate index of image quality, but by itself it cannot build a comparison between the qualities of two different images [12]. It is likely, indeed, that an image with a lower PSNR might be perceived as an image of better quality compared to one with a higher signal to noise ratio.

$$PSNR = 10 \log \frac{(2^n - 1)^2}{MSE} = 10 \log \frac{255^2}{MSE}$$

PSNR is an engineering term for the ratio between the at most possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. For the reason that many signals have a very wide dynamic range, Peak Signal to Noise Ratio is usually expressed in terms of the logarithmic decibel scale.

PSNR is most commonly used to measure the quality of reconstruction of lossy compression codecs (e.g., for image compression). The signal in this case is the original information, and the noise is the error introduced by compression. When evaluate compression codecs, PSNR is a near to human perception of reconstruction quality. Although a higher Peak Signal to Noise Ratio generally indicates that the reconstruction is of higher quality, in some cases it may not. One has to be extremely careful with the range of validity of this metric; it is only conclusively valid when it is used to compare results from the same codec (or codec type) and same content.

3.10.4. Normalized Cross-Correlation (NCC):

Normalized cross correlation is the simplest but effective method as a similarity compute, which is invariant to linear brightness and contrast variations. Its simple hardware implementation makes it useful for real-time applications. Or normalized cross correlation is a measure of similarity of two waveforms as a function of the time lag applied to one of them [28]. The cross correlation is like in nature to the convolution of two functions

$$NK = \sum_{j=1}^{M} \sum_{k=1}^{N} x_{j,k} \cdot x'_{j,k} \Big/ \sum_{j=1}^{M} \sum_{k=1}^{N} x_{j,k}^2$$

3.10.5. Average Difference (AD):

$$AD = \sum_{j=1}^{M} \sum_{k=1}^{N} \left(x_{j,k} - x'_{j,k} \right) \Big/ MN$$

3.10.6. Maximum Difference (MD):

Difference between any two pixels such that the larger pixel appears after the smallest pixel. The large value

of maximum difference means that image is poor in quality

$$MD = Max \left(\left| x_{j,k} - x'_{j,k} \right| \right)$$

3.10.7. Normalized Absolute Error (NAE):

The large value of normalized absolute error means that image is poor quality. Normalized Absolute Error is defined as follows.

$$NAE = \sum_{j=1}^{M} \sum_{k=1}^{N} \left| x_{j,k} - x'_{j,k} \right| \Big/ \sum_{j=1}^{M} \sum_{k=1}^{N} \left| x_{j,k} \right|$$

IV. EXPERIMENT & RESULT ANALYSIS

The Experiment presented in this thesis shows crossbreed wavelet using cosine transformation based "An Amelloriation technique of imaging compression on crossbreeding wavelets using cosine transformation". In this thesis there are different approaches to image quality evaluation and they are based on objective and subjective parameters. The quality of a compressed image is evaluated by analyzing the difference between the original and the compressed one.

The following summarizes the varying factors in the experiments. Here we supply MATLAB for the implementing our work. MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by Maths Works, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and FORTRAN. M ATLAB is widely used in academic and research institutions as well as industrial enterprises. Cleve Moler, the chairman of the computer science department at the University of New Mexico, started developing MATLAB in the late 1970s.

V. CONCLUSIONS

In this paper, the algorithm has been successfully programmed using MATLAB and tested. We were able to successfully implement the transformation. Our MATLAB implementation was somewhat simple to write and run. The result shows that the higher data redundancy helps to achieve more compression. Experimental results show good compression ratio for the image is obtained hence we conclude that hybrid algorithm is efficient technique for image compression and decompression to some extent. Using above result conclude that the maximum compression ratio is 96.27%, but compromise with little bit in quality, but quality is more important than the size then we should use „db7‟ wavelets because it has less compression ratio i.e. 95.91 % but quality parameter values is very good.

As the future work on compression of video for storing and transmitting images can be done by other methods of compression because as we have concluded above, the result of the decompressed image is almost same as that of the input image so that indicates that there is no loss of information during transmission. So other methods of image compression can be carried out as namely JPEG method, Entropy coding, Huffman entropy etc.

Future improvements to our algorithm that we considered are Video Compression. Through this we were able to gain valuable hands on experience with the MATLAB programming procedure.

ACKNOWLEDGMENT

VI. REFERENCES

[1] Anna Saro Vijendran, Vidhya.B, "A Hybrid Image Compression Technique Using Wavelet Transformation - MFOCPN and Interpolation" Global Journal of Computer Science and Technology Volume 11 Issue Version 1.0 March 2011.

[2] Syed Ali Khayam, "The Discrete Cosine Transform (DCT): Theory and Application" Department of Electrical & Computer Engineering, Michigan State University ,March 10th 2003.

[3] Guy E. Blelloch, "Introduction to Data Compression", Computer Science Department, Carnegie Mellon University September 25, 2010.

[4] Kamrul Hasan Talukder and Koichi Harada, "Discrete Wavelet Transform for Image Compression and A Model of Parallel Image Compression Scheme for Formal Verification", Proceedings of the World Congress on Engineering 2007 Vol I WCE 2007, July 2 - 4, 2007, London, U. K

[5] Andrew B. Watson, NASA Ames Research, "Image Compression Using the Discrete Cosine Transform", Mathematica Journal, 4(1),1994, p. 81-88.

[6] Lotfi, A. A. ; Hazrati, M. M. ; Sharei, M. ; Saeb, A. ; "CDF (2, 2) wavelet lossy iamge compression on primitive FPGA (XC9572)"; International

Symposium on Signals, Circuits and Systems, ISSCS, 2005. Volume: 2, Page(s): 445 - 448. .

[7] Nikolay N. Ponomarenko, Vladimir V. Lukin, Karen Egiazarian, Jaakko Astola. DCT Based High Quality Image Compression. In Proceedings of SCIA'2005. pp. 1177~1185

[8] Karen Lees "Image compression using Wavelets", Report of M. S. 2002

[9] Saeid Saryazdi and Mehrnaz Demehr(2005),"A blind DCT domain digital watermarking ",Proceeding of 3rd International Conference: SETIT ,Tunisia:march-2005

[10] Nopparat Pantaesaena, M. Sangworaisl, C. Nantajiwakornchai and T. Phanpraist, "Image compression using vector quantization", ReCCIT, Thialand,2005

[11] Greg Ames,"Image Compression", Dec 07, 2002.

[12] M. Atonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," IEEE Trans. Image Processing, vol. 1, pp. 205–220, Apr. 1992.

[13] R. L. De Valois and K. K. De Valois, Spatial vision, Oxford University Press, 1988.

[14] H. S. Malvar, Signal Processing With Lapped Transforms. Norwood, MA: Artech House, 1992.

[15] Wallace, G.. The JPEG still picture compression standard. Communications of the ACM 34(4): 30-44. 1991.

[16] Liu Chien-Chih, Hang Hsueh-Ming, "Acceleration and Implementation of JPEG 2000 Encoder on TI DSP platform" Image Processing, 2007. ICIP 2007. IEEE International Conference on, Vo1. 3, pp. III-329-339, 2005.

[17] Ken cabeen and Peter Gent, "Image Compression and the Discrete Cosine Transform" Math 45,College of the Redwoods.

[18] Maneesha Gupta, Dr. Amit Kumar Garg "Analysis Of Image Compression Algorithm Using DCT" International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 2, Issue 1, Jan-Feb 2012,pp.515-521

[19] Rafael C. Gonzalez, and Richard E. Woods, Digital image processing, Pearson-Prentice Hall,2005.

[20] H. S. Malvar, "Fast progressive image coding without wavelets," in Proc. 2000 Data Compression Conf., Mar. 2000, pp. 243–252 .