IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727,
PP 47-57
www.iosrjournals.org

Image Compression Supported By Encryption Using Unitary Transform

Arathy Nair[1], Sreejith S[2]
[1](M.Tech Scholar, Department of CSE, LBS Institute of Technology for Women, Thiruvananthapuram, India)
[2](Assistant Professor, Department of CSE, LBS Institute of Technology for Women, Thiruvananthapuram, India)

ABSTRACT: *The purpose of image compression and the image steganography are antagonist.. Even if by definition they are contradictory, they can be combined to improve the security as well as the image compression rate can be increased. The main focus is on the improvement in security and image compression rate of JPEG images using unitary transform. Here an encryption technique by applying unitary transforms according to the encryption key generated from random permutation method at the transformation stage can be used to provide better security. After that the data compression is performed twice; first by using conventional standard JPEG, by taking the advantage of energy compaction using JPEG to reduce redundant data and then by means of steganography which embeds some bit-blocks within its subsequent blocks of the same image. Then after decompression the inverse of the unitary transform can be applied. The two algorithms UNI_STEGO_JPEG encoder and UNI_STEGO_JPEG decoder is introduced to achieve this. The embedded blocks do not increase the file size of the compressed image, but as they are taken from and hidden within the image itself, the file size will be further decreased. Also the original image is encrypted by the unitary transform results in a rotation of the original image and it is compressed This improves the performance by providing better security and image compression, so that it can be used for the services over networks in a secured manner. Experimental results show for this promising technique to have wide potential in image coding.*

Keywords - *DWT, Image compression, JPEG, Steganography, Unitary transform*

I. INTRODUCTION

The image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. It is a technique to reduce irrelevance and redundancy of image data in-order to store or transmit data in an efficient form. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from web pages. A text file or program can be compressed without the introduction of errors, but only up to a certain extent, so no loss of the data. This is called lossless compression. Beyond this point, errors are introduced. In text and program files, it is crucial that compression be lossless because a single error can seriously damage the meaning of a text file, or cause a program not to run. In image compression, a small loss in quality is usually not noticeable. There is no critical point up to which the compression works perfectly, but beyond which it becomes impossible. When there is some tolerance for loss, the compression factor can be greater than it can when there is no loss tolerance. For this reason, graphic images can be compressed more than text files or programs [10]. Lossy compression is a type of compression in which the information is not fully preserved but they have high compression ratio. The quality of the compressed, and subsequently decompressed, data should be as good as possible. Steganography is the art of science of encoding hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of message. The focus of this paper is the improvement of image compression through steganography. Figure 1.1 shows the block diagram of the compression and decompression of an image.

With the advent of the Internet and the need for digital right management systems, Steganography has received a particular interest and wide range of applications, especially when it is used in conjunction with cryptography. When these techniques are combined, the secret data is doubly sheltered; first it is encrypted and then embedded within the target support. There are a number of usual and unusual applications of steganography. The goal of the image compression is to remove the redundancies for minimizing the number of bits required to represent an image while steganography works by embedding the secret data in redundancies of the image in nvisibility manner. One possible unusual application of steganography is image compression, which is the focus of this paper.

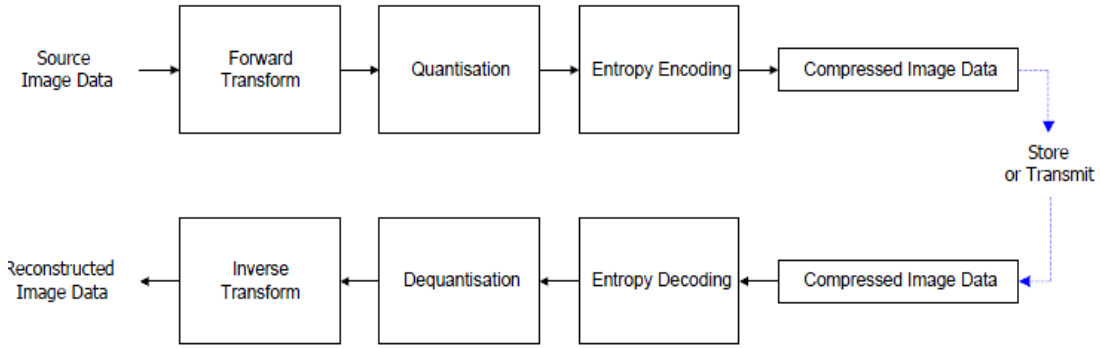

Fig 1: Block diagrams of the JPEG2000 [3]

In the previous works the lossy JPEG compression is based on DCT [2] and DWT [1] and then apply the steganographic method to hide selected compressed blocks into the subsequent blocks. These methods provide better compression rate . even though the previous work did not give much importance to the security of the data through the network. So in order to provide better security and better performance the new method using unitary transform can be introduced.

The focus of this paper is the improvement of image compression through steganography. Even if the purposes of digital steganography and data compression are by definition contradictory, these techniques can be used jointly to compress an image. The combined usage of compression and steganography will help to reduce the file size even though some bits are embedded. This is because the embedded bits are taken from and hidden within the image itself. Along with the combination of the steganography and the compression the security can also be improved in this work by adding an extra encryption before embedding the data. It can be implemented using the two new algorithms UNI_STEGO_JPEG encoder and UNI_STEGO_JPEG decoder. By using these two algorithms it is experimentally proven that the compression rate can be improved along with better security.

The base paper is further divided into different sections. The rest of the paper is organized as follows: in Section 2, a brief description on the related works was provided. In Section 3, the details of the proposed method and in Section 4 the experimental results are given. Finally we end the paper with the concluding remarks.

II. RELATED WORKS

In [4] Christopoulous et al. proposed the JPEG still picture compression standard is used to achieve the image compression. Itis developed because if two applications cannot exchange uncompressed images because they use incompatible color spaces, dimensions etc , a common compression method is required. In DCT based coding, at the encoder side first the source image samples are grouped into 8x8 blocks. Then provide these blocks as the input it to FDCT (Forward DCT). Next it is to be quantized by a quantizer and then entropy encoding is performed to obtain the compressed image. Then the reverse process is done in decoder. This method is introduced due to the high expense of VLSI codecs But here only gray scale images are considered which can be taken as its disadvantage. Fig. 2. shows the processing steps of the DCT based encoder and decoder.

In [3] an image coding algorithm which uses the data hiding techniques to fold an image into itself is proposed. i.e it fold one part of the image into other part. Data hiding is a process of encoding extra information into a host image by making small modifications to its pixels. In this an image is split into two parts of equal size: a host image and a residual image. The host image contains the most relevant part of the image. It is used as the host image in data hiding process. The residual image contains less important information and can be compressed to a very low bit rate. The residue image is used as the data source which is embedded into the host image. In this way, we fold part of the image into the other part. The host image "holds" the residual image. The residual image is compressed before being embedded into the host image. This increases the effective data hiding capacity of the host image. Here a modified embedded zero-tree wavelet coder to compress the residual image. This compression method is chosen because it is very efficient in compression and fully embedded: it can stop at any bit rate without sacrificing its compression efficiency. To embed the compressed residual image into the host image, the host is JPEG coded and decoded with the designated quality of the application. The coded residual image in embedded into the JPEG.

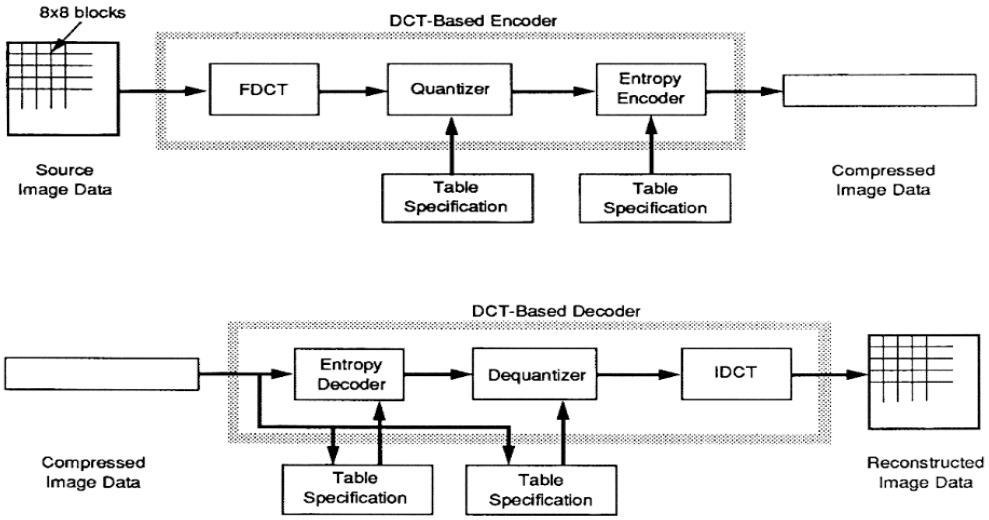

Fig 2: DCT based encoder and decoder processing steps [4]

Reconstructed host image. The host image with the embedded data is then JPEG compressed to the desired quality. The residual image is compressed into a bit stream and then embedded into the host image. The host image, which is 50% of the size of the original image, is coded using standard compression techniques. The embedded data does not increase the bit rate of the coded image. As a result, one may code only 50% pixels of the original image and still have perfect reconstruction.

In [2] a novel image compression scheme, employing steganography to decrease the data file size, is investigated. That is, data compression is performed twice under this point of view. Using at first, the conventional standard JPEG which reduces redundant data, taking advantage of the energy compaction property, and secondly, by means of steganography which embeds some bits blocks within its subsequent blocks of the same image. The embedded bits do not increase the file size of the compressed image, but as they are taken from and hidden within the image itself, the file size will be further decreased. The key idea behind this is to compress the target block k of an image using lossy JPEG, and then hide the resulting bits into subsequent blocks of the compressed image. That is, data compression is performed twice under this point of view. Using at first, the standard JPEG, which reduces redundant data, and finally, by means of steganography which embedded some bits of a given block within its subsequent blocks of the same image. The embedded bits do not increase the file size of the compressed image, but as they are taken from and hidden within the image itself, the file size will be further decreased.

In [3] points out that with the increasing use of multimedia technologies, image compression requires higher performance as well as new features. To address this need in the specific area of still image encoding, a new standard is currently being developed, the JPEG2000. It is not only intended to provide rate-distortion and subjective image quality performance superior to existing standards, but also to provide features and functionalities that current standards can either not address efficiently or in many cases cannot address at all. Lossless and lossy compression, embedded lossy to lossless coding, progressive transmission by pixel Accuracy and by resolution, robustness to the presence of bit-errors and region-of-interest coding, are some representative features. It is interesting to note that JPEG2000 is being designed to address the requirements of a diversity of applications Here the steps used are tiling, decomposed using wavelet transform, quantization, code blocks from the input to the entropy encoder, arithmetic coding, and finally a layered bit stream formation. The new JPEG 2000 standard use wavelet-based compression method, and it can operate at higher compression ratio without generating the characteristic 'blocky and blurry' artifacts of the original DCT-based JPEG standard. Image Compression is a technique to reduce irrelevance and redundancy of image data in -order to store or transmit data in an efficient form. Steganography is the art of science of encoding hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of message.

In [1] a novel image compression scheme, employing steganography to decrease the data file size . Here the data compression is performed twice. First, the advantage of energy compaction using JPEG is used to reduce redundant data. Second, embed some bit blocks within its subsequent blocks of the same image with steganography. The key idea behind this is to compress the target block k of an image using lossy JPEG, and

then hide the resulting bits into subsequent blocks of the compressed image. The reason for the paper [1] is mainly because of inorder to extend previous works for color images and by providing high compression gain with high quality for the jpeg images .If steganography process adds extra data within the target support for authentication purpose, compression attempts to remove redundant data to reduce the original file size. To this end, two image compression algorithms exploring this idea are investigated. The first one is based on the baseline DCT-based JPEG, while the second uses the DWT-based JPEG. The baseline JPEG and DWT-based version of JPEG are still widely used for compression of still images available in the Web and produced by digital cameras. To decrease the original file size, a steganographic scheme is integrated within the compression encoding process. Precisely, after the division of the target image into a set of blocks using JPEG, some blocks are embedded into their subsequent blocks of the same image. That is, compression is performed in two steps. First, the conventional standard JPEG either DCT or DWT is used. And second, by means of steganography which embedded some bits-blocks within its subsequent blocks of the same image. The embedded blocks do not increase the file size of the compressed image, but as they are taken from and hidden within the image itself, the file size will be further decreased. The novelty of this paper is instead of DCT based scheme DWT is used. As a result of DWT decomposition, the tile is divided into 4 subbands LL,LH,HL,HH(here the DWT coefficients are grouped). The target blocks are compressed using DWT-JPEG. After quantization depending on the embedded value(0 or 1) the quantized coefficient is rounded to the nearest odd or even value respectively. The three level DWT decomposition of the image tile is shown in the figure. 3. There are mainly two algorithms has been introduced here for the encoding and decoding purpose, Stego_JPEG encoder and Stego_JPEG decoder. Before embedding process, the target blocks are compressed using DWT-JPEG, including DWT transformation, quantization, and entropy coding .While receiving blocks are only transformed using DWT and quantized. After quantization, receiving blocks accept the bits from the target block as follows. Depending on the embedded bit (0 or 1), the quantized coefficient value of the receiving blocks is rounded to the nearest odd or the even value. If the embedded bit is equal to 0; the quantized coefficient value is rounded to its odd value, otherwise, it will be rounded to the even value. For color images Stego-JPEG(DWT) decoder as well as encoder should be repeated three times for each of the components Y, Cb, and Cr respectively.

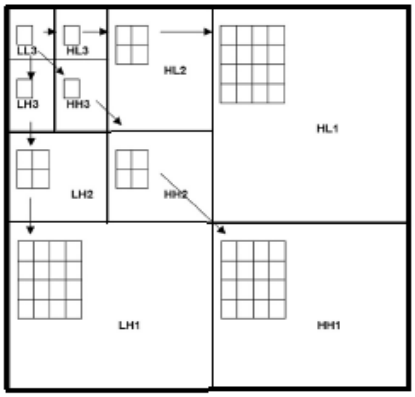

Fig 3: Three level DWT decomposition of image tiles[1].

In [1] and [2] the main focus is on the increasing image compression rate with the help of the steganographic methods for the JPEG images. It provides better image compression rate, even though it does not give much importance to the security. Here the image quality will be drastically reduced when the image is compressed a lot using the JPEG compression and also the security of the data transferred is not considered here to an acceptable level. So in order to overcome that disadvantage or limitation in addition to the existing method as specified in [1] a robust perceptual encryption technique can be used before compression in order to provide better security as well as performance.

A robust perceptual encryption technique can be used by selecting one out of multiple unitary transforms (UT) according to the encryption key generated from the random permutation method at the transformation stage. By the introduction of above said algorithms definitely we achieve an efficient encryption. This methodology will be useful for services over networks. In order to achieve it by using an extra step, an encryption is done prior to the compression and embedding so that the data can be sheltered in a much better

way. By doing so the main aim is to achieve better compression rate along with better performance. A novel approach of using unitary transform along with the combination of image compression and steganography in order to improve the performance is to be implemented.

III. Proposed System

The steganography based compression algorithms must satisfy invisibility, payload, file size requirements and robustness. The embedded data is invisible if a human subject with normal vision is not able to distinguish media that contain hidden data from those that do not. The payload refers to the number of hidden bits while satisfying the invisibility requirement. The embedded data is robust if it can be detected after non intentional modification such as lossy compression. Finally, the last requirement to which attention must be paid after embedding is file size. This requirement is not significant in the usual steganography applications, but for the compression the file size has to be reduced. Two-dimensional unitary transforms have found two major applications in image processing. Transforms have been utilized to extract features from images. Dimensionality reduction in computation is a second image processing application. Stated simply, those transform coefficients that are small may be excluded from processing operations, such as filtering, without much loss in processing accuracy. Another application in the field of image coding is transform image coding, in which a bandwidth reduction is achieved by discarding or grossly quantizing low-magnitude transform coefficients [8].

A unitary transformation is a rotation of a vector in an N-dimension space, i.e., a rotation of basis coordinates [9]. The angles between vectors are Encryption keys generated by random permutation. Unitary Transform implies the following properties: Orthonormality which shows no two basis represent the same information in the image and Completeness shows all information in the image are represented in the set of basic functions. As the n-dimensional space can be spanned by the column vectors of any n by n unitary (orthogonal) matrix, a vector in the space can be represented by any of such matrices, each defining a different transform[9]. Two new algorithms can be proposed here for the introduction of the new robust encryption method along with the existing procedures UNI_STEGO_JPEG encoder and UNI_STEGO_JPEG decoder.

In UNI_STEGO_JPEG encoder, first of all the input image is selected and then its dimensionality is calculated so that the color images and the gray scale images can be differentiated. If it is a color image it can be converted to YC_bC_r components and then generate the encryption key from random permutation and apply unitary transformation to Y,C_B and C_R components based on the generated encryption key. Then the embedding process can be performed as in Stego_JPEG (DWT) encoder [1]. In UNI_STEGO_JPEG decoder the reverse of the encoder section is carried out for the extraction.

3.1 Encrypting and Embedding Process

The target blocks are divided into 8X8 pixels after encryption and some of the blocks are embedded into their subsequent blocks of the same image. This helps to reduce the original file size which uses the steganographic method. The target blocks are the blocks that are hiding to the subsequent blocks called the receiving blocks (as they receive from the target block) [5].Here the key idea is then to compress a target block B_k of an image using lossy JPEG, and then hide the resulting bits into subsequent blocks $B_{k+1}, \ldots, B_l$ of the compressed image. For color images each of the three components Y, C_b and C_r components have to be treated separately. The total numbers of embedded bits are the summation of the number of bits that are embedded in each of the subsequent blocks. It can be denoted as $n = n_{k+1}+n_{k+2}+ \ldots+n_l$, where $n_{k+1},n_{k+2}, \ldots,n_l$ are the embedded bits in each of the subsequent blocks $B_{k+1},\ldots B_l$, where l is a variable number of blocks, and it depends on the number of bits that are coded. Here the data is not hide in the DC component, since contents in this coefficient is very sensitive to human eyes. So the data can be embedded in the high frequency coefficients called AC coefficients. Here the target blocks are fully compressed while the receiving blocks are not fully compressed. So that before embedding, the target blocks

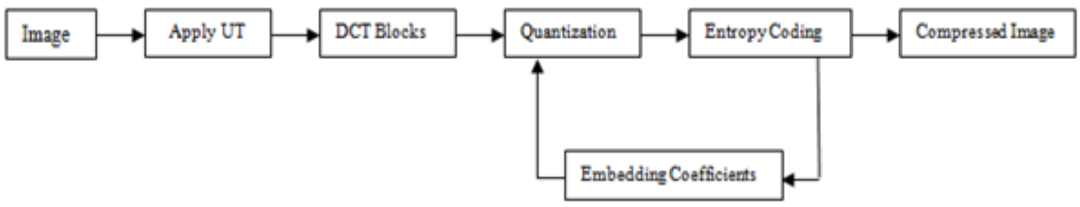

Fig. 4 Block diagram of encryption and embedding process (proposed method).

Undergo JPEG compression steps including DCT, Quantization, Rounding and Coding and the receiving blocks undergo only DCT and Quantization steps. The receiving blocks then receive the bits from the target blocks in a particular format. The block diagram for the embedding process is shown in fig 4.

The embedded bit can be 0 or 1 and depends on that value the quantized coefficient value $\bar{C}(i,j)$ of the receiving blocks are rounded to the nearest odd or even value. If the embedded bit is 0, then it is rounded to the odd value, otherwise it is rounded to even value.

The JPEG quality can be preserved in embedding process by performing in a zigzag way in the receiving blocks. The embedding process is performed with respect to a given threshold *T*. The payload size depends on this threshold value such that if the value of *T* increases, fewer coefficients are used for embedding there by the payload size can be reduced.

In order to increase the compression the steganography technique used here integrates both embedding and the rounding process If $T = 0$ portion of embedding is maximum (with increasing file size), and if $T = 1$ portion of rounding is maximum (while embedding is high without increasing file size). When $T-1 \leq |C_i(i,j)| < T$ (knowing the fact that $T \geq 1$), an ambiguity during data extraction will appear. For example, if $T = 2$ and $C_i(i,j) = 2.4$, then $Q_0(C_i(i,j)) = 2$. In this situation it cannot be decide whether there is hidden data or not. As a solution, the coefficients between *T* and $T - 0.5$ are rounded to $T - 1$.So the *T* value can be chosen as 1 for better compression.

The main steps in JPEG encoding consists of DCT on 8x8 image blocks, Quantization, Zig-zag ordering and run-length encoding and Entropy coding . In DCT the image is divided up into 8x8 blocks and 2D DCT is performed on each block independently. That is why, when a high degree of compression is requested, JPEG gives a "blocky" image result [11]. Quantization in JPEG aims at reducing the total number of bits in the compressed image. Here divide each entry in the frequency space block by an integer (quantization matrix $Q(u, v)$, then round the result. The remaining steps all lead up to entropy coding of the quantized DCT coefficients and these additional data compression steps are lossless. Most of the loss occurred in the quantization step.

In Run-Length Coding (RLC) the AC and DC components are treated differently. This is because after quantization there are many AC components having value = 0v, RLC is a good idea. Most of the zero components are towards the lower right corner (high spatial frequencies). To take advantage of this, use zigzag scanning to create a 64-vector. Now the RLC step replaces values in a 64-vector (previously an 8x8 block) by a pair (RUNLENGTH, VALUE), where RUNLENGTH is the number of zeroes in the run and VALUE is the next non-zero value. Now the DC coefficients are handled. There is only one DC per block and the DC coefficients may vary greatly over the whole image, but slowly from one block to its neighbor (once again, zigzag order). So we have to apply Differential Pulse Code Modulation (DPCM) for the DC coefficients. After that apply entropy coding to the RLC coded AC coefficients and the DPCM coded DC coefficients. The baseline entropy coding method uses Huffman coding on images with 8-bit components [8].It should be noted that for color images, steps 3-7 of algorithm 1 UNI_STEGO_JPEG encoder and steps 1-6 of algorithm 2 UNI_STEGO_JPEG decoder is repeated three time for each of the components Y, C_b and C_r separately.

The encryption and embedding process can be summarized as follows. Here the input is a JPEG image *I* and the output is an encrypted compressed image.

Algorithm 1: UNI_STEGO_JPEG Encoder

1. Compute dimensionality.
2. Generate encryption key from random permutation and apply unitary transformation (separately for *Y*, C_b and C_r components for color images) based on the generated encryption key.
3. Group wavelet coefficients into *m* x *m* blocks B_k from the encrypted image of *N* x *M* pixels.
4. For each of the target block B_k apply quantization and rounding.
5. Compress B_k by using lossy JPEG coding based on DWT.
6. The n coefficients of the subsequent blocks $B_{k+1}, \ldots, B_l$ can be used to embed the current bits of the compressed block B_k.

$$C_i(i,j) = \begin{cases} \text{Round}(C_i(i,j)) & |C_i(i,j)| \geq T \\ C_i(i,j) & T\text{-}1 \leq |C_i(i,j)| < T \\ C_i(i,j) + 0.5 & \text{Otherwise} \end{cases}$$

7. Repeat steps 4-6 until $k > \lfloor N \times M / m \times m \rfloor$, $k = l + 1$.

3.2 Decrypting and Extracting Process

The same method that is used in the encrypting and embedding process can be used to extract the embedded bits from a compressed image in a reverse fashion. First the decompression is performed by using the JPEG decoding procedure. After that, the embedded blocks can be extracted by checking the odd and even values of $C_i(i,j)$ coefficients from n coefficients of sequential blocks. The decoder ignores those coefficients that have a quantized value with magnitude T or small. Next the inverse DWT is used for the extracted blocks and the original blocks. Finally, apply inverse of unitary transform on decompressed image (repeated for Y, C_b and C_r components for color images). The block diagram for the decryption and extracting process is shown in figure. 4.2.

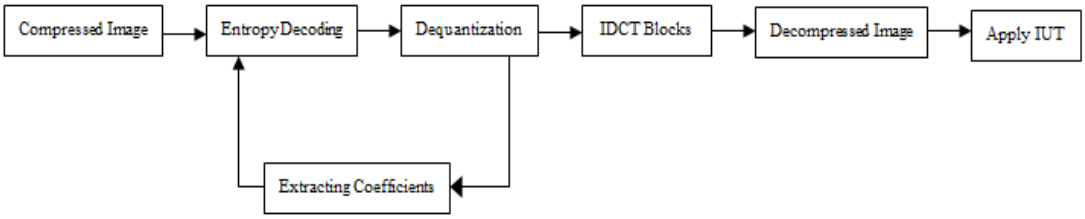

Fig 5: Block diagram of decryption and extracting process (proposed method)

So the decryption and extracting process can be summarized as follows. Here the input is a compressed image and the output is a decompressed image I.

ALGORITHM 2:UNI_STEGO_JPEG decoder

1. Based on DWT, JPEG decoding can be performed for decompress the blocks.
2. Extract the embedded block of $B_{k+1}, \ldots, B_l$ coefficients $|C_i(i,j)| \geq T$.fom the subsequent blocks such that

$$\text{Embedded-bit} = \begin{cases} 1 & C_i(i,j) = \text{odd} \\ 0 & C_i(i,j) = \text{even} \end{cases}$$

3. Apply JPEG decoding based on DWT for embedded block.
4. Repeat 1–3 until $l > \lfloor N \times M / m \times m \rfloor$, $l = l + 1$.
5. Create DWT image form from extracted and original blocks.
6. Apply IDWT.
7. Apply inverse of unitary transform on decompressed image.

These are the two algorithms used here for the compression and decompression.

IV. Experiments And Results

In this paper, different images having different sizes for both color and gray scale images are taken for the experiment. Here the compression performance is assessed using both the compression ratio and the quality of compression. The criterion for the image quality comparison is taken as the resemblance between the original image and the reconstructed image. In this work the measure of image quality in order to compare the images is

taken as Peak Signal to Noise Ratio (PSNR). Also SSIM measure can be chosen which is better for perceptual evaluation [7]. Even though for the JPEG compression, SSIM and PSNR are equivalents [6]. The PSNR can be calculated $10log_{10}(255^2/MSE)$ where MSE is the mean squared error.

The algorithms (Algorithm 1 and Algorithm 2) are implemented and tested with different values of T. The results shows that when T=1 the maximum performance can be achieved as shown in figure 10 and figure 11 which shows the increasing compression ratio and the decrease in PSNR value respectively. Also with T=1 a number of images belonging to a dataset was tested.

Fig 6 is a set of images belonging to this dataset. It contains color images of sizes 256 x 256 pixels and 512 x 512 pixels and gray scale images of sizes 128 x 128 pixels, 256 x 256 pixels and 512 x 512 pixels have been selected. A total of 250 images are taken for the experiment and results are noticed.

.

Fig. 6 Example of images used during compression.

Table 1, 2 and 3 shows the compression ratio and decrease of PSNR on the three different images Lena, Pepper and Baboon on DCT, DWT and by using the proposed method respectively.

Table 1: Increasing compression ratio and decrease of PSNR for color images (512 x 512) based on DCT

	Lena	Pepper	Baboon
Compression ratio	25.2912	27.1626	33.6113
Decrease of PSNR	1.0425	1.1028	1.0014

Table 2: Increasing compression ratio and decrease of PSNR for color images (512 x 512) based on DWT

	Lena	Pepper	Baboon
Compression ratio	27.6688	28.9967	34.5413
Decrease of PSNR	1.0326	1.0193	0.9278

Table 3: Increasing compression ratio and decrease of PSNR for color images (512 x 512) based on the proposed method

	Lena	Pepper	Baboon
Compression ratio	39.0999	38.0268	41.2662
Decrease of PSNR	1.0104	1.0135	0.9051

As from the experiment the decompressed images obtained by using the Stego-JPEG (DWT) and the UNI_STEGO_JPEG for the images having few frequency components, there are no large visual differences can be viewed. It is shown in figure 7 and figure 8 respectively.

(a) (b)

Fig.7. Decompressed DWT based image (a) original image, (b) Stego-JPEG (DWT).

(a) (b)

Fig.8. Decompressed DWT based image (a) original image, (b) UNI_STEGO_JPEG

These differences can be narrowed by reducing the payload or compression ratio. The experiment shows that for color images also there is no major visual difference between the original JPEG image and the reconstructed image as shown in fig 9.

(a) (b)

Fig. 9. Input and output images using the proposed method
(Original image in first column (a) and UNI_STEGO_JPEG in second column (b).)

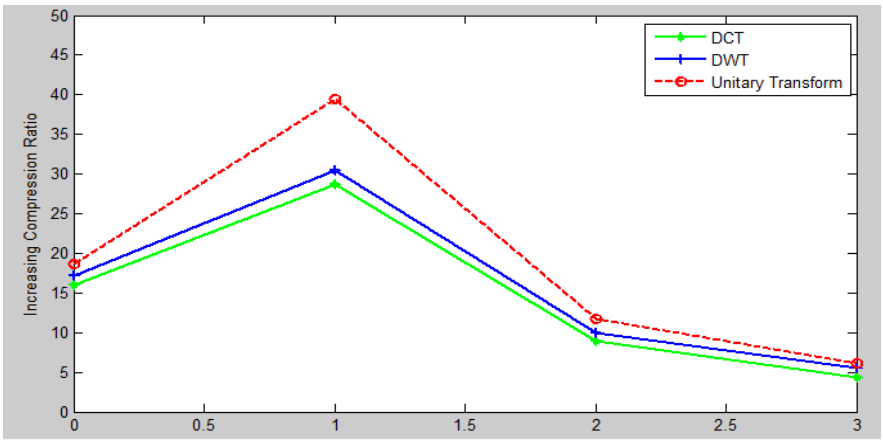

Fig 10 Average compression ratio when T=0, 1, 2 and 3 for Lena, Pepper, and Baboon

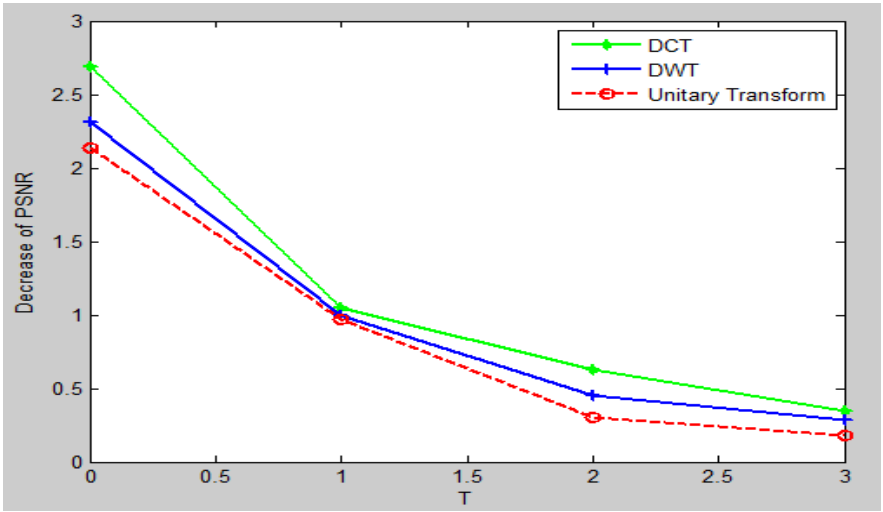

Fig 11 Average decrease of PSNR when T= 0, 1, 2 and 3 for Lena, Pepper, and Baboon

The algorithms have been implemented by MATLAB on Windows7 with Intel Core2 Duo2.40 GHz and 3 GB memory. The experimental results show that the compression rate can be improved along with better security.

V. CONCLUSION

Even if the purposes of steganography and compression are antagonist the steganography can be used efficiently to increase the compression ratio of JPEG based on DCT and DWT. Here these techniques can be used jointly to compress and to hide data within the same image. That is, data compression is performed twice under this point of view. Using at first, the JPEG (based on DCT or DWT), which reduces redundant data, and finally, by means of steganography which embedded some bits of a given block within its subsequent blocks of the same image. The embedded bits do not increase the file size of the compressed image, but as they are taken from and hidden within the image itself, the file size will be further decreased.

The security can be improved by the addition of a robust perceptual encryption technique. For this a unitary transform was employed and by generating a key by using the random permutation method the encryption can be provided. Here the Y, C_B, C_R components for a color image can be rotated in different angles and then the compression and embedding is performed and for the gray scale images only one rotation is performed as its dimensionality is one and then the compression is performed. It will add extra security and also provide an increase in image compression ratio.

The results shows that it not only increases the security but also improve the performance. So the data can be compressed with a better compression rate and also with improved security which can be transferred safely through the network can be achieved. Experimental results show that this method gives better compression rates as well as better performance.

Here for the gray scale images only one key is generated so inorder to provide a better security for grayscale images another method can be adapted. So as a future work a better encryption technique for gray scale images can be used.

REFERENCES

[1]. Reza Jafari, Djemel Ziou, Mohammad Mehdi Rashidib. "Increasing image compression rate using steganography", Elsevier Expert Systems with Applications 40, pp 6918–6927, 2013
[2]. Jafari, R., Ziou, D., & Mammeri, A. "Increasing compression of JPEG images using steganography", In IEEE International Symposium on Robotic and Sensors Environment, pp 226–230, 2011.
[3]. Swanson, M., Zhu, B., & Tewfik, A. H "Image coding by folding", IEEE International Conference on Image Processing , pp 665–668, 1997
[4]. Wallace, G. K. "The JPEG still picture compression standard", IEEE Transaction on Consumer Electronics, 38(1), pp 18–35, April 1992
[5]. Christopoulos, C., Skodras, A., & Ebrahimi, T. "The JPEG2000 still image coding system: an overview", IEEE Transaction on Consumer Electronics, 46(4), pp 1103-1127, November 2000.
[6]. Hore, A., & Ziou, D. "Image quality metric: PSNR vs. SSIM". In 20th International conference on pattern recognition, pp 2366–2369, 2010
[7]. Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. "Image quality assessments: From error visibility to structural similarity". IEEE Transactions on Image Processing, 3(4), pp 600–612, 2004.
[8]. Bernd Jähne, Digital Image Processing, Berlin, Springer Science & Business Media, Volume 1, 2005
[9]. William. K. Pratt, "Chapter 8 Unitary Transforms", Digital Image Processing, 4th edition, PIKS Scientific Inside, John Wiley & Sons, Inc, 2007
[10]. R. C. Gonzalez, R. E. Woods, Digital Image Processing, 3rd edition, India, Prentice Hall, 2008
[11]. W. B. Pennebaker and J. L. Mitcell, JPEG: Still Image Data Compression Standard, Newyork, Van Nostrand Reinhold, 1993.
[12]. M. Antonini, M. Barlaud, P. Mathieu and I.Daubechies: "Image Coding Using the Wavelet Transform", IEEE Trans. Image Proc., pp. 205-220, April 1992.