A Novel Steganography Method for Image Based on Huffman Encoding

Rig Das
Department of Computer Science and Engineering
North Eastern Regional Institute of Science and Technology (Deemed University)
Itanagar, Arunachal Pradesh - 791109, India
rig.das@gmail.com

Themrichon Tuithung
Department of Computer Science and Engineering
North Eastern Regional Institute of Science and Technology (Deemed University)
Itanagar, Arunachal Pradesh - 791109, India
t_tuithung@yahoo.com

***Abstract*— *Steganography* is the science that involves communicating secret data in an appropriate multimedia carrier, e.g., image, audio and video files. It comes under the assumption that if the feature is visible, the point of attack is evident, thus the goal here is always to conceal the very existence of the embedded data. This paper presents a novel technique for image steganography based on Huffman Encoding. Two 8 bit gray level image of size M X N and P X Q are used as cover image and secret image respectively. Huffman Encoding is performed over the secret image/message before embedding and each bit of Huffman code of secret image/message is embedded inside the cover image by altering the least significant bit (LSB) of each of the pixel's intensities of cover image. The size of the Huffman encoded bit stream and Huffman Table are also embedded inside the cover image, so that the Stego-Image becomes standalone information to the receiver. The experimental result shows that the algorithm has a high capacity and a good invisibility. Moreover Peak Signal to Noise Ratio (PSNR) of stego image with cover image shows better result in comparison with other existing steganography approaches. Furthermore, satisfactory security is maintained since the secret message/image cannot be extracted without knowing the decoding rules and Huffman table.**

Keywords- Steganography, Huffman Encoding, LSB, DCT, PSNR

I. INTRODUCTION

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. Steganography, derived from Greek, literally means "*covered writing*" (Greek words "**stegos**" meaning "**cover**" and "**grafia**" meaning "**writing**"). It comes under the assumption that if the feature is visible, the point of attack is evident, thus the goal here is always to conceal the very existence of the embedded data. Therefore Steganography gets a role on the stage of information security [1, 2].

Steganography is the science that involves communicating secret data in an appropriate multimedia carrier, e.g., image, audio and video files. The media with or without hidden information are called ***Stego Media*** and ***Cover Media***, respectively. Steganography can meet both legal and illegal interests, e.g., civilians may use it for protecting privacy while terrorists may use it for spreading terroristic information [3].

Steganography and *Cryptography* are cousins in the spy craft family. Cryptography scrambles a message so it cannot be understood. Steganography hides the message so it cannot be seen. A message in cipher text, for instance, might arouse suspicion on the part of the recipient while an "invisible" message created with steganographic methods will not. [2]

Another technology that is closely related to Steganography is *Watermarking*. *Watermarking* is a protecting technique which protects (claims) the owner's property right for digital media (i.e. images, music, video and software) by some hidden watermark. Therefore, the goal of Steganography is the secret messages while the goal of watermarking is the cover object itself.

The main objective of Steganography is to communicate securely in such a way that the true message is not visible to the observer. That is unwanted parties should not be able to distinguish in any sense between *cover-image* (image not containing any secret message) and *stego-image* (modified cover-image that contains secret message). Thus the stego-image should not deviate much from the original cover-image. Today Steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels. Fig. 1 shows the block diagram of a simple Steganographic system [1].

II. RELATED WORK

Many research works have been carried out on Stego-Security. Different researchers employed different techniques for the purpose of secured secret image embedding. Following are the few related works carried out by various research groups:

A. Steganography by Embedding Message Inside an Image[3]

Steganography can be accomplished by simply feeding into a Windows OS command window, e.g., the following code:

C:\> Copy Cover_Image.jpg /b + Message.txt /b Stego_Image.jpg

What this code does is that it appends the secret message found in the text file "***Message.txt***" into the JPEG image file "***Cover_Image.jpg***" and produces the stego-image "***Stego_Image.jpg***". The idea behind this is to abuse the

978-1-4577-0748-3/12/$26.00 © 2012 IEEE

recognition of **EOF** (End of File). In other words the message is packed and inserted after the EOF tag. When Stego_Image.jpg is viewed using any photo editing application, the later will just display the picture ignoring anything coming after the EOF tag. However, when opened in Notepad, for example, our message reveals itself after displaying some data as shown in Fig. 2. The embedded message does not impair the image quality. Neither image histograms nor visual perception can detect any difference between the two images due to the secret message being hidden after the EOF tag. Unfortunately, this simple technique would not resist any kind of editing to the stego-image nor any attacks by steganalysis experts.

B. Steganography in Spatial Domain [2]

In spatial domain methods a steganographer modifies the secret data and the cover medium in the spatial domain, which involves encoding at the level of the **LSB**s. A simple way of steganography is based on modifying the least significant bit layer of images, known as the LSB technique. The LSB technique directly embeds the secret data within the pixels of the cover image. In some cases LSB of pixels visited in random or in certain areas of image and sometimes increment or decrement the pixel value. Some of the recent research studied the nature of the stego and suggested new methodologies for increasing the capacity. Habes in [4] proposed a new method (4 least significant bits) for hiding secret image inside carrier image. In this method each of individual pixels in an image is made up of a string of bits. He took the 4-least significant bit of 8-bit true color image to hold 4-bit of the secret message/image by simply overwriting the data that was already there.

To hide an image in the **LSBs** of each byte of a **24-bit image**, you can store **3 bits** in each pixel. A **1,024 X 768** image has the potential to hide a total of **2,359,296 bits** (**294,912 bytes**) of information. If you compress the message to be hidden before you embed it, you can hide a large amount of information. To the human eye, the resulting stego-image will look identical to the cover image. E.g. the letter **A** can be hidden in three pixels (assuming no compression). The original raster data for **3 pixels** (**9 bytes**) may be

(00100111 11101001 11001000)

(00100111 11001000 11101001)

(11001000 00100111 11101001)

The binary value for **A** is **10000011**. Inserting the binary value for **A** in the three pixels would result in

(00100111 11101000 11001000)

(00100110 11001000 11101000)

(11001000 00100111 11101001)

The underlined bits are the only three actually changed in the 8bytes used. On average, LSB requires that only half the bits in an image be changed. You can hide data in the least and second least significant bits and still the human eye would not be able to discern it.

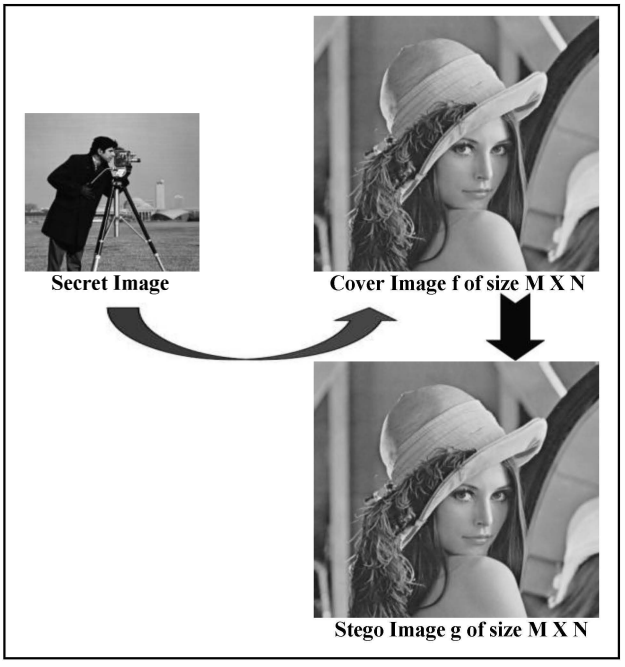

Figure 1. The Block Diagram of Simple Steganographic System

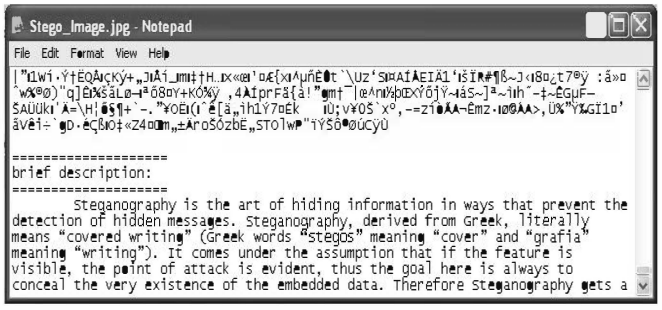

Figure 2. The Secret Message Revealed when the Stego-Image is opened using Notepad

C. Steganography in Frequency Domain [3]

The scheme of the Frequency Domain is to embed the secret data within the cover image that has been transformed such as DCT (discrete cosine transformation). The DCT transforms a cover image from an image representation into a frequency representation, by grouping the pixels into non-overlapping blocks of 8 X 8 pixels and transforming the pixel blocks into 64 DCT coefficients each. A modification of a single DCT coefficient will affect all 64 image pixels in that block. The DCT coefficients of the transformed cover image will be quantized, and then modified according to the secret data. Tseng and Chang in [5] proposed a novel steganography method based on JPEG. The DCT for each block of 8 X 8 pixels was applied in order to improve the capacity and control the compression ratio.

The description of the two-dimensional DCT for an input image F and an output image T is calculated as (1)

$$T_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F_{mn} \cos\frac{\pi(2m+1)p}{2M} \cos\frac{\pi(2n+1)q}{2N} \qquad (1)$$

Where, **0 ≤ p ≤ M-1** and **0 ≤ q ≤ N-1 a**nd,

$$\alpha_p = \begin{cases} 1/\sqrt{M}, & p = 0 \\ \sqrt{2/M}, & 1 \le p \le M-1 \end{cases}$$

$$\alpha_q = \begin{cases} 1/\sqrt{N}, & q = 0 \\ \sqrt{2/N}, & 1 \le q \le N-1 \end{cases}$$

Where M, N are the dimensions of the input image while m and n are variables ranging from 0 to M-1 and 0 to N-1 respectively.

Capacity, security and robustness, are the three main aspects affecting steganography and its usefulness. Capacity refers to the amount of data bits that can be hidden in the cover medium. Security relates to the ability of an eavesdropper to figure the hidden information easily. Robustness is concerned about the resist possibility of modifying or destroying the unseen data.

III. A Novel Image Steganography Algorithm Based on Huffman Encoding

Hiding the secret image/message in the spatial domain can easily be extracted by unauthorized user and in the frequency domain the quality of the extracted secret image deteriorates. In this paper we proposed a spatial domain steganographic technique based on Huffman encoding for hiding a large amount of data with high security, good invisibility and no loss of secret message. The schematic/block diagram of the whole process is given in Fig. 3 and Fig. 4.

The main objective in here is to develop a procedure which will provide a better security to the secret image without compromising on the quality of the stego image. Our algorithm has three main parts. First, it embeds the Huffman encoded bit stream of the secret image into the cover image. Second, it embeds the size of the encoded bit stream into the cover image. Third, it also embeds the Huffman table corresponding to the secret image into the cover image.

A. Four Tier Storage Procedure of "Size of Huffman Encoded Bit Stream"

The size of the Huffman encoded bit stream needs to be embedded inside the cover image to let the decoder know till which pixel's LSB holds the Huffman encoded bit stream. Now the question is how will we get the size of the "Size of Huffman encoded bit stream"? If we store this size of the "Size of Huffman encoded bit stream" then also the same question comes recurrently like what is the size of this size? As a permanent answer to this question we have used a four tier storage procedure to store the "size of Huffman encoded bit stream". In the first tier we find the size of Huffman encoded bit stream and store it into a variable e.g. **A**. Then convert **A** into binary format and find the size of the binary bit stream of **A** and store it into another variable e.g. **B** (second tier). Similarly in the third and fourth tier, we find the size of the bit stream of **B** and store it into another variable e.g. **C** and again

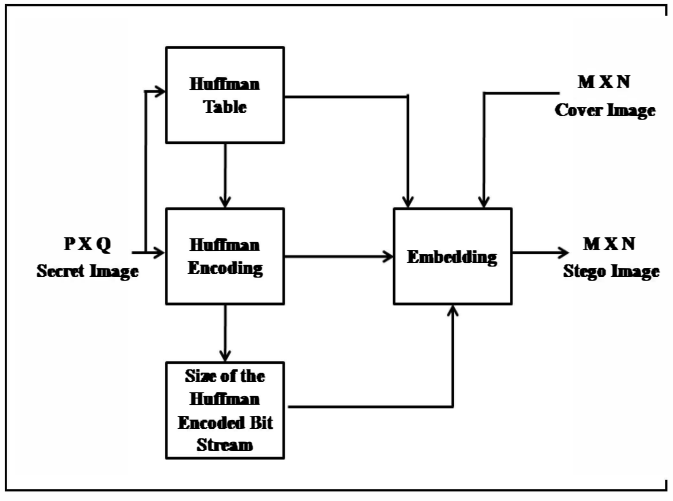

Figure 3. Insertion of the Secret Image/Message into a Cover Image

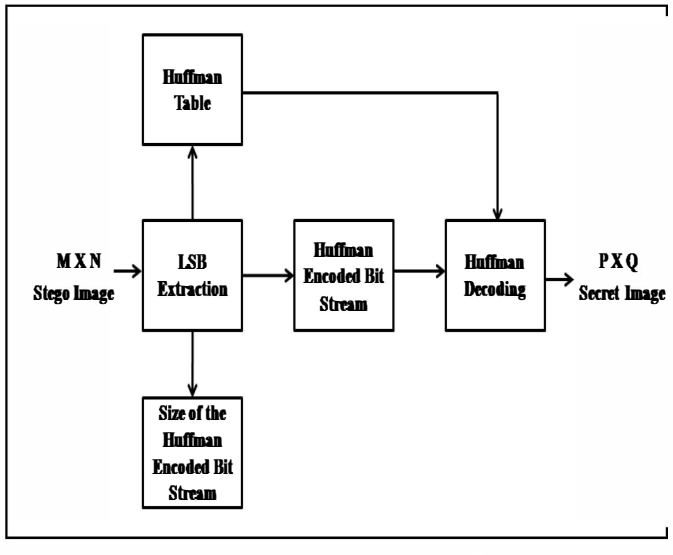

Figure 4. Extraction of the Secret Image from the Stego Image

find the size of the bit stream of **C** and store it into another variable e.g. **D,** respectively. Finally **D** will have a value that can be represented by two binary bits only. Now these two binary bits can be stored in the first or in the last pixel's two **LSB** positions of the first 8 X 8 block of the cover image. The variables **C, B**, **A** also needs to be stored into the cover image one by one sequentially. E.g. of four tier storage of the "size of Huffman encoded bit stream", let us say the size of a Huffman encoded bit stream is **461690** so,

A = 461690 = 1110000101101111010, which is of size 19 bits.

B = 19 = 10011, which is of size 5 bits.

C = 5 = 101, which is of size 3 bits.

D = 3 = 11, which is of size 2 bits

This four tier storage procedure is very important because the size of the Huffman encoded bit stream comes down to **2 bits** from **19 bits**. Variables are stored from **D** to **C**, **B**, **A** sequentially. At the time of LSB extraction first variable **D** is extracted then **C** to **B** to **A** to finally find the actual size of Huffman encoded bit stream.

B. *Proposed Algorithm for Embedding and Extraction of the Secret Image*

Embedding Algorithm:

Input: An M X N carrier image and a P X Q secret message/image.

Output: An M X N stego-image.

Step-1: Read both the Cover Image and the Secret Image and store their intensity value of different pixels in two different arrays.

Step-2: Calculate the size of the Secret Image. The size of the Secret Image multiplied by 8 (for 8 bit images) should be less than the size of the Cover Image. E.g. if the Secret Image size is 256 × 256 = 65536 then after multiplying it by 8 it becomes 523288. This is lesser than the size of Cover Image i.e. 1024 × 1024 = 1048576.

Step-3: Obtain Huffman table of secret message/image.

Step-4: Find the Huffman encoded binary bit stream of secret-image by applying Huffman encoding technique using Huffman table obtained in **Step-3**.

Step-5: Calculate size of Huffman encoded bit stream.

Step-6: Store the size found in **Step-5** using the four tier storage procedure described above by modifying the LSB of pixels of first 8 X 8 block of the cover image.

Step-7: Change the LSBs of the cover image excluding the first 8 X 8 block of pixels for every bit of Huffman encoded bit stream found in **Step-4**.

Step-8: Change the LSBs of the cover image to embed the Huffman table found in **Step-3** excluding the first 8 X 8 block of pixels and the pixels used in **Step-7**.

Step-9: Write the new *Stego Image* into the disk.

Step-10: End.

Extraction Algorithm:

Input: An M X N stego-image.

Output: A P X Q secret image.

Step-1: Read the Stego Image and extract the size of the Huffman encoded bit stream from the first 8 X 8 block by extracting the LSB of the pixels using the procedure described in the four tier storage method like variable **D** to **C** to **B** to **A**.

Step-2: Using the size found in **Step-1** extract the Huffman encoded bit stream by extracting the LSB of pixels excluding first 8 X 8 block and add it into an array.

Step-3: Construct the Huffman table by extracting the LSB of pixels excluding the first 8 X 8 block of pixels and the pixels used at **Step-2**.

Step-4: Decode the array obtained in **Step-2** using the Huffman table obtained in **Step-3** to extract the secret image from the stego image.

Step-5: End.

IV. SIMULATION RESULTS

Some experiments are carried out to prove the efficiency of our proposed algorithm. The measurement of the quality between the cover image **f** and stego-image **g** of size M X N is done using **PSNR** (Peak Signal to Noise Ratio) value and the PSNR is defined as:

$$PSNR = 10 \times \log(255^2/MSE)$$

$$Where, \quad MSE = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} (f(x,y) - g(x,y))^2 / N^2$$

f(x, y) and g(x, y) means the intensity value of pixel at position (x, y) of the cover image and stego image respectively. The PSNR is expressed in dB. Larger PSNR indicates the higher the image quality i.e. there is only little difference between the cover-image and the stego-image. On the other hand, a smaller PSNR means there is huge distortion between the cover-image and the stego image.

All the simulation has been done using the MATLAB 7 program on Windows XP platform. A set of 8-bit grayscale images of size 1024 X 1024 and 256 X 256 are used as the cover-image and secret image respectively to form the stego-image. The Fig. 5(a) - (d) shows the four original cover (carrier) images and Fig. 5(e) shows the original secret image/message. Table I exhibit the PSNR comparison of four cover images with their corresponding stego images and the secret image (original and extracted).

From Table I it is observed that for all cover images, PSNR is greater than 57 dB, so the quality of the stego image is very high and the deterioration in quality due to embedding of secret image cannot be distinguished by naked eye. The PSNR between original secret image and extracted secret image is infinite in all the cases that mean both the images are identical, so 100% recovery of the secret image is being achieved.

Table II shows the PSNR of proposed algorithm is better than the one in [1].

Fig. 6 (a) – (d) shows the stego images of the proposed method and Fig. 6 (e) also shows the secret image retrieved from the stego images.

V. CONCLUSION

In this paper we have proposed a novel image steganography method based on Huffman Encoding. The algorithm improves the security and the quality of the stego image and is better in comparison with other existing algorithms. According to the simulation results, the stego images of our proposed algorithm are almost identical to the cover images and it is very difficult to differentiate between them. We have achieved 100% recovery of the secret image that means original and extracted secret images are identical. The stego image also contains the size of the secret image (embedded using our proposed four tier storage procedure) along with the Huffman Table, so the stego image itself is standalone information to the decoder. The decoder only needs to know the Extraction algorithm to extract the secret image/message. Huffman encoding of the secret image keeps

the image away from stealing, destroying by any unintended users hence the proposed method may be more robust against brute force attack.

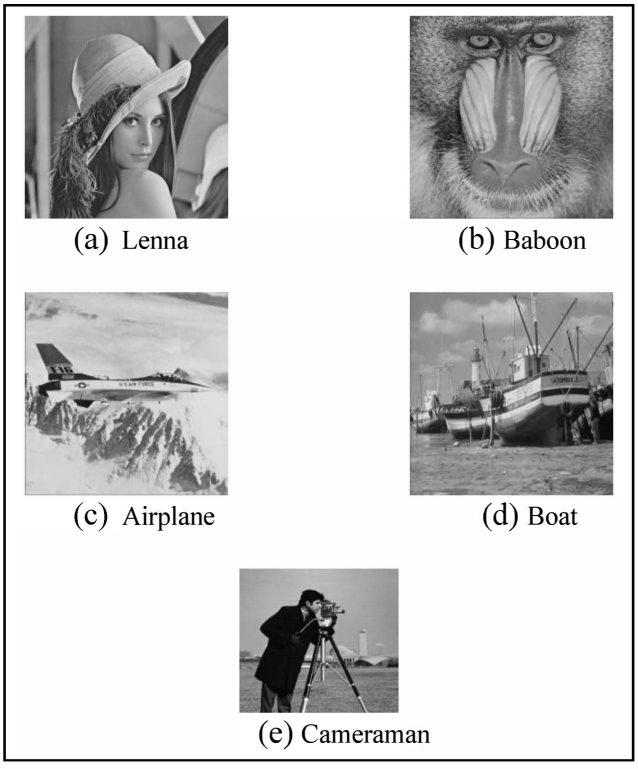

Figure 5. (a) – (d) Four Cover Images for Simulations, (e) Secret Image.

TABLE I. PSNR COMPARISON OF FOUR STEGO IMAGES AND SECRET IMAGE

Cover Image	PSNR (dB)	
	Cover Image and Stego Image	*Original Secret image and Extracted Secret Image*
Lenna	+57.43	Infinite value. Images are Identical.
Baboon	+57.46	Infinite value. Images are Identical.
Airplane	+57.46	Infinite value. Images are Identical.
Boat	+57.46	Infinite value. Images are Identical.

TABLE II. PSNR COMPARISON BETWEEN STEGANOGRAPHY BASED ON BLOCK-DCT AND HUFFMAN ENCODING [3] AND PROPOSED ALGORITHM

Cover Image	PSNR (dB) Between Cover Image and Stego Image	
	Steganography Based on Block-DCT and Huffman Encoding	*Steganography Based on Huffman Encoding*
Lenna	+50.48	+57.43
Baboon	+50.28	+57.46
Airplane	+50.91	+57.46
Boat	+50.36	+57.46

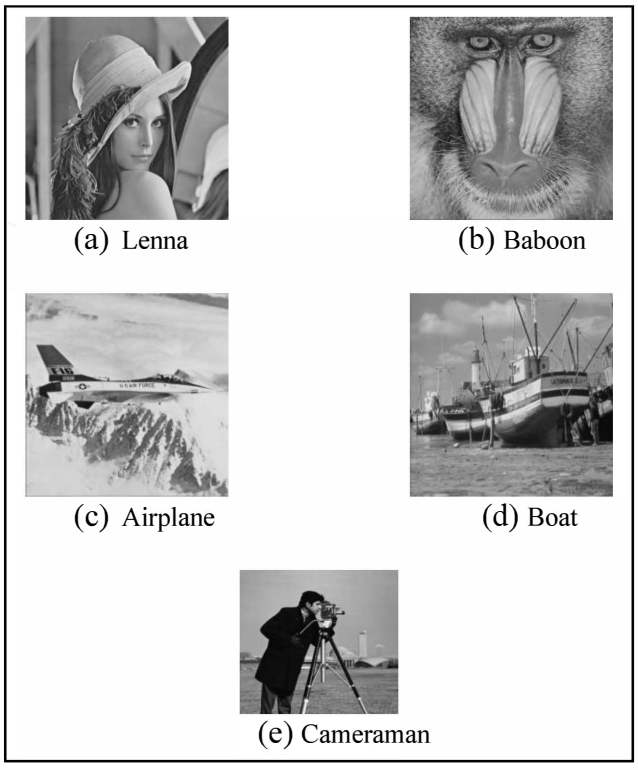

Figure 6. (a) – (d) Four Stego Images of the Proposed Algorithm, (e) Extracted Secret Image.

REFERENCES

[1] A. Nag, S. Biswas, D. Sarkar, P. P. Sarkar, "A Novel Technique for Image Steganography Based on Block-DCT and Huffman Encoding". International Journal of Computer Science and Information Technology, Volume 2, Number 3, June 2010

[2] Neil F. Jhonson, Sushil Jajodia. "Exploring Steganography: Seeing the Unseen". IEEE paper of February 1998.

[3] Abbas Cheddad, Joan Condell, Kevin Curran, Paul Mc Kevitt. "Digital Image Steganography: Survey and Analysis of Current Methods". ELSEVIER Journal on Signal Processing 90 (2010) 727-752.

[4] Alkhrais Habes, "4 least Significant Bits Information Hiding Implementation and Analysis", ICGST Int. Conf. on Graphics, Vision and Image Processing (GVIP-05), Cairo, Egypt, 2005.

[5] C.-C. Chang, T.-S. Chen and L.-Z. Chung, "A steganographic method based upon JPEG and quantization table modification", Information Sciences, vol. 141, 2002, pp. 123-138.

[6] Bin Li, Junhui He, Jiwu Huang, Yun Qing Shi, "A Survey on Image Steganography and Steganalysis". Journal of Information Hiding and Multimedia Signal Processing, Volume 2, Number 2, April 2011

[7] R. Chu, X. You, X. Kong and X. Ba, "A DCT-based image steganographic method resisting statistical attacks", In Proceedings of (ICASSP '04), IEEE International Conference on Acoustics, Speech, and Signal Processing, 17-21 May.vol.5, 2004, pp V-953-6.

[8] Gonzalez, R.C. and Woods, R.E., Digital Image Processing using MATLAB, Pearson Education,India,2006.

[9] Duane Hanselman and Bruce Littlefield, Mastering MATLAB® 7, Pearson Education, India 2008.