Data Hiding in JPEG Images

S.K.Muttoo[1] and **Sushil Kumar**[2]

1. INTRODUCTION

Steganography has been an important subject since people started communicating in writing. Steganography means hiding a secret message (the *embedded message*) within a larger one (*source co*ver) in such a way that an observer cannot detect the presence of contents of the hidden message. Today the growth in the information technology, especially in computer networks such as Internet, Mobile communication, and Digital Multimedia applications such as Digital camera, handset video etc. has opened new opportunities in scientific and commercial applications. But this progress has also led to many serious problems such as hacking, duplications and malevolent usage of digital information. Steganography finds its role in attempt to address these growing concerns. We know that, with the use of steganographic techniques, it is possible to hide information within digital audio, images and video files which is perceptually and statistically undetectable. The method of embedding secret message (which can be plain text, cipher text, or even images) is usually based on replacing bits of useless or unused data in the source cover (can be audio files, sound, text, Disk space, hidden partition, network packets, digital images, software, or circuitry). There are two common methods of embedding: *Spatial embeddin*g in which messages are inserted into the LSBs of image pixels, and *Transform embedding* in which a message is embedded by modifying frequency coefficients of the cover image (result is called the *stego-image*).Transform embedding methods are found to be in general more robust than the Spatial embedding methods which are susceptible to image-processing type of attacks. However with respect to steganography robustness is not a critical property but the perceptibility (i.e., whether the source cover is distorted by embedding information to a visually unacceptable level). There is another important issue of steganography, namely, capacity, i.e., how much information can be embedded relative to its perceptibility [5, 1].

We shall use digital images as the cover object in this paper in which we embed the hidden information. The challenge of using steganography in cover images is to hide as much data as possible with the least noticeable difference in the stego-image. Steganographic algorithms operate on basically three types of images: Raw images (i.e., bmp format), Palette based images (i.e., GIF images) and JPEG images. JPEG images are routinely used in Steganographic algorithms due to the most popular lossy image compression method. Usually it is found that an algorithm used to hide large amounts of information typically result in lower perceptibility (i.e., greater change to the image appearance) and a more robust algorithm result into lower embedding capacity. The JPEG image generation first decomposed the input image into a number of 8 x 8 blocks. Then DCT of each block are computed and the resultant DCT coefficient matrix is quantized using a standard quantization table. Finally the inverse DCT of quantized coefficient matrix are evaluated and the final JPEG image is obtained after rounding the values.

[1]*Reader, Department of Computer Science, University of Delhi, India*
[2]*Reader, Rajdhani College, University of Delhi, New Delhi, India*
E-mail:[1] *skmuttoo@cs.du.ac.in and azadsk2000@yahoo.co.in*

1.1. Jpeg-Jsteg

One of the well known embedding method of steganography based on Transform domain is Jpeg-Jsteg which embeds secret message (that is, in encoded form with help of Huffman codes) into LSB of the quantized DCT coefficients. There is one disadvantage of Jpeg-Jsteg that only few messages can be embedded in the cover-image. Also, Andreas Westfeld and Andreas Pfitzmann [11] noticed that steganographic systems that change LSBs sequentially cause distortionsdetectable by steganalysis methods. They observed that for a given image, the embedding of high-entropy data (often due to encryption) changed the histogram of color frequencies in a predictable way. J.Fridrich [3] has claimed that her method can potentially detect messages as short as any single bit change in a JPEG image.

Chang [2] has proposed a new Steganographic method to increase the message load in every block of the stego-image while retaining the stego-image quality. He has suggested a modified quantization table such that the secret message can be embedded in the middle-frequency part of the quantized DCT coefficients. Moreover, his method is as secured as the original Jpeg-Jsteg.

Neils Provos [7] has proposed another method to counter the statistical attack known as OutGuess. In the first pass, similar to Jsteg, OutGuess embeds message bits using a pseudo-random number generator to select DCT coefficients at random. After embedding, the image is processed again using a second pass. This time, corrections are made to the coefficients to make the stego-image histogram match the cover image histogram.

1.2. T-codes

We know that the best variable-length codes (VLC) are the Huffman codes. They are easy to construct for optimum efficiency if source statistics are known. But if they are used in serial communication, a loss of synchronization often results in a complex resynchronization process whose length and outcome are difficult to predict T-codes provide the solution to this problem.

T-codes are families of variable-length codes (VLC) that exhibit extraordinarily strong tendency towards self-synchronization. The concept of simple T-codes were given by M.R. Titchner [8]. In 1996 [9], Titchner proposed a novel recursive construction of T-codes known as the Generalized T-codes that retain the properties of self-synchronization.

Gavin R. Higgie [4] showed that in situation where codeword synchronization is important, the T-codes can be used instead of Huffman codes, giving excellent self-synchronizing properties without sacrificing coding efficiency. The main advantage of the T-Code is that they are self-synchronizing, so if some bits are lost or modified in a T-code encoded stream, the decoder will regain synchronization automatically. The best T-codes achieve self-synchronization within 1.5 characters following a lock loss. Thus, we can use T-codes in place of Huffman codes in the algorithms such as Jpeg-Jsteg. The advantage of this approach is the ability to send steganographic messages in lossy environment that are robust against detection or attack.

A modified robust steganographical method using T-codes is proposed by Muttoo and Sushil [6] and is compared with steganography methods based on Jpeg-Jsteg and Outguess techniques. They have shown that using of T-codes as source encoding in place of Huffman codes result into better PSNR values.

In this paper we propose T-codes for the encoding of original message and for the entropy encoding of compressed stego-image in place of Huffman codes. The proposed scheme takes advantage of the synchronizing ability of T-codes to increase the robustness of popularly used hiding techniques like Jpeg-Jsteg.

2. PROPOSED ALGORITHM

We have developed a novel steganographic method based on Jpeg-Jsteg, famous hiding-tool based on joint photographic expert group(JPEG).The embedding and extracting algorithms are summarized as under:

Embedding Algorithm 2.1:

Input: secret message, cover image
Procedure:

Step1. Encode the message using the T-codes
Step2. Divide the cover image into 8x8 blocks
Step3. Calculate DCT coefficients for each block
Step4. Quantize the coefficients
Step5. **while** complete message not embedded **do**
5.1 get next DCT coefficient
5.2 **if** DCT ≠ 0 , DCT ≠1 and DCT ≠ -1 **then**
5.2.1 get next bit from message
5.2.2 replace DCT LSB with message bit
end{if}
end{While}
Step6. De-quantize and take inverse DCT to obtain stego-image
End.
Output: Stego- image

Extracting Algorithm 2.2:
Input: Stego image
Procedure:

Step1. Divide the stego image into 8x8 blocks
Step2. Calculate DCT coefficients for each block
Step3. Quantize the coefficients
Step4. **while** secret message not completed **do**
4.1 get next DCT coefficient
4.2 **if** DCT ≠ 0 , DCT ≠1and DCT ≠ -1 **then**
Concatenate DCT LSB to secret message
end{if}
end{while}
Step5. Decode secret message bits using the T-codes
end.
Output: Secret message

---

3. RESULTS AND ANALYSIS

The proposed algorithm has been implemented on number of images. The measures such as message capacity (Codeword length) and PSNR values for the two gray-level cover images of size 128 x 128, namely, 8.tif (figure 3.1) and 0.tif (figure 3.2) and four 256 x 256 pixels, namely, Lena, Baboon, Peppers and tree (figures 3.3 to 3.6) obtained from the proposed algorithm are compared with the corresponding Jpeg-Jsteg method.

The figures 3.7 and 3.8 are stego-images obtained by Jpeg-Jsteg using Huffman and proposed T-codes used Jpeg-Jsteg method.

The results of PSNR values obtained from the existing Jpeg-Jsteg and proposed methos are summarized in two tables: Table 3.1 and table 3.2.

In Table 3.1 , we find that when the embedding capacity is increased, PSNR values of the proposed method are close to the PSNR values of the existing method, showing that the proposed method is as good as the existing method

		Jpeg-Jsteg (Huffman)	Jpeg-Jsteg (T-codes)
Image	Message Capacity	PSNR	PSNR
Lena	4382	37.77	37.69
Baboon	6026	36.49	36.40

		Jpeg-Jsteg (Huffman)	Jpeg-Jsteg (T-codes)
Peppers	4403	37.77	37.83
Tree	5554	36.78	36.70

Table 3.1

In Table 3.2, we notice that PSNR values of the proposed method are better than the existing method

		Jpeg-Jsteg (Huffman)		**Jpeg-Jsteg (T-codes)**
Image	Code word length	PSNR	Code word length	PSNR
8.tif	852	34.97	904	35.26
0.tif	650	34.47	681	35.81

Table 3.2

Test Images (Figure 3.1 to 3.6)

Fig. 3.1 (0.tif (16.3 KB))

Fig. 3.2 (8.tif (4.10 KB))

Fig. 3.3 (Lena (93.7 KB))

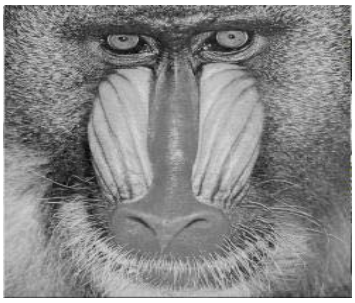

Fig. 3.4 (Baboon(97.0 KB))

Fig.3.5 (Peppers (89.6 KB))

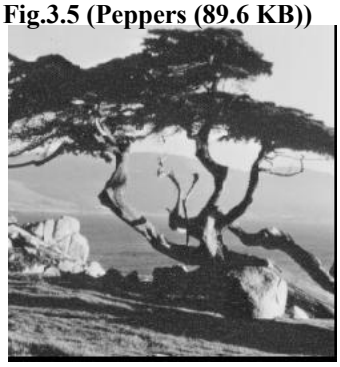

Fig. 3.6 (Tree (82.1 KB))

Stego- images (Figures 3.7 & 3.8):

Fig. 3.7 (PSNR = 37.77 dB) Jpeg-Jsteg (Huffman)

Fig. 3.8 (PSNR = 37.57 dB) Jpeg-Jsteg (T-codes)

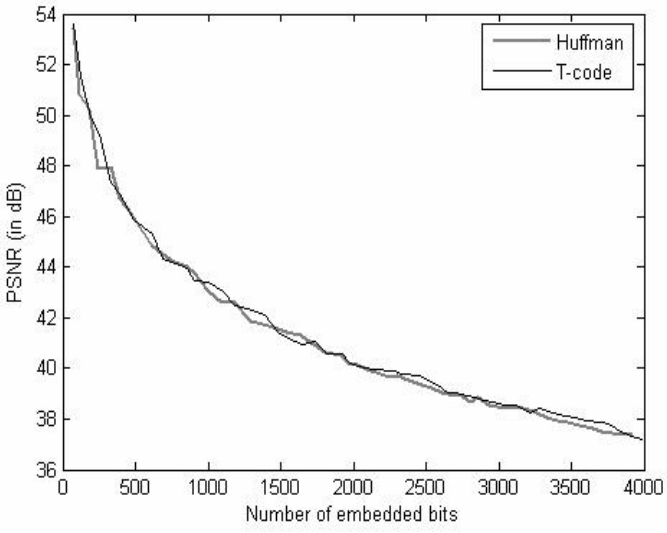

Figure 3.9: PSNR vs embedded bits

In the figure 3.9 we have shown the comparison of PSNR versus embedded message capacity by the original (Huffman based Jpeg-Jsteg) and the modified (T-code based Jpeg_Jsteg). The Dark lines are shown for T-code method where as light gray lines are for the Huffman method. We observe that the variation in PSNR values obtained with increasing values of embedded capacity is almost equal to the original method.

4. CONCLUSION

We observe from our experimental results that PSNR values of the proposed Jpeg-Jsteg algorithm based on T-codes for the different images are almost same as that of original algorithm based on Huffman codes, i.e., there is no change in the stego-image quality. Our method is secure in the way that even if the attacker detects (i.e., statistical attacks) and extracts the embedded message from the stego-image, he/she would not be able to recover the secret message without the encoded key. Moreover due to the inherent property of self-synchronizing of T-codes, our method is more robust as after the extraction process the recovered secret message is decoded and found to be without being much destroyed (for results one may refer to [12])

FUTURE WORK

Our approach has been to develop a Steganographic method that is perceptible and robust. It is known that JPEG approach can be statistically attacked even at one bit embedding. We are in the process of applying 'Best' T-codes as described by Ulrich Gunther [10] in place of simple T-codes to make the algorithm more secure from the attack such as filtering, cropping, noise etc. We are also applying this method to other data hiding techniques. This work will be presented in the form of a research paper in due course of time.

ACKNOWLEDGEMENT

The authors wish to thank their students Ram Kumar Karn, Rohit Choudhary and Kumar Sarth for their help in Matlab implementation.

REFERENCES

[1] Anderson, R.J. and F.A.P. Petitcolas, *" On the limits of steganography"*. IEEE J. Selected Areas in Commun., 16: 4, 1998.

[2] Chin-Chen Chang, Tung-Shou, and Lou-Zo Chung, *"A steganographic method based upon JPEG and quantization table modification"*, Information Sciences 141 ,123-138, 2002

[3] Fridrich J., Goljan M.& DU R, "Steganalysis Based on JPEG Compatibility', Special session on Theoretical and Practical Issues in Digital Watermarking and Data Hiding, SPIE Multimedia Systems and Applications IV, Denver, CO. August 20-24, 2001.

[4] G.R.Higgie, *"Analysis of the Families of Variable-Length Self-Synchronizing Codes called T-codes"*, Ph.D thesis, The University of Auckland, 1991.

[5] Johnson, N.F. and S. Jajodia,*"Exploring steganography: Seeing the unseen"*, IEEE Computer, 31: 26-34,1998.

[6] Muttoo S.K. and Sushil kumar ," Robust Steganography using T-codes", Proceeding of National Conference on Computing for nation development, IndiaCom 2007, Sponsored by AICTE, CSI , and IETE , Delhi, pp.221-223

[7] Niels Provos and Honeyman, *"Hide and Seek: an Introduction to Steganography"*, IEEE Security and Privacy, 32-43, May/June, 2003

[8] Titchener, M.R.,*"Technical note:Digital encoding by way of new T-codes"*, IEE Proc. E. Comput. Digit Tech., 1984, 131, (4),pp. 151-153.

[9] Titchener, M.R., *"Generalised T-codes: extended construction algorithm for self- synchronization codes"*. IEE Proc. Commun.,Vol. 143, No.3, 122-128, 1996

[10] Ulrich Gunther, "Robust Source Coding with Generalised T-codes", HhD Thesis, The University of Auckland, 1998. http;//www.tcs.Auckland.ac.nz/~ultivh/phd.pdf

[11] A. Westfeld and A. Pfitzmann, *"Attacks on steganogrphic systems"*, 3rd International Workshop on Information Hiding, 1999.

[12] Muttoo S.K. and Sushil kumar , "Image Steganography using Self-synchronizing variable codes ", International Conference on quality, reliability and Infocom technology, ICQRIT 2006.