Dct Based Image Steganographic Approach

Sonawane Viraj[1], Mali Jagdish[2], Prof.R.N. Awale[3]
[1,2,3] (Department of Electrical, V.J.T.I, Mumbai)

ABSTRACT

Steganography is the art of invisible communication. Its purpose is to hide the very 8presence of communication by embedding messages into innocuous-looking cover objects. In today's digital world, invisible ink and paper have been replaced by much more versatile and practical covers for hiding messages – digital documents, images, video, and audio files. As long as an electronic document contains perceptually irrelevant or redundant information, it can be used as a "cover" for hiding secret messages.

Each steganographic communication system consists of an embedding algorithm and an extraction algorithm. To accommodate a secret message, the original image, also called the cover-image, is slightly modified by the embedding algorithm. As a result, the stego-image is obtained.

This paper is a review of the DCT based steganography techniques appeared in the literature.

Keywords - DCT,DCT-BIT4, Image stegnography

I. INTRODUCTION

The term Steganography refers to the art of covert communications. By implementing steganography, it is possible for Alice to send a secret message to Bob in such a way that no-one else will know that the message exists. Typically, the message is embedded within another object known as a cover Work, by tweaking its properties. The resulting output, known as a stegogramme is engineered such that it is a near identical perceptual model of the cover Work, but it will also contain the hidden message. It is this stegogramme that is sent between Alice and Bob. If anybody intercepts the communication, they will obtain the stegogramme, but as it is so similar to the cover, it is a difficult task for them to tell that the stegogramme is anything but innocent. It is therefore the duty of steganography to ensure that the adversary regards the stegogramme - and thus, the communication - as innocuous.

In terms of development, Steganography is comprised of two algorithms, one for embedding and one for extracting. The embedding process is concerned with hiding a secret Message within a cover Work, and is the most carefully constructed process of the two. A great deal of attention is paid to ensuring that the secret message goes unnoticed if a third party were to intercept the cover Work. The extracting process is traditionally a much simpler process as it is simply an inverse of the embedding process, where the secret message is revealed at the end.

The entire process of steganography for images can be presented graphically as:

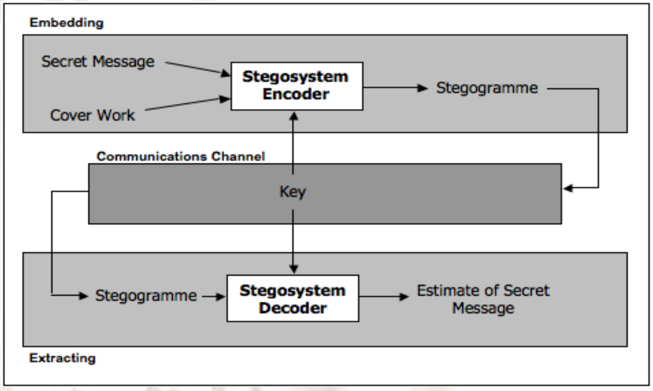

Figure 1. Shows one example of how steganography might be used in practice. Two inputs are required for the embedding process:

1. Secret message - usually a text file that contains the message you want to transfer
2. Cover Work - used to construct a stegogramme that contains a secret message

The next step is to pass the inputs through the Stego-system Encoder, which will be carefully engineered to embed the message within an exact copy of the cover Work, such that minimum distortion is made; the lower the distortion, the better the chances of undetectability. The stego-system encoder will usually require a key to operate, and this key would also be used at the extraction phase. This is a security measure designed to protect the secret message. Without a key, it would be possible for someone to correctly extract the message if they managed to get hold of the embedding or extracting algorithms. However, by using a key, it is possible to randomise the way the stegosystem encoder operates, and the same key will need to be used when extracting the message so that the stego-system decoder knows which process to use. This means that if the algorithm falls into enemy hands, it is extremely unlikely that they will be able to extract the message successfully.

The resulting output from the stego-system encoder is the stegogramme, which is designed to be as close to the cover Work as possible, except it will contain the secret message. This stegogramme is then sent over some communications channel along with the key that was used to embed the message. Both the stegogramme and the key are then fed into the stego-system decoder where an estimate of the secret message is extracted. Note that we can only ever refer to the output of the extraction process as an estimate because when the stegogramme is sent over a communications channel, it may be subjected to noise that will change some of the values. Therefore, we can never be sure that the message extracted is an exact representation of the original. Also, the recipient will obviously never know what the original message was, and so they have nothing to compare it to when it is extracted

The imperceptibility, robustness and security are the design issues for steganography. Robustness is the ability of stego image against different kind of attacks. Successful steganography depends upon the carrier medium not to attract attention. Security is the inability of adversary to detect hidden images accessible only to the authorized user.

II. RELATED WORKS

Image steganography schemes can be divided into two groups: Spatial/image Domain and Frequency/Transform Domain. Spatial domain techniques embed messages in the intensity of the pixels directly, while in transform domain, images are first transformed and then the message is embedded in the image [1]

Least Significant Bit (LSB) is the first most famous and easy spatial domain steganography technique. It embeds the bits of a message in a sequential way in the LSB of the image pixels [2]. But the problem of this technique is that if the image is compressed then the embedded data may be destroyed. Thus, there is a fear for damage of the message that may have sensitive information. Moreover, these kinds of methods are easy to attack by Steganalysis techniques.

Although those spatial hiding methods enable us to embed a great amount of information, they are not robust against attacks. The embedding process can be made in theLSB1, LSB2 or even in more significant bits such as System of Steganography using Bit-4 (SSB-4) by using the fourth bit of the pixel image [3].

Frequency domain steganography:-

Recalling that in spatial domain the data embed inside pixels directly, while in transform domain, images are first transformed and then the message is embedded in the image [1]. On the other hand, in the transform domain the embedding process can usually hide less information into pictures. There is no such an exact limit in the size of the embedded object as in the case of LSB insertion, where the number of pixels and the color depth determine the maximum size of the embedded data, while retaining the invisibility of occurred changes during embedding . However, by imbedding data in the transform domain, the hidden data resides in more robust areas, spread across the entire image, and provides better resistance against signal processing.

There are many techniques used to transform image from spatial domain to frequency domain and lossy image com-pression can be thought of as an application of such trans-form coding. The most common frequency domain methods usually used in image processing are the 2D DCT and Wavelet [4]. In this work, we utilize the DCT as an example of the transform coding technique that can be used.

The DCT helps separate the image into parts of differing importance (with respect to the image's visual quality). In practical, DCT can be carried out by partitioning/sectioning the image into equally size 2D blocks i.e., N × N grids (e.g., 8 × 8 grid containing 64 pixels per grid). With each grid a DCT coefficient for every component in the pixel is calculated. The formula used to calculate the DCT coefficient S(u, v) (for u, v = 0, 1, 2 . . .N − 1) of an image grid of pixels F(x, y) is given in Equation 1 [5] :

$$S(u,v) \quad = \frac{2}{N} C(u)\,C(v) \sum_{n=1}^{N-1} \sum_{n=1}^{N-1} \mathrm{F}(x,y) \left(\cos \frac{\pi u(2x+1)}{2N} \right) \quad * \left(\cos \frac{\pi v(2y+1)}{2N} \right) \tag{1}$$

Where $C(k)=\frac{1}{\sqrt{2}}$ when k=0 ,otherwise C(k)=1, and each F(x, y) pixel value has a level range from0 to 255 in 8 bits monochromic image. It should be noted that for most images much of signal energy lies at low frequencies; these appear in the upper left corner of the grid of DCT coefficients. Note that since these techniques modify only nonzero DCT coefficients, message lengths are defined with respect to the number of nonzero DCT coefficients in the images [10].

To reproduce a grid of image pixels F(x, y), (for x, y = 0, 1, 2 . . .N − 1), from the grid of DCT coefficients S(u,v), we

can use the inverse of the DCT formula given in Equation(2)

$$\mathrm{F}(x,y) = \frac{2}{N} \sum_{u=1}^{N-1} \sum_{v=1}^{N-1} C(u)C(v) * S(u,v) \left(\cos \frac{\pi u(2x+1)}{2N} \right) * \left(\cos \frac{\pi v(2y+1)}{2N} \right) \tag{2}$$

III. ALGORITHM

1. ALGORITHM TO EMBED IN SPATIAL DOMAIN:-

Read the Cover image and the binary message image .Read the Pixel element of the Cover image working from left to right and top to bottom. And simultaneously change the LSB of the pixel according to the corresponding message pixel value.

To retrieve the message image working from left to right and from top to bottom find the last bit of the each pixel and store it as secret message .

2. ALGORITHM TO EMBED IN FREQUENCY DOMAIN:-

Read the cover image and broke the image into the block of 8x8.from left to right and top to bottom take the DCT of the each block .Now embed the message bit into the LSB of the DCT Coefficient. Now write the stego image by taking the IDCT of the coefficient.

To retrieve the image from the stego image .Take the DCT of the stego image block by block .now read the coefficient and calculate the LSB and retrieve then message image .

3. ALGORITHM TO EMBED IN FREQUENCY DOMAIN WITH BIT 4 MODIFICATION:-

This is more or less similar to the above frequency domain embedding method .but the replecment of the bit changed from least significant bit to bit 4 .the changes occurred due to change in bit 4 the number varies by ± 8. To suppress the changes by it here we modify the corresponding number in such way that bit 4 remain same and the other bit altered so as to reduce the error occurred due to modification . If the bit to be embed is same as destination bit4.then no modification is done but, if it won’t matches then bit 1,2,3 &5 are modified in order to suppress the changes[6].

For example

Decimal	Bit to embed	Unmodified outcome	Modified outcome
01111000 (120)	0	01110000 (112)	011101111 (119)
10100111 (167)	1	10101111 (175)	10101000 (168)

IV. RESULTS AND DISCUSSION

Stego image depended on the type of the image and embedding algorithm. Hence we have tried the algorithm on various type of the image. Stego image tested against Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR).

It observed that PSNR of the DCT-LSB method found out to be more than LSB replacement in spatial domain and DCT-BIT 4 technique .but the secret image recovered by the DCT-BIT 4 technique was have more PSNR than any other technique . And it also found to be sustainable against the attack like histogram equalization.

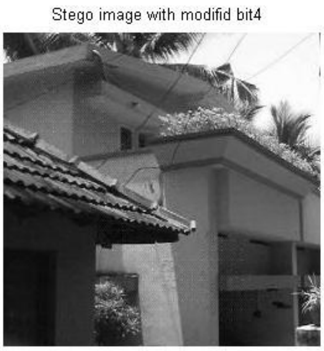

Fig. 1. DCT-BIT4 Stegogramme

PSNR=48.45

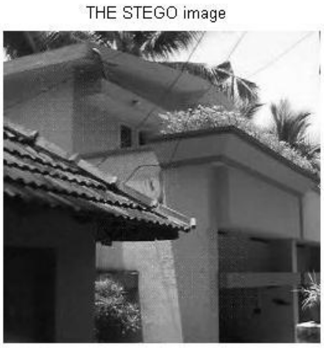

Fig.2 DCT-LSB stegogramme

PSNR=49.34

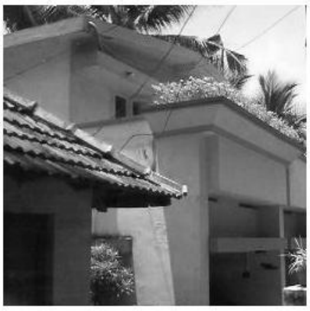

Fig.3- Stegogramme Spatial LSB

PSNR=42.78

Even though LSB replacement in spatial domain in spatial domain is quite easier than other two techniques. PSNR found out to be low compared to other.

V. CONCLUSION

From this paper we can conclude that DCT based embedding is more imperceptible than spatial domain embedding. It is also observed that DCT-BIT 4 more robust than the DCT-LSB based algorithm. As the changes in the higher bit result in spreading of the message bit over wide area. But it is also seen that the PSNR of the DCT-LSB is better than the DCT-BIT4 technique which conclude image quality preserved in better way in the DCT-LSB algorithm, taken as limitation of the DCT-BIT4 algorithm

REFERENCES

[1] S. Dickman, An Overview of Steganography, Research Report JMU-INFOSEC-TR -2007-002, James Madison University, July, 2007.

[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, “Techniques for Data Hiding”, I.B.M. Systems Journal, 35(3-4): pages 313-336, 1996.

[3] J. Rodrigues, J. Rios, and W. Puech “SSB-4 System of Steganography using bit 4”, In International Workshop on Image Analysis for Multimedia WIAMIS, (Montreux, May 2005).

[4] M. Kharrazi06, H. Sencar, and N. Memon, “Performance study of common image steganography and steganalysis techniques,” Communications of the SPIE and IS&T, 15, No.4 , 1017-9909, Oct-Dec., 2006.

[5] N. Provos, and P. Honeyman, “Hide and Seek: An intro-duction to Steganography”, Security & Privacy, IEEE, Vol.1,Issue 3, pp 32-44, May-June 2003.

[6] Nedal M. S. Kafri and Hani Y. Suleiman “Bit-4 of Frequency Domain-DCT Steganography Technique” 2009 IEEE Transaction.