Data Embedding Using Block Truncation Coding

Suhad Ahmed Ali

University of Babylon /Science Collage for Women

suhad_ali2003@yahoo.com

Abstract

In this paper a data hiding method is presented that hide the data in the compression domain. The Block Truncation Coding (BTC) method is used for hiding the information. The key point of the proposed method is to firstly apply traditional Block Truncation Coding and later it applied using Improved Block Truncation Coding (IBTC). Experimental results show that IBTC can embed a large amount of data in the compressed file while maintaining satisfactory image quality.

Keywords: BTC, information hiding, compression, steganography.

الخلاصة

في هذا البحث تم تقديم طريقة إخفاء البيانات التي تخفي البيانات في مجال الضغط. تستخدم طريقة ترميز اقتطاع الكتلة (BTC) لإخفاء المعلومات. النقطة الأساسية في الطريقة المقترحة هي أن يتم تطبيقها أولاً باستخدام ترميز اقتطاع الكتلة التقليدية ثم يتم تطبيقه لاحقًا باستخدام ترميز اقتطاع الكتلة المحسن (IBTC) .تظهر النتائج التجريبية أن IBTC تمكن من اخفاء كمية كبيرة من البيانات في الملف المضغوط مع الحفاظ على جودة صورة مرضية.

الكلمات الدالة: BTC، اخفاء المعلومات، ضغط، إخفاء المعلومات.

1. Introduction

Steganography is the specialty of sending secret data in which correspondence happens. To hide secret information there may be totally different approaches, such as cryptography and steganography. In cryptography, information is a smaller amount protected as a result of it attracts the eye of assaulter, whereas in steganography, the data is safer than the cryptography as a result of it hides the existence of knowledge. The most common technique is to use images to cover the data from the ways of steganography. Hiding the data in an image is termed image steganography [1]. Generally, in steganography, the particular data isn't maintained in its original format and thereby it's regenerate into another equivalent multimedia system file like image, video or audio that successively is being hidden at intervals another object. This apparent message (known as cowl text in usual terms) is distributed through the network to the recipient, wherever the particular message is separated from it. The majority of today's steganographic systems uses multimedia objects like image, audio, video etc. as cover media because people often transmit digital pictures over email and other Internet communication [2]. In recent approach, depending on the nature of cover object, steganography can be divided into five types:

- Text Steganography
- Image Steganography
- Audio Steganography
- Video Steganography
- Protocol Steganography

To use images as cover objects there are two techniques proposed. These techniques can be classified into the following two ways:

(1) Spatial domain techniques

these techniques directly embed secret information in the intensity of the pixels, while in transform domain, images are transformed firstly and then the message is embedded in the image. A steganographer modifies the secret data and the cover medium in the spatial domain [3], [4].

(2) Transform domain techniques.

These techniques use Transformation like Discrete Cosine Transform (DCT) or Discrete Wavelet Transformation (DWT). The cover images are transformed firstly and after that data is hide inside them. In transform domain techniques, data is hidden in mathematical functions [5],[6],[7],[8].

(3) Compression Domain

The compressed data are widely distributed in the internet transmission, the extra message conveying in a secret way also highly raises attention. For this reason, if secret data can be directly embedded into te compressed codes of the image, then we can spare all those decompression and recmpression process. Furthmore the compressed codes transimitied through the internet attract less attention than the raw data itself [9].

In this paper, we develope an image hiding method that can hide the secret data into compressed codes of host image, generated by the (BTC and IBTC) method. The rest of the paper is organized into four sections. The concept of BTC and IBTC comprssion algorithms is introduced in section 3. In section 4, the proposed hiding scheme is presented. Experimental results are given in section 5, and finally some conclusions are made in section 6.

2.Compression Methods

In this section two compression algorithms are explained (BTC and IBTC).

2.1 Block Truncation Coding

Block truncation coding (BTC) is a lossy compression technique for gray-level images proposed by Delp and MitchellIn [10]. In this scheme, the image is divided into non-overlapping blocks of pixels. For each block, threshold and reconstruction values are determined. The threshold is usually the mean of the pixel values in the block. Then a bitmap of the block is derived by replacing all pixels whose values are greater than or equal (less than) to the threshold by a 1 (0). The BTC algorithm involves the following steps:

• **Step1:** The given image is divided into non overlapping rectangular regions. For the sake of simplicity the blocks were let to be square regions of size m x m.

• **Step 2:** For a two level (1 bit) quantizer, the idea is to select two luminance values to represent each pixel in the block. These values are the mean ($\bar{X}$) and standard deviation(σ) which are computed according to the following formulas:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} Xi \qquad , \ldots\ldots \quad (1)$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Xi - \bar{X}i)\text{^}2} \quad , \ldots\ldots\ldots (2)$$

• **Step3:** The two values ($\bar{X}$) and σ are termed as quantizers of BTC. Taking ($\bar{X}$) as the threshold value a two-level bit plane is obtained by comparing each pixel value (x_i) with the threshold. A binary block, denoted by B, is also used to represent the pixels. We can use "1" to represent a pixel whose gray level is greater than or equal to x and "0" to represent a pixel whose gray level is less than.

$$B = \begin{cases} 1 & Xi \geq \bar{X} \\ 0 & Xi < \bar{X} \end{cases} \quad , \ldots\ldots \qquad (3)$$

• Step 4: In the decoder an image block is reconstructed by replacing '1's in the bit plane with (H) and the '0's with (L), which are given by:

$$H = \bar{X} + \sigma\sqrt{\frac{p}{q}} \qquad , \ldots\ldots \qquad (4)$$

$$L = \bar{X} - \sigma \sqrt{\frac{q}{p}} \qquad , \ldots\ldots \qquad (5)$$

$$X = \begin{cases} L & B = 0 \\ H & B = 1 \end{cases} \qquad (6)$$

Where **p** represents number of pixels whose values are greater than or equal to ($\bar{X}$) and **q** is the number of pixels whose values are less than or equal to ($\bar{X}$).

2.2.Improved Block Truncation Coding (IBTC)

In this method, the image have compressed using IBTC algorithm with multilevel quantization scheme [11]. The methodology adopted is shown below:

Step 1: break the original image into blocks of size 4x4 or 8x8 for processing.

Step2: Find the maximum and minimum pixel values in a block and store those in variables m_x and m_n respectively.

Step3: Find the values of thresholds for a particular block by using the formula:

$$thr = m_n + ((m_x - m_n)r/n) \qquad (7)$$

where $\mathbf{m_x}$ and $\mathbf{m_n}$ are the minimum and maximum intensities of the block respectively, ***thr*** represents the $\mathbf{r^{th}}$ value of threshold and **n** is the number of quantization levels.

Step 4: Compare each pixel gray scale value in the block with the threshold values and assign every pixel to one of the n quantization levels depending on the comparison of pixel values and thresholds.

Step 5: Encode the pixels in binary within each Quantization level using two bits. Compute the mean pixel value for each Quantization level.

Step6: Repeat the above steps for each block in the image.

Step 7: To reconstruct the image, assign the respective Quantized grayscale values to the pixels in each Quantization level within each block. In the decoder an image block is reconstructed by replacing '1's in the bit plane with (H) and the '0's with (L), which are given by

$$L = (1/(m-p)) \Sigma xi \quad , xi <= thr \qquad (8)$$

$$H = (1/p) \Sigma xi \quad , xi > thr \qquad (9)$$

Where **p** represents number of pixels whose values are greater than or equal to **thr,** and **m** is total number of pixels in the block .

3. The Proposed Data hiding Method

This section demonstrates how to embed the secret bits into a gray level host image and how to extract the data. The whole process can be divided into two phases: one is the data embedding phase, the other is the data extraction phase.

3.1 Data Embedding Phase

In the data embedding phase, the host image is compressed using BTC as described in section 2.The secret file is a bit stream. To increase the security of our method a block mapping method is used to select the blocks for embedding.

Data Embedding Algorithim

Input: Agray level cover image **(H)** of N×M pixels, secret information (**S)** with length **(L)**

Otutput: A embedding compressed file **(E)**

Begin

Step1: Apply one of compression method (BTC or IBTC) on cover image to get binary array (**B)**

Step3: Apply block mapping on (**B)** according to the following:

Before embedding the secret information a block mapping sequence is used. To get the one to- one mapping sequence equation (10) is used.

$$B' = [F(B) = (K * B) mod\ N] + 1 \qquad (10)$$

Where: B, B′ (∈[0, N _ 1]) is the block number, N(∈ Z– {0}) represent the total number of blocks in the image of size N =2^n × 2^n, and n ∈ N, and k (a prime number and K ∈ Z – { factors of N }),it is a secret key.

Step4: Apply XOR operation between compressed block (**B′**) and secret message **(S)** to get embedding compressed block (E)
Step5: Repeat the above steps until **(L)** secret bits are totally embedded.
Step 6: Return **(E)**
End

3.2 Data Extraction Phase

The extraction of data is relatively simpler than the embedding phase. For each encoded block, the secret block is extracted. The data extraction continue until all **(L)** secret bits are retrieved.

Data Extraction Algorithm:
Input: A gray level cover image **(H)** of N×M pixels , embedding compressed file **(E)**
Output: L secret bits
Begin
Step1: Apply block mapping on **(E)**
Step2: for each block apply XOR between **(H and E)**
Step3: if **L** secret bits are not retrived yet, go to step1.
End

4. **Experimental Results**

To test the suggested method, the standard grey-levels images Lena and Cameraman of size 512×512 are used as host (cover) images. In order to test the performance of the proposed scheme, several benchmarking criteria have been used. These criteria **are Peak Signal to Noise Ratio (PSNR), and Capacity.**

The PSNR measures the quality of the extracted image in comparison to the original image (host). It is a standard way of measuring image fidelity. The PSNR is defined as follows [12]:

$$\text{PSNR}=10\log_{10}\left[\frac{(\text{I}_{\text{MAX}})^2}{\text{MSE}}\right] \qquad (11)$$

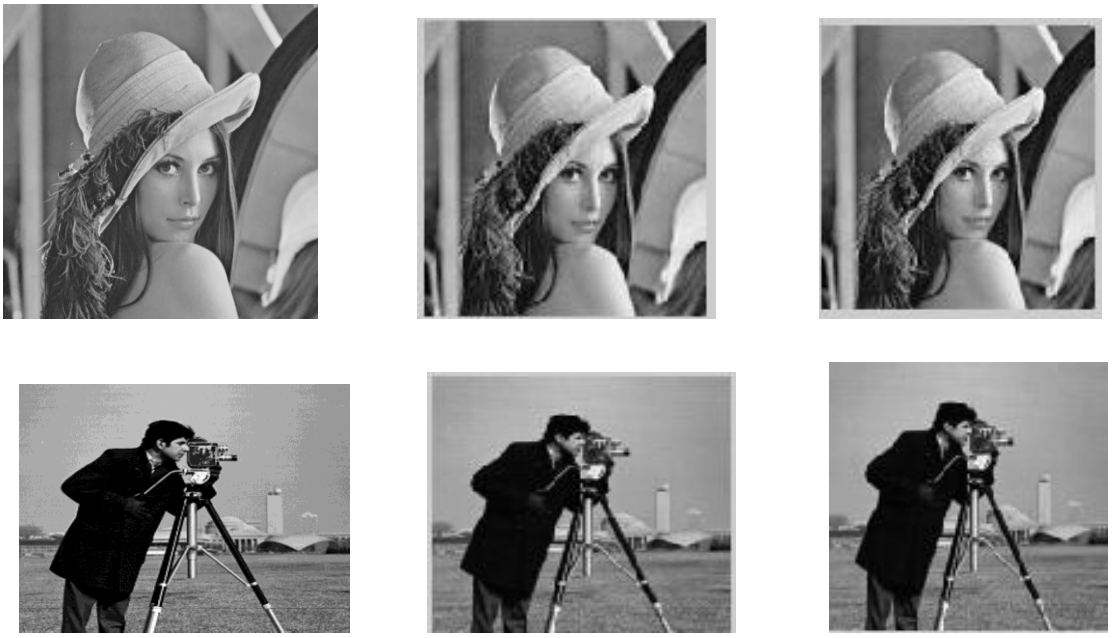
Where I_{MAX} is the maximum grey level of the image. In this case, I_{MAX} can have a maximum value of 255. MSE is Mean Square Error which is defined as follows:

$$\text{MSE}=\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left(p_1(i,j)-p_2(i,j)\right)^2 \qquad (12)$$

Where p1 (i, j) and p2 (i, j) represent two images, m, n represent the dimensions of two images.

Capacity refers to the how much secret information can be hidden into an image. Higher the capacity of the image to hide data better will be the technique. According to K Suresh Babu [13](Suresh Babu et. al., 2008) capacity represented by bpp that is bits per pixel and MHC (Maximum Hiding Capacity) in terms of percentage.

Figure (1) shows the original images of (Lena and cameraman) in (a), the reconstructed image in (b), and the embedding image in (c) by applying traditional BTC algorithm.



(a) **(b)** **(c)**

Figure (1) Results of applying traditional BTC (a) Original image(b) Reconstructed image (c) Embedding image with payload (1024)

The effect of embedding on the perceptual invisibility is tested by using the PSNR. Table (1) shows the amount of payload and the PSNR values for both reconstructed and embedding images.

Table (1) PSNR values by applying BTC with different amount of payload

Image name	***PSNR Btween (orginal and reconstructed) images***	***PSNR Btween (orginal and Embedding)images***	***Payload (bits)***
Lena	28.3688	28.3675	1024
Cameraman	26.6991	26.6986	1024
Lena	25.6773	25.6773	262144
Cameraman	24.9498	24.9498	262144

Figure (2) shows the original images of (Lena and cameraman) in (a), the reconstructed image in (b), and the embedding image in (c) by applying Improved IBTC algorithm.

(a) **(b)** **(c)**

Figure (2) Results of applying traditional IBTC (a) Original image(b) Reconstructed image (c) Embedding image with payload (1024)

Table (2) shows the amount of payload and the PSNR values for both reconstructed and embedding images.

Table (2) PSNR values by applying IBTC with different amount of payload

Image name	***PSNR Btween (orginal and reconstructed) images***	***PSNR Btween (orginal and Embedding)images***	***Payload (bits)***
Lena	33.6190	33.6179	1024
Cameraman	32.1992	32.1988	1024
Lena	33.6190	29.8978	262144
Cameraman	32.1992	29.99	262144

5. Conclusions and Future Works

In this paper, we have proposed an information hiding method to hide secrete data into compression domain of host image, generated by block truncation coding method. Experimental results show that IBTC can embed a large amount of data in the compressed file while maintaining satisfactory image quality. For future works, to more maintain the image quality, it can use a method to select appropriate blocks for embedding.

References

[1] Channalli, S. and Jadhav, A., "Steganography an art of hiding data", International Journal on Computer Science and Engineering Vol.1(3), PP. 137-141, 2009.

[2] B.Pfitzmann,"Information Hiding Terminology", Proc.of First Int. Workshop on Information Hiding, Cambridge, UK, May30-June1, Lecture notes in Computer Science, Vol.1174, Ross Anderson(Ed.), pp.347-350, 1996.

[3] Cheddad, A., Condell, J., Curran, K. and Kevitt,"Digital image steganography: survey and analysis of current methods, Signal Processing Elsevier, vol. 90, no. 3, 2010, pp. 727-752, 2010.

[4] Jassim, F. A.,"modulus method for image transformation", International Journal of Advanced Computer Science and Applications, vol. 4, no. 2, pp. 267–271, 2013.

[5] Juneja, M., Sandhu, P.S,"Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption, International Conference on Advances in Recent Technologies in Communication and Computing, pp. 302-305., 2009.

[6] Krenn. R.," Steganography and Steganalysis", Internet Publication, 2004.

[7] Prabakaran. G, Bhavani. R.,"A Modified Secure Digital Image Steganography Based on Discrete Wavelet Transform, International Conference on Computing, Electronics and Electrical Technologies (ICCEET), pp. 1096-1100, 2012.

[8] Kumar, V. and Kumar, D., "Performance Evaluation of DWT Based Image Steganography", IEEE 2nd International Advance Computing Conference, 2010, pp.223-228.

[9] Chang, Chin-Chen, and Wen-Chuan Wu.:, "A steganographic method for hiding secret data using side match vector quantization." IEICE Transactions on Information and Systems88, no. 9 ,pp.2159-2167, 2005.

[10] E.J. Delp and O.R. Mitchell, "image compression using block truncation coding", IEEE TRANS, Vol.27, PP 1335-1342, 1979.

[11]Shandilya, M., & Shandilya, R, "Implementation of Absolute moment Block Truncation Coding Scheme Based on Mean Square Error Criterion. In Proceeding of the SDR 03 Technical Conference and Product Exposition, , 2003.

[12] Shiva Kumar, K. B., Raja, K. B., Chhotaray, R. K. And Pattnaik, S., "Coherent Steganography using Segmentation and DCT", Computational Intelligence and Computing Research(ICCIC), IEEE-978-1-4244-5967- 4/10, pp. 1-6, 2010.

[13] Suresh Babu, K., Raja, K.B., Kiran Kumar K, Manjula Devi T H, Venugopal K R, Patnaik,L.M., "Authentication of Secret Information in Image Steganography", IEEE Region 10 Conferencepp. 1-6, 2008.