Comparison analysis of spatial Domain and compressed Domain steganographic techniques

Subhashini.D
Asst.prof
MGIT

Nalini.P
Asst.prof
MGIT

Chandrasekhar.G
Sr Analyst
Bank of America

Abstract:

Steganography plays an important role in the past few years due to the increasing need for providing secrecy in an open environment like the internet. Many techniques are used to secure information such as Cryptography that aims to scramble the information sent and makes it unreadable while steganography is used to conceal the information so that no one can sense its existence. In most algorithms, to secure information both steganography and cryptography are used together to secure a part of information. Steganography has many technical challenges such as high hiding capacity and imperceptibility. In this paper we compared two proposed techniques, one with wavelet transforms and other with block truncation coding. First method hides secret data in the integer wavelet coefficients of the cover image with the optimum pixel adjustment (OPA) algorithm. The coefficients used are selected according to a pseudorandom function generator to increase the security of the hidden data. The OPA algorithm is applied after embedding secret message to minimize the embedding error. Second method hides information by Block truncation coding (BTC) by using two quantization levels and a bit plane. Here hiding capacity is independent of the compressed codes. Comparison between two techniques is done by Mean Square Error (MSE), Peak Signal to Noise Rate (PSNR), length of secret message and received hidden secret message.

Keywords - Steganography, adaptive algorithm, spatial domain, integer wavelet transform, discrete wavelet transform, optimum pixel adjustment algorithm, quantization levels, Bit plane formation.

Introduction:

Steganography is the art and science of hiding secret data in plain sight without being noticed within an innocent cover data so that it can be securely transmitted over a network. The word steganography is originally composed of two Greek words Steganos and Grahiia, which means “covered writing “.The use of Steganography dates back to ancient times where it was used by roman’s and ancient Egyptians. The interest in modern digital steganography stated by Simmonsin 1983[1]

Any digital file such as image, video, audio, text or IP packets can be used to hide secret message .Generally the file used to hide data is referred to as cover-object, and the term Stego-object is used for the file containing secret message.

Among all digital file formats available now a days image files are the most popular cover objects because they are easy to find and higher degree of distortion tolerance and high hiding capacity due to the redundancy of digital information of an image data.

There are number of schemes that hide secret message in an image file. We have two popular types of hiding methods which are spatial domain reversible steganography and compressed domain reversible steganography.

In this paper we compared two proposed techniques which are performed in spatial domain and compressed domain. The basic idea behind spatial domain is direct replacement of LSB’s of noisy or unused bits of the cover image with the secret message bits. It provides high hiding capacity and provides a very easy way to control stego image quality [2]. But it has low robustness to modifications made to the stego image[3]and the basic idea behind the compressed domain is that it modifies the coefficients of the compressed code by using optimum pixel adjustment techniques. It offers high computational cost, low hiding capacity and low stego image quality

However most of them are focused on spatial domain, rare of them addressed the reversibility on the compressed domain. In 2007, changet al [4] proposed a new reversible data hiding tech in the VQ –compressed domain which has the benefits of hiding efficiency of embedding and extraction process.

Basic block diagram for steganography:

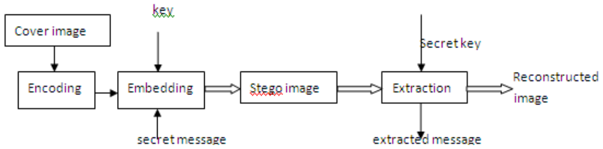

Figure 1.General block diagram for stegnography

Steganography using spatial domain:

In this technology embedding is done by using Integer Wavelet Coefficients. Generally wavelet domain allows to hide data in regions that the Human Visual System (HVS) is less sensitive to the hiding resolution detail band (HL, LH, HH). Hiding data in these regions allows us to increase the robustness while maintaining good visual quality. Integer wavelet transform maps an integer data set into another integer data set. LL sub band in IWT appears to be a close copy with smaller scale of original image[5].

Integer wavelet transform coefficients are calculated by using Haar transform i.e.

$$d_{ln}=s_{0,2n+1}-s_{0,2n},\ s_{l,n}=s_{0,2n}+d_{ln/2}$$

Inverse transform can be calculated by:

$$s_{0,2n}=s_{l,n}+d_{l,n/2},\ s_{0,2n+1}=d_{l,n}+s_{0,2n}$$

Where $s_{i,l}$, $d_{i,l}$ are the nth lower and higher frequency wavelet coefficients at the i^{th} level respect[5]

Processing steps: spatial domain analysis

A) Embedding steps:

1) Read the cover image file into 2 dimensional decimal array.

2)Modify the Histogram to avoid over/under flow by mapping the lowest15 gray scale levels to value of 15 and 15 gray scale levels to the value of 240[6],[7].

3) Divide the cover image into 8*8 non over lapping blocks

4) Apply Haar integer wavelet transform resulting LL1, LH1, HL1and HH1

5) Calculate hiding capacity of each coefficient, we used a modified version of hiding capacity function. The length of LSB's of wavelet coefficients (L) is determined by

$$L=\left\{\begin{matrix} k+3 & if & c_0 \geq 2^{k+3} \\ k+2 & if & 2^{k+2} \leq c_0 < 2^{k+3} \\ k+1 & if & 2^{k+1} \leq c_0 < 2^{k+2} \end{matrix}\right\} 1\leq k \leq 4$$

Else

L=k if $c_0 < 2^{k+1}$

Where C_0 is the absolute value of wavelet coefficients and k is the minimum length to be used in each coefficient. We can use any one of the conditions depending on application.

6) Embed L bits of message into the corresponding randomly chosen coefficients.(Random selection of coefficients provides more security)

7) Apply optimum pixel adjustment algorithm[7], by using below equation:

$\delta_i(x,y)=P_i'(x,y)-P_i(x,y)$ where δ_i is difference between original P_i (x,y)and the modified values $P_i(x,y)$.After calculating the δ_i the algorithm [8]modifies the changed value in the following manner:

case1: (k=1) $-2^k < \boldsymbol{\delta}_i < -2^{k-1}$
If $P_i'(x,y) < 256-2^k$,
Then $P_i'(x,y)^* = P_i'(x,y) + 2^k$
Else $Pi'(x,y)^* = P_i'(x,y)$
case2 (k=2) $-2^{k-1} \leq \boldsymbol{\delta}_i \leq 2^{k-1}$
$P_i'(x,y)^* = P_i'(x,y)$
Case3: (k=4) $2^{k-1} < \boldsymbol{\delta}_i < 2^k$
If $P_i'(x,y) \geq 2^k$,
Then $P_i'(x,y)^* = P_i'(x,y) - 2^k$
Else $Pi'(x,y)^* = P_i'(x,y)$

8) Finally, calculate the Inverse Integer Wavelet Transform on each 8*8 block to restore the image to spatial domain.

B) Extraction steps:

1) Read the image file pixel values to a 2D decimal matrix each value represents the pixel value intensity.
2) Divide the cover image into 8*8 non over lapping blocks.
3) Transform each block to frequency domain with 2D HWT and get 4sub bandsLL1, LH1, HL1, HH1.
4) Calculate the number of bits (L) to hide data of each wavelet co efficient.
5) Use secret key to generate the secret coefficients to embed secret data.
6) Extract L from each selected coefficients.
7) Gather all extracted bits together to form the secret data back in order.

Steganography using compressed reversible domain[9-10]:

In this section, steganography for Block Truncation Coding -compressed images is introduced. For convenience, we defined the notations used in this paper first. The original host image *O* is a grayscale image. The AMBTC-compressed code for *O* is denoted by *C*, and the reconstructed AMBTC compressed image is denoted by *E*. Embedding is done by modifying *C*, and the result is an AMBTC compressed stego-code C'. The original host image is first partitioned into a set of non overlapping blocks of n×n pixels. These image blocks can be viewed as k-dimensional vectors, where k=n×n.Each image block is then compressed by using two quantization levels and b and one bit plane B.let $o_i$be image block i,$o_{i1}, o_{i2}, o_{13,} \dots . o_{ik}$be the pixels of image block i.Two quantization levelsa_i and $b_i$are calculated as follows:

$$a_i=\frac{1}{k-q_i}\sum_{0_{ij<\bar{o}_i}} o_{ij} \text{ and}$$

$$b_i = \frac{1}{q_i}\sum_{o_{ij\geq\bar{o}_i}} o_{ij}$$

Here q_i denotes the number of pixels having a value higher than or equal to the block mean $\bar{o}_i$.

The receiver then uses the restoring process to obtain the AMBTC-compressed stego-image E'. The AMBTC compressed code *C* consists of a sequence of trios (two quantization level *a* and *b*, and a bit plane *B*). Each trio (*a*, *b*, *B*) represents the compressed code for an image block. Note that if we interchange two quantization levels *a* and *b*, and perform Logical NOT operation on the bit plane *B*, the reconstructed AMBTC image blocks will remain the same. Let *R*() denotes the reconstruction function for AMBTC compressed image blocks with 3 parametes *a*, *b* and *B*, then the following equation always hold true for every trio (*a*, *b*, *B*) :

$$R(a, b.B) \equiv R(b, a, \bar{B})$$

where $\bar{B}$ is the resultant of the Logical NOT operation on the bit plane *B*. Above equation implies that the reconstructed image *E* and the recovered stego-image E' are exactly the same, i.e.*E*≡*E*'. Since the interchange of the quantization levels together with logical NOT operation on bit the plane *B* does not change the value of decoded image blocks, we may adopt this property to embedded one bit into each trio without losing any image quality. The embedding procedures are described in the following section.

1)Embedding steps:

The process of producing a stego-image is described below. Suppose an AMBTC compressed code *C*, to be embedded, is composed of $N \times N$ trios. The embedding algorithm of the proposed method follows the steps listed below:
Step 1. The *m*-bits secret data *S* ($m \leq N \times N$) is first appended by 0's of size $N \times N - m$, and then XORed with a random binary sequence generated by a secret key *k* . The encrypted bit stream is denoted by $E_{bs=}\{e_1,e_2,\dots . e_{N\times N}|e_i\varepsilon\{0,1\},i=1,2,\dots\dots\dots N\times N\}$.
Step 2. Sequentially embed secret data into AMBTC encoded blocks. For each AMBTC encoded block *I* with trio (a_i,$b_{i,}$$B_i$), if the corresponding embedded bit e_i=1, then the trio is changed to(b_i,a_i,$\bar{B}_i$) ,Otherwise, leave the trio is remain unchanged.
Step 3. Repeat step 1 and step 2 until the entire encrypted bit stream *E* is embedded.
Step.4. With these new trio values stego image is formed
.

2) Extraction procedure:

Step 1: The data extraction procedures are similar to that of the embedding procedures. For each AMBTC compressed block *i*, if $a_i < bi$, then the embedded secret bit e_i =0 is extracted. Otherwise, *ei* =1 is extracted. This procedure is repeated until all the encrypted secret data E_b have been extracted.
Step 2: Decrypted using the secret key *k* to obtain the original secret data *S*.
Step 3: By doing reverse operation original trio values are obtained, which gives reconstructed image.

Experimental Results and Discussion:

The proposed systems were applied to two typical 512*512 gray scale images Baboon and Lena. The program was implemented using matlab7 running on 1.73G dual core processor under Windows Vista. The test images are given below:

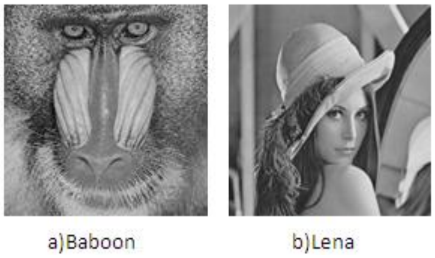
a)Baboon b)Lena

Fig 2 Test images 512*512 pixels

The Peak Signal to Noise Ratio (PSNR)[9] is used to measure the distortion between the original image and the stego image. The computation of PSNR is defined as

$$PSNR = 10\ log_{10} \frac{255^2}{MSE},$$

Where

$$MSE = \frac{1}{M \times N} \sum_1^M \sum_1^N (p(x,y) - p'(x,y))^2$$

MSE is mean square error p(x, y) stands for the image pixel value in the cover image and p'(x,y) stands for the stego image pixel values

A high value of PSNR means better image quality(less distortion), it is recorded that in grayscale images that the human visual system (HVS) can not detect any distortion in stego image having PSNR that goes beyond 36db

Comparison results :

By Integer Wavelet Transform:Lena

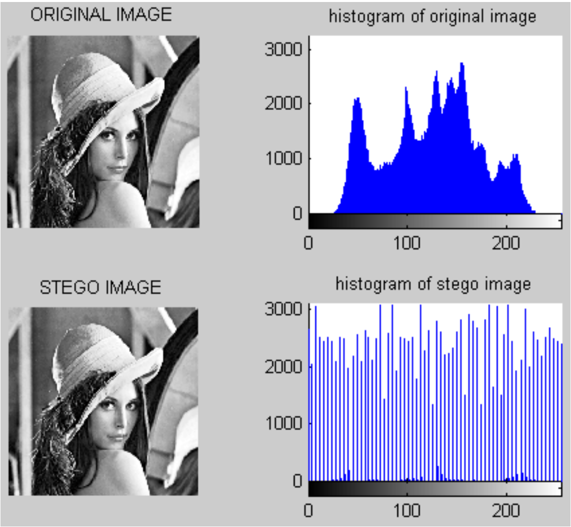

Figure3.Histogram representations for original image and stego image by spatial domain (by integer wavelet transform) with the test image of Lena 512*512 size **PSNR=56.0354**

.

By Integer Wavelet Transform:Baboon

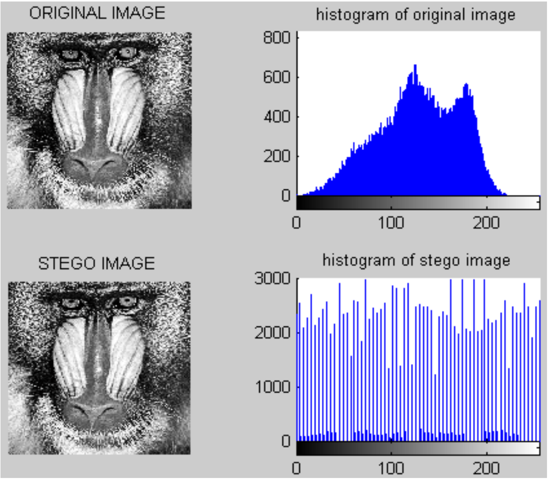

Figure4.Histogram representations for original image and stego image by spatial domain (by integer wavelet transform) with the test image of baboon 512*512 size **PSNR=57.2332**.

By Block Truncation Coding:Lena

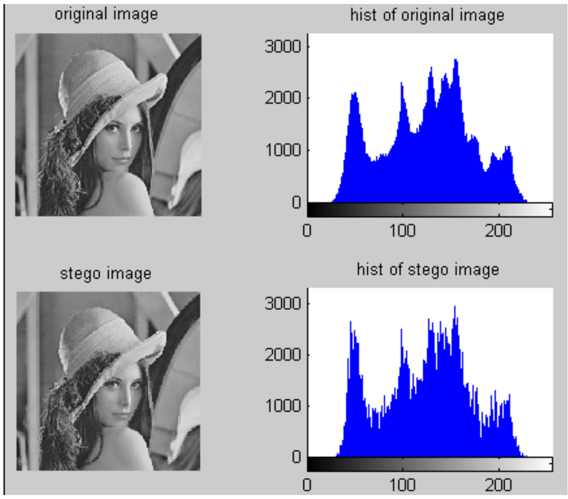

Figure5.Histogram representation for original image and stego image by compressed domain(by Block truncation coding) domain with the test image of Lena 512*512 size **PSNR=43.2103**

Test image	Method	Stego image quality (PSNR)	Em-bedding capacity
Lena	Integer Wavelet transform	57.1487 (key=1)	986408 bits
		57.0018 (key=2)	
		57.2332 (key=3)	
	Block truncation coding	43.2103 (k=4)	16,384 bits
		39.915 (k=8)	
Baboon	Integer Wavelet transform	56.0224 (key=1)	1008593 bits
		56.0089 (key=2)	
		576.035 4 (key=3)	
	Block truncation coding	35.1742 (k=4)	*16,384* bits
		33.9462 (k=8)	

By Block Truncation Coding:Baboon

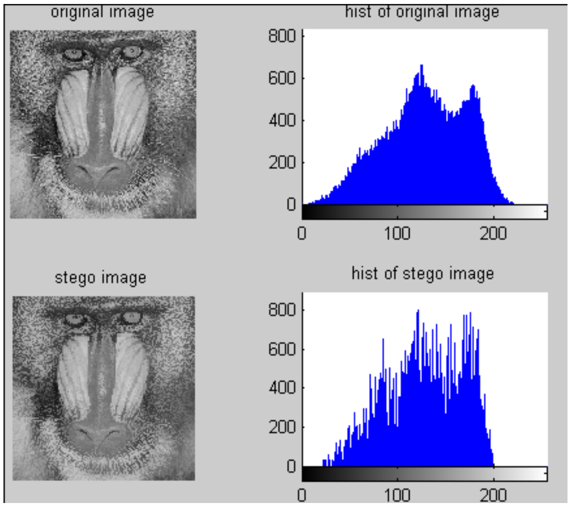

Figure6.Histogram representation for original image and stego image by compressed domain (by Block truncation coding) domain with the test image of baboon 512*512 size **PSNR=39.915**.

Comparison of image quality and embedding capacity:

Conclusions:

In this paper, we compared two proposed steganographic techniques; one is based on spatial domain (by Integer Wavelet Transforms) other one is based on compressed domain (Block Truncation Coding). It is well known that, although the data hiding techniques can recover the original image after the extraction of secret data, the embedding distortion needs to be kept as low as possible in order to achieve perceptually invisible. In our schemes, the process of data embedding does not introduce any image distortion, which should be the best case for steganography.

In AMBTC compressed images, extraction of secret data directly from the compressed domain without decompressing beforehand. Therefore, the data extraction procedures are efficient, and can be applied to real time imaging processing or monitoring. Experimental results show that this scheme can embed secret data in the compressed domain while maintains the same image quality as the original AMBTC-compressed image. Besides, the embedded data can also be completely extracted. Future works may include trying to increase the payloads value and PSNR value.

In Integer Wavelet transforms, it combines an adaptive data hiding technique and the optimum pixel adjustment algorithm to increase the hiding capacity of the system. This technique embeds secret data in a random order using a secret key only known to both sender and receiver. It is an adaptive system which embeds different number of bits in each wavelet coefficients according to a hiding capacity without sacrificing the visual quality of resulting stego image. This technique minimizes the difference between original coefficients values and modified values. This technique provides high hiding capacity but it suffers with low robustness against various attacks such as histogram equalization and jpeg compression.

Further developments are done by using the advantages of two techniques to get higher hiding capacity, good PSNR value and high robustness.

References:

[1] Fabien A. P. Petit colas, Ross J. Anderson and Markus G. Kuhn, “Information Hiding—A Survey,” Proceedings of the IEEE, special issue on protection of multimedia content, 87(7), 1999, pp. 1062–1078.

[2] Shieh, J. M., Lou, D. C. and Chang, M. C., "A Semi-blind Digital Watermarking Scheme Based on Singular Value Decomposition," Computer Standards & Interfaces, Volume 28, Issue 4, 2006, pp. 428-440.

[3] Lin, P. L., Hsieh, C. K. and Huang, P. W., "A Hierarchical Digital Watermarking Method for Image Tamper Detection and Recovery," Pattern Recognition, Volume 38, Issue 12, 2005, pp. 2519-2529.

[4] Chang, C. C., Chou, Y. C. and Lin, C. Y., (2007):"Reversible Data Hiding in the VQ Compressed Domain," IEICE Transactions on Information and Systems, Vol.E90-D No.9, 2007, pp. 1422-1429.

[8] C.K. Chan and L.M. Cheng, "Hiding data in images by simple LSB substitution", pattern recognition, pp.469-474, mar, 2004.

[6][9]H.H.Zayed, "A High Hiding Capacity Technique for Hiding data in Image based on K-bit LSB substitution", the 30th International Conference on Artificial Intelligence Applications (ICAIA-2005) Cairo, Feb. 2005

[10] Chang, C. C., Chou, Y. C. and Lin, C. Y., (2007):"Reversible Data Hiding in the VQ-Compressed Domain," IEICE Transactions on Information and Systems, Vol.E90-D No.9, 2007, pp. 1422-1429.

[11] Chang, C. C. and Lin, C. Y., "Reversible Steganography for VQ-compressed Images Using Side Matching and Relocation," IEEE Transactions on Information Forensics and Security, Vol. 1, No. 4, 2006, pp. 493-501.

[5] s.lee, C.D Yoo and T.Kalker,"Reversible image watermarking based on integer to integer wavelet transforms"IEEE Transactions on information forensic and security ,vol2, No.3, sep 2007, pp321-330

[6]B.lai and L.Chang,"Adaptive Data Hiding for Image Based on Haar Discrete wavelet transform", Lecture Notes in computer science ,volume 4319/2006.

[7]G.Xuan, J.Zhu, Y.Q.Shi, Z.Ni and W.Su "Wavelet Transforms that map integers to integers". Applied and Computational Analysis, vol.5, no.3, pp.332-369,1998.