Spatial Domain Steganography Using Bit- 4 of DCT Coefficients

Nedal Kafri[1], Hani Suleiman[2]
[1]Department of Computer Science and IT
Al Quds University
Main Campus, Abu-Dies
B.O.Box: 20002, Jerusalem
Palestine
[2]Hulul Business Solutions
P.O.Box:4167,Al-Bireh Palestine
nkafri@science.alquds.edu, Hani.Suleiman@hulul.com

ABSTRACT: *Steganography is the art and science of concealing secret information in such a way that no one apart from the sender and intended recipient even realize that there is hidden information. In this paper, we describe a new method of steganography based on embedding message bits in the 4th bit of the coefficients of a transform domain, such as the discrete cosine transform (DCT) and Wavelet, of an image. The proposed technique utilizes the idea of SSB-4 technique in modifying the other bits (i.e., 1st, 2nd, 3rd and/or 5th), to obtain the minimum variation between the original and the modified coefficient. Since this approach uses significant bit, the hidden message resides in more robust areas, spread across the entire stego image, and provides better resistance against compression and steganalysis processes. The obtained experimental results also indicate that; the proposed method is an efficient and acceptable steganogaphy scheme.*

Keywords: Security, Steganography, Image Processing , Spatial Domain, Frequency Domain

Received: 11 September 2010, Revised 31 March 2010, Accepted 2 May 2011

1. Introduction

Security of information is one of the most important issues of information technology and communication. Security of information often lies in the secrecy of its existence and/or the secrecy of how to decode it. Cryptography techniques often use the worst approach assuming that only one of these two conditions holds [6]. It was created as a technique for securing the secrecy of communication. Various methods have been developed to encrypt and decrypt data in order to keep the message secret. Unfortunately, it is not enough to keep the content of the information/message secret, it may also be necessary to keep the existence of the information secret. The technique used to implement this, is called steganography.

Steganography is the science and art of hiding information in another. The definition according to Neil Johnson "Steganography is the art of hiding information in a way that prevent the detection of hidden message" [10]. It is a useful tool that allows covert communication amongst acknowledged parties. The word steganography is derived from the Greek words "stegos" meaning cover/hidden/roof and ``grafia'' meaning writing [2][6][10][14][21][23] defining it as "covered writing" and essentially means "to hide in plain sight". In image steganography the information is hidden exclusively in images. Hiding messages by masking their existence is nothing new. Before the digital era, simple steganographic techniques have been in use for hundreds of years. However, with the emergence of networks and digital technologies and increasing use of communication and files in electronic format, new techniques for information hiding have become possible.

Through history, people have hidden information by a multitude of methods and variations. For example, the Chinese civilization used to write messages in thin silk which was then rolled up to make tiny balls and covered in wax. Greeks wrote text on wax-covered tablets. Another ingenious way was to "tattoo" a message or image on a messenger's shaved head. One of the most renowned steganography skills from the 1st century through World War II invisible ink were often used to conceal hidden messages. In another form, while Paris was under siege in 1870, messages were sent by carrier pigeon. A Parisian Photographer used a microfilm technique to enable each pigeon to carry a higher volume of information which leads to the invention of microdot [6]. Between World War I and W. W. II Germany and later many countries used microdot for passing steganographic messages through insecure postal channels. More historical examples can be found in [2][6][10].

The modern formulation of steganography is often given in terms of the prisoner's problem where Alice and Bob are two inmates who wish to communicate in order to hatch an escape plan. However, all communication between them is examined by the warden, Wendy, who will put them in solitary confinement at the slightest suspicion of covert communication. Specially, in the general model for steganography, we have Alice wishing to send a secret message *m* to Bob. In order to do so, she "embeds" message into a cover-object, and obtains a stego-object.

Steganography relies on hiding covert message in unsuspected text, protocols, images, and multimedia (audio/video) data which is generally used in secret communication between acknowledged parties. Also steganography is a method of encryption that hides data among the bits of a cover file, such as a graphic or an audio file. The technique replaces bits with the secret data. Steganography is widely used in image [5]. The most common image formats include BMP, GIF and JPEG. Therefore, in this paper we focus on hidden message in this media type. Generally, the main idea of the proposed method is to hide message bits in a significant bit of transform domain coefficients of an image and inverse it to spatial domain. The most well-known transform coding techniques used to implement lossy image compression (such as JPEG format) are the discrete cosine transform (DCT) and the Wavelet. As a case of the proposed method, in this paper we use the 4th bit of the DCT coefficients of an image and inverse the DCT coefficients to spread the hidden information through the produced stego image.

The rest of this paper is organized as follows: Section 2 gives a background regarding the main schemes of steganography; spatial domain and frequency domain, and their evaluation techniques. While Section 3 introduces the proposed steganography algorithm Section 4 presents and discusses the obtained experimental results. Finally, Section 5 concludes the paper.

2. Background

Most of the camouflage processes use the redundant bits of an image to embed secret messages. Redundant bits are all bits that can be modified without perceptible change in the visual feature of a digital picture/image. It should be noted that pixels of most images are represented as triples (Red, Green, and Blue contributions), and the Blue one is the most imperceptible to human eye [1]. Each color is represented by a number of bits depending on the desired color of the final image. The number of bits of a color scheme is called the "bit depth" and refers to the number of bits used for each pixel. The typical bit depth used for grayscale and monochrome images is 8 bits, and for digital color image is 24 bits [14][16]. In the case of a 24 bitmap, each base color (Red, Green, and Blue) in a pixel has eight bits. Note that grayscale images can be obtained when the values of the (Red, Green, and Blue) are equivalent (from 0 0 0 to 255 255 255).

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 [6].

2.1 Spatial domain steganography

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 [3][5]. 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 [14]. Moreover, these kinds of methods are easy to attack by steganalysis techniques. LSB has been improved by using a Pseudo-Random Number Generator (PRNG) and a secret key in order to have private access to the embedded information [8]. The embedding process starts with deriving a seed for a PRNG from the user password and generating a random walk through the cover image that makes the steganalysis hard. Another recent improvement based on random distribution of the message was introduced in [23]. In this method they utilize an encryption key to hide information about horizontal and vertical blocks where the secret message bits are randomly concealed.

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 the LSB1, LSB2 or even in more significant bits such as System of Steganography using (SSB-4) by using the fourth bit of the pixel image [19]. SSB-4 steganography approach introduced by Rodrigues, Rios and Puech in 2003 [19]. This approach is based on the observation that a small variation in the channel color value of a colored image (e.g., RGB-24 image) is imperceptible to human eye [1].

SSB-4 steganography technique talks about changing the 4th bit of a pixel in the original image according to the bit message. Then modify the other bits (1^{st}, 2^{nd}, 3^{rd} and/or 5^{th}) to minimize the difference between the new/changed pixel value and the original one. Note that the 4th digit is a significant bit and if the image is compressed the embedded information will not be destroyed [7]. The authors in [19] argued that the difference must be equal or less than four (i.e., ±4). The 4^{th} bit was chosen because it satisfies that changing of ±4 units in the channel color value is imperceptible to human eyes, and it is the most significant bit which provides the minimum change in the pixel values [6][19]. Since changing the values smaller than 4 or greater than 251 can be perceptible to human eye, they usually are not employed to embed information. Moreover, the authors in [19] claimed that the worst case variation, when using SSB-4 steganography technique, is ± 4, which is equivalent to the changing of the 3^{rd} bit (i.e., LSB-3), to hide data.

2.2 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 [6]. 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 [13]. However, as noted in [21], ‘‘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 compression can be thought of as an application of such transform coding. The most common frequency domain methods usually used in image processing are the 2D DCT and Wavelet [11][13][14]. 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* x *N* grids (e.g., 8 x 8 grid containing 64 pixels per grid) [20]. 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 [9][12][18]:

$$S(u,v)=\frac{2}{N}C(u)C(v)\sum_{z=0}^{N-1}\sum_{y=0}^{N-1}S(x,y)\cos\left(\frac{\pi u(2x+1)}{2N}\right)\cos\left(\frac{\pi v(2y+1)}{2N}\right) \qquad (1)$$

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 from 0 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 [11].

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 use the inverse of the DCT formula given in Equation 2:

$$S(x,y)=\frac{2}{N}C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}S(u,v)\cos\left(\frac{\pi u(2x+1)}{2N}\right)\cos\left(\frac{\pi v(2y+1)}{2N}\right) \quad (2)$$

Based on both approaches (spatial and transform domain) the German researchers Pfitzman and Westfeld in 2001 introduced their **F5** algorithm [22] and illustrated in [8][11]. The goal of their research was to develop concepts and a practical embedding method for JPEG images that would provide high steganographic capacity while retaining security. Instead of replacing the LSBs of quantized DCT coefficients with the message bits, the absolute value of the randomly selected coefficient is decreased by one. The authors claimed that this type of embedding cannot be detected using their χ^2 statistical attack.

2.2 Stego-image quality measure

Detecting an embedded message defeats the primary goal of steganography, that of concealing the existence of a hidden message. As the Steganography is based on obscurity, the most important tests are related to the human perception. These types of tests evaluate the invisibility or transparency. The most used tests are the Subjective, the different image (between the original and the modified one) and the Peak-Signal-to-Noise-Ratio PSNR in dB (decibel).

The subjective tests are carried out by people who look for visual differences between the images (original and stego image) trying to find which one of them is the original. If the percentage of success goes 50%, it can be concluded that the message is invisible. The subjective test’s rules and recommendations are defined by the International Telecommunication Union [9][19].

Unlike the subjective approach which is vulnerable to human vision, Peak Signal to Noise Ratio (PSNR) is a technical approach usually used to evaluate the real quality of stego image [4][17]. This technique is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The PSNR is most commonly used to measure the quality of reconstruction in an image; by comparing the stego image with the original image. *PSNR* can be calculated using the mathematical models/formulas in Equation 3, and 4 below. First we calculate MSE using Equation 3:

$$MSE=\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left\|I(i,j)-K(i,j)\right\|^2 \quad (3)$$

Where *MSE* is the Mean Squared Error of *m* x *n* monochrome images *I* and *K*, where one of the images is considered a noisy approximation of the other, where lower is better. Thereafter, we can calculate *PSNR* using Equation 4:

$$PSNR=10.\log_{10}\left(\frac{MAX_i^2}{MSE}\right)=20.\log_{10}\left(\frac{MAX_i}{\sqrt{MSE}}\right) \quad (4)$$

Where, MAX_i is the maximum pixel value of the image. In other words $MAX_i = 2^b$-1, where is the bit depth of the original image (e.g., MAX_i =255 in the case of 8 bits depth grayscale images). Typical values for the PSNR in image and video compression are between 30 and 50 dB, where higher is better.

3. The proposed method

The challenge in this work was to camouflage a secret message bits in a way that prevents the detection of the hidden message without perceptible degrading the image quality. At the same time we aim to distribute the embedded information and its effect in unpredictable and unsystematic way in the stego image. Therefore, we utilize the DCT coefficients as a study case of the transform/frequency domain to hide a secret message. Thereafter, the stego image can be produced using the inverse of the DCT. Thus, the message bits will be distributed in unpredictable manner through the stego image. Furthermore, to avoid possible damage in the secrete information in case of lossy compression of the frequency domain of the stego image we use the idea of SSB-4 technique to embed message bits in the bit-4 of DCT coefficients.

The process of the proposed method can be depicted in Figure 1. The figure shows the image partitioning into 8 x 8 blocks. On each block we apply DCT. Thereafter, the message bits can be embedded in the 4th bit of the successive nonzero DCT coefficients. Finally we apply the IDCT on each block to producing the stego image which can be transferred to the intended recipient. The proposed method modifies the 4th bit of the coefficients of the DCT while retaining the minimum difference between the original value and the modified one. In order to minimize this variation we can modify the 1st, 2nd, 3rd and/or 5th bits. The effect of this variation is distributed across the image by using the inverse of the digital cosine transform (IDCT). This approach is illustrated in details in the following four steps (algorithm):

Step 1: Applying 2D DCT on image pixels
Herein the image is partitioned into 2D 8 x 8 grids/blocks. Thus each grid *F(x,y)* consists of 64 values. If the image is 8 bits depth monochromic, *F(x,y)* consists of the whole pixels' values. In the case of RGB 24-bits depth colored image, each grid *F(x,y)* is constructed from only the least significant bytes (i.e., Blue color channel/contribution) of the successive pixels. This is because the Blue channel is the most imperceptible to human eye. Then we calculate the 64 DCT coefficients *S(u,v)* of each grid *F(x,y)* of the image using Equation 1 (in Sec. 2.2).

Step 2: Embedding message bits In this step, message bits are embedded one by one in the successive nonzero DCT coefficients of the low frequency region of the S(u,v) (i.e., the upper left corner of the grid of the DCT coefficients): if the value of the 4th bit and the message bit are equal, nothing should be made. Otherwise, the 4th bit should be replaced by a message bit and modify the 1st, 2nd, 3rd and/or 5th bits to minimize the difference. This process is illustrated in the following piece of code:

```
for (u=0; u<= 7; u++){
 for ( v=0; v<= 7-u; v++){
          c= 4th bit of the S(u,v)
          if( S(u,v) ≠ 0 && c ≠ message bit) {
                     replace the 4th bit of s(u,v) by a message bit and
                     modify the 1st, 2nd, 3rd and 5th bits }
}
}
```

This step produces an intermediate stego grid *S'(u,v)* that conceals the message bits. It should be noted that using the low frequency region of the DCT coefficients to conceal message bits is to limit the degrading of the visual quality of the produced stego image after applying the next steps (i.e., Step 3 and 4).

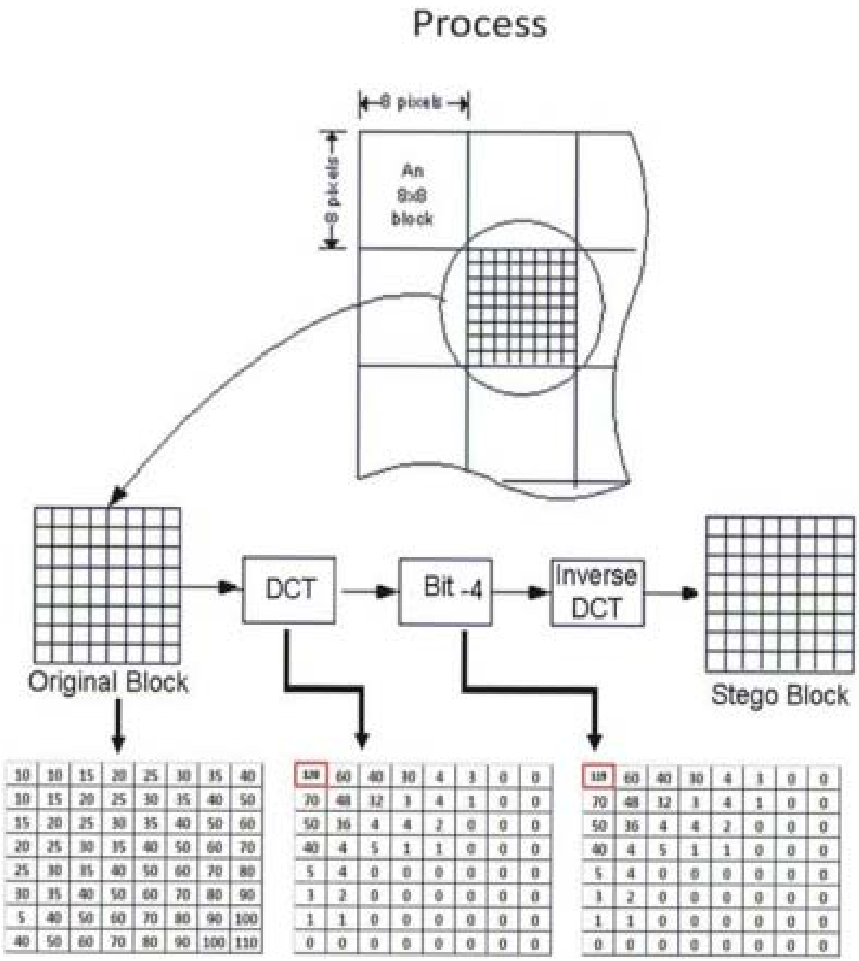
Step 3: Apply the inverse DCT
Now, we apply Equation 2 (i.e., the inverse of DCT or IDCT) on the stego grid *S'(u,v)* generated by Step 2. The result of this process will be the stego grid *F'(x,y)* of the stego image bytes.

Step 4: Construct the stego image
Finally, the stego image is constructed by replacing each original image grid *F(x,y)* by the proper stego grid *F'(x,y)*.

It should be noted that the produced variation/difference in a DCT coefficient by Step 2 can be from 0 to ±8 units. This is depending on the message bit, the replaced bit, and the values of the 5 lest significant bits of a coefficient. For example, consider a coefficient with least significant 5 bits equal to (…00001)2 and a bit message equals to "1". After applying Step 2, the value becomes (…01000)2. It is clear that the resulting difference/variation is 7 units.

Table 1 shows more examples showing the least significant bytes of the original binary values of DCT coefficients before the embedding process (pointed by the letter "A"), and the associated values after insertion a message bit and the best modification on the remainder bits to achieve minimum difference (pointed by the letter "B"). It is clear that the produced difference as a result of embedding message bits "0", "1", and "0" in the 4th bit of the coefficients 120, 83, and 224, respectively, are 1, 4, and 8 units.

Unlike the SSB-4 technique, this change will have no significant effect on the stego grid of the image bytes *F'(x,y)* generated in step 3, because it will be spread across the entire image grids. To reduce this change (e.g., to ±4), as an improvement to the proposed method, we have to avoid using the coefficients with low values to hide the message bits. Consequently, such improvement reduces the amount of information that can be embedded in an image. Furthermore, an extra process and memory should be utilized to keep track of the used coefficients.



10	10	15	20	25	30	35	40
10	15	20	25	30	35	40	50
15	20	25	30	35	40	50	60
20	25	30	35	40	50	60	70
25	30	35	40	50	60	70	80
30	35	40	50	60	70	80	90
5	40	50	60	70	80	90	100
40	50	60	70	80	90	100	110

120	60	40	30	4	3	0	0
70	48	32	3	4	1	0	0
50	36	4	4	2	0	0	0
40	4	5	1	1	0	0	0
5	4	0	0	0	0	0	0
3	2	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

119	60	40	30	4	1	0	0
70	48	32	3	4	1	0	0
50	36	4	4	2	0	0	0
40	4	5	1	1	0	0	0
5	4	0	0	0	0	0	0
3	2	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 1**.** The process of the proposed method

	Binary Value							
Decimal	8	7	6	5	4	3	2	1
A => 120	0	1	1	1	1	0	0	0
B => 119	0	1	1	**1**	***0***	***1***	***1***	***1***
A => 83	0	1	0	1	*0*	*0*	*1*	*1*
B => 79	0	1	0	**0**	***1***	***1***	***1***	***1***
A => 124	1	1	1	**0**	***0***	***0***	***0***	***0***
B => 132	1	1	1	**0**	***1***	***0***	***0***	***0***

Table 1. A=> Original Value, B=> the embedded message bit with best modification of the reminder bits

4. Experimental results

Since the visual detection of stego images is depending on the nature of the image [5], variety of image categories were utilized in the experiments. In order to have significant results in the assays, we have divided the images to 5 categories relevant to the human perception; trees, flowers, mountains, people, and buildings image category. In other words, the images were classified depending on the existence of areas with contiguous homogenous/same luminosity and grayscale level. For example, people and building images have such areas more than trees.

An appropriate collection of 75 BMP 24-bits grayscale images, 15 images for each category (downloaded from websites similar to webshots.com), were used in the experiments. Several randomly selected messages with different short lengths were utilized to be covert and retrieved. We focused on short messages because they are the most challenging to detect [5]. In addition to the proposed steganography technique, for comparison purposes, we utilized two other well-known techniques: the LSB and the LSB with DCT.

In order to evaluate the quality of the stego images generated by the compared techniques, and apart from the expensive and inaccurate subjective tests/judgments which are based on human perception (such as the Subjective test), instead we utilized - the commonly used metric - Peak Signal to Noise Ratio PSNR. Therefore, in this paper the stego image qualities are represented by PSNR (Equation 4, which is calculated from the root mean squared error MSE in Equation 3), introduces in Section 2.3.

The implementations of the compared techniques (LSB, LSB of the DCT coefficients, and the proposed one) and the PSNR tests were carried out using C# Programming language on a PC running on MS Windows XP.

4.1 Comparison with other methods

The obtained results of the experiments are summarized in the following tables and figures by means of the average PSNR in dB (decibel) values, of the 75 images of the 5 categories. Recall that the tested techniques are the proposed technique (Bit-4 of the DCT coefficients), LSB, and LSB of the DCT coefficients.

Table 2 shows some of the obtained results: the average PSNR of the different image categories (trees, flowers, mountains, people, and building) that conceal messages of 10, 20, and 40 bytes respectively. However, the tables show more precisely the decreasing of the PSNRs of stego images as the size of the embedded message increases. From the tables we can see that all of the tested techniques produce acceptable reconstruction of the covering image i.e., greater than 30 dB.

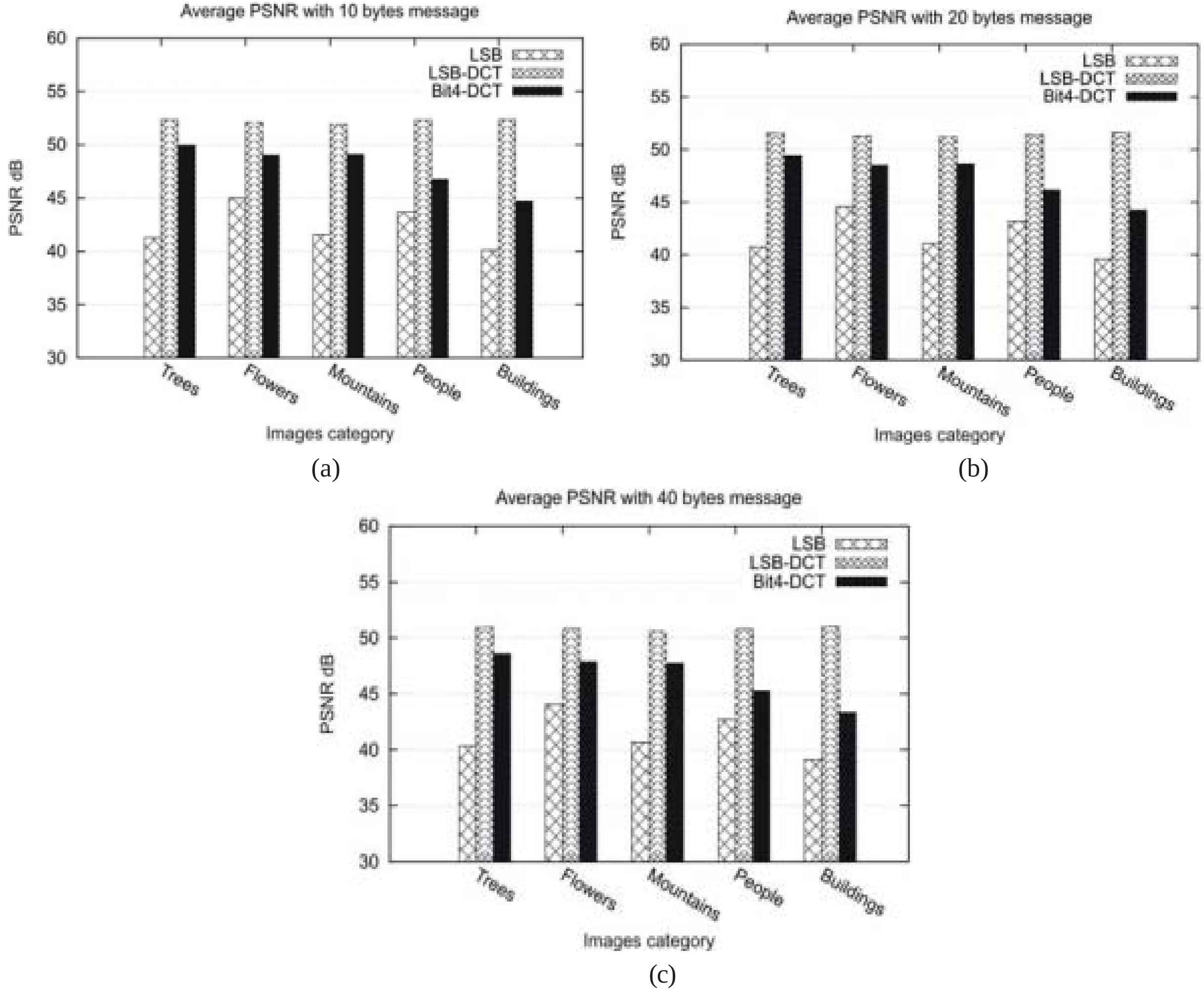
	Average PSNR by the Technique and by the Size of the Embedded Message								
	10 bytes message			20 bytes message			40 bytes message		
Image Type	LSB	LSB with DCT	Bit-4 with DCT	LSB	LSB with DCT	Bit-4 with DCT	LSB	LSB with DCT	Bit-4 with DCT
Trees	41.260	52.338	49.967	40.731	51.559	49.447	40.344	50.991	48.596
Flowers	45.003	52.092	49.044	44.550	51.278	48.501	44.071	50.863	47.878
Mountains	41.572	51.900	49.119	41.059	51.187	48.643	40.673	50.596	47.757
People	43.650	52.311	46.758	43.195	51.393	46.146	42.696	50.836	45.3110
Buildings	40.083	52.345	44.713	39.559	51.629	44.236	39.096	51.045	43.360

Table 2. Average PSNR produced by different techniques by image category and the embedded message size

The results are clearly depicted in Figures 2(a)-(c), where the x-axis represents the categories of the used images by the three tested techniques while the y-axis represents the obtained average PSNRs in dB. Since the higher the SPNR value the better the quality of the stego image, it is clear that the LSB with DCT is the best amongst the tested techniques by means of the PSNR (quality of stego image). However, it is more vulnerable to the loss of information in compression processes and easy to attack by steganalysis techniques. We can see that this technique is almost insensitive to the image category. In addition to its sensitivity to the compression processes by means of the loss of information and the easiest to attack by steganalysis, the LSB technique has the lowest PSNR values. Furthermore, we can see that the stego image produced by LSB technique is sensitive to the covert image category.

Results show that the proposed technique produces almost closed stego image quality to LSB with DCT technique when applied on images with higher frequency changing colors (trees, flowers, and buildings). In general our technique performs better than LSB and slightly lower than LSB with DCT. On the other hand, we can consider the proposed technique the best amongst the tested approaches by means of its robustness against steganalysis attacks and lossy compression techniques in which some data is deliberately discarded to achieve massive reductions in the size of the compressed file. This property is due to the utilization of a higher significant bit and the distribution of the effect of the embedded information in the frequency domain across the stego image grid by applying the inverse DCT process as stated in step 3 of the described algorithm.

As a visual example, Figure 3(a) shows the original well-known image (Lena) and its stego images produced using various tested techniques with its PSNR values (LSB, LSB with DCT, and the proposed technique Bit-4 of DCT coefficients) in Figures 3(b)-(c), respectively. It is clear that the stego images are imperceptible. Therefore, we generated the difference between the stego images and the original one as shown in Figure 4. We can see the proposed method produces stego image which differs more than the produced one using LSB with DCT coefficients. This is due to the high differences that may be produce bay embedding message bits in the 4th bit of the coefficients.



(a)

(b)

(c)

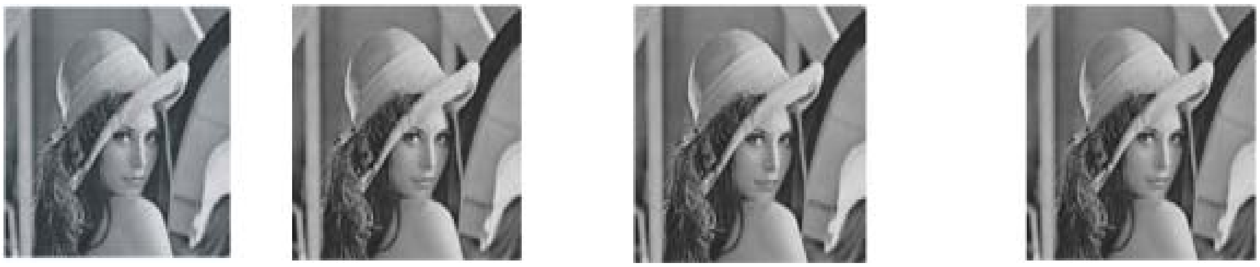
Figure 2. Average PSNR with (a) 10 bytes message (b) 20 bytes message (c) 40 bytes message

Results show that the proposed technique produces almost closed stego image quality to LSB with DCT technique when applied on images with higher frequency changing colors (trees, flowers, and buildings). In general our technique performs better than LSB and slightly lower than LSB with DCT. On the other hand, we can consider the proposed technique the best amongst the tested approaches by means of its robustness against steganalysis attacks and lossy compression techniques in

which some data is deliberately discarded to achieve massive reductions in the size of the compressed file. This property is due to the utilization of a higher significant bit and the distribution of the effect of the embedded information in the frequency domain across the stego image grid by applying the inverse DCT process as stated in step 3 of the described algorithm.

As a visual example, Figure 3(a) shows the original well-known image (Lena) and its stego images produced using various tested techniques with its PSNR values (LSB, LSB with DCT, and the proposed technique Bit-4 of DCT coefficients) in Figures 3(b)-(c),

respectively. It is clear that the stego images are imperceptible. Therefore, we generated the difference between the stego images and the original one as shown in Figure 4. We can see the proposed method produces stego image which differs more than the produced one using LSB with DCT coefficients. This is due to the high differences that may be produce bay embedding message bits in the 4th bit of the coefficients.



Original image

(a)

After insertion using LSB (PSNR: 41.526)

(b)

After insertion using LSB of DCT coefficients (PSNR:49.935)

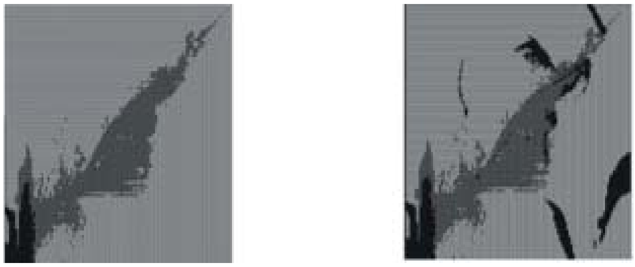
(c)

Using the proposed method Bit-4 of DCT coefficients (PSNR:45.312)

(d)

Figure 3. Original Lena image, Stego image using LSB, LSB with DCT, and Bit-4 of the DCT coefficients

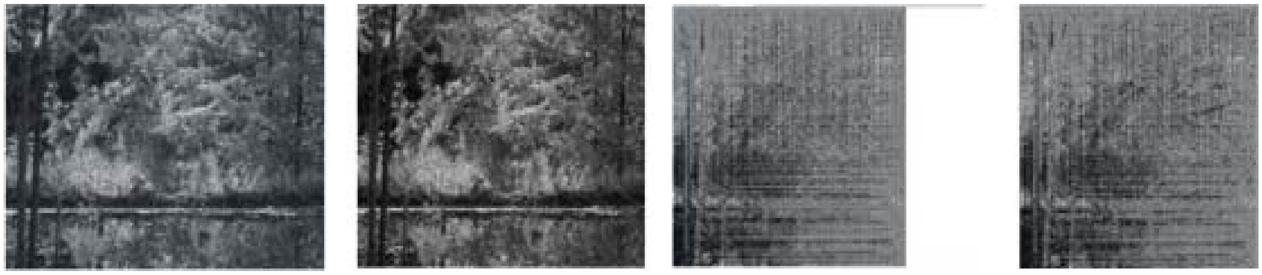
Moreover, Figure 5 (a) and (b) shows another sample example of the produced stego images of the tested picture of trees using the LSB with DCT and the proposed technique. Also, the differences between the stego images and the original image are in Figure 5 (c) while Figure 4 (d) shows the difference between the stego images.



Deference between original Lena and its stego image using LSB of DCT coefficients

Deference between original and Bit-4 of DCT coefficients Stego images (Lena)

Figure 4. The deference between the original Lena image and its produced stego images using LSB and Bit-4 of the DCT coefficients



Stego image of trees using LSB of DCT coefficients

(a)

Stego image of trees using Bit-4 of DCT coefficients

(b)

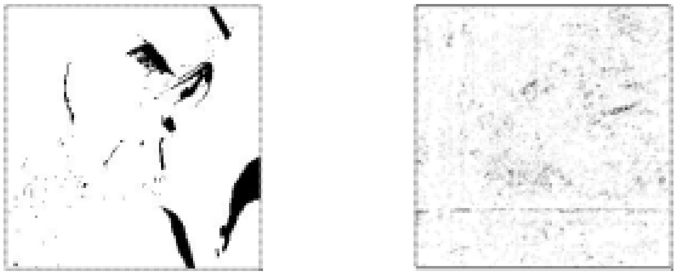
Deference between original and LSB of DCT coefficients Stego images (Trees)

(c)

Deference between original and Bit-4 of DCT coefficients Stego images (Trees)

(d)

Figure 5. Stego image of trees using (a) LSB (b) bit-4 of the DCT coefficients, and the deference between the original and the produced stego images using (c) LSB and (d) Bit-4 of the DCT coefficients of Trees



(a) Lena (b) Trees

Figure 6. The deference between the produced stego images using LSB and Bit-4 of the DCT coefficients

Figure 6 (a) and (b) show the differences between the produced stego images by the LSB of the DCT and Bit-4 of the DCT.

5. Conclusion

In this work we proposed a hybrid steganography technique that applies the frequency domain to hide information in the spatial domain of an image as an intermediate phase. The idea is to utilize a significant bit (4th bit) of the DCT coefficients of a cover image to hide message bits to achieve robustness against lossy compression techniques such as JPEG compression. Thereafter, the information and the variation of the coefficients, affected by the embedding process, are spread in unpredictable manner in the stego image by utilizing the inverse of the DCT process. The obtained experimental results indicate that, the proposed method will be a good and acceptable steganogaphy scheme. Furthermore, by imbedding information in the main significant bits of the DCT domain, the hidden message resides in more robust areas, spread across the entire stego image, and provides better resistance against compression and steganalysis process than other techniques. However, there are many works to do. Therefore, our future work will be focus on the improvement and further development of this technique utilizing different image formats and make more comparisons with other techniques.

References

[1] Ismail Avcibas N. M. and Sankur, B. (2003). Steganalysis using image quality metrics, *IEEE Transactions on Image Processing*, 12 (2).

[2] Bakshi, N. (2007). Steganography, Syracuse University. Available at: http://web.syr.edu/Ünbakshi/

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

[4] Chang, C. C., Chen, T. S., Chung, L. Z. (2002). A steganographic method based upon JPEG and quantization table modification, *Information Sciences*, 141 (1-2) 123-138.

[5] Davidson, L. and Goutam, P. (2004). Locating secret message in images, *In:* ACM SIGKDD International Conference on Knowledge discovery and Data mining, (Seattle, Washington, Aug. 22-25. ACM.

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

[7] Fridrich, J., Goljan, M. (2002). Practical steganalysis: stateofthe-art, *In:* Proceeding of SPIE Photonics West, Electronic Imaging 2002, 4675, p. 1-13.

[8] Fridrich, J., Goljan, M. (2003). Steganalysis of JPEG Images: Breaking the F5 Algorithm, Publisher: *Springer Berlin, Heidelberg,* Lecture Notes in Computer Science, V. 25-78. p. 310-323.

[9] International Telecommunication Union, Information Technology- Digital Compression and Coding of Continuous-Tone Still Images - Requirements and Specifications Recommendation T.81, *ITU* Sept., (1992).

[10] Johnson, N. F., Jajodia, S. (1998). Exploring Steganography: Seeing the Unseen, *IEEE Computer*, 31 (2) 26-34, available at: http://www.jjtc.com

[11] Kharrazi, M., Sencar, H., Memon, N. (2006). Performance study of common image steganography and steganalysis techniques, *Communications of the SPIE and IS&T*, 15 (4).

[12] Krenn, R. (2004). Steganography and Steganalysis, available at: http://www.krenn.nl/univ/cry/steg

[13] Lenti, J.(2002). Steganographic methods, periodic polytechnic *Ser.El.Eng* 44 (3-4) 249-258.

[14] Morkel, T., Eloff, J., Olivier, M. (2005). An overview of image steganography, *In:* Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005), (Sandton, South Africa, Jun/July).

[15] Nikolaidis, N., Pitas, I. (1998). Robust Image Watermarking in the Spatial Domain, *Signal Processing*, 66 (3) 385-403.

[16] Owens, M. (2002). A Discussion of Covert Channels and Steganography, SANS [Online] (March 19), Available at: http://www.sans.org/rr/whitepapers/covert/677.php

[17] Pavlidis, G., Tsompanopoulos, A., Papamarkos, N., Chamzas, C. (2003). JPEG2000 over noisy communication channels thorough evaluation and cost analysis, *Signal Processing: Image Communication*, 18 (6) 497-514.

[18] Provos, N., Honeyman, P. (2003). Hide and Seek: An introduction to Steganography, *Security & Privacy, IEEE*, 1 (3) 32-44.

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

[20] Wang, Y., Moulin, P. (2003). Steganalysis of block-DCT image steganography, *In:* IEEE Workshop on Statistical Signal Processing, p. 339- 342, Oct.

[21] Wang, H., Wang, S. (2004). Cyber warfare: Steganography vs. Steganalysis", *Communications of the ACM* . 47 (10) 76-82.

[22] Westfeld, A. (2001). High Capacity Despite Better Steganalysis (F5- A Steganographic Algorithm). *In: Moskowitz, I.S. (eds.)*: Information Hiding. 4th InternationalWorkshop. Lecture Notes in Computer *Science*, Vol.2137. *Springer-Verlag*, Berlin, Heidelberg, New York, 289-302

[23] Bani Younes, M. A., Jantan, A. (2008). A New Steganography Approach for Image Encryption Exchange by Using the Least Significant Bit Insertion, *IJCSNS, International Journal of Computer Science and Network Security*, 8 (6) June.