International Journal of Applied Science and Engineering
2006. 4, 3: 275-290

A DWT Based Approach for Image Steganography

Po-Yueh Chen* and Hung-Ju Lin

Department of Computer Science and Information Engineering,
National Changhua University of Education,
No. 2 Shi-Da Road, Changhua City 500, Taiwan, R.O.C.

Abstract: In this paper we propose a new steganography technique which embeds the secret messages in frequency domain. According to different users' demands on the embedding capacity and image quality, the proposed algorithm is divided into two modes and 5 cases. Unlike the space domain approaches, secret messages are embedded in the high frequency coefficients resulted from Discrete Wavelet Transform. Coefficients in the low frequency sub-band are preserved unaltered to improve the image quality. Some basic mathematical operations are performed on the secret messages before embedding. These operations and a well-designed mapping Table keep the messages away from stealing, destroying from unintended users on the internet and hence provide satisfactory security.

Keywords: Discrete Wavelet Transform; Security; Steganography.

Introduction

In a highly digitalized world we live today, computers help transforming analog data into digital forms before storing and/or processing. In the mean while, the internet develops very fast and hence becomes an important medium for digital data transmission. However, being a fully open medium, the internet brought us not only convenience but also some hazards and risks. If the data to be transmitted are confidential, it is convenient as well for some malicious users to illegally copy, destroy, or change them on the internet. As a result, information security becomes an essential issue [1][2]. Various schemes for data hiding are developed recently [3][4][5]. According to the purposes of data hiding, these schemes are classified into two categories: watermarking and steganography. Watermarking is a protecting technique which protects (claims) the author's property right for images by some hidden watermarks. On the other hand, steganography techniques apply some cover images to protect the confidential data from unintended internet users. According to the domain where watermarks or confidential data are embedded, both categories can be further classified as the time domain methods and the frequency domain methods [6].

Watermarking designs are usually consistent with the following features [7].(1) Imperceptibility: Human eyes cannot distinguish the difference between the watermarked image and the original version. In other words, the watermarked images still preserve high image quality. (2) Security: The watermarked image cannot be copied, modified, or deleted by any animus observer. (3) Robustness: The watermark still can be extracted out within certain acceptable quality even the image has endured some signal processing or noises before extraction. (4) Statistically undetectable: It is extremely hard (or impossible) to detect the

* Corresponding author; e-mail: pychen@cc.ncue.edu.tw
Accepted for Publication: December 10, 2006

watermark by statistical and/or mathematical analysis. (5) Blind detection: The extracting procedures have not to access the original image.

For spatial domain watermarking methods [8][9], the processing is applied on the image pixel values directly. In other words, the watermark is embedded in image by modifying the pixel values. The advantage of this type of watermarking is easy and computationally fast. The disadvantage is its low ability to bear some signal processing or noises. For frequency domain methods [10], the first step is to transform the image data into frequency domain coefficients by some mathematical tools (e.g. FFT, DCT, or DWT). Then, according to the different data characteristics generated by these transforms, embed the watermark into the coefficients in frequency domain. After the watermarked coefficients are transformed back to spatial domain, the entire embedding procedure is completed. The advantage of this type of watermarking is the high ability to face some signal processing or noises. However, methods of this type are computationally complex and hence slower.

The second category of data hiding is called Steganography. The methods are designed to embed some confidential data into some cover-media (such as texts, voices, images, and videos). After the confidential data are embedded, they are then transmitted together with the cover-media. The major objective is to prevent some unintended observer from stealing or destroying those confidential data. There are two things to be considered when designing a steganography system: (1) Invisibility: Human eyes cannot distinguish the difference between the original image and the stego-image (the image with confidential data embedded in). (2) Capacity: The more data an image can carry the better it is. However, large embedded data usually degrade the image quality significantly. How one can increase the capacity without ruining the invisibility is the key problem. The design of a steganography system also can be categorized into spatial domain methods [11][12] and frequency domain ones [13][14][15]. The advantages and disadvantages are the same as those we mentioned about watermarking methods earlier.

The rest of this paper is organized as follows. Section 2 reviews the related theoretical knowledge. In section 3, the proposed steganography method is described in details step by step. Some numerical examples are illustrated as well. Experimental results and analysis are demonstrated in section 4. Finally, some concluding remarks are provided in section 5.

2. Related knowledge

In this section, some preliminary techniques are reviewed including the LSB (Least Significant Bits) substitution method and the Haar-DWT. The quality of a stego-image is defined as well to perform the experiment analysis.

2.1 LSB substitution

The most frequently used steganography method is the technique of LSB substitution [16]. In a gray-level image, every pixel consists of 8 bits. One pixel can hence display 2^8 =256 variations. The weighting configuration of an 8-bit number is illustrated in Figure 1.

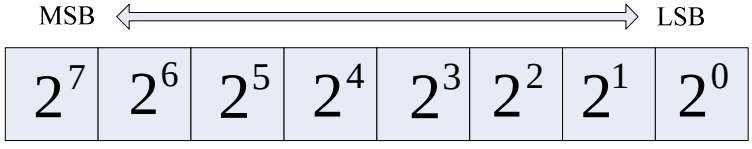


Figure 1. Weighting of an 8-bit pixel

The basic concept of LSB substitution is to embed the confidential data at the rightmost bits (bits with the smallest weighting) so that the embedding procedure does not affect the original pixel value greatly. The mathematical representation for LSB method is:

$$x_i^{'} = x_i - x_i \bmod 2^k + m_i \quad (1)$$

In Equation (1), $x_i^{'}$ represents the i th pixel value of the stego-image, x_i represents that of the original cover-image, and m_i represents the decimal value of the i th block in confidential data. The number of LSBs to be substituted is denoted as k. The extraction process is to copy the k-rightmost bits directly. Mathematically the extracted message is represented as:

$$m_i = x_i^{'} \bmod 2^k \quad (2)$$

Hence, a simple permutation of the extracted m_i gives us the original confidential data. This method is easy and straightforward. However, when the capacity is greatly increased, the image quality decreases a lot and hence a suspected stego-image results. Furthermore, the confidential data might be easily stolen by simply extracting the k-rightmost bits directly.

2.2 PSNR (Peak Signal to Noise Ratio)

How do we determine the quality of a digital image? Human eyes perception is the fastest approach. However, although this criterion is effective in general, the results may differ from person to person. To establish an objective criterion for digital image quality, a parameter named PSNR (Peak Signal to Noise Ratio) is defined as follows:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (3)$$

where MSE (Mean Square Error) stands for the mean-squared difference between the cover-image and the stego-image. The mathematical definition for MSE is:

$$MSE = \left(\frac{1}{M \times N}\right) \sum_{i=1}^{M} \sum_{j=1}^{N} (a_{ij} - b_{ij})^2 \quad (4)$$

In Equation (4), a_{ij} means the pixel value at position (i, j) in the cover-image and b_{ij} means the pixel value at the same position in the corresponding stego-image. The calculated PSNR usually adopts dB value for quality judgment. The larger PSNR is, the higher the image quality is (which means there is only little difference between the cover-image and the stego-image). On the contrary, a small dB value of PSNR means there is great distortion between the cover-image and the stego-image.

2.3 Haar-DWT

The frequency domain transform we applied in this research is Haar-DWT, the simplest DWT [17]. A 2-dimensional Haar-DWT consists of two operations: One is the horizontal operation and the other is the vertical one. Detailed procedures of a 2-D Haar-DWT are described as follows:

Step 1: At first, scan the pixels from left to right in horizontal direction. Then, perform the addition and subtraction operations on neighboring pixels. Store the sum on the left and the difference on the right as illustrated in Figure 2. Repeat this operation until all the rows are processed. The pixel sums represent the low frequency part (denoted as symbol L) while the pixel differences represent the high frequency part of the original image (denoted as symbol H).

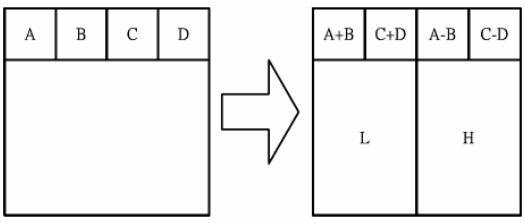


Figure 2. The horizontal operation on the first row

Step 2: Secondly, scan the pixels from top to bottom in vertical direction. Perform the addition and subtraction operations on neighboring pixels and then store the sum on the top and the difference on the bottom as illustrated in Figure 3. Repeat this operation until all the columns are processed. Finally we will obtain 4 sub-bands denoted as LL, HL, LH, and HH respectively. The LL sub-band is the low frequency portion and hence looks very similar to the original image.

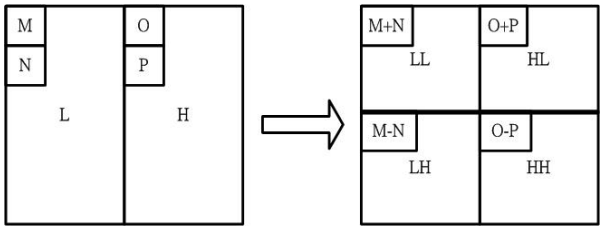


Figure 3. The vertical operation

The whole procedure described above is called the first-order 2-D Haar-DWT. The first-order 2-D Haar-DWT applied on the image " Lena" is illustrated in Figure 4.

(a) (b)

Figure 4. (a) Original image-Lena, (b) Result after the first-order 2-D Haar-DWT

3. Proposed method

3.1. Embedding procedure

According to different application requirements, the proposed scheme is classified as varying mode and fix mode. In varying mode, there is not a specific range for the capacity and we discuss in three cases based on different needs of capacity (different image sizes result in different capacity as well). In fix mode, there is a specific range for required capacity and we discuss in two cases based on different image quality requirements. Let us explain the working principle of these two embedding modes in the next 2 sub-sections respectively.

3.1.1 Varying mode

Case 1： Low embedding capacity requirement

Suppose C is the original 8-bit gray-level cover-image of size $M_C \times N_C$. It is denoted as

$$C = \{x_{ij} \mid 1 \le i \le M_C, 1 \le j \le N_C, \; x_{ij} \in \{0,1,...,255\}\} \quad (5)$$

S is the n-bit secret message represented as

$$S = \{s_i \mid 1 \le i \le n, s_i \in \{0,1\}\} \quad (6)$$

In case 1, we assume $n \le M_c \times N_c \times 8 \times (3/4) \times (2/8)$. (i.e. we exploit at most 2 bits per pixel in the 3 high frequency sub-bands HH, HL, and LH). The upper bound is obtained by multiplying the total number of pixels by 8 (assume 8 bits per pixel) and 3/4 (LL sub-band is kept unaltered) and 2/8 (at most 2 LSBs are applied).

Step 1: Apply DWT on C to obtain the frequency-domain matrix H. The 4 sub-bands obtained are denoted as H_{LL}, H_{HL}, H_{LH}, and H_{HH} (All 4 sub-bands have the same size of $M_C \times N_C / 4$).

Step 2: For the first $1 \le i \le M_C \times N_C$ bits of S, every 2 consecutive bits are combined to form a decimal value ranging from 0 to 3. For example, sequence 010001101111 will be transformed to sequence 101233. Every 2 consecutive values in the resulted decimal sequence are further combined to perform subtraction operation and form a differential sequence ranging from -3 to +3. For example, sequence 101233 results in a differential se-

quence of 1(1-0), -1(1-2), 0(3-3). As shown in Table 1, there are only 4 possible absolute values (0, 1, 2, and 3) for the elements in this differential sequence. Record these absolute values in H_{HH} by substituting the 2 LSBs of coefficients in H_{HH} with 00, 01, 10, and 11 respectively. However, same absolute value might be consequence of different subtraction pairs (For example, 1 could be |3-2|, |2-1|, |1-0|, |2-3|, |1-2|, or |0-1|). Hence, we need more bits to distinguish the subtraction status. In Table 1, the right Table coding is designed to record the possible subtraction pairs. The codes underlined are embedded in H_{LH} and the others are embedded in H_{HL}. Embedding positions in H_{LH} and H_{HL} are just the corresponding positions in H_{HH}. The raster-scan order illustrated in Figure 5 is employed for embedding.

Table 1. Sequence mapping table

Left table: 4 possible absolute values

former \ latter	0	1	2	3
0	|0|	|-1	|-2|	-3|
1	|1|	|0|	|-1	-2|
2	|2|	|1|	|0|	-1|
3	|3|	|2|	|1|	0|

Right table: Status of subtraction pairs

former \ latter	0	1	2	3
0	00	100	10	1
1	000	01	101	11
2	00	001	10	110
3	0	01	010	11

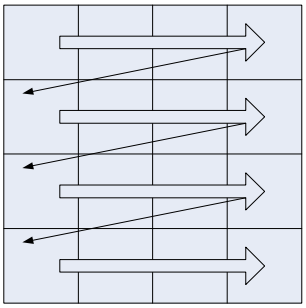


Figure 5. Embed with raster-scan order

Figure 6 demonstrates the embedding rules with some examples. When the value embedded in H_{HH} is 1 (there are 6 possible subtraction pairs), we have to embed 2 more bits at the corresponding position in H_{LH} and 1 more bit at the corresponding position in H_{HL}. If the value embedded in H_{HH} is 3 (only 2 possible subtraction pairs), recording one more bit in H_{LH} is required but no bit is embedded in H_{HL}.

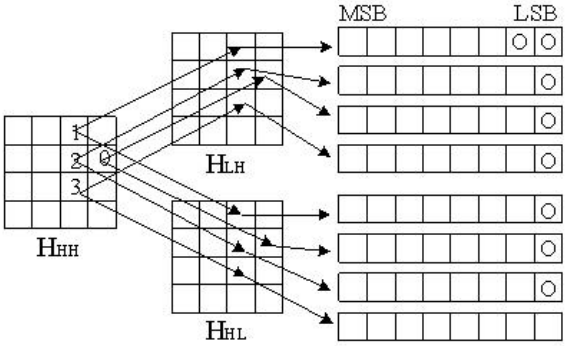


Figure 6. Values embedded in H_{HH} determine whether embed in H_{LH} and H_{HL} or not (O denotes embedding)

Step 3: The remaining bits of S: $\{M_C \times N_C + 1 \le i \le n\}$ are embedded at those unused LSBs (denoted as O in Figure 7) in H_{LH} and then H_{HL} bit by bit. For example, if the value embedded in H_{HH} is 1, we cannot embed any more message bit at the corresponding position of H_{LH} but 1 more bit at the corresponding position in H_{LH}. The embedding positions are illustrated in Figure 7.

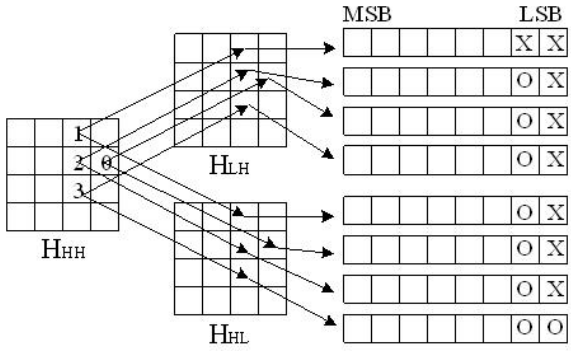


Figure 7. Remaining bits are embedded in H_{LH} and H_{HL}. (O denotes available positions for embedding and X denotes positions occupied already)

Step 4: After embedding all message bits, we obtain the slightly modified coefficients matrix H'. By performing the inverse DWT (IDWT) on H', the stego-image E is obtained. However, due to the LSB substitutions, some pixels in E are not integers ranging from 0 to 255. As shown in Figure 8, we employ a so-called "Key matrix"- K to record the 4 possible non-integer situations (0.0, 0.25, 0.5 and 0.75). The rounded pixel values of E are used to show the stego-image. In order to perfectly reconstruct the secret message bits, K is necessary in the extracting procedure.

$$E = \begin{bmatrix} 173\ .75 \\ 174\ .0 \\ 174\ .25 \\ 174\ .5 \\ 174\ .75 \\ 175\ .0 \end{bmatrix} \Rightarrow K = \begin{bmatrix} 11 \\ 00 \\ 01 \\ 10 \\ 11 \\ 00 \end{bmatrix}$$

Figure 8. Generation of the Key matrix

Step 5: The rounded version of E, denoted as F, is then stored in a specific image file format while K is filled in the unused tags (for example, the "Description" tag in TIFF format or the "Comment" tag in JPG format). The stego-image with secret message embedded is then ready for transmission.

Case 2: Median embedding capacity requirement

$$\begin{pmatrix} M_c \times N_c \times 8 \times (3/4) \times (2/8) \le n \\ \le M_c \times N_c \times 8 \times (3/4) \times (2/8) \\ + M_c \times N_c \times 8 \times (2/4) \times (1/8) \end{pmatrix}$$

Step 1~ Step 3: Identical to case 1.
Step4: Embed the remaining bits of S at the third LSB in H_{LH}, and H_{HL} sub-bands. The embedding positions are illustrated in Figure 9.

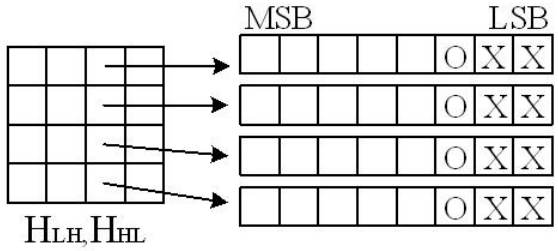


Figure 9. Remaining bits are embedded in H_{LH} and H_{HL}. (O denotes available positions for embedding and X denotes positions occupied already)

Step 5: Identical to step 4 of case 1.
Step 6: Identical to step 5 of case 1.

Case 3: High embedding capacity requirement

$$\begin{pmatrix} M_c \times N_c \times 8 \times [(3/4) \times (2/8) + (2/4) \times (1/8)] \le n \\ \le M_c \times N_c \times 8 \times [(3/4) \times (2/8) + (3/4) \times (1/8)] \end{pmatrix}$$

Step 1~Step 4: Identical to case 2.
Step 5: Embed the remaining bits of S at the third LSB in H_{HH} sub-band. The embedding positions are illustrated in Figure 10.

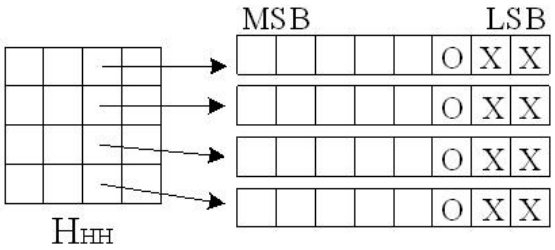


Figure 10. Remaining bits are embedded in H_{HH} (O denotes available positions for embedding and X denotes positions occupied already)

Step 6: Identical to step 5 of case 2.
Step 7: Identical to step 6 of case 2.

3.1.2 Fix mode

For fix embedding capacity requirement ($n = M_c \times N_c \times (2/4) \times 4$), the following two cases are analyzed.

Case 1：

Step 1: Identical to step 1 of case 1 in varying mode.

Step 2: For the first $1 \le i \le M_C \times N_C$ bits of S, every 2 consecutive bits are combined to form a decimal value ranging from 0 to 3. For example, sequence 010001101111 will be transformed to sequence 101233. Every 2 consecutive values in the resulted decimal sequence are further combined to perform subtraction operation and form a differential sequence ranging from -3 to +3. For example, sequence 101233 results in a differential sequence of 1(1-0), -1(1-2), 0(3-3). As shown in Table 1, there are only 4 possible absolute values (0, 1, 2, and 3) for the elements in this differential sequence. Next, embed the absolute values and the status values in three sub-bands (LH, HL and HH) utilizing LSBs substitution method. The embedding order still uses the raster-scan order illustrated in Figure 5 as well.

In this case, we embed the absolute difference values at two rightmost LSBs in H_{LH} and H_{HL} (H_{LH} sub-band first, then H_{HL} sub-band) and the status values at those in H_{HH} and further more (the third one, and the fourth one if needed) LSBs in H_{LH} and H_{HL} sub-bands. In Table 1, the coding on the right Table is designed to record the possible subtraction pairs. The codes underlined are embedded in the third or fourth LSBs in H_{LH} and H_{HL} sub-bands while the others are embedded in H_{HH} (and the first LSB for H_{LH} and the second for H_{HL} if needed). The embedding positions in H_{HH} are just those corresponding positions in H_{LH} and H_{HL}. The embedding positions are illustrated in Figure 11. For example, if the absolute value embedded in H_{LH} is 3 (the first LSB of H_{HH} is available for embedding) and that embedded in H_{HL} is 1, then we can embed the H_{LH} status (only one bit needed) in the first LSB of H_{HH}. The pairing status of H_{HL} (three status bits needed, since the value is 1) is recorded at the second LSB in H_{HH} , the third, and the fourth LSBs in H_{HL} .

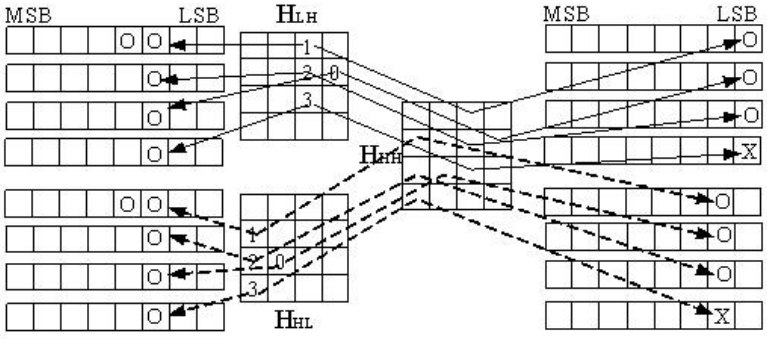


Figure 11. The values embedded in H_{LH} and H_{HL} determine whether embed in H_{HH} or not (O denotes embedding and X denotes no embedding)

Step 3: After embedding all message bits, we obtain the slightly modified coefficients matrix H'. By performing the inverse DWT (IDWT) on H', the stego-image E is obtained. Again, due to the LSB substitutions, some pixels in E are not integers ranging from 0 to 255. The "Key matrix"- K is used to record the 4 possible non-integer situations (0.0, 0.25, 0.5 and 0.75).
Step 4: The rounded version of E, denoted as F, is then stored in a specific image file format while K is filled in the unused tags (for example, the "Description" tag in TIFF format and the "Comment" tag in JPG format). The stego-image with secret message embedded is then ready for transmission.

Case 2：
Step 1: Identical to case 1.
Step 2: Embed the absolute values at LSBs in H_{LH} and H_{HL} (H_{LH} sub-band first, then H_{HL} sub-band) and the status values in H_{HH} and further more LSBs in H_{LH} and H_{HL} sub-bands. In Table 1, the coding on the right table is designed to record the possible subtraction pairs. In this case, the codes underlined are embedded in H_{HH} (the first and second LSBs are for H_{LH} status; the third and fourth ones are for H_{HL} status) the others (which are not underlined) are embedded in the third LSB in H_{LH} or H_{HL} sub-band. The embedding positions in H_{HH} are just the corresponding positions in H_{LH} and H_{HL}. The embedding positions are illustrated in Figure 12. For example, if the absolute value embedded in H_{LH} is 3 and that embedded in H_{HL} is 1, then we can embed the H_{LH} status (only one bit needed) in the first LSB of H_{HH}. The pairing status of H_{HL} (three status bits needed, since the value is 1) is recorded at the second LSB in H_{LH}, the second, and the third LSBs in H_{HH}.

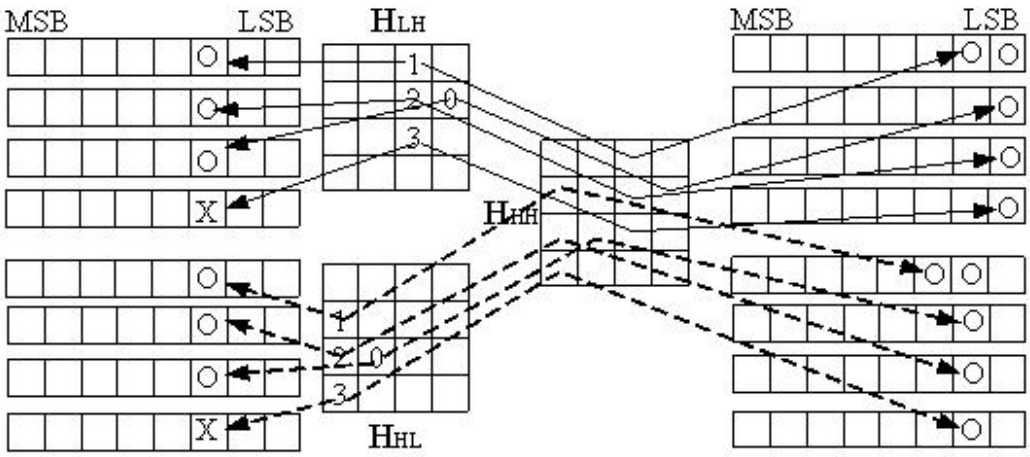


Figure 12. The values embedded in H_{LH} and H_{HL} determine whether embed in H_{HH} or not (O denotes embedding and X denotes no embedding)

Step 3: Identical to case 1.
Step 4: Identical to case 1.

3.2 Extracting procedure

The message extracting is explained as follows.

3.2.1 Varying mode

Case 1:
The 8-bit gray-level stego-image of $M_F \times N_F$ pixels is represented as

$$F = \{y_{ij} \mid 1 \le i \le M_F, \; 1 \le j \le N_F, y_{ij} \in \{0,1,...,255\}\} \quad (7)$$

Step 1: Extract the Key Matrix from file tag of F as

$$K = \{k_{ij} \mid 1 \le i \le M_F, \\ 1 \le j \le N_F, k_{ij} \in \{00,01,10,11\}\} \quad (8)$$

Transform all elements of K into 0, 0.25, -0.5 and -0.25 to form K', which is represented as

$$K' = \{k'_{ij} \mid 1 \le i \le M_F, \\ 1 \le j \le N_F, k'_{ij} \in \{0, 0.25, -0.5, -0.25\}\} \quad (9)$$

Step 2: Obtain matrix H' by performing DWT transform on E (which is calculated by F + K').

Step 3: Extract the absolute values (0, 1, 2 and 3) from the 2 rightmost LSBs in H'_{HH} sub-band. According to the value extracted, LSBs of corresponding positions in H'_{LH} and H'_{HL} are used to determine the subtraction pair. Figure 13 illustrated this rule. Base on the mapping rules defined in Table 1, we can reconstruct 2 values (former and latter) of the decimal sequence. Cue these decimal values in correct order and then expand them to a binary bit stream.

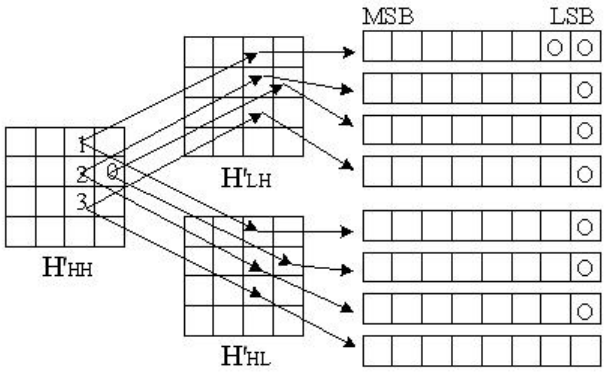


Figure 13. Extracting positions in H'_{LH} and H'_{HL} depend on the absolute difference value (O denotes extraction)

Step 4: By extracting some more second LSBs in H'_{LH} and H'_{HL} according to the rule illustrated in Figure 14, we obtain the remaining portion of S. Cascade it with the sequence obtained in step 3, the whole message bit stream S is completely extracted.

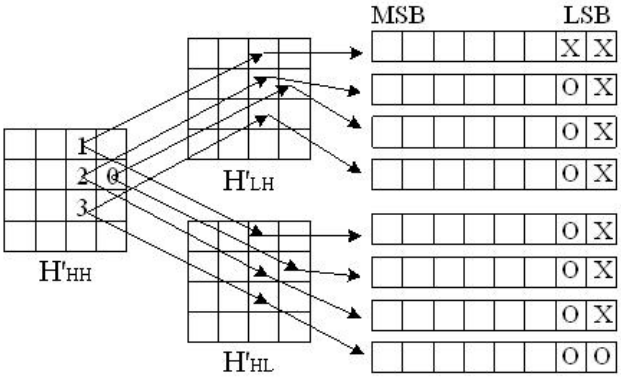


Figure 14. Extracting positions in H'_{LH} and H'_{HL} (O denotes extraction and X denotes no extraction)

Case 2:
Step 1~Step 4: Identical to case 1.
Step 5: Extracting the last portion of S from the third LSB in H'_{LH} and H'_{HL} and cascade it with the result obtained in step 4. Figure 15 shows the extracting positions in this step.

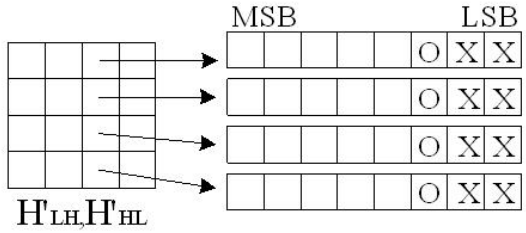


Figure 15. Extracting positions in H'_{LH} and H'_{HL} (O denotes extraction and X denotes no extraction)

Case 3:
Step 1~Step 5: Identical to case 2.
Step 6: Extracting the last portion of S from the third LSB in H'_{HH} and cascade it with the result obtained in step 5. Figure 16 shows the extracting positions in step 6.

3.2.2 Fix mode

Case 1:
Step 1: Identical to the step 1 of case 1 in varying mode.
Step 2: Identical to the step 2 of case 1 in varying mode.
Step 3: Extract the absolute values (0, 1, 2, or 3) from the 2 rightmost LSBs in H'_{LH} sub-band. According to the value extracted, the rightmost LSB of corresponding positions in H'_{HH} and the third (together with the fourth in some cases) LSB(s) of H'_{LH} are used to determine the subtraction pair. Figure 17 illustrates this rule. Base on the mapping rules defined in Table 1, we can reconstruct 2 values (former and latter) of the decimal sequence. Cue these decimal values in correct order and then expand them to a binary bit stream. While H'_{LH} sub-band is finished, keep extracting absolute values from H'_{HL} sub-band and perform the same decoding processes to obtain the remaining portion of S. Cascade it with the sequence obtained in H'_{LH} sub-band, the whole message bit stream S is completely extracted.

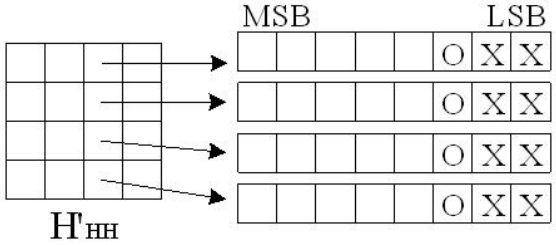


Figure 16. Extracting positions in H'_{HH} (O denotes extraction and X denotes no extraction)

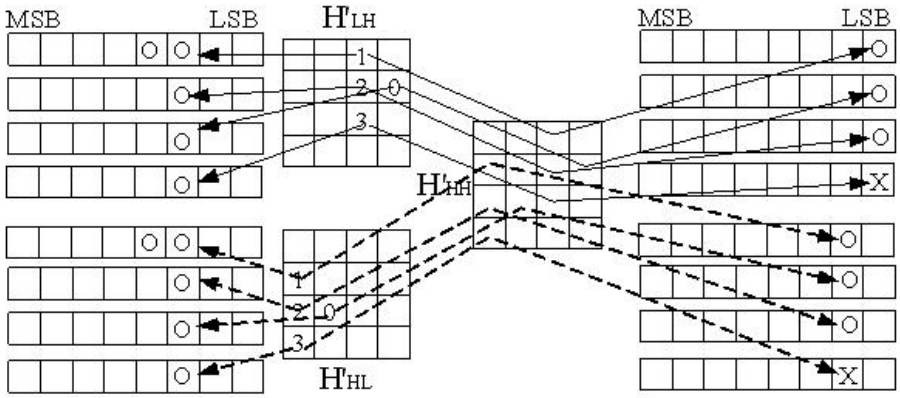


Figure 17. The extracting positions for H'_{LH}, H'_{HL} and H'_{HH} (O denotes extraction and X denotes no extraction)

Case 2:
Step 1: Identical to case 1.
Step 2: Identical to case 1.
Step 3: Extract the absolute values (0, 1, 2, or 3) from the 2 rightmost LSBs in H'_{LH} sub-band. According to the value extracted, the first (together with the second) LSB(s) of corresponding positions in H'_{HH} and the third LSB of H'_{LH} (if the absolute value is not 3) determine the original subtraction pair. Figure 18 illustrates this rule. Base on the mapping rules defined in Table 1, we can reconstruct 2 values (former and latter) of the decimal sequence. Cue these decimal values in correct order and then expand them to a binary bit stream. While H'_{LH} sub-band is finished, keep extracting absolute values from H'_{HL} sub-band and perform the same decoding processes to obtain the remaining portion of S. Cascade it with the sequence obtained in H'_{LH} sub-band, the whole message bit stream S is completely extracted.

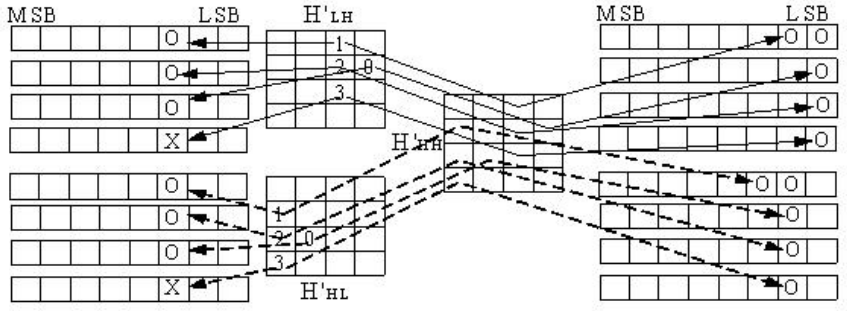


Figure 18. The extracting positions for H'_{LH}, H'_{HL} and H'_{HH}
(O denotes extraction and X denotes no extraction)

4. Simulation results and analysis

In this section, we demonstrate the simulation results and perform some data analysis. The proposed system is designed using Matlab 7.0 for programming. Six TIFF formatted images "Airplane"（254Kbyte）,"Baboon"（258Kbyte）,"Boat"（256Kbyte）,"Girl"（255Kbyte）,"Lena"（258Kbyte）, and"Pepper"(256Kbyte)are employed as the cover-images. All of them are of size 512×512, 8-bit gray-level images as shown in Figure 19. Haar-DWT [18] is applied for simplicity.

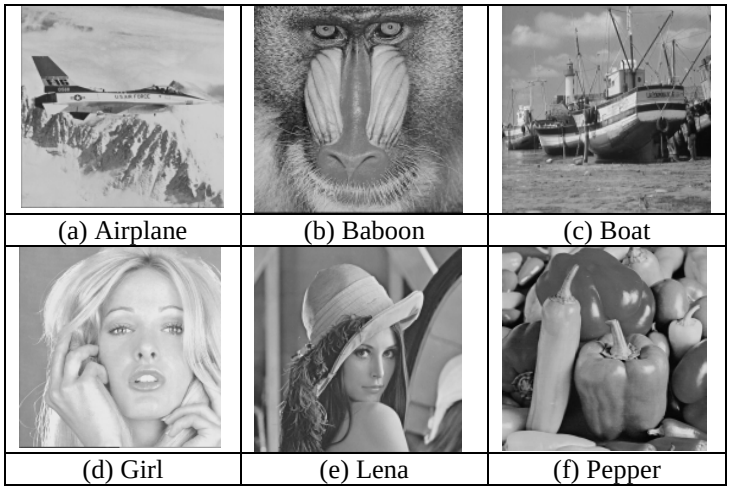


Figure 19. Six cover-images for simulations

4.1 Simulation results

The secret message S adopts pseudo-random numbers and is denoted as:

$$S = \{s_i \mid 1 \le i \le 900^2, s_i \in \{0,1\}\} \quad (10)$$

The performance in terms of capacity (in bit) and PSNR (in dB) are demonstrated for two operational modes in the following sub-sections respectively.

4.1.1 Varying mode

Six resulted stego-images are listed in appendix. It is difficult to tell the difference between the stego-image and the corresponding cover-image. Table 2 and 3 exhibit the capacity and PSNR respectively for these six images in three cases.

Table 2. Capacity of 6 images in 3 cases

Cases / Images	Case 1 Capacity	Case 2 Capacity	Case 3 Capacity
Airplane	376710	507856	573206
Baboon	376835	507670	573392
Boat	376598	507867	573318
Girl	377038	507940	573422
Lena	376942	507856	573550
Pepper	377125	507946	573184

Table 3. PSNR of 6 images in 3 cases

Cases / Images	Case 1 PSNR	Case 2 PSNR	Case 3 PSNR
Airplane	50.8554	45.9961	44.7683
Baboon	50.7647	46.1948	44.9664
Boat	50.7499	46.1385	44.9260
Girl	50.7746	46.0763	44.8842
Lena	50.8021	46.0882	44.9011
Pepper	50.7975	46.0793	44.8973

4.1.2 Fix mode

Since the capacity is fixed, only the PSNR results are demonstrated here. Six resulted stego-images are listed in appendices. For 512×512 images, The capacity is fixed at 256×256 (size of one sub-band) $\times 4$ (embedded bits per pixel) $\times 2$ (two sub-bands are used)=524288bits for both cases. Hence, PSNR (stego-image quality) is the only factor to be analyzed. Table 4 exhibits the PSNR of six images in two cases.

4.2 Performance analysis

As an example of analysis, in case 1 of varying mode, we assume all embedded coefficients in HH sub-band are value 1's which results in the worst embedding capacity. As a result, the LH sub-band cannot embed any more secret bits in the second LSB. However, the remaining secret bits can be embedded in the second LSB of the whole HL sub-band. Hence the first portion of embedding capacity is $256 \times 256 \times 4 = 262144$ bits and the total embedded capacity is $262144 + 65536 = 327680$ bits.

This is a satisfactory amount for capacity. Why the image quality becomes better while maintaining certain capacity value? The 4 sub-bands of DWT coefficients in frequency domain have different properties. The proposed method keeps the coefficients in LL sub-band (which is the most important part of an image) unaltered. Although we do modify the LH sub-band (horizontal edges), HL sub-band (vertical edges), and HH sub-band (diagonal edges), only the edge parts of an image are slightly changed. This is the main reason our approach outperforms the other ones. As another example, for image "Baboon" in case 3 of varying mode, the worst situation is LH, HL and HH sub-bands changes 3 bits after the embedding of secret messages. That means the difference between the original coefficients and the embedded ones is 7. In this worst case, the PSNR is 38.4694 dB. This is also a satisfactory amount for PSNR. To demonstrate the superiority of the proposed scheme, Table 5 compares its performance (in terms of PNSR with same capacity) with that of the side match method [12].

Table 4. PSNR of 6 images in 2 cases

Cases / Images	Case 1 PSNR	Case 2 PSNR
Airplane	43.2206	46.7523
Baboon	39.0033	46.5443
Boat	42.3820	46.7215
Girl	42.7842	46.8093
Lena	43.2741	46.8369
Pepper	43.4479	46.7818

Table 5. PSNR comparison with the side match method [12]

Images / Methods	Lena		Baboon	
	PSNR	Capacity	PSNR	Capacity
Three-sided	45.03dB	267242	34.93dB	483758
Proposed	52.78dB	267242	46.74dB	483758
Four-sided	48.18dB	164538	38.56dB	298413
Proposed	54.94dB	164538	52.03dB	298413

About the security issue, an untended observer cannot correctly solve the E matrix without being aware of the "Key matrix". Even one may accidentally find the "Key matrix" in some tags of the image format they are just a lot of meaningless disordered numbers. Besides, the original secret messages cannot be extracted from some rightmost LSBs of any sub-band. The untended observe needs to know the mapping rules before performing the decoding. In case 2 and case 3 of varying mode, embedding the third LSB in a random order can provide further security needs. Because most of coefficients in HH sub-band are either 0 or very small values in practical cases, some suspicion may arise from the modification of the third LSBs. However, the rightmost LSBs in HH are just storing the decimal values of some subtraction operations and hence the secret messages are still secure.

5. Conclusions and future works

5.1 Conclusions

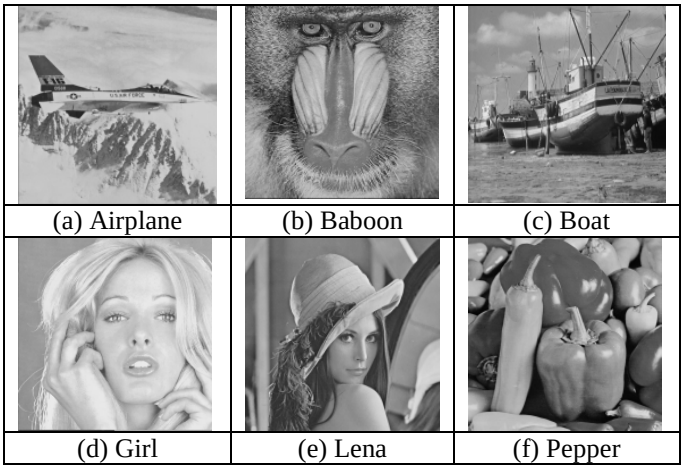
A new Image Steganography scheme is proposed in this paper. Based on different requirements of applications, two operation modes and 5 cases are provided for selection. According to the simulation results, the PSNR is still a satisfactory value even the highest capacity case is applied. This is due to the different characteristics of DWT coefficients in different sub-bands. Since the most essential portion (the low frequency part) is kept unchanged while the secret messages are embedded in the high frequency sub-bands (corresponding to the edges portion of the original image), better PSNR is not a surprising result. Furthermore, respectable security is maintained as well since no message can be extracted without the "Key matrix" and decoding rules.

5.2 Future works

To reduce the extra data in the stego-images, We have to compress the size of "Key matrix" as far as possible. Some novel coding schemes are available for this kind of problem. As a result, the file sizes of the original image and that of the corresponding stego-image will not differ too much. Another issue is to efficiently integrate the proposed scheme in the JPEG2000 flow which is based on DWT as well.

Appendix : the resulted stego-images

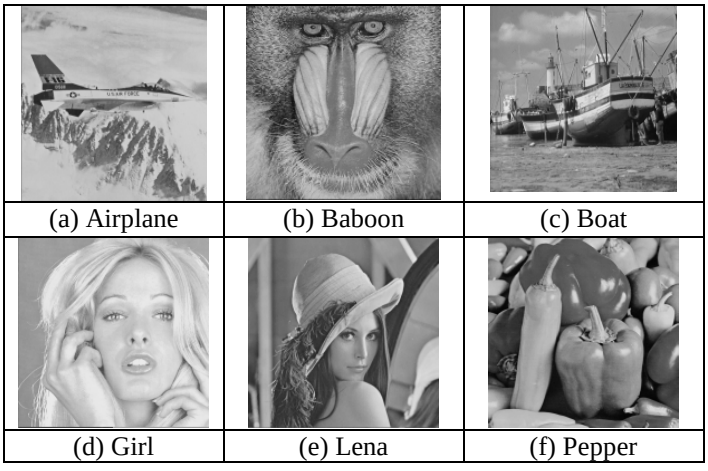
A.1. Six resulted stego-images for varying mode (case 1).



(a) Airplane (b) Baboon (c) Boat

(d) Girl (e) Lena (f) Pepper

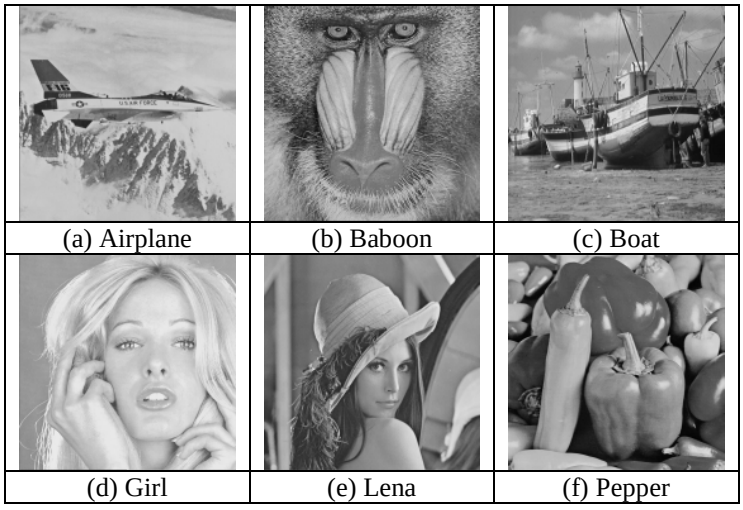
A.2. Six resulted stego-images for varying mode (case 2).



(a) Airplane (b) Baboon (c) Boat

(d) Girl (e) Lena (f) Pepper

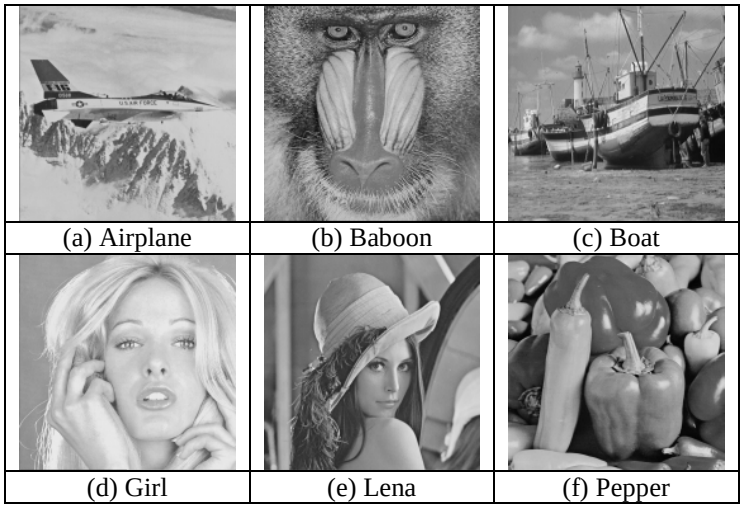
A.3. Six resulted stego-images for varying mode (case 3).



(a) Airplane (b) Baboon (c) Boat

(d) Girl (e) Lena (f) Pepper

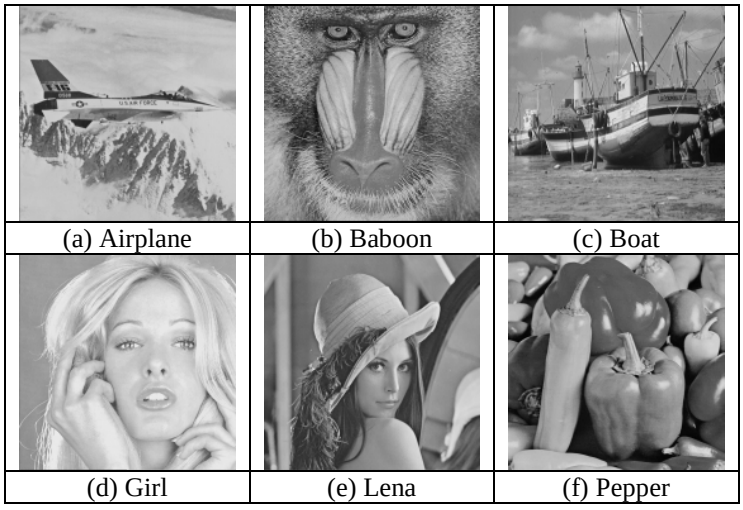
A.4. Six resulted stego-images for fix mode (case 1).



(a) Airplane (b) Baboon (c) Boat

(d) Girl (e) Lena (f) Pepper

A.5. Six resulted stego-images for fix mode (case 2).



(a) Airplane (b) Baboon (c) Boat

(d) Girl (e) Lena (f) Pepper

References

[1] Chang, C. C. and Chuang, L. Z. 2004. Image Steganography., *Communication of the Chinese Cryptology and Information Security Association (CCISA)*, 10, 1: 108-122.

[2] Chang, C. C. and Chuang, L. Z. 2004. Introduction to the Visual Cryptography, *Communication of the Chinese Cryptology and Information Security Association (CCISA)*,10, 2: 1-14.

[3] Bender, W., Gruhl, D., Morimoto, N., and Lu, A. 1996. Techniques for data hiding. *IBM Systems Journal*, 35, 3 and 4: 313-336.

[4] Johnson, N. F. and Jajodia, S. 1998. Steganography: Seeing the Unseen. *IEEE Computer*, February: 26-34.

[5] Lou, D. C. and Liu, J. L. 2002. Steganography Method for Secure Communications. *Elsevier Science on Computers & Security*, 21, 5: 449-460.

[6] Petitcolas, F. A. P. Anderson, R. J., and Kuhn, M. G. 1999. Information Hiding – A Survey, *Proceeding of IEEE*, 87, 7: 1062-1078.

[7] Chang, C. C., Chen, T. S., and Hwang, K. F. 2000. Electronic Image Techniques. Taipei: Unalis.

[8] Lu, Z. M., Wu, H. T., Xu, D. G., and Sun, S. H. 2003. A Multipurpose Image Watermarking Method for Copyright Notification and Protection. *IEICE Transfusion. Information & Syst*ems, E86-D, 9: September: 1931-1933.

[9] Shih F. Y. and Wu Y.T. 2003. Combinational image watermarking in the spatial and frequency domains. *Pattern Recognition*, 36: 969-975.

[10] Hsu, C. T. and Wu, J. L. 1999. Hidden digital watermarks in images. *IEEE Transactions on Images Processing*, 8: 58-68.

[11] Chan, C. K. and Cheng, L. M. 2003. Hiding data in image by simple LSB substitution. *Pattern Recognition*, 37: 469-474.

[12] Chang, C. C. and Tseng, H. W. 2004. A Steganographic method for digital images using side match. *Pattern Recognition Letters*, 25: 1431-1437.

[13] Chen, T. S., Chang, C. C., and Hwang, M. S. 1998. A virtual image cryptosystem based upon vector quantization. *IEEE Transactions on Image Processing*, 7, 10: 1485-1488.

[14] Chung, K. L., Shen, C. H. and Chang, L. C. 2001. A novel SVD- and VQ-based image hiding scheme. *Pattern Recognition Letters*, 22: 1051-1058.

[15] Iwata, M., Miyake, K., and Shiozaki, A. 2004. Digital Steganography Utilizing Features of JPEG Images, IEICE Transfusion Fundamentals, E87-A, 4:929-936.

[16] van Schyndel, R. G., Tirkel, A. Z., and Osborne, C. F. 1994. A digital watermark. *IEEE International. Conf. Image Processing*, 2: 86-90.

[17] Chen, T. S., Chang, C. C., and Hwang, K. F., 2002. *Digital Image Processing*, Taipei: Flag.

[18] Chen, P. Y. and Liao, E. C. 2002. A New Algorithm for Haar Wavelet Transform. *IEEE International Symposium on Intelligent Signal Processing and Communication System*: 453-457.