DCT-based Reversible Data Hiding Scheme

Chia-Chen Lin
Department of Computer Science and Information Management, Providence University
200 Chung-Chi Rd., Taichung, Taiwan
Email:mhlin3@pu.edu.tw

Pei-Feng Shiu
Department of Computer Science and Information Management, Providence University
200 Chung-Chi Rd., Taichung, Taiwan
Email:s9472012@pu.edu.tw

***Abstract*—To enhance the hiding capacity of Chang et al.'s reversible DCT-based data hiding scheme, in this paper we propose another method called a layer-1 data embedding strategy. Our proposed layer-1 strategy considers some areas not used by Chang et al.'s scheme, which we call layer-2 data embedding. To achieve our objective, we applied Tian's pixel expansion method to design our layer-1 data embedding strategy. Experimental results confirm that the hiding capacity provided by combining our strategy with Chang et al.'s is higher than that provided by the Chang et al. approach alone. Moreover, the image quality of stego-images with our proposed scheme remains above 30 dB for most test images, which is better than the best image quality offered by Chang et al.'s scheme. Finally, the security and reversibility of Chang et al.'s scheme is unaffected when their layer-2 scheme is combined with our proposed layer-1 scheme.**

Index Terms—Reversible data hiding, DCT, JPEG

I. INTRODUCTION

Protecting data transmitted over the Internet has become a critical issue driven by the progress in data digitalization and communications networking over the past decade. To ensure transmitted data are secure and cannot be tampered with or eavesdropped by malicious attackers, two approaches are proposed here. One is a traditional cryptography approach, in which secret messages are transformed into unrecognizable form by using secret information shared between senders and authorized receivers. In this approach, only authorized users can retransform a secret message back to its original form. Many well-known encryption schemes, such as RSA [15], DES [8] and the like are widely used in the commercial market. The second approach is steganography, which enables senders to transmit secret message via meaningful cover media in order to avoid attracting attackers' attention. Hiding the subjects with steganographic techniques involves the spatial [1, 4, 13, 14, 16, 18] and frequency [3, 11, 12] domains of these cover images.

In the spatial domain approach, the secret message is embedded directly into the pixels of the cover images. Least significant bit (LSB)-based hiding strategies are the most commonly used in this approach. For example, in Lee and Chen's scheme [13], the LSB of each pixel in a cover image is modified to embed the secret message. In Chang et al.'s scheme [4], a dynamic programming strategy is used to find the optimal LSB substitution to hide images. In addition to LSB-based hiding strategies, several schemes that use different strategies to hide secret messages in the spatial domain of cover images have been proposed in the past decade. For example, Chung et al. offered singular value decomposition (SVD)-based hiding scheme [6], and Tsai et al. used the bit plane of each block truncation coding (BTC) block to embed secret messages [17].

In the frequency domain [3, 11, 12], cover images must be transformed using a frequency-oriented mechanism such as discrete cosine transformation (DCT), discrete wavelet transformation (DWT) or similar mechanisms first, after which the secret can be combined with the coefficients in the frequency-form images to achieve embedding. For example, in Chang et al.'s scheme [3], the medium-frequency coefficients of DCT-transformed cover images are used to embed a secret message. The quantization table of the JPEG is also modified to further protect the embedded secret message. Similarly, Iwata et al. uses the boundaries between zero and non-zero DCT coefficients to hide secret data [11].

To extend the application of data hiding to some sensitive domains such as military, medical and fine arts, which require the embedded cover images to be properly covered, reversible data hiding has become another new branch of this field. In the spatial domain, Tian explored the difference expansion between two neighboring pixels to hide one secret bit [16], and Mehmet used the generalized LSB of a pixel in a cover image to design a lossless data embedding system [1]. In the compression domain, Chang et al. modified the codeword selection method of the side-mach quantization vector (SMVQ) and further proposed a reversible data hiding scheme [5]. In the frequency domain, Fridrich et al. [9] presented an invertible watermarking scheme for authenticating digital images in the JPEG domain. This scheme uses an order-2 function, which is an inverse function, to modify the quantization table to enable lossless embedding of one bit per chosen DCT coefficient. Later, Xuan et al. [19] proposed a high-capacity distortion-free data hiding technique based on the integer wavelet transform. In addition, histogram modification is used in their scheme to embed secret data into the middle frequency of the wavelet domain. Xuan et al.'s scheme can also be applied in JPEG2000-compressed images because

doi:10.4304/jsw.5.2.214-224

JPEG2000 is based on the wavelet transform domain. In 2007, Chang et al. extended Iwata et l.'s idea [11] and then presented a lossless steganography scheme for hiding secret data in each block of quantized discrete cosine transformation (DCT) coefficients in JPEG images [2]. In Chang et al.'s scheme, the two successive zero coefficients of the medium-frequency components in each block are used to hide the secret data. Furthermore, they modified the quantization table to maintain the quality of the stego-image while concealing higher payload compared with previous schemes. Thus, their scheme achieves reversibility and acceptable image quality of the stego-image simultaneously. However, their scheme can embed secret bits only into the zero coefficient which located in the successive zero coefficient in medium area; non-zero coefficients in the medium can not been used for data embedding.

Inspired by Chang et al.'s idea [2], we propose a DCT-based reversible data hiding scheme that improves on the hiding capacity of Chang et al.'s scheme while maintaining acceptable image quality of stego-images. To explore the relationship between hiding capacity and quantization table, the standard quantization table and a modified quantization table proposed by Chang et al.'s scheme are used in our experiments. Experimental results confirm that the proposed scheme successfully enhances the hiding capacity while the image quality of stego-image and reversibility are maintained.

The rest of this paper is organized as follows. In Section 2, we briefly review the DCT transform, Iwata et al.'s data hiding scheme [11] and Chang et al.'s data hiding scheme [2]. Our proposed reversible and high hiding capacity data embedding scheme is then illustrated in Section 3. Then, Section 4 presents our experimental results. Finally, concluding remarks appear in Section 5.

II. RELATED WORKS

In this section, we briefly review the DCT transform, Iwata et al.'s data hiding scheme [11] and introduce Chang et al.'s [2] reversible data hiding scheme.

A. Discrete Cosine Transform (DCT) and Quantization

DCT is a widely used mechanism for image transformation adopted to compress JPEG images. Figure 1 shows the JPEG compression process, which consists of five phases: transforming an RGB image to a YCbCr image, composition of minimum coding units, 2-dimensional DCT, quantization of DCT coefficients, runlength coding and Huffman coding.

In the 2-dimensional DCT phase, each 8×8 non-overlapping block is transformed into the DCT domain by the following Equation (2):

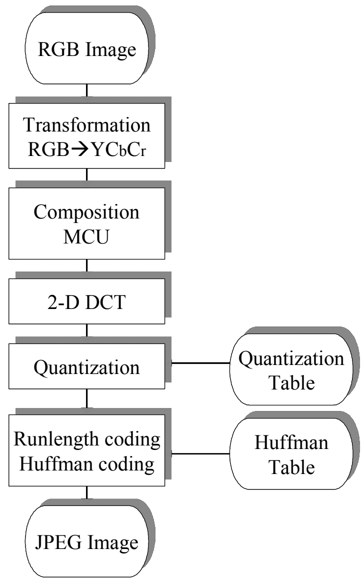


Figure 1. Flowchart of JPEG compression

$$F(u,v) = \frac{c(u)c(v)}{4}\sum_{i=0}^{7}\sum_{j=0}^{7}\cos\left(\frac{(2i+1)u\pi}{16}\right)\cos\left(\frac{(2j+1)v\pi}{16}\right)f(i,j), \quad (1)$$

$$\text{where } c(e) = \begin{cases} \frac{1}{\sqrt{2}}, & if\ e = 0 \\ 1, & if\ e \neq 0 \end{cases}.$$

Here, $F(u,v)$ and $f(i,j)$ present a DCT coefficient at the (u,v) coordinate and a pixel value at the (i,j) coordinate, respectively. $F(0,0)$ is the DC component, which corresponds to an average intensity value of each block in the spatial domain. $F(u,v)$ is the AC component, in which $u \neq 0$ and $v \neq 0$. For data reduction during the quantization phase, DCT coefficients are quantized by using the standard quantization table shown in Figure 2.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Figure 2. Standard quantization table

The human vision system is much more sensitive to the values in low-frequency components than those in the higher frequencies. Thus, distortion in high-frequency components is visually acceptable and perceptible. Therefore, the upper left values in the quantization table are small enough to avoid large alteration. In contrast, the lower right values in the table are large and can be altered.

B. Iwata et al.'s Data Embedding Scheme

In 2004, Iwata et al. discovered that the values of AC coefficients tend to be zero after the quantization phase of JPEG compression; therefore, they designed a data hiding strategy which embeds secret information into high-frequency components based on the length of zero sequences after quantization of the DCT coefficients [11]. By using Iwata et al.'s modification strategy, a JPEG coded stego-image is obtained after the runlength coding and Huffman coding. The modified DCT coefficients are reserved after the secret data is extracted from the JPEG-coded stego-image because both runlength coding and Huffman coding involve lossless compression. Iwata et al.'s data hiding scheme is broken down into two procedures: data embedding and data extracting procedures. In the following subsections, detailed descriptions of the two procedures are given.

B.1. Data Embedding Procedure

First of all, Iwata et al. defined a set for embedding one bit as shown in Figure 3, for data embedding.

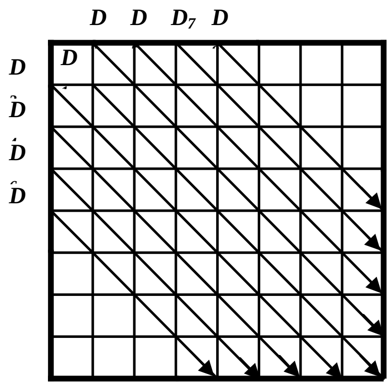


Figure 3. Sets for Iwata et al.'s data sets for data embedding

In Figure 3, set D_i ($1 \le i \le l_i$) contains quantized DCT coefficients on the line labeled "D_i". Le l_i be the length of a zero sequence of higher frequency components on the "D_i" line. The odd or even state of l_i indicates what secret data are embedded into D_i. Let $(d_{i,1},\ d_{i,2},\ d_{i,3},\ldots,\ d_{i,k_i})$ be the coefficient sequence in set D_i with k_i components from low frequency to high frequency, $d_{i,j}$ be the non-zero value of the highest frequency component of set D_i, where $1 \le j \le k_i$, and T be a predetermined threshold. Based on above sets of a block, Iwata et al. designed four modification strategies for four cases. Basically, their idea is to modify the length of a zero sequence of higher frequency components on line "D_i" to let l_i be even when a secret bit is 1, and let l_i be odd when a secret bit is odd. The detailed modification strategies for different scenarios are described as follows:

Scenario 1: When $\left|d_{i,j}\right| > T$ the coefficient of the location $d_{i,j+1}$ is replaced by 1 or -1, where -1 and 1 are randomly selected.

Scenario 2: When $\left|d_{i,j}\right| \le T$ and $d_{i,j-1}$ is equal to 0, $d_{i,j}$ and $d_{i,j-1}$ will be replaced with 0 and 1, respectively.

Scenario 3: When $\left|d_{i,j}\right| \le T$ but $d_{i,j-1}$ does not exist or is not equal to 0, $d_{i,j}$ will be set at 0.

Scenario 4: When $d_{i,j}$ does not exist, it means all components in set D_i are zero, a 1 or -1 is randomly assigned to the lowest coefficient in the set D_i.

B.2. Data Extracting Procedure

After the receiver receives a JPEG stego-image from a sender, s/he conducts the following steps to extract the hidden secret information from each 8×8 block in a JPEG stego-image:

Step 1. Obtain 8×8 non-overlapping blocks of the quantized DCT coefficients of the Y component from a JPEG stego-image after Huffman decoding and runlength decoding.

Step 2. Extract secret bits from an 8×8 non-overlapping block by using Equation (2).

$$w_i = \begin{cases} 0, when\ l_i\ \ is\ even. \\ 1, when\ l_i\ \ is\ odd. \end{cases} \tag{2}$$

Here, w_i ($1 \le i \le l_i$) is the hidden bit in the set D_i.

By using the proposed data hiding strategy, Iwata et al. successfully embedded a secret bit into a set in an 8×8 block of a JPEG image and caused only a slight difference in the histogram of quantized DCT coefficients. Later, Chang et al. [2] found the weakness of Iwata et al.'s scheme, that is Iwata et al.'s scheme lacks reversibility. To conquer above weakness, Chang et al. proposed a reversible data hiding for JPEG coded images in 2007. Detailed descriptions of Chang et al.'s proposed scheme is presented in subsection 2.3.

C. Chang et al.'s Reversible Data Embedding Scheme

As we mentioned in previous subsection, Chang et al.'s scheme successfully embedded secret bits into a DCT-based compressed image and restored the original DCT coefficients after the secret bits are extracted [2], which can not be achieved by using Iwata et al.'s scheme [11]. Their scheme can be broken down into three procedures: data embedding, data extracting and restoring procedures. Note that their scheme requires a preprocessing procedure because the secret data are hidden in the DCT coefficients of a cover image. The preprocessing procedure in the proposed scheme involves first partitioning a cover image into non-overlapping blocks of 8×8 pixels, then performing the 2-dimensional DCT to transform each block into an 8x8 block of DCT coefficients. Later, the quantized coefficients are obtained through the 8x8 quantization table shown in Figure 2. After the DCT coefficients are quantized, the proposed data embedding procedure can begin.

C.1. Data Embedding Procedure

Chang et al.'s scheme defines several sets R_i $(1 \le i \le 9)$ for embedding one bit, as shown in Figure 4. In each set, one secret bit is embedded in the successive zero sequence, which runs from the highest frequency component to the lower frequency components and ensures that there are at least two zeros in each set R_i ($1 \le i \le 9$). Let b_i ($1 \le i \le 9$) be the length of ceaseless zeros in order from the highest frequency component to the lower frequency components in set R_i. The value of b_i is key to deciding whether set R_i can hide a secret bit in Chang et al.'s scheme. The estimation rule is set as follows: if $b_i \geq 2$, set R_i can hide a secret bit; otherwise, set R_i cannot hide a secret bit. In the example given in Figure 5, four continuous zero sequences exist from the highest frequency component to the lower frequency components in set R_1, and b_1 equals 4 because the length of ceaseless zeros in order from the highest frequency component to the lower frequency components in set R_1.is 4. Similarly, we can obtain $b_2 = 2$ and $b_3 = 1$, respectively.

Following these rules, in set R_i, if $b_i \geq 2$, $z_{i,1}$ represents the zero value of the lowest frequency of set R_i, and $z_{i,2}$ represents the lower right component of $z_{i,1}$, respectively. Note that $z_{i,2}$ does not exist once b_i is less than 2 in set R_i (e.g., in the set R_3 shown in Figure 6). Let $(r_{i,1},\ r_{i,2},\ ..,\ r_{i,k_i})$ be the coefficient sequence in set R_i with k_i components from high frequency to low frequency, and s_i be the secret bit we want to embed in set R_i. Refer to set R_1 in Figure 5. In set R_1, the coefficient sequence is represented as $(r_{i,1},\ r_{i,2},\ r_{i,3},\ r_{i,4},\ r_{i,5},\ r_{i,6},\ r_{i,7})$, and the values of set R_1 are (0, 0, 0, 0, 2, 2, 3), individually. According to the definition just given, $z_{1,1}$ stands for $r_{1,4}$=0 and $z_{1,2}$ stands for $r_{1,3}$ in set R_1.

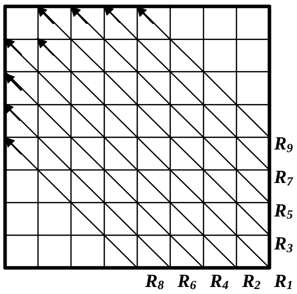


Figure 4. Coefficient sets for embedding

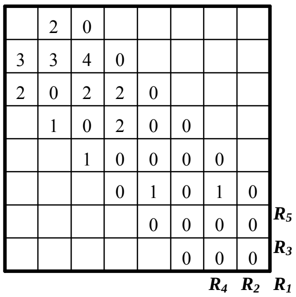


Figure 5. Example of quantized coefficients

To make sure the embedded coefficients can be successfully restored during the restoration phase, Chang et al.'s data embedding strategies and embedding rules for ambiguous conditions are listed below.

Rule 1: If $b_i \geq 2$, we use the value of $z_{i,2}$ to indicate the hidden secret bit in set R_i ($1 \leq i \leq 9$). We modify the value of $z_{i,2}$ to hide secret bit by using the Equation (3):

$$z_{i,2} = \begin{cases} 0, & when\ s_i\ is\ 0, \\ 1\ or\ -1, & when\ s_i\ is\ 1, \end{cases} \tag{3}$$

Note that 1 or -1 is randomly selected to $z_{i,2}$ when s_i is 1.

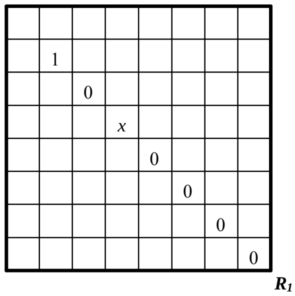


Figure 6. Example of an ambiguous condition when x =1 or -1

Ambiguous Condition A and its remedial solution: Before data hiding, Chang et al.'s scheme must eliminate any potentially ambiguous conditions. If the sequence of set R_i is (0, 0,…, *x*, 0) and all coefficients of $(r_{i,1},\ r_{i,2},\ ..,\ r_{i,j-2})$ are zeros, where $x \neq 0$, $4 \leq j \leq k_i$. According to our definition, $z_{i,2}$ is $r_{i,j-3}$ in set R_i. Once secret bit s_i equals 1 and *x* is 1 or -1, the receiver might make a false judgment while extracting data from the set R_i. Figure 6 demonstrates an ambiguous condition for *x*=1 or -1 and the coefficients located in the components at higher frequencies than *x* are all zeros. In addition, the value of the upper left component of *x* is also zero.

To avoid this ambiguous condition and guarantee that the original coefficient can be successfully restored, the coefficient $r_{i,j-1}$ is modified as shown in Equation (4) before the secret bit can be hidden.

$$r_{i,j-1}' = \begin{cases} r_{i,j-1}+1, & \text{when } r_{i,j-1} > 0, \\ r_{i,j-1}-1, & \text{when } r_{i,j-1} < 0, \end{cases} \quad where\ 3 \leq (j-1) \leq k_i \tag{4}$$

Return to set R_2 in Figure 5. The corresponding coefficient sequence ($r_{2,1}, r_{2,2}, r_{2,3}, r_{2,4}, r_{2,5}, r_{2,6}, r_{2,7}$) of set R_2 is (0, 0, **1**, 0, 0, 0, 3). In set R_2, $r_{2,3}$ is $z_{i,2}$, so we must modify $r_{2,3}$ to hide the secret bit. However, once we modify $r_{2,3}$ according to Equation (3), the receiver may make the misjudgment that the hidden bit is $r_{2,3}$ instead of $r_{2,1}$. To avoid this potential misjudgment, we must change the value of $r_{2,3}$ from 1 to 2 according to Equation (4). The modified coefficient sequence of set R_2 is then presented as (0, 0, **2**, 0, 0, 0, 3). In general, the successful embedding of each set of a DCT-quantized coefficient block causes no more than two coefficients to be modified.

Rule 2: If b_i **< 2** and both $z_{i,1}$ and $z_{i,2}$ do not exist, none secret bits can be hidden in a set R_i.

Two ambiguous conditions may exist and therefore two remedial measures for eliminating them are described below.

Ambiguous Condition B and its remedial solution: If the two highest coefficients $r_{i,1}$ and $r_{i,2}$ of set R_i are *x* and 0, respectively, $r_{i,1}$ is changed according to Equation (5) to demonstrate no secret data are hidden and make sure the ambiguous condition does not occur again.

$$r_{i,1}' = \begin{cases} r_{i,1}+1, & \text{when } r_{i,1} > 0, \\ r_{i,1}-1, & \text{when } r_{i,1} < 0. \end{cases} \tag{5}$$

Ambiguous Condition C and its remedial solution: If the three highest coefficients $r_{i,1}$,$r_{i,2}$ and $r_{i,3}$ of set R_i are 0, x and 0, respectively, the value of $r_{i,2}$ is modified according to Equation (6).

$$r_{i,2}' = \begin{cases} r_{i,2}+1, & \text{when } r_{i,2} > 0, \\ r_{i,2}-1, & \text{when } r_{i,2} < 0. \end{cases} \qquad (6)$$

Example of Chang et al.'s Data Embedding:

The example demonstrated in Figure 7 shows the embedded coefficients without ambiguous conditions in Figure 7(b) when Chang et al.'s data embedding strategies are used. Note that R_2 and R_3 shown in Figure 7(a) are the hidden coefficients without ambiguous conditions, but R_1, R_4 and R_5 have ambiguous conditions after data embedding and they may lead to the restoration phase not working properly; therefore, the elimination solutions must apply to these three sets. The eliminated results are shown in Table I.

	2	0					
3	3	4	0				
2	0	2	2	0			
	1	0	2	0	0		
		1	0	0	0	0	
			0	1	0	1	0
				0	0	0	0
					0	0	0

R_5 R_3 R_4 R_2 R_1

(a) Original DCT coefficient in a block

	2	0					
3	3	4	0				
2	0	2	2	0			
	1	0	2	0	0		
		1	0	0	0	0	
			0	2	0	2	0
				0	0	0	0
					0	0	0

R_5 R_3 R_4 R_2 R_1

(b) Modified coefficient block without an ambiguous situation

Figure 7. Example of hiding four bits into five sets in a block

TABLE I.
SETS OF FIGURE 7(D), SECRET BITS AND FINAL HIDDEN RESULTS

Set	k_i	Coefficients ($r_{i,1} \rightarrow r_{i,k}$)	$z_{i,2}$	s_i	Modified coefficients ($r'_{i,1} \rightarrow r'_{i,k}$)
R_1	7	(0, 0, 0, 0, 2, 2, 3)	$r_{1,3}$	0	(0, 0, **0**, 0, 2, 2, 3)
R_2	7	(0, 0, 2, 0, 0, 0, 3)	$r_{2,1}$	0	(**0**, 0, 2, 0, 0, 0, 3)
R_3	7	(0, 2, 0, 0, 2, 4, 2)	not exist	not exist	(0, 2, 0, 0, 2, 4, 2)
R_4	6	(0, 0, 0, 1, 1, 2)	$r_{4,2}$	1	(0, **1**, 0, 1, 1, 2)
R_5	6	(0, 0, 0, 0, 0, 0)	$r_{5,5}$	1	(0, 0, 0, 0, **1**, 0)

C.2. Data Extracting Procedure

Chang et al.'s extracting procedure consists of five steps. Detailed descriptions of the five steps follow.

Step 1. Obtain non-overlapping 8×8 blocks of quantized DCT coefficients of the Y components from a JPEG stego-image after Huffman decoding and runlength decoding.
Step 2. Scan each block according to a predetermined order.
Step 3. For each set R_i in a block, let $r_{i,j}$ be the highest frequency non-zero component, where $1 \le i \le 9$ and $1 \le j \le k_i$.
Step 4. Extract s_i from set R_i by using the following rules:

Rule 1: If $r_{i,j}$ = 1 or -1 and $r_{i,j+1}$ = 0, then s_i is 1 and mark $r_{i,j}$ as $z_{i,2}$.
Rule 2: If $r_{i,j}$ = 1 or -1, $r_{i,j+1} \neq 0$, $r_{i,j-1}$ = 0 and $r_{i,j-2}$ = 0, then s_i is 0 and mark $r_{i,j-2}$ as $z_{i,2}$ where $j-2 \ge 1$.
Rule 3: If $r_{i,j}$ = 1 or -1 and $r_{i,j+1} \neq 0$ for $j \le 2$, none secret bit in set R_i. That is, s_i does not exist in set R_i.
Rule 4: If $r_{i,j} \neq$ 1 or -1, $r_{i,j-1}$ = 0 and $r_{i,j-2}$ = 0, then s_i is 0 and mark $r_{i,j-2}$ as $z_{i,2}$, where $j-2 \ge 1$.
Rule 5: If $r_{i,j} \neq$ 1 or -1 and $j \le 2$, none secret bits are in set R_i. That is, s_i does not exist in set R_i.
Rule 6: If $r_{i,j}$ does not exist, then s_i is 0 and mark $r_{i,1}$ as $z_{i,2}$.

Step 5. Repeat Steps 3 and 4 until all blocks are processed.

Take Table I for example. In set R_1, the highest frequency non-zero value is $r_{1,5}'$ and the pair ($r_{1,3}'$, $r_{1,4}'$) is (0, 0), which satisfies Rule 4, so secret bit s_1 is 0. The secret data in set R_2 are extracted in the same way as R_1 and secret bit s_{2i} is 0. No secret bit is hidden in set R_3 because $r_{3,2}'$ does not equal 1 or -1 and Rule 5 is satisfied. In set R_4, the highest frequency non-zero coefficient is $r_{4,2}'$ and equals 1. Moreover, the value of $r_{4,3}'$ is 0, which satisfies Rule 1; therefore, the secret bit is 1. Based on the same rule, the secret bit extracted from set R_5 is also the same as R_4's.

TABLE II.
SETS OF FINAL HIDDEN RESULT, EXTRACTED SECRET BITS AND RELATED DATA

Set	k_i	Modified coefficients ($r'_{i,1} \rightarrow r'_{i,k}$)	$z_{i,2}$	s_i
R_1	7	(0, 0, 0, 0, 2, 2, 3)	$r_{1,3}'$	0
R_2	7	(0, 0, 2, 0, 0, 0, 3)	$r_{2,1}'$	0
R_3	7	(0, 2, 0, 0, 2, 4, 2)	not exist	not exist
R_4	6	(0, 1, 0, 1, 1, 2)	$r_{4,2}'$	1
R_5	6	(0, 0, 0, 0, 1, 0)	$r_{5,5}'$	1

C.3. Data Restoring Procedure

Once Chang et al.'s extraction procedure is completed, the restoring procedure can begin. As Table II shows, some sets may not contain any secret bit because their b_i's are less than 2. During the extraction procedure, their scheme has already recognized the corresponding $z_{i,2}$ for each embeddable set. In the restoring procedure, the proposed scheme must first replace $z_{i,2}$ in each embeddable set with 0. Later, the original value of the modified coefficient in each set can be restored based on the following three rules. Let the location of $z_{i,2}$ in set R_i be $r_{r,j}'$ in each embeddable set

Rule 1: If s_i exists and $r_{i,j+3}'=0$, where $4 \le (j+3) \le k_i$, then the original value of $r_{i,j+2}'$ is restored by using Equation (7).

$$r_{i,j+2} = \begin{cases} r_{i,j+2}'-1, & when\ r_{i,j+2}' > 0, \\ r_{i,j+2}'+1, & when\ r_{i,j+2}' < 0, \end{cases} where\ 3 \le (j+2) < k_i. \quad (7)$$

Rule 2: If s_i does not exist and the two highest coefficients $(r_{i,1}',\ r_{i,2}')$ of set R_i equal $(x, 0)$, where $x \ne 0$, then the original value of $r_{i,1}'$ is restored by using Equation (8).

$$r_{i,1} = \begin{cases} r_{i,1}'-1, & when\ r_{i,1}' > 0, \\ r_{i,1}'+1, & when\ r_{i,1}' < 0. \end{cases} \quad (8)$$

Rule 3: If s_i does not exist and the pair having the three highest coefficients $(r_{i,1}',\ r_{i,2}', r_{i,3}')$ of set R_i equals $(0, x, 0)$, where $x \ne 0$, then the original value of $r_{i,2}'$ is restored by using Equation (9).

$$r_{i,2} = \begin{cases} r_{i,2}'-1, & when\ r_{i,2}' > 0, \\ r_{i,2}'+1, & when\ r_{i,2}' < 0. \end{cases} \quad (9)$$

	2	0						
3	3	4	1					
2	0	2	2	0				
	1	0	2	0	0			
		1	0	0	0	0		
			0	2	0	2	0	
				1	0	0	0	R_5
					0	0	0	R_3
					R_4	R_2	R_1	

(a) Hidden results

	2	0						
3	3	4	0					
2	0	2	2	0				
	1	0	2	0	0			
		1	0	0	0	0		
			0	2	0	2	0	
				0	0	0	0	R_5
					0	0	0	R_3
					R_4	R_2	R_1	

(b) Replacing zi,2 with zero

	2	0						
3	3	4	0					
2	0	2	2	0				
	1	0	2	0	0			
		1	0	0	0	0		
			0	1	0	1	0	
				0	0	0	0	R_5
					0	0	0	R_3
					R_4	R_2	R_1	

(c) Restored DCT coefficients

Figure 8. Example of restoring DCT coefficients

Figure 8(a) presents the final hidden results and Figure 8(b) presents the results after $z_{i,2}$ is replaced with 0 in each embeddable set. Let us take set R_2 as an example. During the extraction procedure, $r_{2,1}'$ is then recognized as $z_{2,2}$ in set R_2 in Table II. In set R_2, the coefficient sequence is (0, 0, 2, 0, 0, 0, 3) running from high frequency to low frequency, as shown in Figure 8(b). Because a secret bit exists and $r_{2,4}$ is 0, which satisfies Rule 1 for restoring the original coefficient, $r_{2,3}$ is changed to 1 by Equation (6). Set R_3 is not an embeddable set but its three highest coefficients are (0, 2, 0), which satisfies Rule 3; therefore, $r_{3,2}$ is also replaced with 1 by Equation (8). The restored DCT coefficient block is shown in Figure 8(c). Note that Figure 8(c) is exactly the same as Figure 7(a).

III. Our Proposed High Capacity and Reversible Data Embedding Scheme

As we mentioned at the beginning, we discovered that Chang et al.'s scheme can achieve reversibility and acceptable image quality at the same time, but does not effectively use the coefficients located in the medial frequency because its hiding strategies modify only the zero coefficient located in successive zero coefficients.

The following subsections give a detailed description of our proposed reversible data embedding scheme.

A. Data Embedding Procedure

Since our proposed scheme is an extension of Chang et al.'s scheme, the embedding procedure combines two data hiding strategies. One is Chang et al.'s data embedding strategy called Layer-2 data embedding, and the other is our proposed data embedding strategy called Layer-1 data embedding. In essence, both data embedding strategies achieve reversibility. Because we introduced Chang et al.'s data embedding strategy in previous subsection 2.C, in this subsection we focus on our proposed data embedding strategy. To make sure readers have a clear picture of our embedding approach, the embedding flowchart is shown in Figure 9.

In our data embedding method, we also inherit the set definition for DCT coefficients in an image block given in subsection A. The basic idea behind our proposed data embedding strategy is to hide one secret bit into two neighboring coefficients in a set. To implement such a data embedding mechanism, Tian's pixel expansion method is adopted as a part of our proposed data embedding strategy [2]. The only problem that must be solved is setting up an indicator to distinguish a coefficient as the hidden result when its value is "0". In the following paragraphs, we first introduce how Tian's pixel expansion method is applied to hide one secret bit into two neighboring DCT coefficients in a set. Then, we describe our elimination solution for ambiguous conditions that may occur during our data embedding strategy.

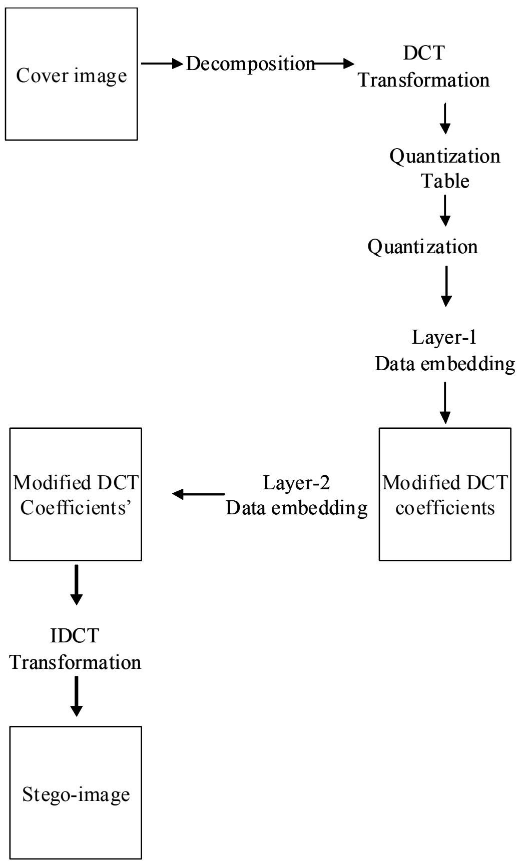


Figure 9. Flowchart of the proposed data embedding

Basically, three equations from Tian's scheme are used to design our data embedding strategy in Layer-1 data embedding, as follows:

$$avg=\left\lfloor \frac{P1+P2}{2} \right\rfloor, \tag{10}$$

$$d=P1-P2 \text{ and } d'=2*d+s, \tag{11}$$

$$P1'=avg+\left\lfloor \frac{d'+1}{2} \right\rfloor \text{ and } P2'=avg-\left\lfloor \frac{d'}{2} \right\rfloor, \tag{12}$$

The following two examples give detailed explanations of Tian's embedding based on Equations (10), (11) and (12). Let us assume that $P1=35$, $P2=30$, and $s=1$. Equations (10) to (12) yield the embedded pixels listed below.

$$P1' = avg + \left\lfloor \frac{d'+1}{2} \right\rfloor = 32 + \left\lfloor \frac{11+1}{2} \right\rfloor = 38, \text{ and}$$

$$P2' = avg - \left\lfloor \frac{d'}{2} \right\rfloor = 32 - \left\lfloor \frac{11}{2} \right\rfloor = 27.$$

The corresponding extraction equations are shown below.

$$avg=\left\lfloor \frac{P1'+P2'}{2} \right\rfloor, \tag{13}$$

$$s = d' - 2\times\left\lfloor \frac{d'}{2} \right\rfloor \text{ and } d = \left\lfloor \frac{d'}{2} \right\rfloor, \tag{14}$$

$$P1 = avg + \left\lfloor \frac{d+1}{2} \right\rfloor \text{ and } P1 = avg - \left\lfloor \frac{d}{2} \right\rfloor. \tag{15}$$

By using Equations (12) to (14), the hidden secret can be extracted and two embedded pixels can be successfully restored as follows. $d' = P1' - P2' = 38 - 27 = 11$, and $avg = \left\lfloor \frac{P1'+P2'}{2} \right\rfloor = \left\lfloor \frac{38+27}{2} \right\rfloor = 32.$ Next, d can be derived from d' as $d = \left\lfloor \frac{d'}{2} \right\rfloor = 5.$ Later, the embedded data s can be extracted from d' by computing $s = d' - \left\lfloor \frac{d'}{2} \right\rfloor \times 2 = 11 - 10 = 1$. Finally, the original pixels $P1$ and $P2$ are restored as $P1 = avg + \left\lfloor \frac{d+1}{2} \right\rfloor = 32 + \left\lfloor \frac{5+1}{2} \right\rfloor = 35$, and $P2 = avg - \left\lfloor \frac{d}{2} \right\rfloor = 32 - \left\lfloor \frac{5}{2} \right\rfloor = 30.$

To verify that the hidden secret bit can be extracted during the extracting procedure and two embedded neighboring coefficients can be successfully restored in the restoring procedure, we set the embedding conduction as follows: in a set, a non-zero DCT coefficient and a zero DCT coefficient (x, 0) are selected as the embedding pair when they are next to successive zero coefficients. Take R_1 :(0, 0, 0, 0, 2, 2, 3) shown in Figure 7(a) for example. In R_1, (2,0) is an embeddable pair when the coefficients pattern is one of following: N N x 0 0 0 or N N 0 x 0 0, where N denotes an integer and x denotes a integer except 0 and -1. Once an embeddable pair is determined, Equations (10) to (12) are used to hide a secret bit s in the embedding pair. For each embeddable pair, the next zero coefficient must be modified so that the misjudgment does not occur during the extraction procedure.

To avoid an ambiguous condition occurring after data embedding, we designed four ambiguous solutions in the proposed Layer-1 data embedding strategy, as defined in the following rules.

Rule 1: If an embeddable pair (x, 0, 0) is changed to (y, 0, 0) where x and y are non-zero coefficients which do not equal to -1 after data embedding, the next zero coefficient is changed to "-1". It means the embedded pair (y, 0, 0) is changed to (y, 0, -1) finally.

Rule 2: If an embeddable pair (x, 0, 0) is changed to (0, y, 0) where x and y are non-zero coefficients which do not equal to -1 after data embedding, the next zero coefficient is changed to "-1". It means the embedded pair (0, y, 0) is changed to (0, y, -1) finally.

Rule 3: If an embeddable pair (x, 0, 0) is changed to (y, -1, 0) where x and y are non-zero coefficients which do not equal to -1 after data embedding, the next zero coefficient is changed to "-1". It means the embedded pair (y, -1, 0) is changed to (y, -1, -1) finally.

Rule 4: The un-embeddable pair (N, N, -1) is changed to (N, N, 1) to make sure the misjudgment does not occur during the extraction procedure.

B. Data Extracting Procedure

Because we used Tian's method to design our embedding strategy for two neighboring coefficients (x, 0) that are close to successive zero coefficients. Tian's extraction procedure can also be used to extract the hidden secret bit from an embedded pair. Tian's extraction procedure can be easily implemented by using Equations (13) to (15) combined with our extraction principles, given below.

Principle 1: If the coefficients pattern (N, N, N, x, 0, -1) is found, where N and x denote an integer coefficient and a non-zero coefficient except -1, respectively, during the extracting procedure. Equations (13) to (15) are used to extract a hidden secret bit from (x, 0) in the embedded pair.

Principle 2: If the coefficients pattern (N, N, N, x, -1, -1) is found, where N and x denote an integer coefficient and a non-zero coefficient except -1, respectively, during the extracting procedure. Equations (13) to (15) are used to extract a hidden secret bit from (x, -1) in the embedded pair.

Principle 3: If the coefficients pattern (N, N, N, 0, x, -1) is found, where N and x denote an integer coefficient and a non-zero coefficient except -1, respectively, during the extracting procedure. Equations (13) to (15) are used to extract a hidden secret bit from (0, x) in the embedded pair.

Principle 4: If the coefficients pattern (N, N, N, x, y, 0) is found, where N denotes an integer coefficient and x and y denote a non-zero coefficient except -1, during the extracting procedure. Equations (13) to (15) are used to extract a hidden secret bit from (x, y) in the embedded pair.

Principle 5: If the coefficients pattern (N, N, N, 0, y, 0) is found, where N and y denote an integer coefficient and a non-zero coefficient except -1, respectively, during the extracting procedure, it means none secret is contained in the coefficient pattern.

Principle 6: If the coefficients pattern (N, N, N, N, N, y) is found, where N and y denote an integer coefficient and a non-zero coefficient except -1, respectively, during the extracting procedure, it means none secret is contained in the coefficient pattern.

C. Restoring Procedure

Once the conditions in principles 1-3 in the extracting procedure are met, the restoring procedure can begin. The corresponding restoring processing is listed below.

Principle 1: If the embedded coefficients pattern (N, N, N, x, 0, -1) is found, where N and x denote an integer coefficient and a non-zero coefficient except -1, respectively, during the extracting procedure. The indicator "-1" is changed to "0" in the coefficient pattern during the restoring procedure.

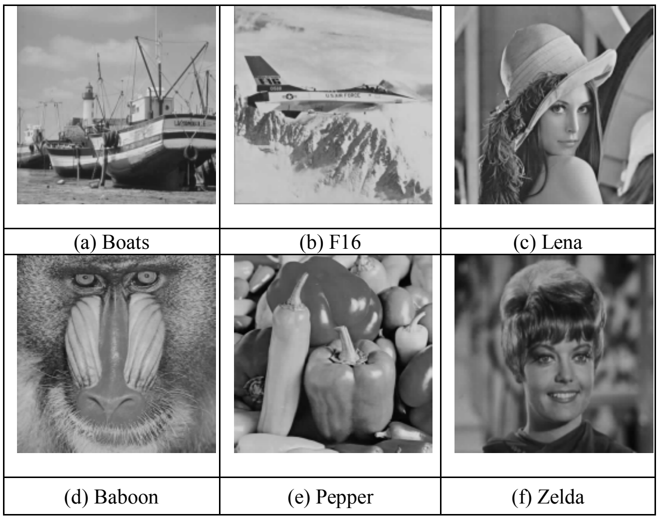
Principle 2: If the embedded coefficients pattern (N, N, N, x, -1, -1) is found, where N and x denote an integer coefficient and a non-zero coefficient except -1, respectively, during the extracting procedure. The indicator "-1" is changed to "0" in the coefficient pattern during the restoring procedure.

Principle 3: If the embedded coefficients pattern (N, N, N, 0, x, -1) is found, where N and x denote an integer coefficient and a non-zero coefficient except -1, respectively, during the extracting procedure. The indicator "-1" is changed to "0" in the coefficient pattern during the restoring procedure.

In addition, if the coefficients pattern (N, N, N, x, y, 1) is found, where N denotes an integer coefficient and x and y denote a non-zero coefficient except -1, during the extracting procedure. Change (N, N, N, x, y, 1) to (N, N, N, x, y, 0) during the restoring procedure.

IV. EXPERIMENTS

To prove that the hiding capacity of our hybrid data embedding scheme is higher than can be achieved with Chang et al.'s scheme and still maintain acceptable image quality of the stego-images, in this section, we further discuss hiding capacity and image quality of stego-images. Our hiding, extracting and restoring algorithms were developed using C programming language. Our simulation platform is Microsoft Windows XP, Pentium 4.3 with 3 GHz memory.



(a) Boats | (b) F16 | (c) Lena
(d) Baboon | (e) Pepper | (f) Zelda

Figure 10. Six 512×512 gray-level images served as test images

In our first experiment, we used the six 512×512 gray-level images shown in Figure 10 as our cover images, to which we applied the following algorithm.

Input: A 512×512 gray-level image as a cover image.

Output: A compressed image.

Step 1: Divide cover image into several non-overlapping blocks.

Step 2: Use DCT to transform each 8×8 block into DCT coefficients for each block.

Set 3: Perform quantization with the standard quantization table and our modified quantization table, as shown Figures 2 and 11, respectively.

Step 4: Use inverse DCT process to transform each block into the spatial domain.

Although the cover images used in our scheme are only achieved by quantization, they remain the same as those generated by JPEG compression because quantization is the only lossy process in JPEG compression. Therefore, the following experimental results are very similar to those achieved using JPEG compression images as cover images.

16	11	10	16	24	40	51	61
12	12	14	19	26	41	60	55
14	13	16	17	28	40	48	56
14	17	22	20	36	61	56	43
18	22	26	39	48	76	72	54
24	25	39	45	57	73	79	64
49	64	55	61	72	85	84	71
72	92	95	69	78	70	72	70

Figure 11. Modified quantization table

The peak signal to noise ratio (*PSNR*) used to evaluate the image quality is defined in Equation (16):

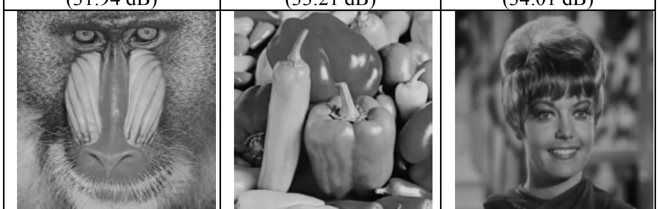
$$PSNR = 10\log_{10}\frac{255^2}{MSE}, \tag{16}$$

where the mean square error (MSE) for an $M \times N$ gray-level image is defined in Equation (17):

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{i,j} - x'_{i,j}\right)^2, \tag{17}$$

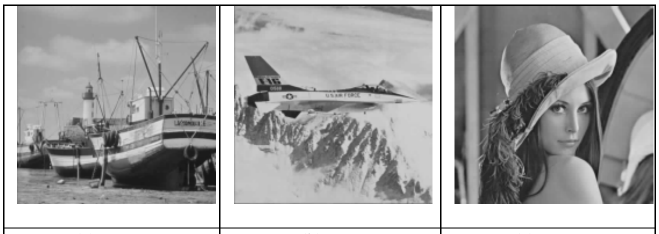
And where $x_{i,j}$ and $x'_{i,j}$ are the pixel values of the cover and stego-images, respectively. Note that Iwata et al.'s scheme can not offer reversibility and its average hiding capacity is less than 40000 bits; therefore, Iwata et al.'s experimental results are not included in the following comparisons.

(a) Boats (31.94 dB)	(b) F16 (33.21 dB)	(c) Lena (34.01 dB)



(d) Baboon (26.70 dB)	(e) Pepper (34.09 dB)	(f) Zelda (36.33 dB)

Figure 12. Six stego-images when our proposed hybrid embedding scheme and the standard quantization table are used, and their corresponding *PSNR*s



(a) Boats (32.23 dB)	(b) F16 (33.44 dB)	(c) Lena (30.30 dB)

(d) Baboon (27.20 dB)	(e) Pepper (34.30 dB)	(f) Zelda (36.60 dB)

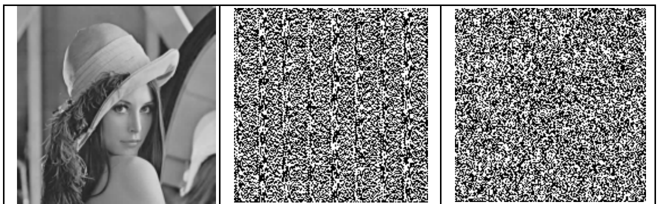
Figure 13. Six stego-images when our proposed hybrid embedding scheme and the modified quantization table are used, and their corresponding *PSNR*s

Comparing the six stego-images in Figures 12 and 13, it is obvious that the *PSNR*s of the latter images are higher than in the earlier images.

TABLE III.
COMPARISONS OF MEAN PSNRS OF THE SIX STEGO IMAGES WITH STANDARD AND MODIFIED QUANTIZATION TABLES

Cover images	Chang et al.'s scheme		Our scheme	
	Standard quantization table	Modified quantization table	Standard quantization table	Modified quantization table
Boats	36817	36710	58357	58426
Jet	36852	36817	57889	57966
Lena	36861	36850	57644	57717
Baboon	36094	35402	66048	66075
Pepper	36842	36804	56936	56981
Zelda	36864	36861	55184	55190

Furthermore, Table III shows that, on average, our modified quantization table can offer higher hiding capacity than that can be provided by Chang et al.'s scheme with the standard quantization table or the modified quantization table.

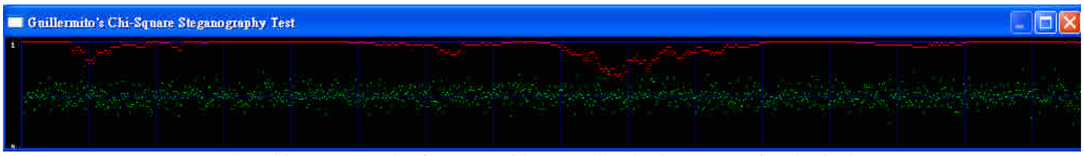


(a) Lena	(b) Enhanced LSBs of (a) while secret data is	(c) Enhanced LSBs of stegano-image of (a)

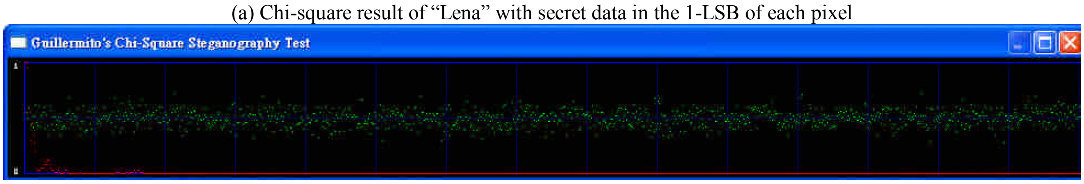
	carried in the LSBs	

Figure 14. Example of "Lena" for visual attack using enhancing LSBs.

To prove our proposed data embedding strategy does not affect the security of Chang et al.'s scheme, we used two general attacks, statistical attack and visual attack, to analyze stego-images generated by our proposed scheme. Enhancing the least significant bits of an LSB-based stego-image reveals several regular patterns once the secret data are inside, as Figure 14(b) shows. Our proposed scheme, on the other hand, hides secret data by modifying the DCT coefficients rather than by directly modifying the least significant bits of each pixel in a cover image. Therefore, no regular patterns appear in our stego-image, as can be seen in Figure 14(c). To further prove that our proposed scheme can withstand a Chi-square attack, we used a Chi-square steganography test program provided by Guillermito [10] to perform steganography analyses. Figure 15 shows the test results. In Figure 15, the red curve is the result of the Chi-square test. It is close to 1, so the probability for a random embedded message is high. The second output is green curve that presents the average value of the LSBs. In Figure 15(a), the green curve stays at about 0.5, which means a random message is embedded. Note that in Figure 15(b), the green average of LSBs varies considerably and the Chi-square red output is flat at zero all along the picture. In other words, nothing is hidden in our stego-image.



(a) Chi-square result of "Lena" with secret data in the 1-LSB of each pixel



(b) Chi-square result of stego-image of "Lena"

Figure 15 Example of "Lena" for statistical attack using Chi-square analysis.

V. CONCLUSIONS

DCT is a widely used mechanism for frequency transformation. To extend the variety of cover images and for the sake of repeated usage, Chang et al. proposed a lossless data hiding scheme for DCT-based compressed images in 2007. In this paper, we enhance the hiding capacity provided by Chang et al.'s scheme while maintaining acceptable image quality of stego-images.

By combining our proposed data embedding method with Chang et al.'s scheme, we successfully designed a hybrid data embedding scheme offering both reversibility and high hiding capacity properties while maintaining acceptable image quality of stego-image about 30 dB. Experimental results confirm that the modified quantization table offers higher hiding capacity and better image quality of stego-images. They further demonstrate that our proposed scheme outperforms Chang et al.'s scheme in hiding capacity and image quality of stego-images.

ACKNOWLEDGMENTS

Some of the results demonstrated in this paper are sponsored by the National Science Council, Project Number NSC-97-2221-E-126-010.

REFERENCES

[1] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless Generalized-LSB Data Embedding," *IEEE Transactions on Image Processing*, Vol. 14, No. 2, Feb. 2005, pp. 253-266.

[2] Chin-Chen Chang, Chia-Chen Lin, Chun-Sen Tseng and Wei-Liang Tai, "Reversible hiding in DCT-based compressed images," *Information Sciences*, Vol. 177, Issue 13, July 2007, pp. 2768-2786.

[3] C. C. Chang, T. S. Chen and L. Z. Chung, "A Steganographic Method Based upon JPEG and Quantization Table Modification," *Information Sciences*, Vol. 141, 2002, pp.123-138.

[4] C. C. Chang, J. Y. Hsiao, and C. S. Chan. "Finding Optimal Least-Significant-Bit Substitution in Image Hiding by Dynamic Programming Strategy," *Pattern Recognition*, Vol. 36, No. 7, July 2003, pp. 1595-1683.

[5] C. C. Chang, W. L. Tai, and C. C. Lin, "A Reversible Data Hiding Scheme Based on Side Match Vector Quantization," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 10, pp. 1301-1308, 2006.

[6] K. L. Chung, C. H. Shen and L. C. Chang, "A Novel SVD- and VQ-based Image Hiding Scheme," *Pattern Recognition Letters*, Vol. 22, No. 9, 2001, pp.1051-1058.

[7] B. V. Dasarathy, "Image Data Compression: Block Truncation Coding," IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 164-173.

[8] W. Diffie and M. E. Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard," *IEEE Computer*, Vol. 10, 1977, pp. 74-84.

[9] J. Fridrich, M. Goljanb, R. Du, "Invertible Authentication Watermark for JPEG Images," *IEEE Int. Conf. Information Technology: Coding and Computing*, Las Vegas, Nevada, April 2-4, 2001, pp. 223-227.

[10] Guillermito, Chi-square Steganography Test Program, http://www.guillermito2.net/stegano/tools/index.html.

[11] M. Iwata, K Miyake, and A. Shiozaki, "Digital Steganography Utilizing Features of JPEG Images," *IEICE Transactions on Fundamentals*, Vol. E87-A, No. 4, April 2004, pp.929-936.

[12] H. Kobayashi, Y. Noguchi and H. Kiya, "A Method of Embedding Binary Data into JPEG Bitstreams," *IEICE Transactions on Fundamentals*, J83-D2, Vol. 6, 2000, pp. 1469-1476.

[13] Y. K. Lee and L. H. Chen, "High Capacity Image Steganographic Model," *Proc. of IEE International Conference on Vision, Image and Signal Processing*, Vol. 147, No. 3, 2000, pp. 288-294.

[14] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, "Reversible Data Hiding," *Proc.s of the 2003 International Symposium on Circuits and Systems (ISCAS '03)*, Vol. 2, May 2003, pp. II-912-II-915.

[15] R.L. Rivest, A. Shamir, L. Adelman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystem," *Communications of the ACM*, Vol. 21, No. 2, Feb. 1978, pp. 120-126.

[16] J. Tian, "Reversible Data Embedding Using a Difference Expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 8, Aug. 2003 pp. 890-896.

[17] P. Tsai, Y. C. Hu and C. C. Chang, "An Image Hiding Technique Using Block Truncation Coding," *Proc. of Pacific Rim Workshop on Digital Steganography*, Kitakyushu, Japan, July 2002, pp. 54-64.

[18] R. Z. Wang, C. F. Lin and J. C. Lin, "Image Hiding by Optimal LSB Substitution and Genetic Algorithm," Pattern Recognition, Vol. 34, No. 3, 2001, pp.671-683.

[19] G. Xuan, J. Zhu, J. Chen, Y.-Q. Shi, Z. Ni, and W. Su, "Distortionless Data Hiding Based on Integer Wavelet Transform," *IEE Electronics Letters*, vol. 38, no. 25, December 2002, pp. 1646-1648.