Available online at www.sciencedirect.com
ScienceDirect
Information Sciences 177 (2007) 2768–2786
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Reversible hiding in DCT-based compressed images

Chin-Chen Chang [a,b], Chia-Chen Lin [c,*], Chun-Sen Tseng [b], Wei-Liang Tai [b]

[a] *Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan, ROC*
[b] *Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan, ROC*
[c] *Department of Computer Science and Information Management, Providence University, Taichung 433, Taiwan, ROC*

Received 19 July 2005; received in revised form 15 February 2007; accepted 15 February 2007

Abstract

This paper presents a lossless and reversible steganography scheme for hiding secret data in each block of quantized discrete cosine transformation (DCT) coefficients in JPEG images. In this scheme, the two successive zero coefficients of the medium-frequency components in each block are used to hide the secret data. Furthermore, the scheme modifies the quantization table to maintain the quality of the stego-image. Experimental results also confirm that the proposed scheme can provide expected acceptable image quality of stego-images and successfully achieve reversibility.

Keywords: Discrete cosine transform; Reversible data hiding; Lossless steganography

1. Introduction

With the digitalization of data and the networking of communication, the issue of security over the Internet is becoming more and more crucial. In essence, the Internet is an open channel and security problems such as interception, modification and others are very real. Several approaches have been proposed to make communication via the Internet secure. In such schemes, a secret message is protected by transforming it into an unrecognizable form. Only an authorized user can retransform it back to its original form by using secret information shared between senders and receivers. Many famous encryption schemes, such as RSA [16], DES [9] and the like have been widely used in the commercial market. However, the meaningless form could leave a clue and thus allow an unauthorized user to expose the original message. Another approach, called steganography, hides a secret message in a widespread cover material to avoid detection. The concept of steganography is similar to the concept of camouflage, which many animals use to protect against attack.

* Corresponding author. Tel.: +886 4 26328001x18108; fax: +886 4 26324045.
E-mail addresses: ccc@cs.ccu.edu.tw (C.-C. Chang), mhlin3@pu.edu.tw (C.-C. Lin), tcs92@cs.ccu.edu.tw (C.-S. Tseng), taiwl@cs.ccu.edu.tw (W.-L. Tai).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2007.02.019

Basically, hiding the subjects of steganographies involves the spatial [1,3,5,14,15,17,19] and frequency [2,12,13] domains of host images.

In the spatial domain manner, the secret is hidden directly in the pixels. The most commonly used methods in the spatial domain approach are least significant bit (LSB) or LSB-like embedding. For example, in Lee and Chen's scheme [14], the least significant bit (LSB) of each pixel in the cover image was modified to embed the secret message. In Chang et al.'s scheme [3], a dynamic programming strategy was employed to find the optimal LSB substitution for image hiding. In 2006, by incorporating both run-length encoding and modular arithmetic, Chang et al. [5] proposed two efficient data hiding methods for embedding bitmap files and general gray scale files, respectively, into gray scale images. Aside from LSB or LSB-like embedding methods, several other schemes exist that utilize different hiding methods to embed secret data in the spatial domain of a cover image. For example, Chung et al. offered a singular value decomposition (SVD)-based hiding scheme [7], and Tsai et al. utilized the bit plane of each block truncation coding (BTC [8]) block to embed secret messages [18].

In the frequency domain [2,12,13], cover images must first be transformed using a frequency-oriented mechanism such as discrete cosine transformation (DCT), discrete wavelet transformation (DWT) or similar mechanisms, after which the secret is then combined with the related coefficients in the frequency-form images to achieve embedding. For example, in Chang et al.'s scheme [2], the medium-frequency coefficients of the DCT-transformed cover image were employed to embed a secret message. The quantization table of the JPEG was also modified to further protect the embedded secret message. In the same way, Iwata et al. also utilized the boundaries between zero and non-zero DCT coefficients to hide secret data [12].

In addition to hiding secret data in the frequency domain, in recent years another branch of research known as reversible data hiding has been explored for use in some sensitive applications such as military, medical and fine arts data. In the spatial domain, Tian explored redundancies in the digital content to achieve a reversible stego-image [17], 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 side-mach quantization vector (SMVQ) and further proposed two reversible data hiding scheme [4,6]. In the frequency domain, Fridrich et al. [10] presented an invertible watermarking scheme for authenticating digital images in the JPEG domain. This scheme used 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. [20] proposed a high-capacity distortion-free data hiding technique based on the integer wavelet transform. Histogram modification was used in Xuan et al.'s scheme to embed secret data into the middle frequency of the wavelet domain. Their scheme can also be applied in JPEG2000-compressed images because JPEG2000 is based on the wavelet transform domain.

Although Iwata et al. [12] and Fridrich et al. [10] have tried to hide secret information in JPEG images, the former does not achieve reversibility and the latter is reversible only at the cost of limited hiding capacity. In this paper, we propose a reversible data hiding scheme for DCT-based compressed images that increases the security of hidden secret data and enhances hiding capacity. To maintain the quality of stego-images, we focus on modifying the DCT-quantized coefficients of the middle frequency components in each block because the human vision system is more sensitive to noise in the lower frequency. Moreover, we modify the quantization table to improve the quality of stego-images without significantly decreasing hiding capacity. Experimental results confirm that the proposed scheme can achieve both satisfactory image quality and high hiding capacity in stego-images.

The rest of this paper is organized as follows. In Section 2, we briefly review the DCT transform and Iwata et al.'s data hiding scheme. Our proposed reversible data hiding scheme, including the hiding phase, the extracting phase, and the reversing phase, are then illustrated in Section 3. After that, Section 4 presents our experimental results and demonstrates the superiority of our proposed scheme over others. Finally, concluding remarks appear in Section 5.

2. Related works

In this section, we briefly review the DCT transform and introduce how Iwata et al. [12] hide secret data in the DCT coefficients.

2.1. Discrete cosine transform (DCT) and quantization

DCT is a widely used mechanism for image transformation and has been adopted by JPEG to compress images. The flowchart in Fig. 1 illustrates the JPEG compression process, which consists of five phases: transforming an RGB image to an YC_bC_r image, composition of minimum coding units, 2-dimensional discrete cosine transform (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 2-D DCT equation:

$$F(u,v) = \frac{c(u)c(v)}{4}\sum_{i=0}^{7}\sum_{j=0}^{7}\cos\left(\frac{(2i+1)u\pi}{16}\right)\cos\left(\frac{(2j+1)v\pi}{16}\right)f(i,j),$$
$$c(e) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } e = 0, \\ 1, & \text{if } e \neq 0. \end{cases} \tag{1}$$

Here, $F(u,v)$ and $f(i,j)$ present a DCT coefficient at the coordinate (u,v) and a pixel value at the coordinate (i,j), respectively. $F(0,0)$ is called the direct current (DC) component, which corresponds to an average intensity value of each block in the spatial domain. $F(u,v)$ is called the alternating current (AC) component, in which $u \neq 0$ and $v \neq 0$.

For data reduction during the quantization phase, DCT coefficients are quantized by using a quantization table as shown in Fig. 2. In general, the human vision system is much more sensitive to the values in the low-frequency components than those in the higher frequencies; distortion in high-frequency components is visually acceptable and non-imperceptible. Therefore, the quantization process takes advantage of this feature to reduce the number of DCT coefficients. That is, 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.

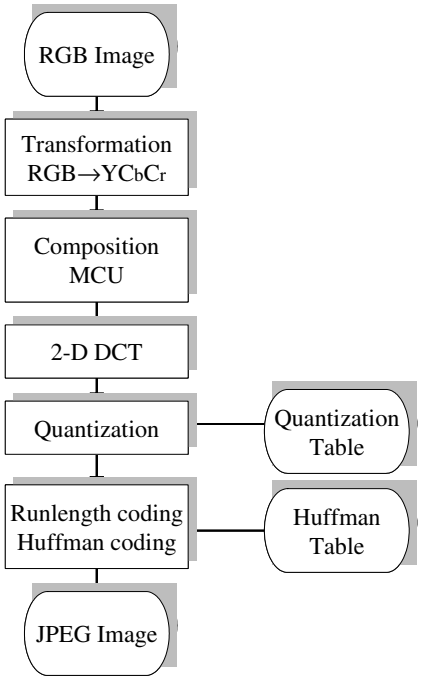

Fig. 1. Flowchart of JPEG compression.

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

Fig. 2. Standard quantization table.

2.2. Iwata et al.'s data hiding scheme

Iwata et al. discovered that the values of AC coefficients tend to be zero after the quantization phase of JPEG compression; therefore, their scheme hides secret information in high-frequency components by the length of zero sequences after quantization of the DCT coefficients [12]. According to their modification strategy, a JPEG-coded stego-image is generated after the runlength coding and Huffman coding. The modified DCT coefficients are reserved once 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 consists of embedding and extracting procedures. Detailed descriptions of the two procedures follow.

2.2.1. Embedding procedure

To hide secret data, Iwata et al. defined a set for embedding one bit as shown in Fig. 3.

The set D_i $(1 \leqslant i \leqslant l_i)$ contains quantized DCT coefficients on the line labeled "D_i" in Fig. 3. We assume that l_i is the length of a zero sequence of higher frequency components on the "D_i" line in Fig. 3. The odd or even state of l_i indicates what secret data are embedded in 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 $1 \leqslant j \leqslant k_i$, and T be a predetermined threshold.

Based on these sets of a block, Iwata et al. proposed four modification strategies for different cases. Their kernel concept 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 are described as follows:

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

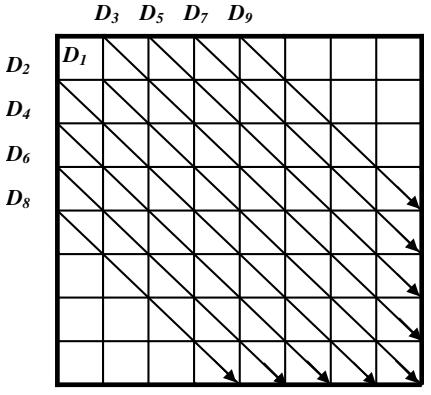

Fig. 3. Sets for Iwata et al.'s embedding.

Case 2: If $|d_{i,j}| \leqslant T$ and $d_{i,j-1}$ equals 0, $d_{i,j}$ and $d_{i,j-1}$ will be replaced with 0 and 1, respectively.
Case 3: If $|d_{i,j}| \leqslant T$ but $d_{i,j-1}$ does not exist or does not equal 0, $d_{i,j}$ will be set at 0.
Case 4: If $d_{i,j}$ does not exist, which means all components in set D_i are zero, a 1 or -1 is assigned to the lowest coefficient in the set D_i, where -1 and 1 are randomly selected.

2.2.2. Extracting procedure

Upon receiving a stego-image from a sender, the receiver must take the following steps to extract hidden secret information from each 8×8 block:

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 the following equation:

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

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

Iwata et al. successfully embedded a secret bit into a set in an 8×8 block of a JPEG image and caused only a tiny difference in the histogram of quantized DCT coefficients. However, their hiding scheme is not reversible. That is, the original cover image cannot be restored even after the hidden secret data are extracted. To enhance reversibility in the Iwata et al.'s scheme, we propose a reversible data hiding scheme for JPEG-coded images. Detailed descriptions of our proposed scheme appear in Section 3.

3. Proposed scheme

The proposed scheme embeds secret bits into a DCT-based compressed image and restores the original DCT coefficients after the secret bits have been extracted. Our scheme can be divided into three procedures: embedding, extracting and restoring. The preprocessing of 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 8×8 block of DCT coefficients. Later, the quantized coefficients are obtained by the 8×8 quantization table shown in Fig. 2.

After quantizing the DCT coefficients, the proposed scheme embeds secret data into the successive zero sequence in the middle-frequency components. To eliminate any potential misjudgment occurring during the extracting phase, we propose three elimination measures. The related embedding, extracting and restoring procedures are presented in the following subsections.

3.1. Embedding procedure

To hide secret data, the proposed scheme defines several sets R_i $(1 \leqslant i \leqslant 9)$ for embedding secret bits, as shown in Fig. 4. In each set, we embed secret data into 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 $(1 \leqslant i \leqslant 9)$. Let b_i $(1 \leqslant i \leqslant 9)$ 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 the key to deciding whether set R_i can hide a secret bit. The estimation rule is straightforward: if $b_i \geqslant 2$, set R_i can hide a secret bit; otherwise, set R_i cannot hide a secret bit. Let us use the example in Fig. 5 to further explain. In Fig. 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. Because the values of b_1 and b_2 are larger than or equal to 2, sets R_1 and R_2 can be used to hide secret data. Conversely, R_3 cannot hide secret data because its b_3 is less than 2.

Following the above rules, in set R_i, if $b_i \geqslant 2$, $z_{i,1}$ 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 $z_{i,2}$ will not exist once b_i is less than

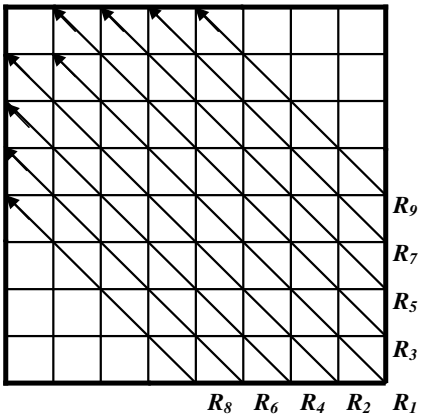

Fig. 4. Sets for embedding.

	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	R5
					0	0	0	R3
						R4	R2	R1

Fig. 5. Example of quantized coefficients.

2 in set R_i (e.g., in the set R_3 shown in Fig. 5). Let $(r_{i,1}, r_{i,2}, \ldots, r_{i,k_i})$ 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 into set R_i. Refer to set R_1 in Fig. 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.

The embedding strategies and elimination measures for ambiguous conditions are as follows:

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

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

where 1 or -1 is randomly selected.

3.1.1. Ambiguous condition A and its remedial measure

Before hiding data, we must eliminate any potentially ambiguous conditions. If the sequence of set R_i is $(0,0,\ldots,x,0)$ and all coefficients of $(r_{i,1}, r_{i,2}, \ldots, r_{i,j-2})$ are zeros, where $x \neq 0$, $4 \leqslant j \leqslant k_i$. 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. Fig. 6 presents an ambiguous condition for $x = 1$ or -1 and the

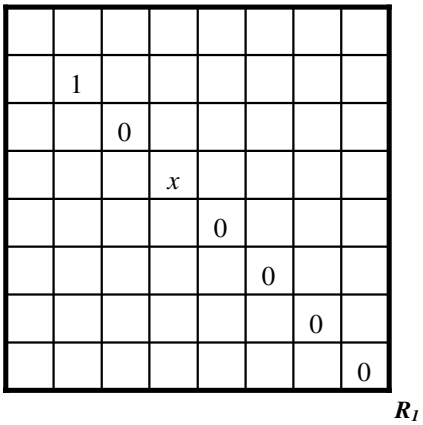

Fig. 6. Example of an ambiguous condition when $x = 1$ or -1.

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 restored successfully, the coefficient $r_{i,j-1}$ is modified as shown in Eq. (4) before the secret bit can be hidden.

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

Return to set R_2 in Fig. 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 $(0, 0, \mathbf{1}, 0, 0, 0, 3)$. 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 Eq. (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, the value of $r_{2,3}$ must be changed from 1 to 2 according to Eq. (4). The modified coefficient sequence of set R_2 is then presented as $(0, 0, \mathbf{2}, 0, 0, 0, 3)$. In general, the successful embedding of each set of a DCT-quantized coefficient block will cause no more than two coefficients to be modified.

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

Although none secret bits can be embedded into a set R_i when $b_i < 2$ and both $z_{i,1}$ and $z_{i,2}$ do not exist, some ambiguous conditions still exist and may lead receivers to extract a non-existing secret bit. Two ambiguous conditions may exist, and therefore two remedial measures for eliminating them are described below.

3.1.2. Ambiguous condition B and its remedial measure

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 to demonstrate no secret hidden to eliminate the ambiguous as follows:

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

3.1.3. Ambiguous condition C and its remedial measure

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 as follows:

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

3.1.4. Example of embedding

To demonstrate the proposed embedding strategies in greater detail, Fig. 7 shows the original coefficients of a DCT block, the modified coefficients for eliminating ambiguous conditions and the final hidden results.

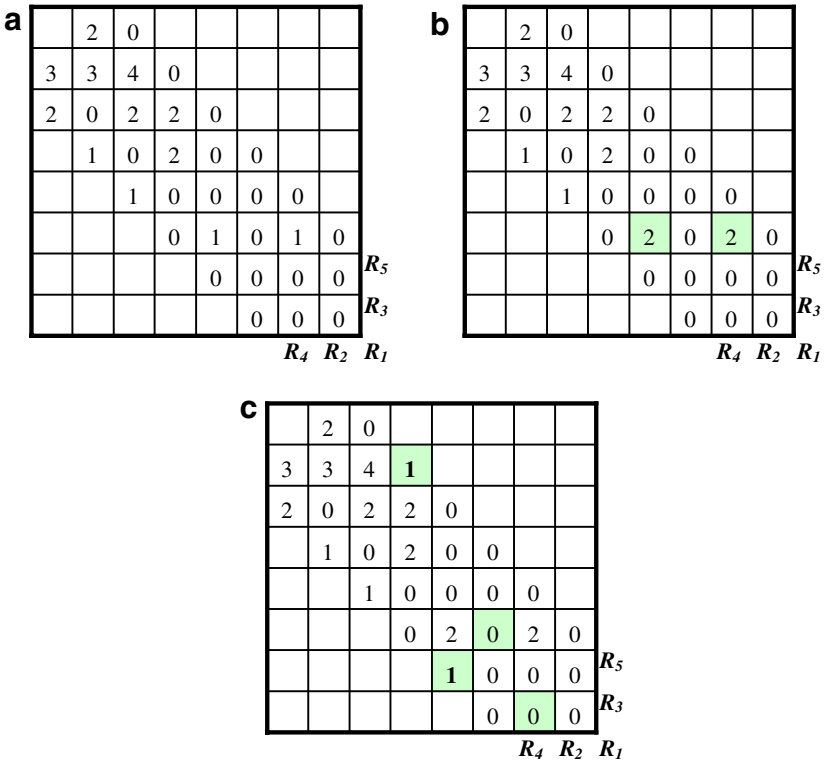

Fig. 7. Example of hiding four bits into five sets in a block: (a) original coefficient block, (b) modified coefficient block after elimination of ambiguous situations and (c) hidden results.

Before embedding secret information, the ambiguous conditions must be first examined and eliminated. Then, for embeddable set R_i $(1 \leqslant i \leqslant 9)$, the value of $z_{i,2}$ is modified according to Eq. (3).

After examining for ambiguous conditions, we can see that the four highest frequency coefficients of set R_2 are 0, 0, 1 and 0, respectively. Furthermore, the three highest coefficients of set R_3 are 0, 1 and 0, respectively. Both conditions could lead receivers to fail in extracting secret bits. Because set R_2 is in ambiguous condition A, the value of $r_{2,3}$ must be changed from 1 to 2 according to Eq. (4). Set R_3 is also in ambiguous condition B, so the value of $r_{3,2}$ is changed from 1 to 2 according to Eq. (6). Fig. 7b shows the modified coefficients that can be restored once receivers extract the secret bits. Because the modified set R_3 does not contain two successive zeros from the highest frequency component to the lower frequency components, we can only hide four bits into four sets: R_1, R_2, R_4 and R_5 of Fig. 7b. Let us assume four secret bits, 0, 0, 1 and 1, as listed in the s_i column of Table 1. We can hide those four bits into sets R_1, R_2, R_4 and R_5, individually. For each embeddable set, we first recognize its $z_{i,2}$. We then modify $z_{i,2}$ in each embeddable set to hide a secret bit. For set $R_1, r_{1,3}$ represents its $z_{i,2}$ and its corresponding secret bit is "0"; therefore, we do not need to modify the value of $r_{1,3}$ because the

Table 1
Sets of Fig. 7a, secret bits, hidden results and related data

Set	k_i	$(r_{i,1}, r_{i,2}, r_{i,3}, \ldots, r_{i,k_i})$	$z_{i,2}$	s_i	$(r'_{i,1}, r'_{i,2}, r'_{i,3}, \ldots, r'_{i,k_i})$
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)

value of $r_{1,3}$ is the same as s_1. For set R_5, $r_{5,5}$ represents its $z_{i,2}$. Because secret bit s_4 is "1" but the value of $r_{1,3}$ is 0, we must change the value of $r_{1,3}$ from 0 to 1 according to Eq. (3).

Table 1 presents both the sequences of coefficient sets shown in Fig. 7b and the hidden result of secret bit s_i. To hide secret data in each set except for set R_3, we find the corresponding $z_{i,2}$ for each set and replace it with the value of s_i. When we compare the hidden results $(r'_{i,1}, r'_{i,2}, r'_{i,3}, \ldots, r'_{i,k_i})$ in Table 1 with those in Fig. 7c, we see that they are the same.

3.2. Extracting procedure

The secret information can be extracted from a DCT-based stego-image as follows:

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 $1 \leqslant i \leqslant 9$ and $1 \leqslant j \leqslant k_i$.
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 $j - 2 \geqslant 1$.
Rule 3. If $r_{i,j} = 1$ or -1 and $r_{i,j+1} \neq 0$ for $j \leqslant 2$, 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 $j - 2 \geqslant 1$.
Rule 5. If $r_{i,j} \neq 1$ or -1 and $j \leqslant 2$, none secret bit 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.

Let us take Fig. 7c as an 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 bit in set R_2 is 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 means Rule 1 is satisfied; 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. All extracted secret bits are listed in the last column in Table 2.

3.3. Restoring procedure

The proposed restoring procedure only starts once the extraction procedure is completed. As shown in Table 2, some sets may not hide any secret bit because their b_i's are less than 2. During the extraction procedure, we already recognized the corresponding $z_{i,2}$ for each embeddable set. In the restoring procedure, we must first replace $z_{i,2}$ in each embeddable set with 0. We then restore the original value of the modified coefficient in each set as described below. Let the location of $z_{i,2}$ in set R_i be $r'_{r,j}$ in each embeddable set

Table 2
Sets of Fig. 7c, extracted secret bits and related data

Set	k_i	$(r'_{i,1}, r'_{i,2}, r'_{i,3}, \ldots, r'_{i,k_i})$	$z_{i,2}$	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

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

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

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

$$r_{i,1} = \begin{cases} r'_{i,1} - 1, & \text{when } r'_{i,1} > 0, \\ r'_{i,1} + 1, & \text{when } r'_{i,1} < 0. \end{cases} \tag{8}$$

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 $x \neq 0$, then the original value of $r'_{i,2}$ is restored by using Eq. (9)

$$r_{i,2} = \begin{cases} r'_{i,2} - 1, & \text{when } r'_{i,2} > 0, \\ r'_{i,2} + 1, & \text{when } r'_{i,2} < 0. \end{cases} \tag{9}$$

Fig. 8a shows the hidden results and Fig. 8b shows 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, we recognized $r'_{2,1}$ as $z_{2,2}$ in set R_2 in Table 2. 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 Fig. 8b. 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 Eq. (7). In set R_3, it 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 Eq. (9). The restored DCT coefficient block is shown in Fig. 8c. Note that Fig. 8c is exactly the same as Fig. 7a.

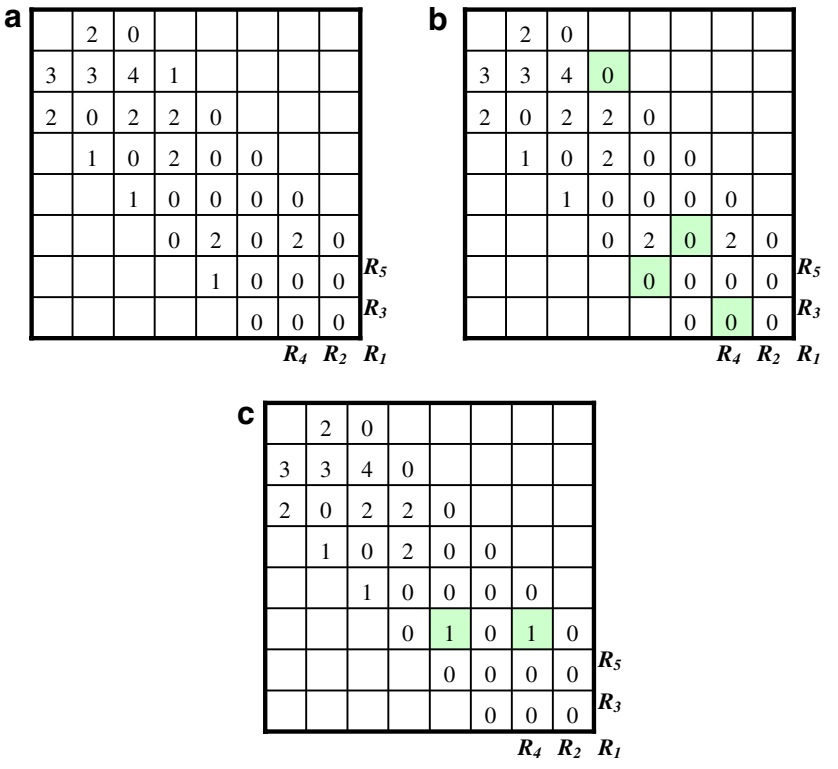

Fig. 8. Example of restoring DCT coefficients: (a) hidden results, (b) replacing $z_{i,2}$ with zero and (c) restored DCT coefficients.

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

Fig. 9. Modified quantization table.

3.4. Modifying quantization table for better image quality and hiding capacity

Tsai et al.'s scheme [2] proposed a hiding scheme based on JPEG compression with a modified quantization table. By modifying the quantization table, Tsai et al.'s scheme provides a large hiding capacity and achieves the expected acceptable quality of stego-images. Inspired by Tsai et al.'s scheme, we diminished the values in the medium- and high-frequency components of the general quantization table to enhance hiding capacity and increase the image quality of stego-images. Fig. 9 illustrates the modified quantization table. In this table, for sets R_i's, where $1 \leqslant i \leqslant 9$, coefficients in the medium and high frequencies are multiplied by 0.7. Section 4 presents supportive experimental results to prove that the modified quantization table can offer better image quality in stego-images while maintaining the same hiding capacity as the standard quantization table.

4. Experimental results

Proof of the reversibility of our proposed scheme has been provided in Section 3.3. In this section, we further discuss hiding capacity, image quality of stego-images, compression performance with and without secret data and security issues.

In our first experiment, we used six gray-level images as our cover images by applying the following algorithm.

Input: A 512×512 gray-level image C.
Output: A compressed image.
Step 1: Divide C into non-overlapping blocks.
Step 2: Use DCT to transform each 8×8 block into DCT coefficients.
Step 3: Perform quantization with standard quantization table and our modified quantization table, as shown Figs. 1 and 8, respectively.
Step 4: Use IDCT to transform each block into the spatial domain.

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

In Fig. 10, the six gray-level cover images are all 512×512 pixels in size. Each DCT coefficient block is 8×8 pixels in size. The secret data are a set of randomly generated bits. The peak signal to noise (PSNR) used to evaluate the image quality is defined as

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

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

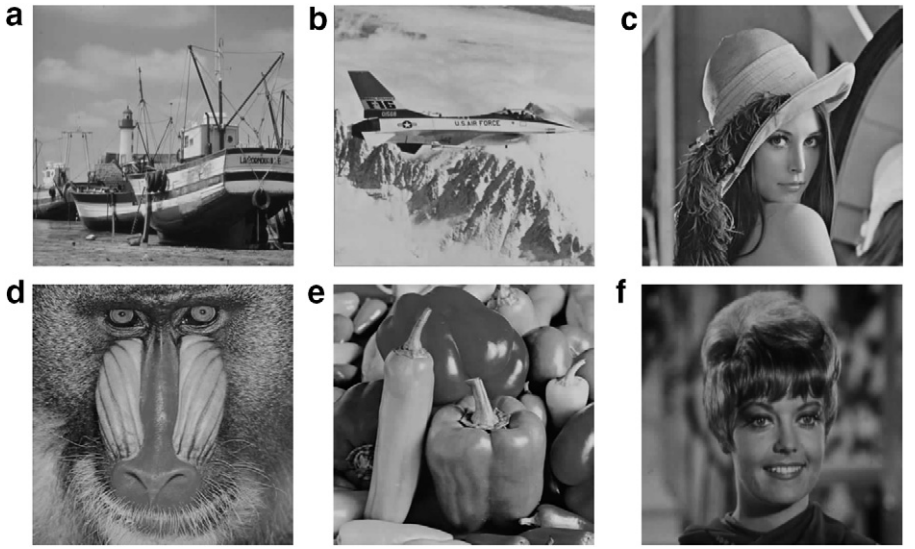

Fig. 10. Original cover images achieved with the standard quantization table: (a) Boats, (b) Jet, (c) Lena, (d) Mandrill, (e) Pepper and (f) Zelda.

$$\mathrm{MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(x_{i,j} - x'_{i,j})^2, \tag{11}$$

where $x_{i,j}$ and $x'_{i,j}$ are the pixel values of the cover and stego-images, respectively.

Let L be the number of bits we want to embed into each block (e.g., if $L = 3$, we will use sets R_1, R_2, and R_3 to hide information in each block). Fig. 11 presents the six hidden results employing the standard quantization table when $L = 9$. Secret information hidden in these six stego-images is imperceptible to the human eye.

Table 3 lists the PSNRs of the six stego-images. Although the PSNRs are not very high, the image quality in Fig. 11 is still visually acceptable for the human vision system. The corresponding hiding capacities of the stego-images are presented in Table 4. From Tables 3 and 4, we can see that a more complex image such

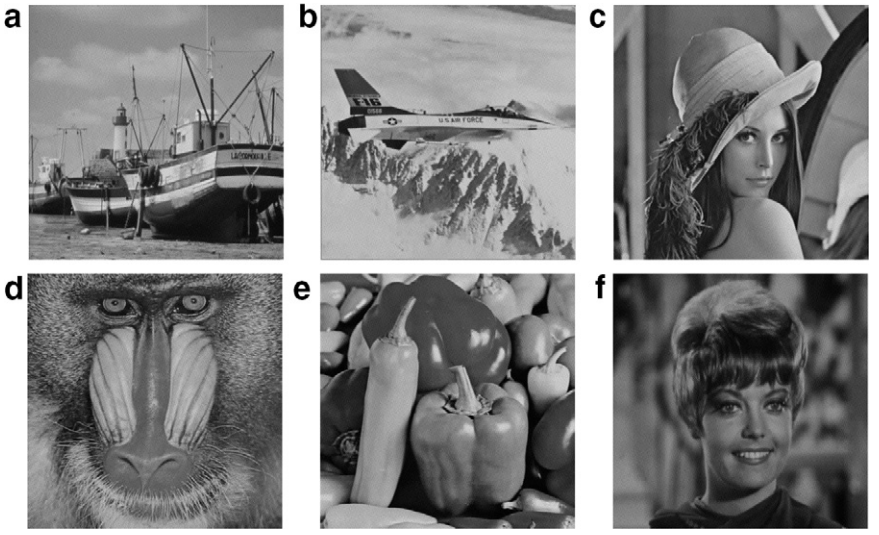

Fig. 11. Six stego-images when $L = 9$ and the standard quantization table are used, and their corresponding PSNRs: (a) Boats (27.49 dB), (b) Jet (27.73 dB), (c) Lena (28.13 dB), (d) Mandrill (24.22 dB), (e) Pepper (28.54 dB) and (f) Zelda (29.5 dB).

Table 3
PSNRs of stego-images when the standard quantization table is used

Cover images	L								
	1	2	3	4	5	6	7	8	9
Boats	38.77	35.08	32.70	31.33	29.35	28.90	28.45	28.07	27.49
Jet	38.68	35.21	33.10	31.97	29.73	29.22	28.81	28.37	27.73
Lena	38.55	34.93	32.91	31.77	29.65	29.24	28.84	28.55	28.13
Mandrill	34.84	30.75	29.01	28.25	27.20	26.32	25.68	24.99	24.22
Pepper	39.66	36.26	34.19	32.98	30.33	29.85	29.54	29.10	28.54
Zelda	41.78	38.26	35.93	34.51	31.14	30.76	30.53	30.14	29.5

Table 4
Hiding capacity using our proposed scheme with the standard quantization table

Cover images	L								
	1	2	3	4	5	6	7	8	9
Boats	4096	8192	12,288	16,370	20,433	24,529	28,625	32,721	36,817
Jet	4096	8192	12,288	16,384	20,479	24,575	28,671	32,766	36,852
Lena	4096	8192	12,288	16,383	20,479	24,575	28,671	32,767	36,861
Mandrill	4096	8192	12,288	16,380	20,474	24,570	28,647	32,607	36,094
Pepper	4096	8192	12,288	16,384	20,480	24,576	28,672	32,768	36,842
Zelda	4096	8192	12,288	16,384	20,480	24,576	28,672	32,768	36,864

as "Mandrill" has the lowest hiding capacity and the worst image quality compared with others of less complexity. That is because our data hiding scheme utilizes the zeros in the high frequencies of the DCT-quantized blocks. When an image is more complex, the fewer zero coefficients are contained in its high frequency. In this case, it is more difficult to find the embeddable sets in each block, and modifying the coefficients to hide secret bits causes distortions in a stego-image. Furthermore, there is a higher probability for "Mandrill" to hide no bit in a block compared with other images. However, our experimental results demonstrate that such a case is quite rare. Even for "Mandrill," our proposed scheme still can hide 36,094 bits.

To compare the performance of our modified quantization table with that of the standard quantization table, we also generated the six cover images in Fig. 12 using our modified quantization table in Fig. 9. Fig. 13 presents the corresponding stego-images that hide the same secret information as in Fig. 11.

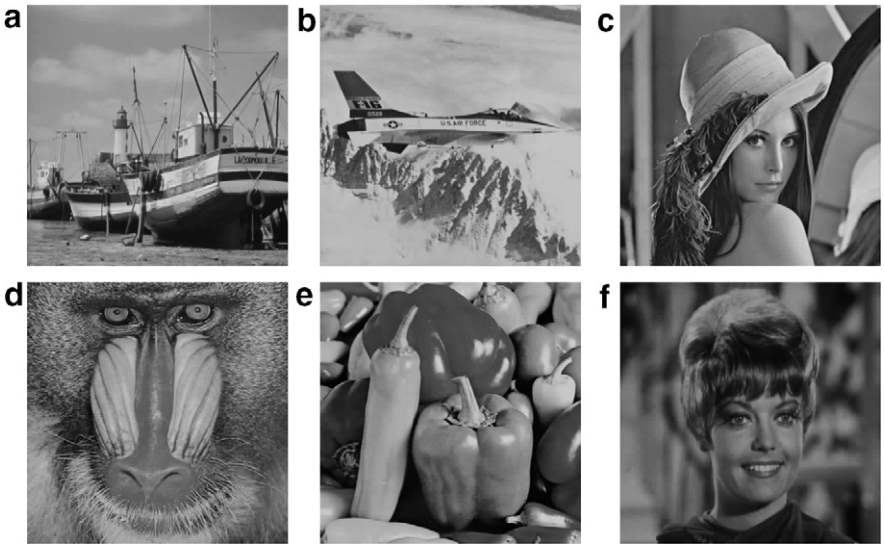

Fig. 12. Six original cover images with our modified quantization table: (a) Boats, (b) Jet, (c) Lena, (d) Mandrill, (e) Pepper and (f) Zelda.

Comparing the six stego-images in Fig. 12 with those in Fig. 13, it is obvious that the PSNRs of the latter images are higher than in the earlier images. Tables 5 and 6 confirm that the tiny difference between our modified quantization table and the standard quantization table does not significantly influence hiding capacity. On average, our scheme still can embed 36574 bits by hiding nine secret bits in each block when our modification table is adopted.

Furthermore, Fig. 14 shows that, on average, our modified quantization table can offer a PSNR value 2.2 times higher than the value that can be achieved with the standard quantization table.

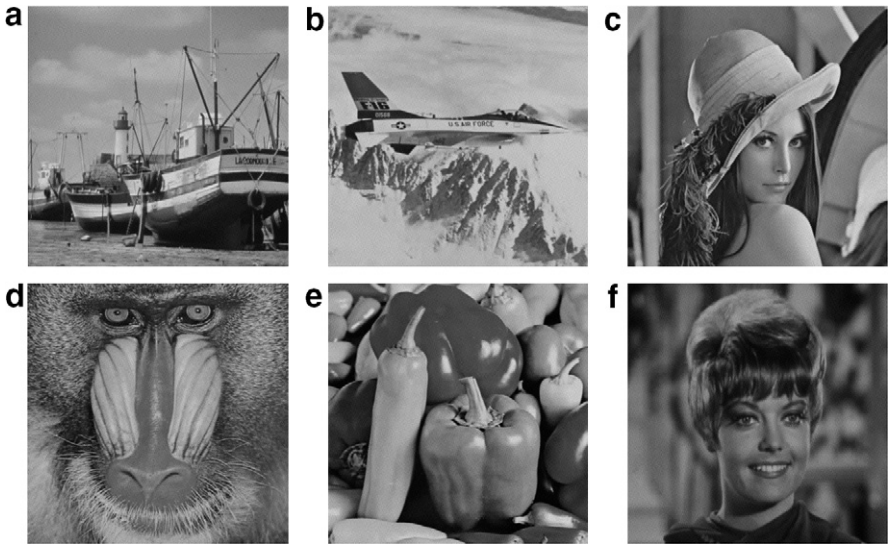

Fig. 13. Six stego-images when $L = 9$ and our modified quantization table are used, and their corresponding PSNRs: (a) Boats (29.75 dB), (b) Jet (29.98 dB), (c) Lena (30.34 dB), (d) Mandrill (26.46 dB), (e) Pepper (30.65 dB) and (f) Zelda (31.64 dB).

Table 5
PSNRs of stego-images when our modified quantization table is used

Cover images	L								
	1	2	3	4	5	6	7	8	9
Boats	40.55	36.86	34.92	33.41	31.76	31.17	30.81	30.24	29.75
Jet	40.26	36.95	35.22	33.94	32.09	31.49	31.03	30.56	29.98
Lena	40.49	37.15	35.15	33.86	32.13	31.56	31.20	30.71	30.34
Mandrill	35.95	32.65	31.34	30.54	29.59	28.75	28.00	27.22	26.46
Pepper	41.41	38.16	36.32	34.83	32.74	32.14	31.75	31.21	30.65
Zelda	43.43	40.36	38.04	36.17	33.55	33.00	32.63	32.15	31.64

Table 6
Hiding capacity using our proposed scheme with our modified quantization table

Cover images	L								
	1	2	3	4	5	6	7	8	9
Boats	4096	8192	12,288	16,343	20,339	24,435	28,531	32,626	36,710
Jet	4096	8192	12,288	16,383	20,472	24,568	28,664	32,759	36,817
Lena	4096	8192	12,288	16,381	20,473	24,569	28,665	32,761	36,850
Mandrill	4096	8192	12,288	16,367	20,437	24,526	28,503	32,249	35,402
Pepper	4096	8192	12,288	16,384	20,479	24,575	28,669	32,761	36,804
Zelda	4096	8192	12,288	16,384	20,479	24,575	28,671	32,767	36,861

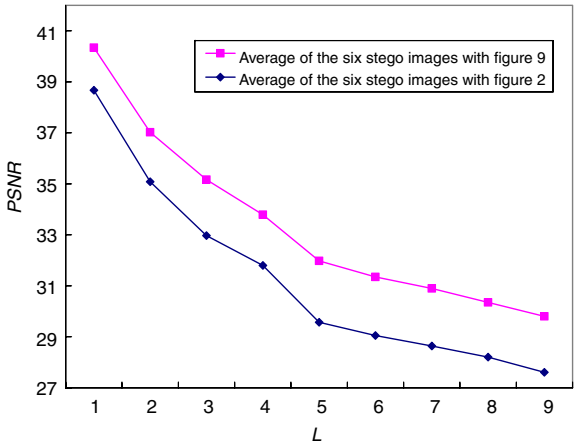

Fig. 14. Comparison of mean PSNRs of the six stego-images with standard and modified quantization tables.

In theory, the hiding capacity and image quality of stego-images can be influenced by a data hiding scheme focusing on reversibility. To see the side effects of reversibility on the hiding capacity and image quality of stego-images, we also compared our proposed scheme with the Iwata et al.'s scheme described in Section 2.2. For example, for "Lena," comparisons of image quality and hiding capacity of the stego-images are shown in Figs. 15 and 16, respectively.

Because the Iwata et al. scheme involves hiding data without distortion, it only modifies bits to hide secret data without providing the extra data required for restoring the original coefficients. In contrast, whether a secret bit is hidden or not, our proposed scheme must always modify an extra bit to avoid receiver misjudgments during the extracting and restoring procedures. Therefore, in Fig. 15 we can see that our reversibility and hiding capacity are achieved at the cost of image quality in the stego-images. The PSNR of the "Lena" stego-image using our proposed scheme with a standard quantization table is less than 5 dB of that of Iwata et al.'s scheme.

Although the image quality of our proposed scheme is significantly less than those of Iwata et al.'s scheme, the image quality of a stego-image using our proposed scheme with the standard quantization table still can

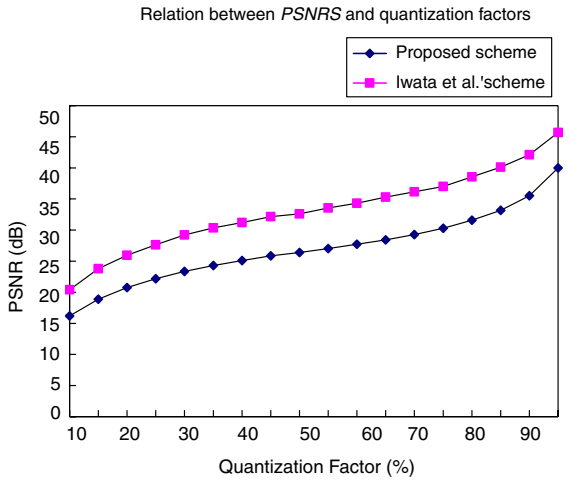

Fig. 15. Comparison between our proposed scheme and Iwata et al.'s scheme of image quality of stego-images with different quantization factors and $L = 9$.

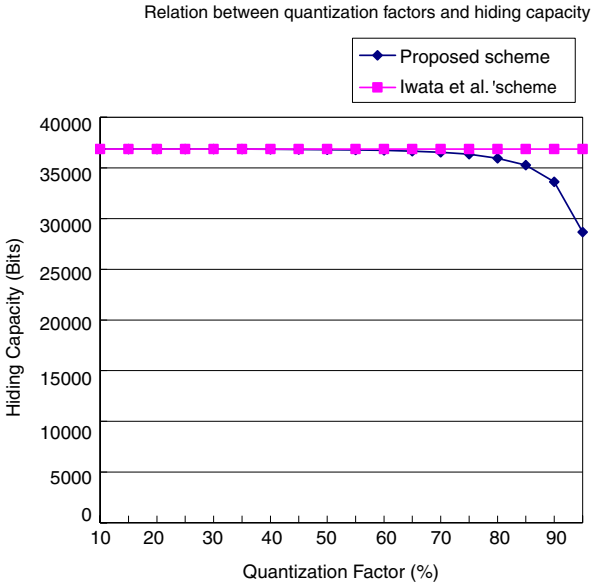

Fig. 16. Comparison between proposed scheme and Iwata et al.'s scheme of hiding capacities with different quantization factors and $L = 9$.

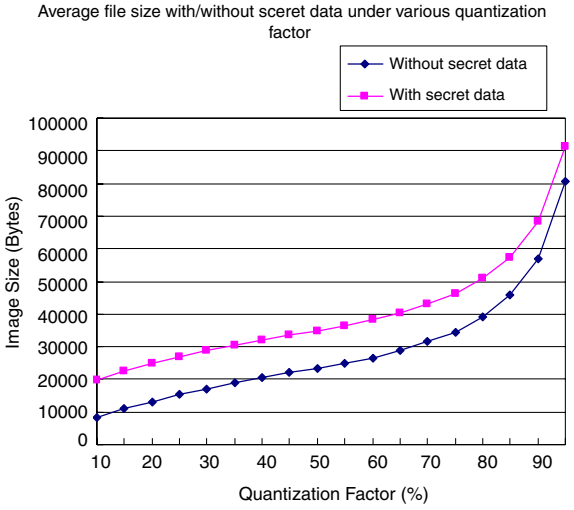

Fig. 17. Average file size of JPEG compressed stego-images with and without secret data.

maintain about 30 dB when the quantization factor is set at 70%. In addition, as Fig. 16 shows, our proposed scheme can offer the same hiding capacity as Iwata et al.'s scheme.

In our second experiment, we compressed six cover images with and without secret data to see whether the hidden data affect our compression performance. The comparisons are presented in Fig. 17. On average, the size of our JPEG compressed stego-image is larger than that of an image without secret data. The difference is about 10,000 bytes.

Because the average size of JPEG compressed stego-images in our proposed scheme is about two times that of an image without secret data, attackers may feel suspicious after they compare the sizes of JPEG compressed files transmitted over the Internet. Once they are suspicious of the JPEG compressed files

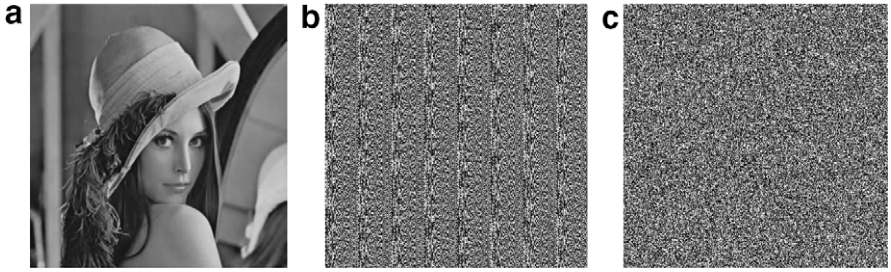

Fig. 18. Example "Lena" for visual attack using enhancing LSBs: (a) original "Lena"; (b) enhanced LSBs of "Lena" with secret bits in the least significant bits of each pixel; (c) enhanced LSBs of stego-image of "Lena".

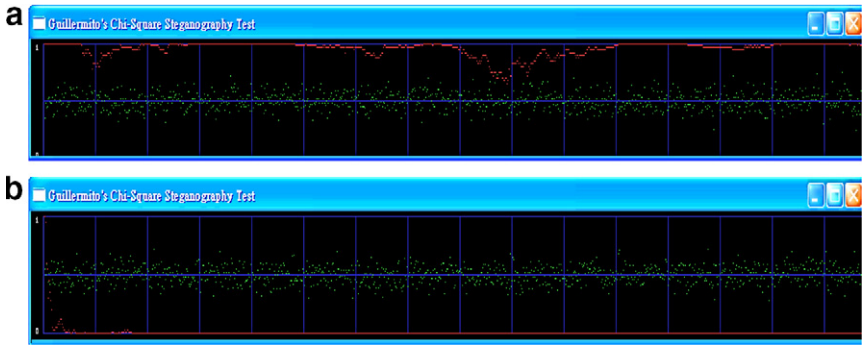

Fig. 19. Example of "Lena" for statistical attack using Chi-square analysis: (a) Chi-square result of "Lena" with secret data in the 1-LSB of each pixel; (b) Chi-square result of stego-image of "Lena".

generated by our proposed scheme, they may try to analyze them to extract secret data. However, our proposed scheme still is better able to resist visual and statistical attacks because we hide secret data in the frequency domain rather than in the spatial domain. To prove this assertion, in our third experiment we used two general attacks, statistical attack and visual attack, to analyze stego-images generated by our proposed scheme.

Basically, visual and statistical attacks enable attackers to discover whether an LSB-based stego-image contains secret data, primarily because the hidden data are embedded in the least significant bits of each pixel in a cover image. Whether the secret data are randomly generated or encrypted by a modern encryption technique, there is an almost 50% probability for each bit to be 0 or 1. Enhancing the least significant bits of an LSB-based stego-image reveals several regular patterns once the secret data are inside, as Fig. 18b 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 Fig. 18c.

To further prove that our proposed scheme can withstand a Chi-square attack, we used a Chi-square steganography test program provided by Guillermito [11] to perform steganography analyses. Fig. 19 shows the test results. In Fig. 19, 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[1] curve that presents the average value of the LSBs. In Fig. 19a, the green curve stays at about 0.5, which means a random message is embedded. Note that in Fig. 19b, 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.

[1] For interpretation of color in figures, the reader is referred to the Web version of this article.

Combining the two experimental results presented in Figs. 18 and 19, we can see that even when attackers use enhancing LSBs and Chi-square analyses, they cannot obtain a clue to hidden secret data. Certainly, our future work is planned to further reduce the size of our JPEG compressed stego-image to enhance the security of hidden data.

5. Conclusions

DCT is a widely used mechanism for frequency transformation. To extend the variety of cover images and for the sake of repeated usage, we offer a lossless data hiding scheme for DCT-based compressed images in this paper. Using a modified quantization table and our proposed embedding strategy, our proposed scheme can maintain the image quality of stego-images, with a PSNR value 2.2 times higher than that offered by a standard quantization table without affecting hiding capacity. Experimental results further demonstrate that our proposed scheme provides stego-images with acceptable image quality and similar hiding capacity to those can be achieved with the Iwata et al. scheme.

Although the size of our JPEG compressed stego-image is about two times the size of a regular image, our third experiment proves that our proposed scheme can withstand visual and statistical attacks. In other words, even though attackers may feel suspicious about our stego-images they cannot obtain further evidence to support their suspicion. Certainly, significantly reducing the size of our JPEG compressed stego-image is the next step in enhancing security. At the same time, we plan to increase the hiding capacity of our proposed scheme to extend its applications.

References

[1] M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber, Lossless generalized-LSB data embedding, IEEE Transactions on Image Processing 14 (2) (2005) 253–266.
[2] C.C. Chang, T.S. Chen, L.Z. Chung, A steganographic method based upon JPEG and quantization table modification, Information Sciences 141 (2002) 123–138.
[3] C.C. Chang, J.Y. Hsiao, C.S. Chan, Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy, Pattern Recognition 36 (7) (2003) 1595–1683.
[4] C.C. Chang, W.L. Tai, C.C. Lin, A reversible data hiding scheme based on side match vector quantization, IEEE Transactions on Circuits and Systems for Video Technology 16 (10) (2006) 1301–1308.
[5] Chin-Chen Chang, Chih-Yang Lin, Yu-Zheng Wang, New image steganographic methods using run-length approach, Information Sciences 176 (2006) 3393–3408.
[6] Chin-Chen Chang, Chih-Yang Lin, Reversible steganographic method using SMVQ approach based on declustering, Information Sciences, Available online 24 October, 2006.
[7] K.L. Chung, C.H. Shen, L.C. Chang, A novel SVD- and VQ-based image hiding scheme, Pattern Recognition Letters 22 (9) (2001) 1051–1058.
[8] B.V. Dasarathy, Image Data Compression: Block Truncation Coding, IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 164–173.
[9] W. Diffie, M.E. Hellman, Exhaustive cryptanalysis of the NBS data encryption standard, IEEE Computer 10 (1977) 74–84.
[10] J. Fridrich, M. Goljanb, R. Du, Invertible authentication watermark for JPEG images, IEEE International Conference on Information Technology: Coding and Computing, Las Vegas, Nevada, April 2–4, 2001, pp. 223–227.
[11] Guillermito, Chi-square Steganography Test Program. <http://www.guillermito2.net/stegano/tools/index.html>.
[12] M. Iwata, K. Miyake, A. Shiozaki, Digital steganography utilizing features of JPEG images, IEICE Transactions on Fundamentals E87-A (4) (2004) 929–936.
[13] H. Kobayashi, Y. Noguchi, H. Kiya, A method of embedding binary data into JPEG bitstreams, IEICE Transactions on Fundamentals J83-D2, 6 (2000) 1469–1476.
[14] Y.K. Lee, L.H. Chen, High capacity image steganographic model, in: Proceedings of the IEE International Conference on Vision, Image and Signal Processing, vol. 147 (3), 2000, pp. 288–294.
[15] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible Data Hiding, in: Proceedings of the 2003 International Symposium on Circuits and Systems (ISCAS'03), vol. 2, May 2003, pp. II-912–II-915.
[16] R.L. Rivest, A. Shamir, L. Adelman, A method for obtaining digital signatures and public-key cryptosystem, Communications of the ACM 21 (2) (1978) 120–126.
[17] J. Tian, Reversible data embedding using a difference expansion, IEEE Transactions on Circuits and Systems for Video Technology 13 (8) (2003) 890–896.

[18] P. Tsai, Y.C. Hu, C.C. Chang, An image hiding technique using block truncation coding, in: Proceedings of the Pacific Rim Workshop on Digital Steganography, Kitakyushu, Japan, July 2002, pp. 54–64.
[19] R.Z. Wang, C.F. Lin, J.C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern Recognition 34 (3) (2001) 671–683.
[20] G. Xuan, J. Zhu, J. Chen, Y.-Q. Shi, Z. Ni, W. Su, Distortionless data hiding based on integer wavelet transform, IEE Electronics Letters 38 (25) (2002) 1646–1648.