Steganography Using JPEG-Compressed Images

Hsien-Wen Tseng and Chin-Chen Chang
Department of Computer Science and Information Engineering
National Chung Cheng University
Chaiyi, Taiwan 621, R.O.C.
E-mail: {hwtseng,ccc}@cs.ccu.edu.tw

Abstract

In this paper, a novel steganographic method based on JPEG is proposed. *We take advantage of the quantization error resulting from processing the JPEG-compressed image with two different scaling factors. One of the scaling factors is used to control the bit rate of the stego-image while the other is used to guarantee the quality of the stego-image. Our experimental results show that the proposed steganographic method can provide a high information hiding capacity and successfully control the compression ratio and distortion of the stego-image.*

1. Introduction

Nowadays, daily communications of all kinds over the Internet have become incredibly popular and convenient. However, message transmissions over the Internet still have to face all kinds of security problems. To the best of our knowledge, all the most popular forms of security protection heavily rely on encryption, which refers to the process of encoding secret information in such a way that only the person with the right key can successfully decode it. However, encryption leaves obviously noticeable marks on the message, making it suspicious enough to attract eavesdroppers' attention. A practical way to solve this problem is to hide the secret information behind some kind of a cover such as a digital image or sound clip so that it draws no special attention. This technique of information security, called steganography [1], is applicable to many situations in which invisible communications take place.

So far, many methods for hiding data in digital images have been proposed: some embed secret data in the spatial domain [2,3], and others hide them in the frequency domain [4]-[6]. In this paper, we shall focus on steganography in the DCT domain of the JPEG-compressed image. JPEG [7] is the most commonly used image format on the Internet, and therefore JPEG-compressed images are suitable cover images for the hidden secret information. However, throughout the literatures concerned, little can be found about steganographic techniques having to do with JPEG-compressed images [8]-[13]. The reason is that JPEG usually results in the concentration of energy upon a small set of transformed coefficients only, which means a tiny change to these coefficients can lead to noticeable degradation in image quality. As a result, almost all the steganographic techniques are applicable to JPEG-compressed images are not so good at providing a high information hiding capacity. To make things worse, the distortion caused by embedding cannot be easily controlled. The image quality of the stego-image is never known until the stego-image is produced.

In this paper, we shall propose a novel steganographic technique to hide secret information behind the cover of a JPEG-compressed image. The proposed method will provide a flexible mechanism to balance the image quality, the information hiding capacity, and the bit rate, and the choice will be all up to the user. Moreover, the distortion of stego-image will be controlled and the compression ratio of the stego-image will be predictable.

The rest of this paper is organized as follows. In Section 2, some related works will be described. Then, in Section 3, we shall present our new method, followed by the experimental results in Section 4. Finally, the conclusions will appear in Section 5.

2. Relative Works

In 1999, Kobayashi et al. [8,9] proposed a method to hide data into JPEG-compressed images. Their method embeds only one secret bit into the 64th

Proceedings of the Fourth International Conference on Computer and Information Technology (CIT'04)
0-7695-2216-5/04 $20.00 © 2004 IEEE

quantized DCT coefficient of a DCT block in zigzag order. Besides, a different quantization table is offered to the JPEG decoder so as to reduce the noise caused by the secret data. Since a small change of the quantized DCT coefficient will lead to significant distortion in the decoded image, the value in the quantization table for the position the embedded data is in is changed to 1. Their method is able to produce a stego-image with small distortion. However, the information hiding capacity is very limited. A 512×512 gray level image can hold only 4096 bits.

Jpeg-Jsteg [13] is another famous secret data hiding tool for embedding secret information into JPEG-compressed images. Jpeg-Jsteg embeds one secret bit in the LSB of the quantized DCT coefficients whose values are not 0, 1, or –1. The information hiding capacity of Jpeg-Jsteg is improved but is still limited. The number of bits that can be embedded becomes smaller when the compression ratio gets higher.

In the literatures concerned, there are few data hiding methods [10]-[12] that hide secret data in JPEG-compressed images. The main drawbacks of these methods, however, are that the information hiding capacity is low and that neither the bit rate nor the distortion of the stego-image can be controlled.

3. The Proposed Method

In the encoding process of JPEG, an $N \times N$ image is divided into 8×8 blocks, and then a discrete cosine transform (DCT) is performed on each block. The transformed coefficients are quantized and coded by using a combination of run-length and Huffman coding. During the quantization step, each number in the DCT coefficient block is divided by the corresponding quantization value, and the result is rounded to the nearest integer. Consider an 8×8 block in the image. Let F be the corresponding DCT block with $F(i, j)$ denoting the $(i, j)^{\text{th}}$ entry of F, where $0 \le i \le 7$ and $0 \le j \le 7$. The DCT block F contains one DC coefficient at position (0,0) and 63 AC coefficients. After the DCT block F is calculated, it is quantized. The quantized DCT coefficients $Fq(i, j)$ are computed by

$$Fq(i, j) = \text{Round}\left(\frac{F(i, j)}{Q(i, j)}\right), \qquad (1)$$

where Q is a quantization table. In general, JPEG uses a default quantization table $Q\text{default}(i, j)$ and a quantizing factor q to control the compression ratio; i.e., $Q(i, j) = \text{Round}\left(\frac{Q_{\text{default}}(i, j) \times q}{50}\right)$. Owing to the large values in the middle and high frequency regions, most corresponding DCT coefficients will be truncated to zero after quantization. This is the step where the information loss occurs.

Let F' be the dequantized DCT block, and then the dequantized DCT coefficient $F'(i, j)$ is given by

$$F'(i, j) = Fq(i, j) \times Q(i, j). \qquad (2)$$

The difference between $F(i, j)$ and $F'(i, j)$ is called the quantization error. It is used to describe the difference introduced when a component of the DCT is recovered from its quantized value. Let QET be the Quantization Error Table, and then $QET(i, j)$ is defined as

$$QET(i, j) = F(i, j) - F'(i, j). \qquad (3)$$

For example, Fig. 1(a) is a DCT block of the "Lena" image. Fig. 1(b) is the reconstructed DCT block using default quantization table with scaling factor $q = 50$. Fig. 2 is the QET produced by Fig. 1(b) – Fig. 1(a).

667	-27	-15	3	-5	1	-1	1
31	6	-18	5	0	3	-2	0
-9	16	-3	1	-2	2	-2	1
-1	4	-1	2	-1	5	0	0
0	3	-3	2	1	1	1	1
-4	-1	-1	1	-2	0	-1	0
-1	-1	-2	-1	0	1	0	0
2	1	-2	3	0	0	0	1

(a) DCT block

672	-22	-20	0	0	0	0	0
36	12	-14	0	0	0	0	0
-14	13	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b) Dequantized DCT block

Fig. 1. Quantization and dequantization

-5	-5	5	3	-5	1	-1	1
-5	-6	-4	5	0	3	-2	0
5	3	-3	1	-2	2	-2	1
-1	4	-1	2	-1	5	0	0
0	3	-3	2	1	1	1	1
-4	-1	-1	1	-2	0	-1	0
-1	-1	-2	-1	0	1	0	0
2	1	-2	3	0	0	0	1

Fig. 2. The *QET* table

The quantization error represents the information loss during JPEG compression. From another viewpoint, it can also be said to represent the tolerant error range within which a DCT value can be modified. In this section, we will propose a new steganographic technique to hide secret information in a JPEG-compressed image while decreasing the quantization error. Our new method is composed of three parts: DCT coefficient selection, information embedding, and modification of quantization table.

3.1. DCT Coefficient Selection

The proposed method hides secret information in JPEG-compressed images according to the *QET*. However, not all DCT coefficients will be used. From our experiments, we observe that many DCT coefficients tend to be zero after the quantization step in JPEG compression. In order to reduce the error produced during quantization and not to enlarge the bit rate of the stego-image, the DCT coefficients that turn out to be zero after quantization are selected for embedding the secret information. First, an $N \times N$ image is divided into 8×8 blocks, and each block is DCT-transformed and quantized independently. For every DCT coefficient (i, j) in each block, the count of zero coefficients $Fq(i, j)$ is calculated. If the total exceeds a threshold Γ, the DCT coefficient (i, j) is selected for embedding the secret information. The DCT coefficient selection method provides an easy way to control the information hiding capacity. The more zero DCT coefficients after quantization (high compression ratio) are, the more embedded secret information will be.

3.2. Information Embedding

After the DCT coefficient selection step, the secret information can be embedded into the selected DCT coefficients. The selected DCT coefficients are dequantized and the $F'(i, j)$ is obtained. We then embed secret data d into $F'(i, j)$ and obtain a new value $F''(i, j)$. The embedded DCT coefficient $F''(i, j)$ is computed as follows:

$$F''(i,j) = \begin{cases} F'(i,j) + d, & \text{if } QET(i,j) > 0 \\ F'(i,j) - d, & \text{if } QET(i,j) < 0 \end{cases}, \qquad (4)$$

where $0 \le d \le |QET(i, j)|$. The distortion caused by embedding is no more than that caused by JPEG compression because $F''(i, j)$ is closer to $F(i, j)$ than $F'(i, j)$. This guarantees that the distortion is under control. Besides, the value of d is limited within the range from 0 to $|QET(i, j)|$, and therefore the number of bits to be embedded into the selected DCT coefficient is:

$$E(i,j) = \text{Log2}(|QET(i,j)| + 1). \qquad (5)$$

Finally, the embedded DCT block is coded using a combination of run-length and Huffman coding.

For example, Fig. 1(a) is the 1000th DCT block in the "Lena" image by top-down and left-right scanning. The *QET* for this block is shown in Fig. 2. According to the coefficient selection method in Section 3.1, the selected coefficients are AC9 ~ AC63 in zigzag order. Suppose that the following data is the secret information to be embedded:

0|01|00|11|10|0|1|1|00|10|0|1|00|0|1|00|1|1|00|1|1…

…

(without the vertical bars). The vertical bars are used to demarcate the number of bits embedded in each selected DCT coefficient. To give an example, AC9 (at position (3,0)) is the selected DCT coefficient, and the corresponding value in the *QET* (Fig. 2) is –1. So the number of bits to be embedded into this coefficient is obtained by Eqn. (5) as follows:

$E(3,0) = \text{Log2}(|QET(3,0)| + 1) = \text{Log2}(|-1| + 1) = 1.$

Then we pick out the first bit (0) from the secret information and embed it into AC9 in accordance with Eqn. (4):

$F''(3,0) = F'(3,0) - d = 0 - 0 = 0.$

AC10 has no secret bit embedded in because the corresponding value in the *QET* is zero. AC11 (at position (3,1)) can hold 2 bits because the corresponding value in the *QET* is 4. Thus, the following two bits (01) are extracted from the secret information and embedded into AC11, and so the value of AC11 is changed to 1. The quantized DCT block after embedding is shown in Fig. 3.

42	-2	-2	0	-2	0	-1	1
3	1	-1	3	0	0	0	0
-1	1	0	1	-1	0	-1	1
0	1	-1	0	0	0	0	0
0	0	0	1	1	1	0	1
-2	-1	0	1	0	0	0	0
0	0	0	-1	0	0	0	0
0	1	0	3	0	0	0	1

Fig. 3. Quantized DCT block after embedding

Take Fig. 2 as an example. Using modified quantization table as shown in Fig. 4, then Fig. 5 is the dequantized DCT block. The quantization error table for this embedded block is calculated as shown in Fig. 6. Compared with Fig. 2, the quantization error is reduced; i.e., the reconstructed image quality is

improved after embedding, and the distortion caused by embedding is under control.

16	11	10	16	1	1	1	1
12	12	14	1	1	1	1	1
14	13	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Fig. 4. Modified quantization table

672	-22	-20	0	-2	0	-1	1
36	12	-14	3	0	0	0	0
-14	13	0	1	-1	0	-1	1
0	1	-1	0	0	0	0	0
0	0	0	1	1	1	0	1
-2	-1	0	1	0	0	0	0
0	0	0	-1	0	0	0	0
0	1	0	3	0	0	0	1

Fig. 5. Dequantized DCT block

-5	-5	5	3	-3	1	0	0
-5	-6	-4	2	0	3	-2	0
5	3	-3	0	-1	2	-1	0
-1	3	0	2	-1	5	0	0
0	3	-2	1	0	0	1	0
-2	0	-1	0	-2	0	-1	0
-1	-1	-2	0	0	1	0	0
2	0	-2	0	0	0	0	0

Fig. 6. The *QET* table of embedded DCT block

3.3. Modification of Quantization Table

In a JPEG-compressed image, a slight change of the quantized coefficients can cause serious degradation in reconstructed image quality. The reason is in the quantization table. According to Eqn. (2), the dequantized DCT coefficient $F'(i, j)$ is recovered from its quantized value and the quantization table. In our proposed method, the corresponding entries in the quantization table of the selected DCT coefficients are changed to 1 for the purpose of avoiding significant distortion in the reconstructed image. Fig. 4 is an example of our modified quantization table, where AC9 ~ AC63 are our selected coefficients. This explains why we cannot use all DCT coefficients to embed the secret information. Because all the entries in the quantization table are changed to 1, the quantization step in JPEG is meaningless.

3.4. Generalized Hiding Method

In the earlier subsections, we have presented a hiding method that provides controlled distortion in the stego-image. However, according to our experimental results, the bit rate of the stego-image seems to be a little too high. Now we modify the previous method so that it can gain control over both the bit rate and image quality of the stego-image. A JPEG compressor with a low scaling factor $q1$ is applied first to a digital image in order to produce a high-bound image. Then, we apply JPEG with a high scaling factor $q2$ to the same digital image in order to produce a low-bound image. The *QET* is obtained from the difference between the high-bound image and the low-bound image. Five new equations are defined as follows:

$$F_{q1}(i, j) = \text{Round}\left(\frac{F(i, j)}{Q1(i, j)}\right). \quad (6)$$

$$F'_1(i, j) = F_{q1}(i, j) \times Q1(i, j). \quad (7)$$

$$F_{q2}(i, j) = \text{Round}\left(\frac{F(i, j)}{Q2(i, j)}\right). \quad (8)$$

$$F'_2(i, j) = F_{q2}(i, j) \times Q2(i, j). \quad (9)$$

$$QET(i, j) = F'_1(i, j) - F'_2(i, j). \quad (10)$$

$Q1$ and $Q2$ use the same default quantization table but different quantizing factors $q1$ and $q2$, respectively ($q1 < q2$). Note that $F_{q1}(i, j) = F(i, j)$ when $Q1(i, j) = 1$. Then, $F'_1(i, j) = F(i, j)$. It conforms to the method we have just presented.

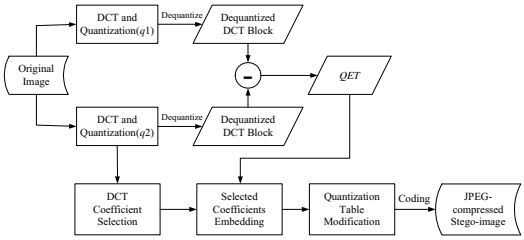

Fig. 7. Block diagram of proposed method

The quantizing factor in JPEG is used to control the bit rate or the image quality. Here, the quantizing factor $q1$ is used to control the bit rate, and $q2$ is used to control the image quality. Besides, $q1$ is always smaller than $q2$, and the difference between them is

used to control the hiding capacity. Fig. 7 shows the steps to take for embedding the secret information into a JPEG-compressed image.

3.5. Extracting Method

The extraction of secret information requires the original image, the stego-image, and the quantizing factors $q1$ and $q2$. The extraction steps are described as follows.

A. DCT Coefficient Identification

The stego-image is a JPEG-compressed image. Thus, we can identify the selected DCT coefficients from the quantization table. The components whose value is 1 are the coefficients where we are to hide the secret information.

B. Generation of *QET*

Apply lossy JPEG compression to the original image with the quantizing factors $q1$ and $q2$. Then, dequantize the DCT blocks and calculate the difference between them to produce the *QET*.

C. Information Extraction

To extract the secret information from the selected DCT coefficient (i, j), Eqn. (4) is used to obtain the embedded digit d. Then, we convert the digit d into a binary string whose length is $E(i, j)$. Then we collect all the resulting bit strings to form the original embedded secret information.

4. Experimental Results

The standard 512×512 gray scale image "Lena" was used as the test image in our experiments. We employed the peak signal-to-noise ratio (PSNR) as a measure of the stego-image quality. It is defined as follows:

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{\text{MSE}} \text{dB}, \qquad (11)$$

where MSE is the mean square error between the original image and the reconstructed image. A larger PSNR value means that the stego-image preserves the original image quality better.

Table 1. Experimental results with $q1=0$

q	JPEG		Jpeg-Jsteg			Proposed method		
	bpp	PSNR	bpp	PSNR	HC	bpp	PSNR	HC
5	2.68	45.86	2.67	44.86	67000	3.84	47.92	127414
10	1.77	43.13	1.77	42.12	43655	3.31	45.58	172622
25	0.98	39.90	0.98	38.62	22023	3.09	42.54	238968
50	0.63	37.51	0.62	35.86	14183	3.03	40.24	278879
75	0.48	36.10	0.48	34.08	11074	3.06	38.88	301849
100	0.41	35.02	0.41	32.53	9287	3.08	37.78	316259

Two experiments were performed. In the first experiment, the scaling factor $q1$ was set to be zero. We did not actually apply any zero to the quantization table but modified all the components in the quantization table to one. That is to say, the DCT coefficients in the high-bound image were not quantized. The results are shown in Table 1, where the HC denotes the information hiding capacity in bits. In the second experiment, the scaling factor $q1$ was used as it is in the standard JPEG compressor. The results are shown in Table 2. For each experiment, the secret information to embed was generated by a pseudo random number generator with a fixed seed.

In Table 1, the PSNR of the stego-image and the resulting JPEG-compressed image are shown. Here, the DCT coefficients in the high-bound image are free from quantization, and the DCT coefficients in the low-bound image are quantized with the scaling factor q. According to the results in Table 1, the PSNR of the proposed hiding method is always larger than that of JPEG. This means the distortion caused by embedding information in the stego-image does not exceed that caused by JPEG. Furthermore, the information hiding capacity of the proposed method will become larger when the scaling factor gets bigger. But the bit rate (bpp) of the stego-image is always higher than 3.

Table 2. Experimental results with $q1>0$

$q1$	$q2$	bpp	PSNR	HC
5	25	2.54	42.24	151856
5	50	2.39	40.09	183916
5	75	2.45	38.81	209024
10	25	1.80	41.41	71541
10	50	1.91	39.58	128977
10	75	1.81	38.27	138204
25	50	1.01	38.48	28142
25	75	1.05	37.22	45732
25	100	1.04	36.31	54543
50	100	0.66	35.77	19004

Table 2 shows that the proposed hiding method can control the bit rate and the image quality at the same time. The scaling factors $q1$ and $q2$ are used to control the bit rate and the stego-image quality, respectively. For example, the bit per pixel (bpp) value is around 1.8 when $q1 = 10$, which is about the same as JPEG with $q = 10$ (as shown in Table 1). For $q1 = 25$, the bpp value is around 1, and the effect is about the same as JPEG with $q = 25$ (as shown in Table 1). Besides, the hiding capacity is controlled by the difference between $q1$ and $q2$. More secret information can be embedded when $q1$ is small and the difference between $q1$ and $q2$ is large.

In general, the proposed method tends to embed fewer bits at a large $q1$ because most DCT coefficients are quantized to zero.

As to the threshold Γ employed in the DCT coefficient selection step, the value is up to the user. A small Γ value leads to the result that more DCT coefficients can be chosen from, so the hiding capacity is increased. However, the cost is that the bit rate is relatively raised.

5. Conclusions

In this paper, we have proposed a novel steganographic method to embed secret information into a JPEG-compressed image. In our method, a JPEG compressor with a low scaling factor is applied first to a digital image to produce the high-bound image. Then, we apply JPEG with a high scaling factor to the same digital image to produce the low-bound image. The quantization error is calculated between these two images. We embed secret information into the selected DCT coefficients on the basis of quantization error. This guarantees that the error caused by embedding does not exceed the low-bound image and that the bit rate of the stego-image is about the same as that of the high-bound image. Our experimental results have shown that the proposed method can successfully gain control over the bit rate and the distortion.

References

[1] S. Katzenbeisser, F.A.P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, Inc., Boston, 2000.

[2] H. M. Chao, C. M. Hsu, and S. G. Miaou, "A Data-Hiding Technique with Authentication, Integration, and Confidentiality for Electronic Patient Records," *IEEE Trans. Information Technology in Biomedicine*, vol. 6, no. 1, 2002, pp. 46-53.

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

[4] J. Cox, J. Kilian, F. T. Leighton, T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. Image Processing*, vol. 6, no. 12, 1997, pp. 1673-1687.

[5] R. B. Wolfgang, C. I. Podilchuk, E. J. Delp, "Perceptual Watermarks for Digital Images and Video," *Proceeding of the IEEE*, vol. 87, no. 7, 1999, pp. 1108-1126.

[6] X. G. Xia, C. G. Bongelet, G. R. Arce, "A Multiresolution Watermark for Digital Images," *Proceedings of IEEE International Conference on Image Processing*, Santa Barbara, CA, USA, Oct. 1997, vol. 3, pp. 548-551.

[7] W. Pennebaker, J. Mitchell, *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, 1993.

[8] H. Kobayashi, Y. Noguchi, H. Kiya, "A Method of Embedding Binary Data into JPEG Bitstreams," *IEICE Trans. Information and Systems*, vol. J83-D, no. 2, 1999, pp. 1469-1476.

[9] Y. Noguchi, H. Kobayashi, H. Kiya, "A Method of Extracting Embedded Binary Data from JPEG Bitstreams Using Standard JPEG Decoder," *Proceedings of IEEE International Conference on Image Processing*, Vancouver, BC, Canada, Sept. 2000, Vol. 1, pp. 577-580.

[10] C. C. Chang, T. S. Chen, L. Z. Chung, "A Steganographic Method Based upon JPEG and Quantization Table Modification," *Information Sciences*, vol. 141, no. 1, 2002, pp. 123-138.

[11] P. H. W. Wong, O. C. Au and J. W. C. Wong, "Data Hiding and Watermarking in JPEG Compressed Domain by DC Coefficient Modification," *Proc. of SPIE Conference on Security and Watermarking of Multimedia Contents* II, San Jose, CA, USA, Jan. 2000, Vol. 3971, pp. 237-244.

[12] P. H. W. Wong, O. C. Au, J. W. C. Wong, "A Data Hiding Technique in JPEG Compressed Domain," *Proc. of SPIE Conference on Security and Watermarking of Multimedia Contents* III, San Jose, CA, USA, Jan. 2001, Vol. 4314, pp. 309-320.

[13] N. Johnson, S. Jajodia, "Steganalysis of Images Created Using Current Steganography Software," *Proceedings of Information Hiding Workshop*, Portland, Oregon, USA, Apr. 1998, LNCS 1525, pp. 273-289.