INFORMATICA, 2004, Vol. 15, No. 1, 127–142

High Capacity Data Hiding in JPEG-Compressed Images

Hsien-Wen TSENG, 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

Received: March 2003

Abstract. The JPEG image is the most popular file format in relation to digital images. However, up to the present time, there seems to have been very few data hiding techniques taking the JPEG image into account. In this paper, we shall propose a novel high capacity data hiding method based on JPEG. The proposed method employs a capacity table to estimate the number of bits that can be hidden in each DCT component so that significant distortions in the stego-image can be avoided. The capacity table is derived from the JPEG default quantization table and the Human Visual System (HVS). Then, the adaptive least-significant bit (LSB) substitution technique is employed to process each quantized DCT coefficient. The proposed data hiding method enables us to control the level of embedding capacity by using a capacity factor. According to our experimental results, our new scheme can achieve an impressively high embedding capacity of around 20% of the compressed image size with little noticeable degradation of image quality.

Key words: JPEG, data hiding, steganography, HVS, Jpeg–Jsteg, LSB substitution.

1. Introduction

Image, audio, video, and many other kinds of data are nowadays mostly passed from person to person or from place to place in a digital form. It is often desirable to embed data into the digital contents for copyright control and authentication, or for secret data hiding. Data-embedding techniques designed to take care of such tasks are commonly classified as watermarking or data hiding techniques in accordance with their functionalities. Watermarking techniques are often further divided into two groups: robust watermarking methods and fragile watermarking methods. In robust watermarking methods, the hidden information remains robust against manipulations from any possible sources including hostile ones. Hence such methods are usually developed to protect copyright. On the other hand, fragile watermarking methods are usually designed to easily get broken so that common content processing operations, if there are any at all, can be found. Therefore, such methods are good for tampering detection and authentication. As for those classified as data hiding techniques, they are sometimes called steganographical methods, where the secret message blends in a common digital content, so that eavesdroppers will not have any idea that the secret message is there, and so they will not have the slightest

intention of trying to break the protection. Under such circumstances, robustness seems to be less stringent, and the major issues here are the embedding capacity and invisibility. In other words, a good data hiding method should be one that can embed as much data as possible, and the perceptual distortion of the digital content after the embedding procedure should be as little as possible.

Current methods for the embedding of data into the cover image fall into two categories: spatial-based schemes (Adelson, 1990; van Schyndel *et al.*, 1994; Wang *et al.*, 2001) and transform-based schemes (Cox *et al.*, 1997; Wolfgang *et al.*, 1999; Xia *et al.*, 1997).

Spatial-based schemes embed the data into the pixels of the cover image directly, while transform-based schemes embed the data into the cover image by modifying the coefficients in a transform domain, such as the Discrete-Cosine Transform (DCT). In this paper, we will focus upon data hiding in the DCT domain as well as quantized DCT coefficients. We shall embed the data into a JPEG (Pennebaker and Mitchell, 1993) compressed image, for most digital images are stored and transmitted in the JPEG compressed format. Surprisingly, in the literature, data hiding techniques that deal with the JPEG compressed image (Kobayashi *et al.*, 1999; Noguchi *et al.*, 2000; Johnson and Jajodia, 1998; Chang *et al.*, 2002) are astonishingly few and far between. Kobayashi *et al.* (1999) presented a method to hide data into JPEG bitstreams. However, the embedding capacity is very limited. Jpeg–Jsteg (Johnson and Jajodia, 1998) is another famous hiding tool for embedding data into the JPEG compressed image. The secret data is embedded into the LSB of the quantized DCT coefficients. In the scheme proposed by Chang *et al.* (2002), the secret data is embedded in the middle-frequency part of the quantized DCT coefficients. The scheme provides a larger embedding capacity than Jpeg–Jsteg, but the compression ratio of image is bounded.

In general, for the purpose of avoiding too much distortion to the embedded image, the quantized DCT coefficients should be modified as little as possible. Furthermore, the AC coefficients become zeros mostly after quantization. These zeros are usually incapable of embedding and the embedding capacity of the JPEG compressed image is thus limited. To improve the embedding capacity, a high capacity hiding method based on the adaptive least-significant bit (LSB) substitution method and the human visual system (HVS) (Daly, 1994; Wandell, 1995) will be proposed. The number of bits embedded in the DCT components is computed through a predefined equation which is built up in accordance with the features of HVS to ensure the embedded image still preserves good image quality. Indeed, a high percentage modification in DCT components will certainly lead to significant distortion in the embedded image. Therefore, how to find the optimal balance between both ends (namely image degradation and embedding capability) is the topic of study in this paper.

To keep consistency in this paper, we shall define some terms for later use. The data to be embedded is called the secret data, for it has usually been processed by such encryption methods as DES (DES Encryption Standard, 1977) under some security requirement. The image responsible for carrying the hidden secret data is called the cover image, and the image containing the hidden secret data is called the stego-image.

The rest of this paper is organized as follows. Both concepts of data hiding, by Kobayashi *et al.*, by Jpeg–Jsteg, and by Chang *et al.*, will be briefly introduced in Section 2. Then, in Section 3, our method based on adaptive LSB and HVS will be presented. Finally, the experimental results will be given in Section 4, followed by the conclusions in Section 5.

2. Relative Works

Fig. 1 shows the block diagram of the data hiding JPEG encoder. It is a generic model of data hiding based on JPEG. The cover image is broken down into a set of 8×8 blocks, and then the discrete cosine transform (DCT) is performed on each block. The transformed coefficients are quantized in accordance with the default quantization table of JPEG (Fig. 2). The secret data is then embedded into the quantized coefficients and coded by using a combination of the run-length coding and Huffman coding. Fig. 3 shows the block diagram of how the secret data gets hidden into the JPEG compressed image directly. First, the entropy decoding method is used to process the JPEG image, and then the secret data can be hidden into the quantized DCT coefficients. Finally, to produce the JPEG compressed stego-image, the entropy encoding method is employed.

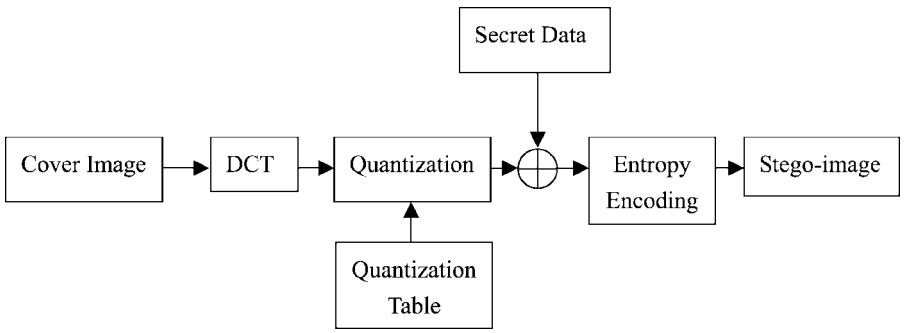


Fig. 1. Block diagram of data hiding JPEG encoder.

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. Default quantization table of JPEG.

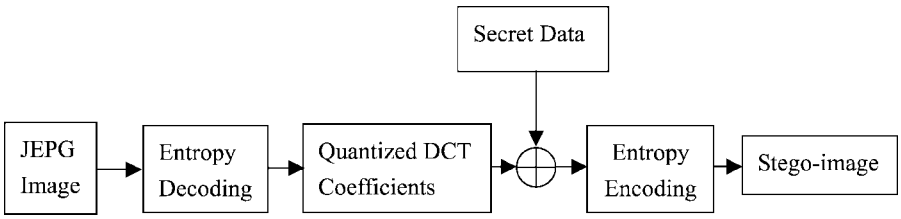


Fig. 3. Block diagram of JPEG image embedding.

2.1. *Kobayashi et al.'s Method*

Kobayashi *et al.* embed only one secret bit into one 8×8 DCT block. The embedded one-bit binary data is replaced with the kth quantized DCT coefficient through zigzag scanning. It is denoted by QDCT(k), where $0 \leqslant k < 64$, and the quantized DCT coefficient in the same position is replaced. Kobayashi *et al.* believe that the high-frequency components are better places to hide the secret data in than low-frequency components (see Fig. 4). The first reason is that the high-frequency components often become zeros after quantization, and there is no need to change the values of the coefficients if the data to be embedded is zero. And the second reason is that high-frequency components are more visually resistant to noises than low-frequency components. Therefore, by following their method, we can reduce the quality degradation of the stego-image.

Furthermore, Kobayashi *et al.* prepare a different quantization table for the JPEG decoder so as to reduce the noise caused by the secret data. As shown in Fig. 2, the values of the high-frequency area are so big. A small change done to the DCT coefficients of this area will lead to significant distortion in the decoded image. Therefore, the value in the quantization table for the position the embedded data is in is changed to 1. Fig. 5 presents the modified quantization table of QDCT(63). Finally, the modified quantization table is sent to the decoder in the standard JPEG bitstream header.

According to the simulation results provided by Kobayashi *et al.*, the distortion of the stego-image is very small. Besides, the secret data can be extracted from the JPEG

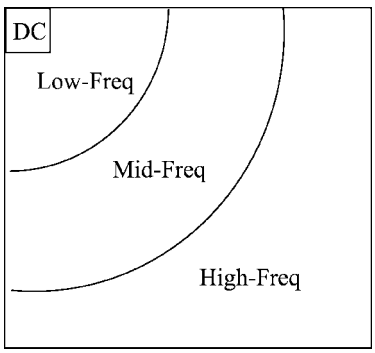


Fig. 4. Frequency distribution in a DCT block.

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

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	1

Fig. 5. Modified quantization table.

bitstreams by using the standard JPEG decoder. However, the embedding capacity is very limited; a 512×512, 8-bit gray level image can hold only 4096 bits. Besides, due to the JPEG encoding method, embedding the secret data in the high-frequency components also increases the size of the JPEG compressed image, which will reveal in our experimental results later.

2.2. *Jpeg–Jsteg*

Jpeg–Jsteg is a famous hiding tool based on JPEG. In Jpeg–Jsteg, the secret data is embedded into the LSB of the quantized DCT coefficients whose values are not 0, 1, or -1. The constraints on the value of coefficients are meant to avoid the otherwise possible ambiguity in the secret data extracting process. For example, if the secret data is 0 and the quantized DCT coefficient is 1, then the quantized DCT coefficient is changed to 0 after embedding. In the meantime, other coefficients with the original value 0 don't have any secret data embedded in them. This results in an ambiguous condition when the secret bits are extracted from these coefficients with the value 0.

Jpeg–Jsteg embeds one secret bit in the LSB of the quantized DCT coefficients whose absolute values are greater than 1. As shown in Fig. 6, the available quantized DCT

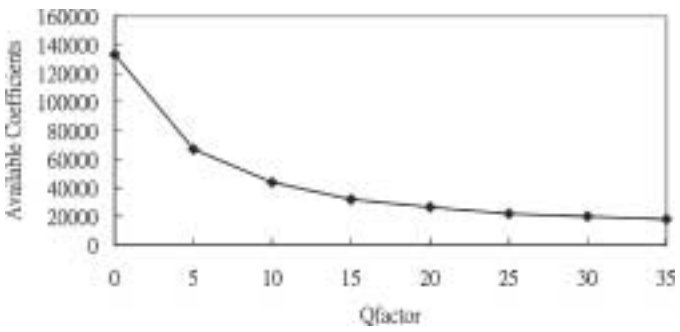


Fig. 6. Available DCT coefficients for embedding.

Table 1
The embedding capacity of Jpeg–Jsteg

	Embedded data (bits)	Compressed file (bytes)	Embedding Capacity	PSNR (dB)
Jpeg–Jsteg (1)	43655	57860	9.4%	42.11
Jpeg–Jsteg (2)	51094	57812	11%	41.14
Jpeg–Jsteg (3)	47415	57968	10.2%	39.44

coefficients for embedding in the 512×512 JPEG compressed Lena image are smaller and smaller in number when the compression ratio gets higher and higher, where Q factor is the so-called quality factor that controls the ratio of compression. Thus high capacity data hiding in a JPEG compressed image is a hard work. If we want to increase the embedding capacity of the JPEG compressed image, only extending the embedding bits is obviously not enough. Table 1 shows the result of hiding data into a 512×512 Lena image compressed by JPEG at Q factor 10. In the table, Jpeg–Jsteg (1) refers to the original Jpeg–Jsteg, Jpeg–Jsteg (2) embeds two secret bits in the LSBs of the quantized DCT coefficients, and Jpeg–Jsteg (3) embeds three secret bits in the LSBs of the quantized DCT coefficients. As the table shows, the embedding capacity of Jpeg–Jsteg (2) is only a bit higher than that of Jpeg–Jsteg (1). However, it is quite surprising to find that the embedding capacity of Jpeg–Jsteg (3) is worse than that of Jpeg–Jsteg (2). As a result, we reckon that we could use another good embedding method to increase the embedding capacity of the JPEG compressed image.

2.3. *Chang et al.'s Method*

Chang *et al.* embed the secret data into the middle-frequency part of the quantized DCT coefficients; meanwhile, the corresponding components in the quantization table of JPEG are changed to 1. Two secret bits are embedded in the least two-signification bit of the quantized DCT coefficients that are located in the middle-frequency part. There are 26 coefficients in each block are selected for embedding, thus a cover image of 512×512 pixels can embed $26 \times 2 \times (512 \times 512)/(8 \times 8) = 212992$ secret bits into it. The embedding capacity of Chang *et al.*'s method is larger than that of Jpeg–Jsteg. However, the compression ratio is quite restricted. It cannot be adjusted freely based on the choices of Q factor.

3. The Proposed Method

According to the descriptions and discussions in the previous section, an adaptive LSBs substitution data hiding method should be developed, and that is exactly what we have done. In our new method, we do not embed the secret data in the high-frequency components in order not to expand the size of the stego-image. Besides, the LSB number in each DCT coefficient used for data hiding depends on the characteristics of the image

according to HVS; as a result, the embedding capacity of the JPEG compressed image can be raised while avoiding significant stego-image distortion. Meanwhile, the capacity formulas for the DC and AC components should be different due to the discrepancy between them. This will be discussed further in detail in Subsection 3.2.

3.1. *Capacity Estimation*

To decide the maximum number of bits that the least significant bits of each DCT coefficient can handle while avoiding perceptual distortion in the stego-image, a capacity table and a capacity factor α are used. The capacity table is an 8×8 table derived from the JPEG default quantization table (Fig. 2). To achieve our goal, each component in the JPEG default quantization table is chosen as a perceptual threshold, or "just noticeable difference", for the visual contribution of its corresponding cosine-basis function. The table is tuned for most natural images according to the results of perceptual experiments. Since the design of the JPEG default quantization table is based on a simple HVS model, it can be easily applied to embedding capacity estimation. As we know, the image information (energy) is usually concentrated in the low frequency region after DCT transformation. Therefore, major modifications in the low frequency region will lead to significant degradation of image quality, while the high frequency region can allow more changes. The capacity table goes along with the JPEG default quantization table; low frequency components are less capable of holding secret bits and so score lower in the table as to the magnitude, while high frequency components are more distortion-tolerant and therefore deserve higher magnitude scores in the table, meaning that they have greater embedding capacity. Besides, we also employ the capacity factor α to control the level of embedding capacity. As the value of α becomes larger, more secret data can be embedded. Let Q be the JPEG default quantization table with $Q(i,j)$ denoting the (i,j)th entry of Q, and let $C(i,j)$ be the (i,j)th entry of the capacity table C. Then $C(i,j)$ can be derived by the following expression:

$$C(i,j) = \operatorname{Log}_2\big(\alpha * Q(i,j)\big), \quad \text{where } 0 \leqslant i,j < 8. \tag{1}$$

Furthermore, the embedded bits should be limited by the coefficient magnitude in order to avoid ambiguity later when they are extracted. Let F be the quantized DCT block with $F(i,j)$ denoting the (i,j)th entry of F. Here, the maximum number of bits that can be embedded in the least significant bits of each quantized DCT coefficient is defined as

$$M(i,j) = \operatorname{Log}_2\big(|F(i,j)|\big), \quad \text{where } 0 \leqslant i,j < 8. \tag{2}$$

In summary, let E be the capacity table with $E(i,j)$ denoting the number of bits that can be embedded in the LSBs of $F(i,j)$. Then, $E(i,j)$ is given by

$$E(i,j) = \min\big\{C(i,j), M(i,j)\big\}. \tag{3}$$

3.2. *The DC Components*

As we know, the magnitude of the DC component is positively proportional to the average pixel value in the original block, and the magnitude of the DC component is much larger than that of any AC component. Embedding data can be viewed as embedding a set of signals onto a larger set of background signals. The embedded signals can only be detected by HVS when they surpass the detection threshold of HVS. To be more precise, the Weber–Fechner law states that the detection threshold of visibility for an embedded signal is proportional to the magnitude of the background signal. Thus, compared with AC components, DC components can be modified by a larger quantity. However, even so, DC components still cannot be changed by a large percentage because significant block artifacts are still very likely to happen in that case. Fig. 7 compares the quality of the stego-image processed by embedding secret bits in the DC components and in the AC components (the first bit, first 2 bits, and first 3 bits), where the cover image is a 512×512, 8-bit gray level image Lena. According to the result, embedding secret bits in the DC components does not necessarily cause significant degradation when the compression ratio (Q factor) is low, but it leads to noticeable distortion when the compression ratio is high, which differs from what happens to the AC components. Therefore, we should define another capacity formula for DC components. The following formula is derived experimentally.

$$C(0,0) = \mathrm{Log}_2\Big(\frac{\alpha * Q(0,0) * 2}{\mathrm{Log}_2(Q\ \mathrm{factor})}\Big). \tag{4}$$

3.3. *Block Classification*

The definition of HVS sensitivity depends on the anatomy of the eye, its limitations and imperfections. According to the texture masking phenomena of HVS, the stronger the texture of the background, the lower the visibility of the embedded signal. That is to say, high-activity regions tolerate higher distortions than flat ones. Therefore, we classify the

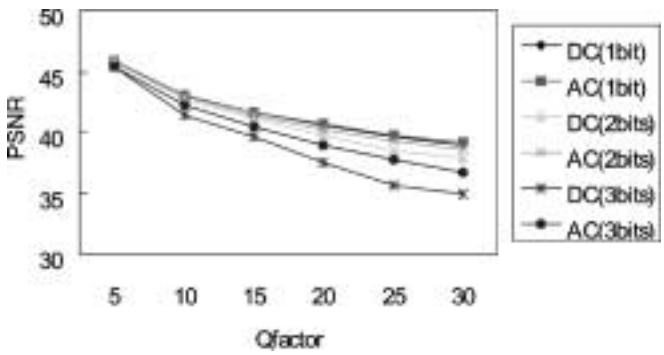


Fig. 7. Comparison of PSNR when embedding in DC and AC components.

image blocks into two classes: uniform blocks and non-uniform blocks. The factor α in Eq. 1 is a capacity parameter which can control the level of embedding capacity. The uniform block has a smaller α value, while the non-uniform block has a larger α value. As to the block classification, a preset threshold T is employed. The energy intensity measure G of a block is calculated by the following expression:

$$G = \sqrt{\sum_{i=1}^{63} (AC_i)^2}, \tag{5}$$

where AC_i is the ith AC component of the DCT block. If the value of G is smaller than a threshold T, then the block is a uniform block. Otherwise, the block is a non-uniform block.

However, in order for the extractor to identify the types of blocks correctly, one bit will be added to each block as the block type indicator. We employ method proposed by Kobayashi *et al.* to embed this bit. If the block is a uniform block, then a 0 bit is embedded in the last AC component. Otherwise, a 1 bit is embedded in the last AC component. The reason for choosing the last AC component is that the number of uniform blocks is greater than that of non-uniform blocks in the normal case. And of course the modification quantization table is used here.

3.4. *Embedding Algorithm*

The algorithm of secret data embedding into the JPEG compressed image includes the following 4 steps.

1. Apply entropy decoding to the JPEG compressed image. For each block, *Step* 2 and *Step* 3 are then executed.
2. Let F be the quantized DCT block with $F(i, j)$ denoting the (i, j)th entry of F, where $0 \leqslant i, j < 8$. For each $|F(i, j)| > 1$, calculate $E(i, j)$ in Eq. 3. Embed the secret data with length $E(i, j)$ in the LSBs of $F(i, j)$.
3. If the block is a uniform block, a 0 bit is embedded in the last AC coefficient. Otherwise, a 1 bit is embedded in the last AC coefficient.
4. Put the modified quantization table in the header of the JPEG file, and then apply JPEG entropy encoding. This way, the stego-image is produced.

3.5. *Extracting Algorithm*

The way to extract the secret data from the stego-image is the same as that to embed data. The capacity factor α should be passed to the extractor first. Then JPEG entropy decoding is performed. And the secret data is extracted by using the reverse of the embedding method. The capacity factor α can be viewed as another private key essential to secret data extracting. If α is kept secret, then no illegal user gets to know the exact number of bits embedded in each DCT component.

4. Experimental Results

The algorithms presented in the previous section were implemented and tested on two standard images "Lena" and "Jet", both of which were 256 gray scale, 512×512 images. In our experiment, we used the JPEG code from the Stanford University Portable Video Research Group (Hung, 1993). Besides, we employed the peak signal-to-noise ratio (PSNR) as a measure of the stego-image quality. It is defined as follows:

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{MSE} \text{dB}, \tag{6}$$

where MSE is the mean-square error. For an $N \times N$ image, its MSE is defined as

Table 2
Comparison of embedding capacity in Lena image when Q factor$= 5$

	Capacity Factor (α)	Embedded data (bits)	Compressed file (bytes)	Embedding Capacity	PSNR (dB)
JPEG	–	–	87891	–	45.85
Jpeg–Jsteg	–	67000	87552	9.56%	44.85
Chang *et al.*	–	212992	88912	29.9%	33.71
Proposed method	0.18	68871	89436	9.63%	43.97
	0.36	107035	89235	15.0%	43.06
	0.5	122032	89199	17.1%	42.53
	0.75	134437	89185	18.8%	41.70
	1.0	142820	89160	20.0%	40.84

Table 3
Comparison of embedding capacity in Lena image when Q factor$= 15$

	Capacity Factor (α)	Embedded data (bits)	Compressed file (bytes)	Embedding Capacity	PSNR (dB)
JPEG	–	–	45321	–	41.75
Jpeg–Jsteg	–	31933	45071	8.85%	40.70
Chang *et al.*	–	212992	88108	30.0%	29.64
Proposed method	0.2	33790	48647	8.68%	40.21
	0.4	51161	48593	13.2%	38.91
	0.6	54652	48636	14.0%	38.10
	0.8	62612	48575	16.1%	36.87
	1.0	63083	48602	16.2%	36.56

Table 4
Comparison of embedding capacity in Jet image when Q factor$= 5$

	Capacity Factor (α)	Embedded data (bits)	Compressed file (bytes)	Embedding Capacity	PSNR (dB)
JPEG	–	–	89184	–	45.60
Jpeg–Jsteg	–	66186	88817	9.3%	44.43
Chang *et al.*	–	212992	89632	29.7%	32.56
Proposed method	0.18	71037	90101	9.85%	43.25
	0.36	108687	89956	15.1%	42.21
	0.5	123533	89968	17.2%	41.75
	0.75	135824	89925	18.9%	40.95
	1.0	144989	89979	20.1%	40.13

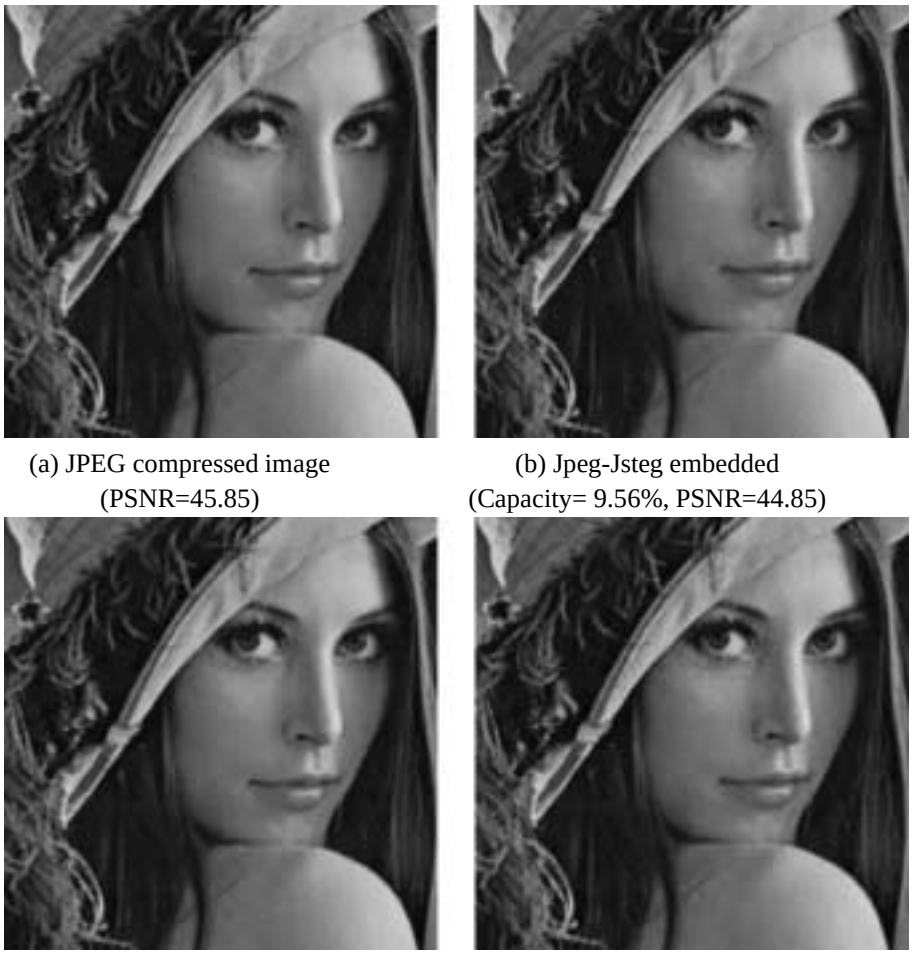
Table 5
Comparison of embedding capacity in Jet image when Q factor$= 15$

	Capacity Factor (α)	Embedded data (bits)	Compressed file (bytes)	Embedding Capacity	PSNR (dB)
JPEG	–	–	45259	–	41.15
Jpeg–Jsteg	–	32868	45060	9.11%	40.07
Chang *et al.*	–	212992	89018	29.9%	28.16
Proposed method	0.2	35678	48137	9.26%	39.38
	0.4	53532	48137	13.9%	38.12
	0.6	58254	48101	15.1%	37.32
	0.8	65457	48222	17.0%	36.29
	1.0	66363	48213	17.2%	35.77

$$MSE = \left(\frac{1}{\mathrm{N}}\right)^2 \times \sum_{i=1}^{N}\sum_{j=1}^{N}\left(x[i,j] - \bar{x}[i,j]\right)^2. \tag{7}$$

Here, $x[i,j]$ and $\bar{x}[i,j]$ denote the original and decoded gray levels of the pixel $[i,j]$ in the image, respectively. A larger PSNR value means that the stego-image preserves the original image quality better.

Our method employs the capacity factors α to control the level of embedding capacity. Users can adjust it to balance between the image quality (PSNR) and the embedding capacity. If the capacity factor is selected as a large number, then the embedding capacity can be raised, but the cost is that the compression ratio of the image gets low. On the con-



(a) JPEG compressed image (PSNR=45.85)

(b) Jpeg-Jsteg embedded (Capacity= 9.56%, PSNR=44.85)

(c) Our method embedded (Capacity=9.65%, PSNR=43.97)

(d) Our method embedded (Capacity=20.0%, PSNR=40.84)

Fig. 8. Experimental results for Lena using JPEG with Q factor 5.

trary, the capacity factor should be selected to be a small number if too much distortion is to be avoided with the compression ratio maintained high. Through quite a number of experiments, the capacity factor α is finally selected for uniform blocks, and $1.2 * \alpha$ for non-uniform blocks. They apply to a wide variety of images.

In addition, we also conducted some experiments to show the flexibility of our method. Experimental results are in Tables 2 through 5. Tables 2 and 3 are for the Lena image compressed by JPEG with a Q factor of 5 and 15, respectively. Tables 4 and 5 are for the Jet image compressed by JPEG with a Q factor of 5 and 15, respectively. We selected a proper capacity factor so that our embedding capacity is about the same as that of Jpeg–Jsteg, and the results showed that the stego-image quality of our method was as good as that of Jpeg–Jsteg. However, the size of the compressed file produced by

JPEG compressed image (PSNR=41.75)

(b) Jpeg-Jsteg embedded (Capacity= 8.85%, PSNR=40.70)

(c) Our method embedded (Capacity=8.68%, PSNR=40.20)

(d) Our method embedded (Capacity=16.2%, PSNR=36.56)

Fig. 9. Experimental results for Lena using JPEG with Q factor 15.

our method was a bit larger than that of Jpeg–Jsteg because we embedded one bit in the last AC component to indicate the block type. But the size of the compressed file did not expand when the embedding capacity increased. The embedding capacity of Chang *et al.* is larger than others, but the compression ratio is bounded and the stego-image quality is not good enough. Generally speaking, our experimental results show that the proposed method is able to achieve the embedding capacity of around 20% of the stego-image with little or no noticeable degradation of image quality when the compression ratio is low. Of course the embedding capacity is lower when the compression ratio increases. Figs. 8 and 9 show the stego-image quality.

5. Conclusions

In this paper, a high capacity data hiding method is proposed. Our method embeds the secret data into the JPEG compressed image directly. Traditional schemes embed fixed-size secret data in the quantized DCT components, and therefore the embedding capacity is quite restricted. To improve the embedding capacity of the JPEG compressed image, we conduct an adaptive capacity estimation for each DCT component based on HVS. The JPEG default quantization table and texture masking phenomena are exploited in our algorithm to estimate the capacity of each DCT component. To sum up, ours is an adaptive data hiding method with which one can adjust capacity factor to balance between the image quality and the embedding capacity dynamically. Furthermore, the proposed method is securer than most of its predecessors.

Experimental results show that our method indeed provides acceptable image quality and adjustable embedding capacity. The distortion of the stego-image caused by our method at low embedding capacity is approximately the same as that by Jpeg–Jsteg. High embedding capacity of around 20% of the JPEG compressed image size is achieved with little noticeable degradation of image quality when the compression ratio is low. The proposed method is very practical for most image files that are stored and transmitted in the JPEG format.

References

Adelson, E. (1990). *Digital Signal Encoding and Decoding Apparatus*. U.S. Patent, No. 4939515.

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

Cox, I.J., J. Kilian, F.T. Leighton and T. Shamoon (1997). Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Processing*, **6**(12), 1673–1687.

Daly, S. (1994). A visual model for optimizing the design of image processing algorithms. In *Proceedings of IEEE International Conference on Image Processing*, Austin, Texas, U.S.A., Nov., Vol. II. pp. 16–20.

DES Encryption Standard (DES), National Bureau of Standards (U.S.) (1977). *Federal Information Processing Standards Publication*, **46**, National Technical Information Service, Springfield, VA.

Johnson, N., and S. Jajodia (1998). Steganalysis of images created using current steganography software. In *Proceedings of Information Hiding Workshop*, Portland, Oregon, USA, April, LNCS **1525**. pp. 273–289.

Hung, A.C. (1993). *PVRG–JPEG CODEC*, Technical Report, Portable Video Research Group, Stanford University.

Kobayashi, H., Y. Noguchi and H. Kiya (1999). A method of embedding binary data into JPEG bitstreams. *IEICE Trans. Information and Systems*, J83-D-II, 1469–1476.

Noguchi, Y., H. Kobayashi and H. Kiya (2000). A method of extracting embedded binary data from JPEG bitstreams using standard JPEG decoder. In *Proceedings of IEEE International Conference on Image Processing*, Vancouver, BC, Canada, 10–13 Sept., Vol. 1. pp. 577–580.

Pennebaker, W., and J. Mitchell (1993). *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York.

Van Schyndel, R.G., A.Z. Tirkel and C.F. Osborne (1994). A digital watermark. In *Proceedings of the First IEEE International Conference on Image Processing*, Austin, Texas, USA, Vol. 11. pp. 86–90.

Wandell, B.A. (1995). *Foundations of Vision*. Sinuaer, Sunderland, MA.

Wang, R.Z., C.F. Lin and J.C. Lin (2001). Image hiding by optimal LSB substitution and genetic algorithm. *Pattern Recognition*, **34**, 671–683.

Wolfgang, R.B., C.I. Podilchuk and E.J. Delp (1999). Perceptual watermarks for digital images and video. In *Proc. IEEE 87*. pp. 1108–1126.

Xia, X.G., C.G. Bongelet and G.R. Arce (1997). A multiresolution watermark for digital images. In *Proceedings of IEEE International Conference on Image Processing*, Santa Barbara, CA, 26–29 Oct 1997, Vol. 3. pp. 548–551.

H.-W. Tseng received the BS degree in computer science and information engineering from Tamkang University, Taipei County, Taiwan, in 1986, and his MS degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1988. Since 1989 to 2000 he worked as a computer engineer. He is currently working toward to the PhD degree in computer science and information engineering at National Chung Cheng University, Chaiyi, Taiwan. His research interests include image processing, data hiding, and error resilient coding.

C.-C. Chang was born in Taichung, Taiwan, on Nov. 12, 1954. He received his BS degree in applied mathematics in 1977 and his MS degree in computer and decision science in 1979 from National Tsing Hua University, Hsinchu, Taiwan. He received his PhD degree in computer engineering in 1982 from National Chiao Tung University, Hsinchu, Taiwan. From 1983 to 1989, he was among the faculty of Institue of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. Since August 1989, he worked as a professor of Institute of Computer Science and Information Engineering at National Chung Cheng University, Chaiyi, Taiwan. Dr. Chang is a fellow of IEEE and a member of Chinese Language Computer Society, Chinese Institute of Engineers of Republic of China, and Phi Tau Phi Society of Republic of China. His reaserch interests include computer cryptography, data engineering, and image compression.

Efektyvus duomenų slėpimas JPEG kompresijoje

Hsien-Wen TSENG, Chin-Chen CHANG

JPEG yra vienas iš populiariausių skaitmeninių vaizdų failų formatų. Tačiau iki šiol buvo pasiūlyti tik keli duomenų slėpimo (steganografijos) metodai, naudojantys JPEG formatą. Straipsnyje pateikiamas naujas efektyvus duomenų slėpimo metodas, pagrįstas JPEG. Šis metodas remiasi našumo lentele, kurios pagalba įvertinama, kiek bitų galima paslėpti kiekvienoje DCT komponentėje, neiškreipiant vaizdo. Našumo lentelė išvedama iš JPEG kvantavimo lentelės ir spalvų modelio HVS. Po to naudojamas adaptyvus mažiausiai reikšminio bito pakeitimo algoritmas, perskaičiuojant DCT koeficientus. Šitoks metodas našumo faktoriaus dėka leidžia valdyti vaizdo kokybę. Rezultatai, patikrinti eksperimentiniais skaičiavimais, rodo siūlomo algoritmo efektyvumą.