High performance JPEG steganography using quantization index modulation in DCT domain

Hideki Noda [a,*], Michiharu Niimi [a], Eiji Kawaguchi [b]

[a] *Department of Systems Innovation and Informatics, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka-shi, 820-8502 Japan*

[b] *Keio University, 2-15-45 Mita, Minato-ku, Tokyo, 108-8345 Japan*

Abstract

This paper presents two JPEG steganographic methods using quantization index modulation (QIM) in the discrete cosine transform (DCT) domain. The two methods approximately preserve the histogram of quantized DCT coefficients, aiming at secure JPEG steganography against histogram-based attacks. In comparison with F5 as a representative JPEG steganography, the proposed methods show high performance with regard to embedding rate, PSNR of stego image, and particularly histogram preservation.

Key words: Steganography, Information hiding, Information security, Quantization index modulation, JPEG, DCT, Histogram matching

1 Introduction

Steganography is the practice of hiding or camouflaging secret data in an innocent looking dummy container. This container may be a digital still image, audio file, or video file. Once the data has been embedded, it may be transferred across insecure lines or posted in public places. Therefore, the dummy container should seem innocent under most examinations. Steganography provides good security in itself and when combined with encryption becomes an extremely powerful security tool.

* Corresponding author. Tel.: +81-93-884-3247; fax: +81-93-871-5835
Email address: noda@know.comp.kyutech.ac.jp (Hideki Noda).

The JPEG compression using the discrete cosine transform (DCT) is still the most common compression standard for still images, though the JPEG2000 standard has already been released. Getting less attention, JPEG images are therefore the most suitable dummy (cover) images for steganography using compressed images. Several steganographic techniques using JPEG images have already been proposed such as J-Steg (Upham 1997), F5 (Westfeld 2001) and OutGuess (Provos 2001). It is well known that embedding by J-Steg is detectable using the chi-square attack (Westfeld 2001) since it is based on flipping the least significant bits (LSBs). F5 cannot be detected by the chi-square attack. However it can be detected by a specific technique (Fridrich et al. 2003) which exploits a significant change on the histogram of quantized DCT coefficients caused by embedding. J-Steg and F5 are known as high-capacity steganographic methods. OutGuess has been presented to counter the chi-square attack. OutGuess embeds data by a similar way as J-Steg, however the LSB flipping is applied to a part of usable coefficients. The rest part is used to make the after-embedding histogram of quantized DCT coefficients match the histogram of cover image. Therefore the embedding capacity of OutGuess is much lower than that of J-Steg. Note that OutGuess can preserve only the global histogram for all frequencies, and cannot preserve histograms for individual frequencies.

Recently, another high-capacity JPEG steganographic method has been presented (Eggers et al. 2002) where quantization index modulation (QIM) (Chen and Wornell 2001), in fact spirit of QIM, is utilized to embed data into quantized DCT coefficients. Its main feature is that histograms of DCT coefficients for individual frequencies after embedding are kept same as those before embedding [1] and therefore it is robust against histogram-based attacks. Data embedding by QIM using two different quantizers is generally carried out in such a way that in order to embed zero, one of two quantizers is used at the quantization step of DCT coefficients, and the other quantizer is used to embed one. In histogram-preserving data mappings (HPDM) by Eggers et al. (2002), the data mapping technique is applied to quantized DCT coefficients. Given two different mappings which map DCT coefficients onto the different sets of representatives of two quantizers, one mapping is used to embed zero and the other mapping is used to embed one. The histogram preserving embedding is realized under the condition that the probability of zero in binary data to be embedded is equal to the probability of quantized DCT coefficients of cover image belonging to representatives of one of two quantizers. Note that the condition also means that the probability of one in binary embedding data is equal to the probability of quantized coefficients belonging to representatives of the other quantizer. However the condition cannot be met in general. In order to overcome this problem, secret data are processed by two steps to

[1] Note that this is true in a statistical sense and therefore a realization of after-embedding histogram is not guaranteed to be identical to the original histogram.

meet the condition. First, binary encryption is applied to the secret data to make the probability of zero equal to that of one, i.e., both probabilities 0.5. Then, the entropy decoding is applied to the encrypted data to conform both probabilities of zero and one to those on DCT coefficients. Note that different entropy decoder should be prepared for each frequency component since the two probabilities on DCT coefficients differ significantly among frequency components.

This paper presents two steganographic methods using QIM (or its spirit) in the DCT domain. In the two methods, embedding is carried out just during quantization of DCT coefficients (not after quantization step) and it is different from previously proposed methods including HPDM. The first one is a histogram preserving method using two quantizers, where the representatives of each quantizer are given in advance and the intervals for each representative are set so as to preserve the histogram of cover image. The method can meet the aforementioned condition by a different way from (Eggers et al. 2002) and does not need the entropy decoder to meet the condition. The second one is a histogram quasi-preserving method which uses QIM in a straightforward way with a device not to change the after-embedding histogram excessively.

The rest of this paper is organized as follows. In Section 2, direct application of QIM to DCT coefficients is described with a conclusion that it cannot be allowed for secure steganography because of significant histogram change caused by embedding. In Section 3, the proposed histogram preserving and quasi-preserving methods using QIM are presented. Experimental results for the proposed methods are given in Section 4, and conclusions are addressed in Section 5.

2 QIM in DCT Domain

Consider to apply QIM (Chen and Wornell 2001) using two different quantizers to embed binary data at the quantization step of DCT coefficients in JPEG compression. Each bit (zero or one) of binary data is embedded in such a way that one of two quantizers is used for quantization of a DCT coefficient , which corresponds to embed zero, and the other quantizer is used to embed one. Given a quantization table and a quality factor for JPEG compression, quantization step size $\Delta_k, 1 \leq k \leq 64$ for each frequency component can be decided. Then, two codebooks, C^0 and C^1, for two quantizers are chosen as $C^0 = \{2j\Delta_k; j \in Z\}$ and $C^1 = \{(2j+1)\Delta_k; j \in Z\}$ for k-th frequency. Given a k-th frequency DCT coefficient x, $2q$ with $q = \arg\min_j \parallel x - 2j\Delta_k \parallel$ becomes the quantized coefficient in case of embedding zero, for example, and $2q + 1$ with $q = \arg\min_j \parallel x - (2j + 1)\Delta_k \parallel$ in case of embedding one.

Assuming that the probabilities of zero and one are same in binary data to be embedded, consider how histograms of quantized DCT coefficients change after embedding. From now on, we assume that DCT coefficients belonging to k-th frequency are divided by its quantization step size Δ_k in advance and then two codebooks, C^0 and C^1 can be defined as $C^0 = \{2j; j \in Z\}$ and $C^1 = \{2j+1; j \in Z\}$ for all frequency components. Let $h_i, i \in Z$ denote the number of DCT coefficients whose values x are in the interval $i - 0.5 < x < i + 0.5$. Let h_i^- and h_i^+ denote the number of DCT coefficients in the interval $i - 0.5 < x < i$ and $i < x < i + 0.5$, respectively, and therefore $h_i^- + h_i^+ = h_i$. After embedding by QIM, the histogram h_i is changed to $h_i^{'}$ as

$$h_i^{'} = \frac{1}{2}h_i + \frac{1}{2}(h_{i-1}^+ + h_{i+1}^-). \quad (1)$$

The change in (1) can be understood as follows. If i is an even number, i.e., $i \in C^0$ and C^0 is used for embedding zero, half of DCT coefficients in the interval $i-0.5 < x < i+0.5$ are used for embedding zero and their quantized coefficients are unchanged after embedding. However the other half, $(h_i^- + h_i^+)/2$ coefficients are used for embedding one, resulting in that $h_i^-/2$ coefficients are quantized to $i-1$ and $h_i^+/2$ coefficients to $i+1$. Alternatively, $h_{i-1}^+/2$ coefficients from the bin $i-1$ and $h_{i+1}^-/2$ coefficients from the bin $i+1$ are quantized to i for embedding zero. With similar consideration, it is easily understood that the change shown in (1) holds true also for odd number i.

A typical example of before- and after-embedding histograms is shown in Fig. 1. These histograms are those of (2,2) frequency component among 64 components $(k, l), 1 \leq k, l \leq 8$ for "Lena" image (512×512 pixels in size, 8 bit per pixel (bpp)) compressed with quality factor 80. It is seen that the numbers of quantized coefficients for low absolute values around zero change significantly after embedding. Eq. (1) indicates that if $h_i = h_{i-1}^+ + h_{i+1}^-$, then the number in the bin i does not change. In particular for $i = 0, \pm 1$, however, much difference between h_i and $h_{i-1}^+ + h_{i+1}^-$ causes the significant change on $h_i^{'}$ after embedding. Therefore a straightforward application of QIM in the DCT domain cannot be allowed for secure steganography against histogram-based attacks.

3 Histogram Preserving and Quasi-Preserving JPEG Steganography

In this section, two JPEG steganographic methods using QIM (or its spirit) in the DCT domain are presented. The first one is a histogram preserving method based on histogram matching similar in its nature to HPDM (Eggers et al. 2002). The histogram matching scheme may not be directly

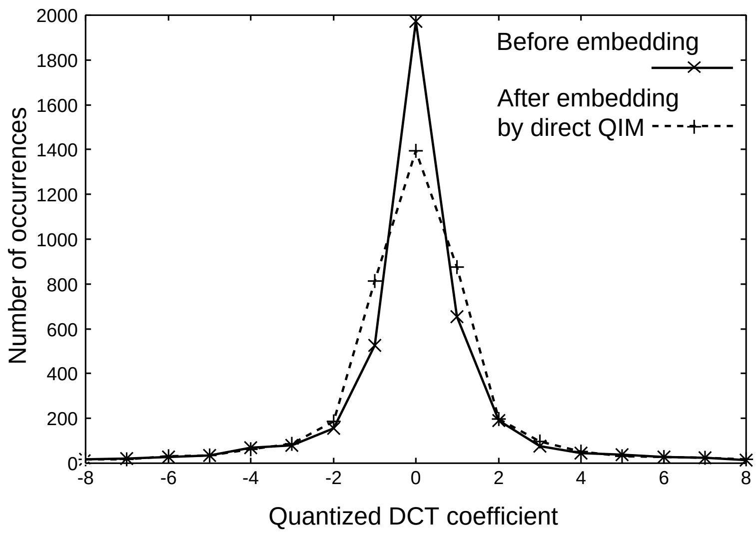

Fig. 1. Before- and after-embedding histograms of (2,2) frequency component. QIM was directly applied for embedding.

related with QIM in the sense that the closest representative of a relevant quantizer to a given DCT coefficient is not necessarily chosen as a quantized coefficient. Our method differs from HPDM in that embedding is carried out during quantization of DCT coefficients and the condition mentioned in Introduction to match after-embedding histogram with before-embedding one is met by a different way. The latter relieves us of using the entropy decoder to meet the condition and sending a receiver the information on probabilities of zero (or one) for each frequency component. The second one is a histogram quasi-preserving method which uses QIM in a straightforward way with a device not to change the after-embedding histogram excessively. Here we call the first method histogram matching JPEG (HM-JPEG) steganography and the second one QIM-JPEG steganography.

3.1 HM-JPEG Steganography

Consider histogram matching at quantization step of DCT coefficients, assuming that the probabilities of zero and one are same in binary data to be embedded. This assumption is quite natural since any compressed data has such property. Histogram matching is here considered separately for positive coefficient part and negative one, since there sometimes exists asymmetry between both parts. In the following, the matching for positive part is only described (negative part can be treated in the same way).

Two quantizers, $Q^0(x)$ and $Q^1(x)$ are prepared: the former used to embed zero and the latter to embed one.

$$Q^0(x) = 2j,\ t_j^0 < x < t_{j+1}^0,\ j \in \{0, 1, 2, \ldots\}, \tag{2}$$

$$Q^1(x) = 2j + 1,\ t_j^1 < x < t_{j+1}^1,\ j \in \{0, 1, 2, \ldots\}, \tag{3}$$

where x is a positive DCT coefficient and $t_0^0 = t_0^1 = 0$. The decision threshold values $t_j^0, j \in \{1, 2, \ldots\}$ for $Q^0(x)$ are set so that they satisfy

$$\frac{1}{2}N(t_j^0 < x < t_{j+1}^0) = \begin{cases} h_0^+ & \text{for} \ \ j = 0 \\ h_{2j} & \text{for} \ \ j \in \{1, 2, \ldots\}, \end{cases} \tag{4}$$

where $N(t_j^0 < x < t_{j+1}^0)$ depicts the number of coefficients in the interval $t_j^0 < x < t_{j+1}^0$. Note that $1/2$ in (4) means that half of relevant coefficients are used for embedding zero and its number is adjusted to h_0^+ or h_{2j} of cover image to preserve the histogram of cover image. The decision threshold values $t_j^1, j \in \{1, 2, \ldots\}$ for $Q^1(x)$ are similarly set as they satisfy

$$\frac{1}{2}N(t_j^1 < x < t_{j+1}^1) = h_{2j+1},\ j \in \{0, 1, 2, \ldots\}. \tag{5}$$

Fig. 2 illustrates the aforementioned histogram matching using two quantizers. From Eqs. (4) and (5), it is found that the histogram preservation can be realized if $h_0^+ + \sum_{j=1}^{\infty} h_{2j} = \sum_{j=0}^{\infty} h_{2j+1} = N(0 < x < \infty)/2$. The condition $h_0^+ + \sum_{j=1}^{\infty} h_{2j} = \sum_{j=0}^{\infty} h_{2j+1}$, i.e., $\sharp even = \sharp odd$ is infrequently approximately satisfied in histograms for very low frequency components, but does not hold true in general. The relation $\sharp even > \sharp odd$ usually holds true for AC components, and in high frequency components, $\sharp even \gg \sharp odd$ because h_0^+ is much larger than others.

Consider how to match after-embedding histogram with before-embedding one under the relation of $\sharp even > \sharp odd$. We introduce a dead zone, $0 < x < t_d$ ($t_d < 0.5$) in which DCT coefficients are not used for embedding[2]. t_d is determined as it fulfills

$$N_d = N(0 < x < t_d) = \sharp even - \sharp odd. \tag{6}$$

[2] In case that $\sharp even < \sharp odd$, we cannot introduce the dead zone, i.e., $t_d = 0$, and cannot fully match two histograms. In such a case, the difference between $\sharp even$ and $\sharp odd$ is generally small, and partial matching is possible to match for smaller absolute value coefficients (bins). In practice, matching for larger absolute value bins is not needed since the number of samples in such a bin is small and mismatch in such a bin is not distinguishable by steganalysis.

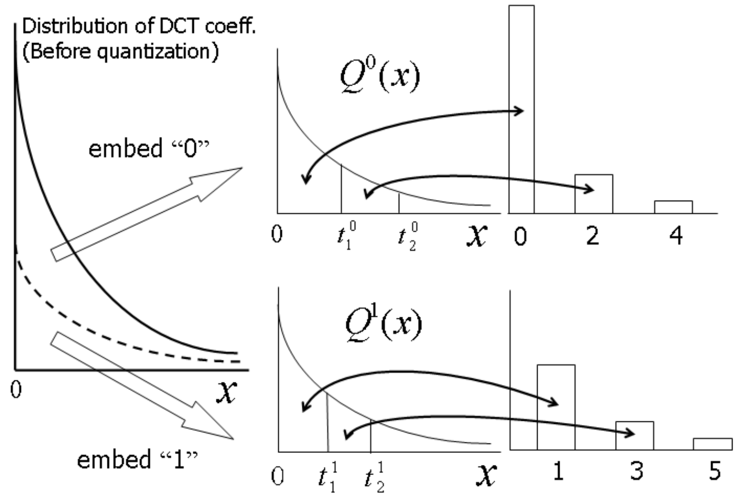

Fig. 2. Illustration of histogram matching using two quantizers.

Eq. (6) means that $\sharp odd$ is equal to $\sharp even$ with least N_d coefficients removed. Then using t_d and N_d, the decision threshold values $t_j^0, t_j^1, j \in \{1, 2, \ldots\}$ for $Q^0(x)$ and $Q^1(x)$ are set so that they satisfy

$$\frac{1}{2} N(t_d < x < t_1^0) = h_0^+ - N_d \tag{7}$$

$$\frac{1}{2} N(t_j^0 < x < t_{j+1}^0) = h_{2j},\ j \in \{1, 2, \ldots\}, \tag{8}$$

$$\frac{1}{2} N(t_d < x < t_1^1) = h_1 \tag{9}$$

$$\frac{1}{2} N(t_j^1 < x < t_{j+1}^1) = h_{2j+1},\ j \in \{1, 2, \ldots\}, \tag{10}$$

respectively. Eqs. (7) to (10) indicate that $\sharp odd$ and $\sharp even - N_d$ are equal to $N(t_d < x < \infty)/2$ and then histogram matching becomes possible. Fig. 3 illustrates the aforementioned histogram matching using two quantizers with a dead zone.

Note that in the proposed histogram preserving steganography, quantized values 0s cannot be treated as zeroes embedded in them, because they cannot be discriminated from 0s in the dead zone. Therefore, if 0 value is selected to embed a zero at quantization step, the embedding process to embed the zero should be continued until other even value except 0 is chosen. This causes a serious problem on the histogram that a surplus of even coefficients is produced resulting in a significant change of the after-embedding histogram (particularly the number of 0 coefficients increased, those of 1 and -1 coefficients decreased

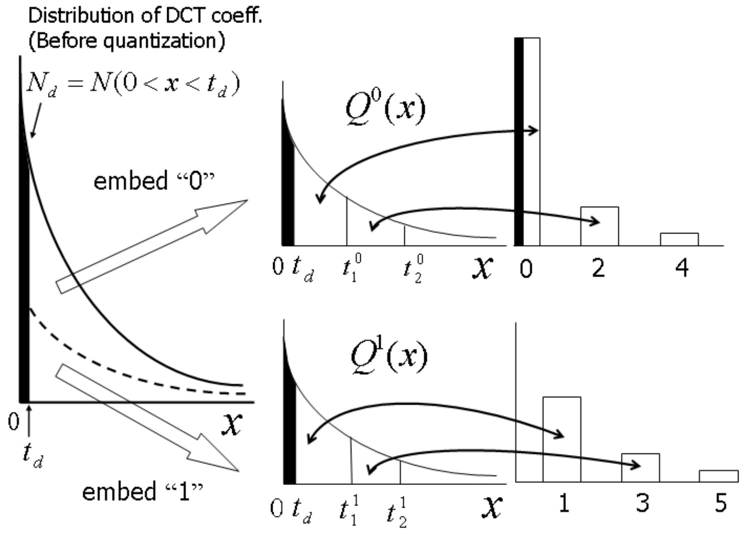

Fig. 3. Illustration of histogram matching using two quantizers with a dead zone. The black area corresponds to the dead zone.

as shown in Fig. 4). This problem can be resolved by applying the exclusive OR (XOR) operation to a message bit with a random bit whenever embedding a message bit. That is, an embedding bit is the XOR-applied message bit. When 0 coefficient is selected to embed a zero of embedding bit, which means no message embedded, an embedding bit at subsequent embedding is not necessarily kept same though the message bit is same until non-0 coefficient is selected. The message bits can be recovered from the embedded XOR-applied message bits using the same random binary sequence as one used in embedding. Fig. 4 shows that HM-JPEG steganography with XOR operation can preserve the before-embedding histogram.

3.2 QIM-JPEG Steganography

The aforementioned HM-JPEG steganography is a histogram matching method using two quantizers, where the representatives of each quantizer are given in advance and the intervals for each representative are set so as to preserve the histogram of cover image. Though the HM-JPEG has spirit of QIM in the usage of two quantizers, it is different from QIM in that the chosen representative of a relevant quantizer is not necessarily closest to a given input. In this section we consider to apply QIM in a straightforward way, where the closest representative to a given DCT coefficient is chosen as a quantized coefficient. We can expect to obtain less distorted stego images by QIM-JPEG than by HM-JPEG.

As shown in Fig. 1, the most significant changes caused by direct application of

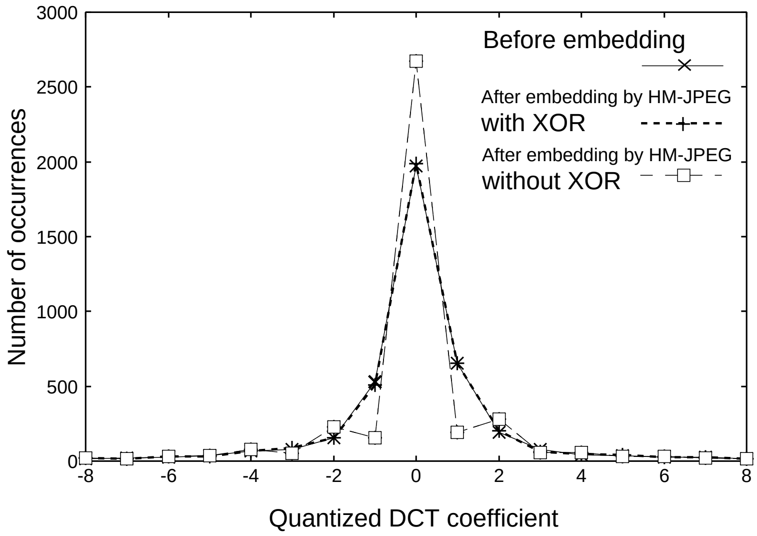

Fig. 4. Before- and after-embedding histograms of (2,2) frequency component. In HM-JPEG steganography, the XOR operation should be applied to embedding message to achieve histogram preservation.

QIM are decrease of h_0 and increase of h_1 and h_{-1}. Let us consider preserving h_0 after embedding. Here we consider only positive coefficient part to preserve h_0^+. Negative part can be treated in the same way which preserves h_0^-.

By direct application of QIM, h_0^+ is changed to

$$h_0^{+'} = \frac{1}{2}h_0^+ + \frac{1}{2}h_1^-. \tag{11}$$

The relation $h_0^+ > h_1^-$ generally holds true, and in high frequency components, $h_0^+ \gg h_1^-$. Therefore $h_0^{+'}$ becomes smaller than h_0^+, and much smaller in high frequency components. We can generally preserve h_0^+ by introducing a dead zone, $0 < x < t_d$ ($t_d < 0.5$) where DCT coefficients are not used for embedding[3]. t_d is determined as it fulfills

$$N_d = N(0 < x < t_d) = h_0^+ - h_1^-. \tag{12}$$

Then it is easy to see the preservation of h_0^+ as follows.

$$h_0^{+'} = N_d + \frac{1}{2}(h_0^+ - N_d) + \frac{1}{2}h_1^-$$

[3] In case that $h_0^+ < h_1^-$ which is infrequently observed in very low frequency components, we can not introduce the dead zone, i.e., $t_d = 0$. However such a case does not cause a serious problem since in such a case the difference between h_0^+ and h_1^- is generally small and therefore the difference between $h_0^{+'}$ and h_0^+ is also small.

$$= N_d + h_1^- = h_0^+$$

A typical example of after-embedding histogram by QIM-JPEG steganography is shown in Fig. 5. Comparing with Fig. 1, it is seen that the preservation of h_0 is achieved and the increase of h_1 and h_{-1} is suppressed. In fact $h_1{}'$ and $h_{-1}{}'$ become a bit smaller than h_1 and h_{-1}. Note that in QIM-JPEG steganography the XOR operation should also be applied to embedding message as applied in HM-JPEG steganography.

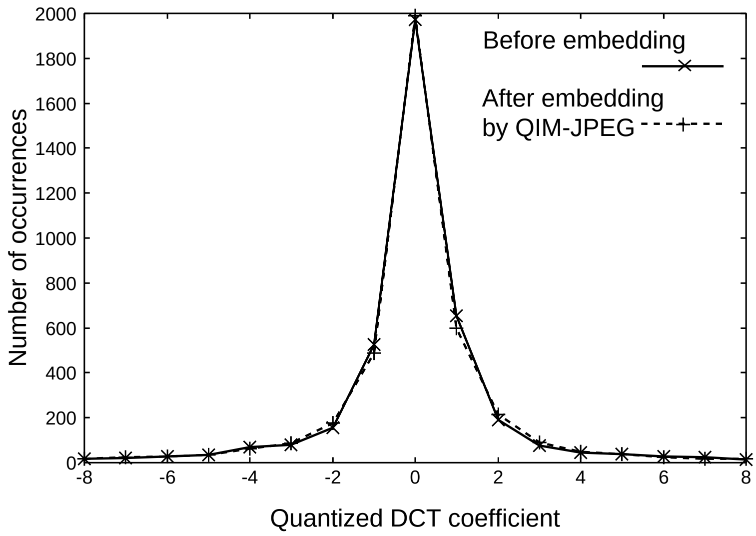

Fig. 5. Before- and after-embedding histograms of (2,2) frequency component. QIM-JPEG steganography can achieve histogram quasi-preservation.

4 Experiments

The proposed HM-JPEG and QIM-JPEG steganographic methods were implemented using image processing software (Yagi et al. 2001), which produces a JPEG-compressed bitstream instead of a complete JPEG file with headers. Therefore the bpp values of compressed images in the following experimental results are not included the sizes for headers. Embedding performance of the proposed methods were evaluated comparing with HPDM and F5 using nine standard images. These images are 512×512 pixels in size, 8 bpp gray images, and were compressed with quality factor 80 since around 80 is a typical value usually used in JPEG steganography. The least 21 frequency components, including the DC component, in the zigzag scan of 64 components were used for embedding experiments, since in higher frequency components there is very little or almost no space for embedding in the used nine images except "Mandrill". For the DC component, QIM was directly applied for embedding, since we could hardly observe discriminating changes on the after-embedding histogram.

The histogram change can be measured by Kullback-Leibler divergence, which is defined as

$$\begin{aligned} D_{KL} &= \sum_i P_i \log \frac{P_i}{P_i'} \\ &= \sum_i \frac{h_i}{N_{total}} \log \frac{h_i}{h_i'}, \end{aligned} \tag{13}$$

where P_i and P_i' are probabilities of quantized coefficient i before and after embedding, respectively, and N_{total} is the total number of coefficients for each frequency component, i.e., here $N_{total} = 4096$. Here Eq. (13) was evaluated only for nonzero h_i and nonzero h_i'.

Table 1 shows experimental results with maximum amount of embedding performed. Fig. 6 shows four histograms of (1,1) frequency component for "Lena": histogram of cover image (before embedding) and after-embedding histograms by HM-JPEG, QIM-JPEG and F5. The KL divergence value shown in Table 1 is the mean for the least 21 frequency components except the DC component. The KL divergence value by F5 is much larger than those by HM-JPEG, QIM-JPEG and HPDM. Smaller KL divergence values represent better histogram preservation. Furthermore, HM-JPEG and QIM-JPEG achieved higher embedding rates with less degradation of image quality (higher PSNR vales) than F5. HPDM achieved much higher embedding rate than others with the cost of the lowest PSNR value. Such a large capacity is because of many 0 coefficients being used for embedding in HPDM.

Table 2 shows experimental results with equal amount of embedding performed. In HM-JPEG, QIM-JPEG and HPDM, the amount was adjusted to that for F5 by randomly selecting DCT coefficients used for embedding. Regarding histogram preservation, HPDM is the best, HM-JPEG is comparable to HPDM and QIM-JPEG is worse than HM-JPEG. HPDM achieved the best embedding efficiency (the number of embedded data divided by the number of coefficients whose values changed by embedding) since 0 coefficients can be used for embedding in HPDM. The decrease of embedding efficiency in HM-JPEG and QIM-JPEG is probably due to the introduction of a dead zone by which 0 coefficients cannot be treated as zeroes embedded. In terms of quality of stego image, HM-JPEG and QIM-JPEG are superior to HPDM and F5. Among the two proposed methods, QIM-JPEG produced a bit higher PSNR stego images than HM-JPEG, probably because in QIM-JPEG the closest representative of a relevant quantizer to a given DCT coefficient is chosen as a quantized coefficient.

Table 1
Results of embedding experiments with maximum amount of embedding performed

image	method	max. embedded data size (bits)	compressed image (bpp)	embedding rate (%)	PSNR (dB)	KL divergence
Lena	(no embedding)	-	1.129	-	38.55	-
	HM-JPEG	45908	1.149	15.2	37.41	0.00138
	QIM-JPEG	45825	1.158	15.1	37.61	0.00397
	HPDM	77693	1.168	25.4	35.84	0.00114
	F5	35174	0.990	13.6	37.26	0.04783
Barbara	(no embedding)	-	1.517	-	36.92	-
	HM-JPEG	53645	1.543	13.3	35.91	0.00106
	QIM-JPEG	53672	1.547	13.2	36.07	0.00344
	HPDM	81372	1.566	19.8	34.28	0.00125
	F5	43555	1.392	11.9	35.69	0.04117
Mandrill	(no embedding)	-	2.364	-	32.63	-
	HM-JPEG	71234	2.369	11.5	32.04	0.00124
	QIM-JPEG	71313	2.378	11.4	32.14	0.00343
	HPDM	85808	2.371	13.8	31.60	0.00163
	F5	61481	2.258	10.4	31.89	0.04543

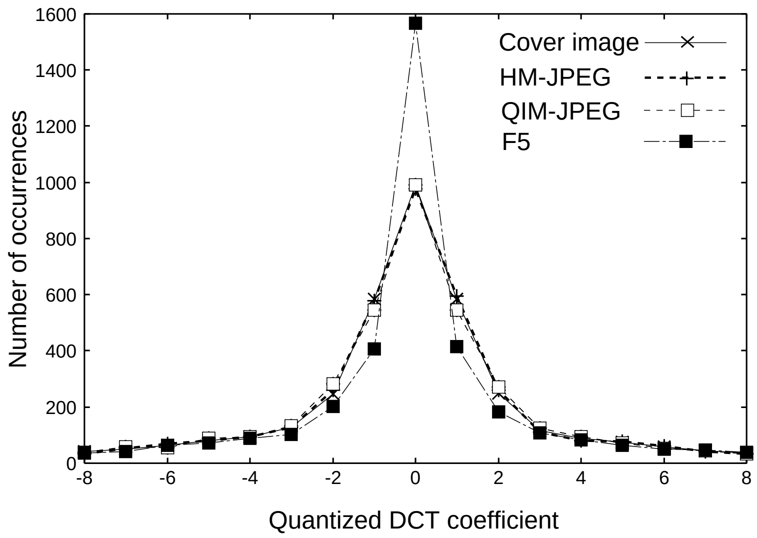

Fig. 6. Histograms of (1,1) frequency component for cover image and stego images by HM-JPEG, QIM-JPEG and F5.

Table 2
Results of embedding experiments with equal amount of embedding performed

image	method	embedded data size (bits)	compressed image (bpp)	embedding efficiency	PSNR (dB)	KL divergence
Lena	HM-JPEG	35206	1.143	1.46	37.65	0.00119
	QIM-JPEG	35304	1.155	1.53	37.81	0.00287
	HPDM	35112	1.150	2.01	37.13	0.00084
	F5	35174	0.990	1.52	37.26	0.04783
Barbara	HM-JPEG	43620	1.541	1.56	36.09	0.00115
	QIM-JPEG	43641	1.541	1.60	36.21	0.00266
	HPDM	43523	1.551	1.96	35.29	0.00119
	F5	43555	1.392	1.62	35.69	0.04117
Mandrill	HM-JPEG	61647	2.370	1.68	32.11	0.00111
	QIM-JPEG	61438	2.374	1.73	32.21	0.00281
	HPDM	61631	2.370	1.98	31.86	0.00123
	F5	61481	2.258	1.71	31.89	0.04543
Airplane	(no embedding)	-	1.162	-	39.35	-
	HM-JPEG	36775	1.184	1.53	38.15	0.00117
	QIM-JPEG	36782	1.190	1.58	38.43	0.00265
	HPDM	36826	1.203	1.92	37.33	0.00098
	F5	36806	1.042	1.62	37.77	0.03641
Boat	(no embedding)	-	1.250	-	38.20	-
	HM-JPEG	38703	1.282	1.53	37.26	0.00101
	QIM-JPEG	38780	1.282	1.61	37.36	0.00207
	HPDM	38747	1.300	1.89	36.36	0.00095
	F5	38763	1.139	1.62	36.73	0.03243
Goldhill	(no embedding)	-	1.451	-	36.55	-
	HM-JPEG	46694	1.468	1.55	35.72	0.00117
	QIM-JPEG	46685	1.474	1.60	35.83	0.00310
	HPDM	46787	1.471	2.01	35.30	0.00106
	F5	46698	1.311	1.59	35.34	0.05318
Peppers	(no embedding)	-	1.184	-	37.04	-
	HM-JPEG	36743	1.196	1.43	36.32	0.00117
	QIM-JPEG	36753	1.208	1.50	36.47	0.00339
	HPDM	36731	1.200	2.01	35.94	0.00083
	F5	36759	1.036	1.52	36.07	0.05654
Washsat	(no embedding)	-	1.136	-	38.12	-
	HM-JPEG	40591	1.146	1.43	37.10	0.00133
	QIM-JPEG	40649	1.155	1.50	37.27	0.00356
	HPDM	40490	1.145	2.02	36.74	0.00063
	F5	40525	0.975	1.49	36.73	0.07172
Zelda	(no embedding)	-	0.932	-	40.06	-
	HM-JPEG	31369	0.946	1.41	39.02	0.00104
	QIM-JPEG	31374	0.954	1.49	39.20	0.00251
	HPDM	31374	0.948	2.08	38.59	0.00086
	F5	31301	0.795	1.51	38.63	0.04799

5 Conclusions

We have realized high performance JPEG steganography using QIM in the DCT domain. Two methods, HM-JPEG and QIM-JPEG steganography, have been presented. HM-JPEG is a histogram preserving method based on histogram matching using two quantizers with a dead zone and QIM-JPEG is a histogram quasi-preserving method using QIM in a straightforward way with a dead zone. In comparison with F5, the two methods show high performance with regard to embedding rate, PSNR of stego image, and particularly histogram preservation. In comparison with HPDM, the two methods produce high PSNR stego images though accompanied with low embedding efficiency. In terms of quality of stego image QIM-JPEG seems to be superior to HM-JPEG, though in terms of histogram preservation HM-JPEG outperforms QIM-JPEG. The proposed methods are promising candidates for high performance as well as secure JPEG steganography against histogram-based steganalysis.

An important distinctive feature in the proposed methods is that embedding is carried out just during quantization of DCT coefficients, i.e., embedding is not carried out by modifying already quantized coefficients. This allows us to incorporate side information (Costa 1983), (Eggers and Girod 2002) from an uncompressed cover image in the form of pre-rounded coefficient values. Incorporating such side information is an important issue for further research.

6 Acknowledgement

This work was partly supported by the International Communications Foundation, Japan.

References

[Chen and Wornell 2001] Chen, B., Wornell, G.W., 2001. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Trans. on Information Theory 47, 1423-1443.

[Costa 1983] Costa, M.H.M., 1983. Writing on dirty paper. IEEE Trans. on Information Theory 29, 439-441.

[Eggers and Girod 2002] Eggers, J.J., Girod, B., 2002. Informed Watermarking. Kluwer.

[Eggers et al. 2002] Eggers, J.J., Bauml R., Girod, B., 2002. A communications approach to image steganography. Proc. of SPIE 4675, 26-37.

[Fridrich et al. 2003] Fridrich, J., Goljan M., Hogea, D., 2003. New methodology for breaking steganographic techniques for JPEGs. Proc. of SPIE 5020, 143-155.

[Provos 2001] Provos, N., 2001. Defending against statistical steganalysis. 10th USENIX Security Symposium.

[Upham 1997] Upham, D., 1997. http://ftp.funet.fi/pub/crypt/cypherpunks/steganography/jsteg/

[Westfeld 2001] Westfeld, A., 2001. F5 - A steganographic algorithm: high capacity despite better steganalysis. Lecture Notes in Computer Science 2137, 289-302.

[Yagi et al. 2001] Yagi, N., et al., 2001. Practical Digital Image Processing Program by C Language V1.1. Ohmsha.