IEICE TRANSACTIONS

on Information and Systems

VOL. E98-D NO. 7
JULY 2015

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY
The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Method of Spread Spectrum Watermarking Using Quantization Index Modulation for Cropped Images

Takahiro YAMAMOTO†, ***Student Member and*** **Masaki KAWAMURA**†a), ***Senior Member***

SUMMARY We propose a method of spread spectrum digital watermarking with quantization index modulation (QIM) and evaluate the method on the basis of IHC evaluation criteria. The spread spectrum technique can make watermarks robust by using spread codes. Since watermarks can have redundancy, messages can be decoded from a degraded stego-image. Under IHC evaluation criteria, it is necessary to decode the messages without the original image. To do so, we propose a method in which watermarks are generated by using the spread spectrum technique and are embedded by QIM. QIM is an embedding method that can decode without an original image. The IHC evaluation criteria include JPEG compression and cropping as attacks. JPEG compression is lossy compression. Therefore, errors occur in watermarks. Since watermarks in stego-images are out of synchronization due to cropping, the position of embedded watermarks may be unclear. Detecting this position is needed while decoding. Therefore, both error correction and synchronization are required for digital watermarking methods. As countermeasures against cropping, the original image is divided into segments to embed watermarks. Moreover, each segment is divided into 8×8 pixel blocks. A watermark is embedded into a DCT coefficient in a block by QIM. To synchronize in decoding, the proposed method uses the correlation between watermarks and spread codes. After synchronization, watermarks are extracted by QIM, and then, messages are estimated from the watermarks. The proposed method was evaluated on the basis of the IHC evaluation criteria. The PSNR had to be higher than 30 dB. Ten 1920×1080 rectangular regions were cropped from each stego-image, and 200-bit messages were decoded from these regions. Their BERs were calculated to assess the tolerance. As a result, the BERs were less than 1.0%, and the average PSNR was 46.70 dB. Therefore, our method achieved a high image quality when using the IHC evaluation criteria. In addition, the proposed method was also evaluated by using StirMark 4.0. As a result, we found that our method has robustness for not only JPEG compression and cropping but also additional noise and Gaussian filtering. Moreover, the method has an advantage in that detection time is small since the synchronization is processed in 8×8 pixel blocks.

key words: *digital watermarking, spread spectrum technique, quantization index modulation, cropping, JPEG compression*

1. Introduction

Digital watermarking is an information hiding technique that embeds information into digital content. It can protect and increase the value of digital content. The content may be attacked by lossy compression, clipping, geometric transform, and so on. Therefore, the robustness of digital watermarking methods is evaluated against these attacks. StirMark [1]–[3] is one of the more popular attacking tools. A lot of watermarking methods have been evaluated by using StirMark and have been compared with each other. However, since StirMark is an attacking tool, it cannot give us the best way to develop methods. The IHC Committee [4] in IEICE proposed the information hiding criteria (IHC), which provide a specific guideline for developing watermarking methods.

IHC evaluation criteria ver. 2.0 is defined by the IHC Committee [4]. JPEG compression and cropping are included as attacks in the criteria. Since JPEG compression is lossy compression, watermarks are also damaged. There are some errors in the estimated watermarks. Moreover, when a stego-image is cropped, the watermarks in the image are desynchronized. That is, the positions embedding the watermarks may be unclear. It is necessary for watermarks to be synchronized while decoding. Thus, both an error correcting capability and synchronization process are required for watermarking methods. Suhail and Obaidat [5] proposed a method in which an image is segmented on the basis of a Voronoi diagram, but the method cannot decode from a cropped image.

Against a cropping attack, some synchronization codes or markers such as templates are embedded with watermarks in order to synchronize them [6]–[8]. Using the synchronization codes, the embedding position can be easily detected from the cropped image. However, the codes must be embedded into the image in addition to watermarks. Since the total number of embedding bits is increased, image quality is degraded. Hakka *et al.* [7] proposed a DCT-OFDM based method, which could achieve high-quality images in the IHC. Since both watermarks and synchronization codes are embedded into one block in the DCT domain, the size of the block cannot be smaller. Therefore, it takes a lot of time to synchronize blocks. If we could detect the embedding position without synchronization codes, the image quality would become better.

The spread spectrum technique is used for digital watermarking [9]–[11]. Watermarks to be embedded into content are generated from messages and spread codes. Since the messages are spread by using the codes, the watermarks have an error correcting capability. In existing spread spectrum techniques for images, the watermarks are added to the pixel values or frequency coefficients, e.g., DCT and DWT [8]–[16]. To extract the watermarks without original images, the watermarks need to be embedded strongly by using these additive embedding techniques. However, high-quality stego-images and the robustness of the watermarks

Manuscript received November 20, 2014.
Manuscript revised March 17, 2015.
Manuscript publicized April 16, 2015.

†The authors are with the Graduate School of Science and Engineering, Yamaguchi University, Yamaguchi-shi, 753–8512 Japan.
a) E-mail: m.kawamura@m.ieice.org

DOI: 10.1587/transinf.2014EDP7390

are required by the IHC evaluation criteria. These techniques cannot achieve high-quality stego-images. Therefore, these existing spread spectrum techniques are inappropriate for this purpose.

There are some embedding methods, e.g., quantization index modulation (QIM) [17]–[21], dither modulation (DM) [22], and wet paper code [23]. Wet paper code [23] is a method in which embedding positions can be selected from a binary bit sequence arbitrarily and no one needs to know the positions in decoding.

QIM is a method that quantizes coefficients, and then, the quantized coefficients stand for the indexes that correspond to the watermarks. Since this method gives little redundancy to messages, it can correct no errors. However, no original image is needed for decoding. QIM-JPEG steganography [20] is a method that has robustness for histogram attacks. It keeps the histogram of the DCT coefficients. Adaptive spread transform QIM (ST-QIM) [21] is a method that uses QIM and has robustness for JPEG compression since both the quantization step size of the QIM and projection vectors are determined by using an improved perceptual model. M-ary amplitude modulation based QIM (AM-QIM) [22] is a method that uses DM. In AM-QIM, messages are embedded into DWT coefficients. The larger the number of dither vectors, M, is, the longer the decoding time is. ST-QIM and AM-QIM embed the messages into a frequency domain for robustness against JPEG compression. However, robustness for cropping is not effectively handled in these methods.

There are many methods for converting messages into watermarks and for embedding the watermarks into an image. The spread spectrum technique can correct errors, but it requires the original image in decoding. Therefore, to decode without the original image or blind decoding, we propose a method in which watermarks are converted from messages by using the spread spectrum technique and embedded into DCT coefficients by QIM.

Our method is a method of spread spectrum digital watermarking with QIM. There are two advantages of using spread codes. One is that watermarks have an error correcting capability. The other is that it is possible to synchronize blocks by using spread codes. By using QIM, the messages are decoded without original images. By using the optimal quantization step size of QIM, high robustness can be achieved for JPEG compression. Since each 8×8 pixel block, i.e. a DCT block, is processed by JPEG compression, we choose the 8×8 pixel block as an embedding block. A one-bit watermark is embedded into one DCT block. The optimal quantization step size of QIM is equal to the value of the quantization table in the JPEG compression.

The proposed method meets the IHC evaluation criteria. Moreover, digital watermarking methods that have a lower computational complexity are preferred in terms of usefulness. To synchronize the blocks quickly, a smaller block size is better. Since the block size of the proposed method is much smaller than that of the DCT-OFDM based method [7], the proposed method can quickly synchronize when detecting an embedding position.

In summary, the existing watermarking methods [7]–[22] use either the spread spectrum technique or QIM. Our proposed method uses both techniques and thus meets the IHC evaluation criteria. That is, the proposed method can quickly synchronize without synchronization codes, decode watermarks without an original image, and has robustness for JPEG compression. Moreover, robustness against other attacks is also evaluated by StirMark [1]–[3]. In Sect. 2, an embedding process including the image segmentation and spread spectrum techniques are explained. In Sect. 3, a decoding process including the detection of the embedding position and estimating watermarks is explained. In Sect. 4, the results with both the IHC evaluation criteria and StirMark 4.0 [1]–[3] are shown. We conclude in Sect. 5.

2. Embedding Process

The embedding process is shown in Fig. 1. An original image is divided into some segments and blocks to embed watermarks. The watermarks are converted from messages by using the spread spectrum technique, and then, they are embedded into DCT coefficients in 8×8 pixel blocks.

2.1 Image Segmentation

Images are cropped by cropping. We assume that the original image is $W \times H$ pixels and the cropped image is $I_W \times I_H$ pixels. When there are enough watermarks within the cropped image, the messages can be decoded from the image. Therefore, in our method, the original image is divided into $I_W \times I_H$ pixel segments as shown in Fig. 2, and the watermarks are embedded into each segment. Areas within the frame in Fig. 2 are segments. Note that the watermarks can be decoded from different sizes of cropped images if the size is larger than the $I_W \times I_H$ pixel segment. There are $\left\lceil \frac{W}{I_W} \right\rceil \times \left\lceil \frac{H}{I_H} \right\rceil$ segments in the images, where $\lceil x \rceil$ stands for the ceiling function of x.

Watermarks are embedded in each segment. To detect

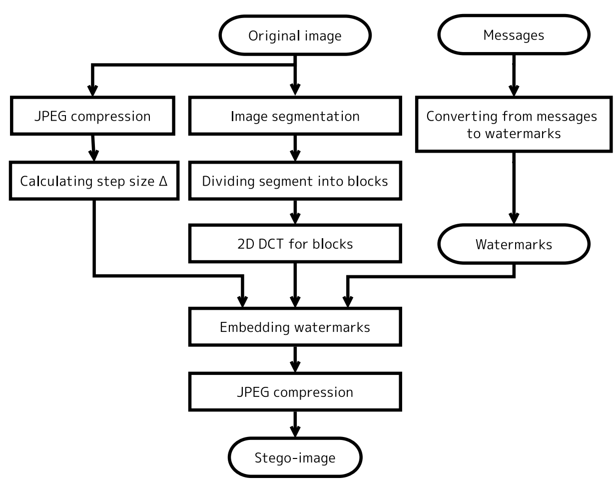

Fig. 1 Embedding process.

Fig. 2 Image segmentation: image is divided into segments. Same watermarks are embedded into each segment.

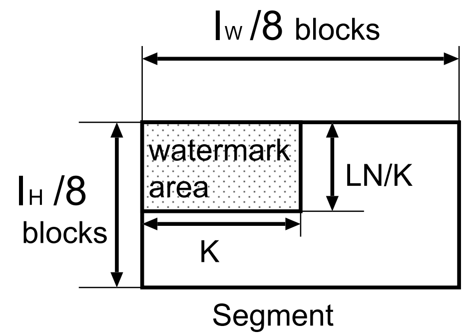

Fig. 3 Watermark area in a segment. Size of watermarks is $\left\lceil \frac{LN}{K} \right\rceil \times K$ blocks.

the watermarks within the segment size, the watermarks are arranged in a rectangular region. The size of the region is smaller than one of the segments. Consequently, watermarks are embedded into the whole image in a cyclic way. Since the same embedding process is executed for each segment, we only describe the embedding process for a segment.

2.2 Spread Spectrum Technique

In the spread spectrum technique, an L bit message $\boldsymbol{m} = (m_1, m_2, \cdots, m_L)^\top$ is spread by using spread codes, where $m_l \in \{+1, -1\}$, $l = 1, 2, \cdots, L$. Since the watermarks have redundancy, some errors can be corrected. In general, the watermarks are added into an image. Therefore, the original image is often required to decode messages in decoding [9]–[16].

A spread code $\boldsymbol{\xi}^l = (\xi_1^l, \xi_2^l, \cdots, \xi_N^l)^\top$ is generated from a maximal length sequence (M-sequence). The spread code length is N. The M-sequence is a pseudo random number consisting of 0 and 1. Since our spread code uses +1 and −1, we convert the M-sequence $(0, 1)$ into the spread code $(+1, -1)$. By using the l-th spread code for message bit m_l, the μ-th watermark bit ω_μ^l is generated by

$$\omega_\mu^l = m_l \xi_\mu^l, \quad \mu = 1, 2, \cdots, N, \tag{1}$$

where spread code $\xi_\mu^l = \pm 1$. Cox *et al.* [10] proposed a method in which watermarks are embedded into images by addition. For example,

$$\tilde{C}_\mu^l = C_\mu^l + \alpha \omega_\mu^l, \tag{2}$$

where C_μ^l is a pixel value or certain coefficient, $\tilde{C}_\mu^l$ is a modified value, and α is a weight coefficient.

2.3 Embedding Watermarks

In our method, watermark bit ω_μ^l in (1) is embedded by QIM. The segment is divided into 8×8 pixel blocks. A watermark bit is embedded into one block. From message length L and spread code length N, LN blocks are used to embed the watermarks. The watermark area shown in Fig. 3 is a rectangular region, of which the size is $\left\lceil \frac{LN}{K} \right\rceil \times K$ blocks. The width of this area is K. An LN bit watermark is sequentially embedded in this area.

Each 8×8 pixel block is transformed by using a 2D discrete cosine transform (DCT). The position of the DC component is $(0, 0)$. Since the conditions of the watermark competition were very tight, e.g., a high compression rate must be accomplished, the watermark bit is embedded at a low frequency in the DCT domain. Embedding it at intermediate and high frequencies in the DCT domain often causes decoding errors. To avoid these errors, the spread code length N must be large. However, this causes a low image quality. We should therefore embed watermarks at a low frequency. Each bit of watermark is embedded into a fixed position in the DCT domain since there is no information about embedded positions in the cropped regions. To achieve the best performance for the IHC evaluation criteria, we selected the $(1, 1)$ position in the DCT domain to embed. We explain the embedding and decoding algorithm with this position.

To decode messages without the original image, watermarks are embedded by QIM. QIM [18] is a method that quantizes DCT coefficients. The quantized coefficient is multiples of the quantization step size Δ and is indexed by using a watermark bit $\{0, 1\}$. Although no errors are corrected by QIM, no original image is necessary for decoding messages. To embed by QIM, the watermark bit $\omega_\mu^l \in \{1, -1\}$ is converted to binary $w_\mu^l \in \{0, 1\}$, which is given by

$$w_\mu^l = \frac{1}{2}\left(\omega_\mu^l + 1\right). \tag{3}$$

The embedded DCT coefficient $\tilde{C}_\mu^l$ is given by

$$\tilde{C}_\mu^l = 2\Delta \left\lfloor \frac{C_\mu^l}{2\Delta} - \frac{w_\mu^l}{2} + 0.5 \right\rfloor + \Delta w_\mu^l, \tag{4}$$

where $\lfloor x \rfloor$ stands for the floor function, which returns the largest integer not greater than x. Δ is the quantization step size. The optimal step size Δ can be chosen by using improved perceptual models [21]. We choose this value of Δ on the basis of the quantization table in the JPEG file. Both the sender and receiver share this value Δ.

The watermark embedder or sender is allowed to assume how much the stego-image is compressed by JPEG compression. Therefore, the value of the quantization table for the assumed compression ratio can be used in the embedding process. After embedding the watermarks in DCT coefficients, pixel values are calculated by using the inverse DCT. All blocks are operated in the same manner.

3. Decoding Process

Figure 4 shows the decoding process. The process consists of two processes: detecting the embedding position and estimating the messages. Due to cropping the image, the embedding position of a segment is not clear. Note that we cannot refer to the original image. To find the embedding position, additional synchronization code may be used. Since the amount of embedded bits increases, image quality is degraded. Therefore, our method does not use the synchronization code but spread codes themselves to find the position.

3.1 Detecting Embedding Position with Spread Codes

First, the embedding position of a segment should be detected. Since an arbitrary rectangular region is cropped from the stego-image, 8×8 pixel blocks are out of synchronization. We want to synchronize these blocks in the rectangular region with the blocks in the original image. There are $8 \times 8 = 64$ possible embedding positions. By calculating the correlations between the spread codes and watermark candidates for all 64 possible positions, the embedding position can be detected.

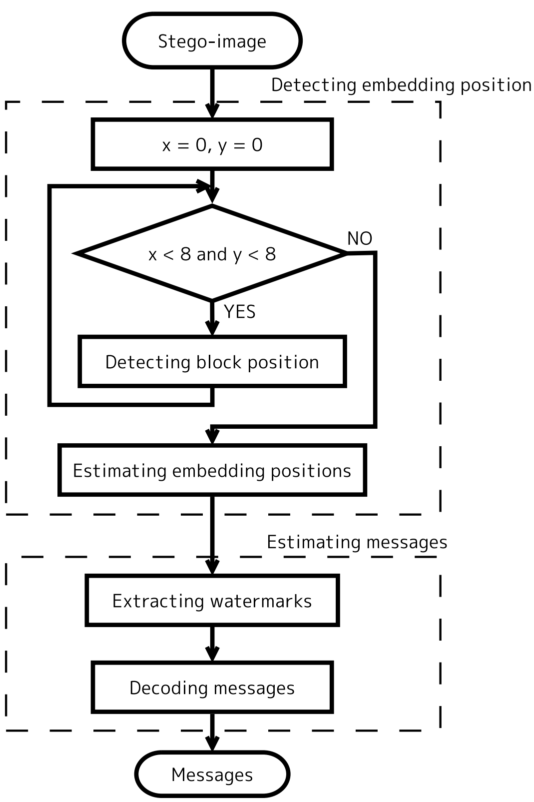

Fig. 4 Decoding process.

Since all correlations are calculated in the same manner, we describe the detection process for a certain embedding position (x, y) in a block, where $0 \leq x, y < 8$. The starting position in the region is (x, y). From this position, the rectangular region is divided into 8×8 pixel blocks. All blocks are transformed to a frequency domain by 2D DCT. We assume that the candidate of a watermark area begins from block position (r, c) in a segment, as shown in Fig. 5. The watermark area consists of $\left\lceil \frac{LN}{K} \right\rceil \times K$ blocks. Watermark candidate $\tilde{\omega}^l_\mu \in \{+1, -1\}$ is extracted from this area.

Since the watermarks are embedded into the $(1, 1)$ position in the DCT domain, the watermark candidates are also extracted from the same position. Let the value of the DCT coefficient be $\hat{C}^l_\mu$. The extracted value $\tilde{w}^l_\mu \in \{0, 1\}$ is obtained by

$$\tilde{w}^l_\mu = \left\lfloor \frac{\left|\hat{C}^l_\mu\right|}{\Delta} + 0.5 \right\rfloor \mod 2, \tag{5}$$

where Δ is the quantization step size. From the extracted value $\tilde{w}^l_\mu \in \{0, 1\}$, the watermark candidate $\tilde{\omega}^l_\mu \in \{+1, -1\}$ becomes

$$\tilde{\omega}^l_\mu = (-1)^{\tilde{w}^l_\mu}. \tag{6}$$

Let us detect the embedding position by calculating the correlations between the spread codes ξ^l_μ and watermark candidates $\tilde{\omega}^l_\mu$. The l-th correlation $R^l(r, c)$ at the block position (r, c) is given by

$$R^l(r, c) = \frac{1}{N} \sum_{\mu=1}^{N} \xi^l_\mu \tilde{\omega}^l_\mu, \; l = 1, 2, \cdots, L, \tag{7}$$

where L is the message length. All L correlations are not always necessary to detect the position. We therefore use the first 20 correlations. The correlation $R_{x,y}(r, c)$ at a starting position (x, y) is given by

$$R_{x,y}(r, c) = \sum_{l=1}^{20} \left| R^l(r, c) \right|. \tag{8}$$

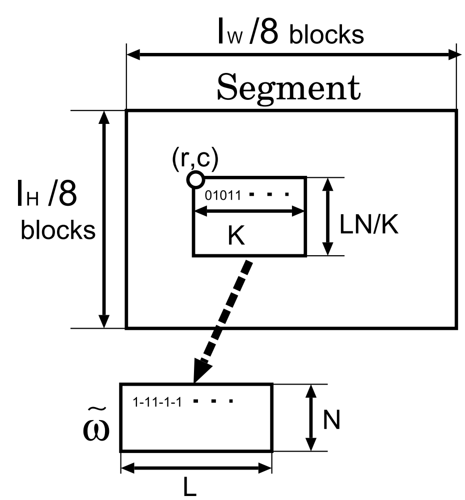

Fig. 5 Extraction of watermark candidate in a segment.

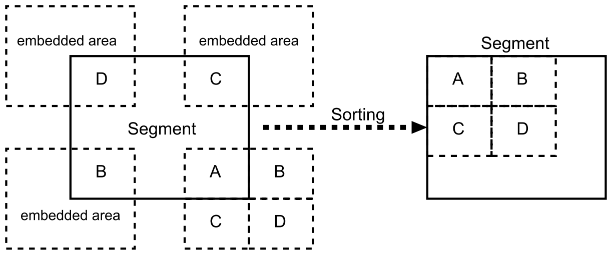

Fig. 6 Reconstructing watermark area by sorting.

When the watermark candidates $\tilde{\omega}^l_\mu$, $\mu = 1, 2, \cdots, N$ are equal to the spread codes ξ^l_μ for the l-th message, the correlation $R_{x,y}(r, c)$ takes the maximum. Therefore, estimated block position $(\hat{r}, \hat{c})$ for a starting position (x, y) is given by

$$(\hat{r}, \hat{c}) = \arg\max_{(r,c)} R_{x,y}(r, c). \tag{9}$$

For all 64 possible embedding positions, the estimated block positions $(\hat{r}, \hat{c})$ are calculated. Then, the estimated starting position $(\hat{x}, \hat{y})$ is given by

$$(\hat{x}, \hat{y}) = \arg\max_{(x,y)} R_{x,y}(\hat{r}, \hat{c}). \tag{10}$$

By using the estimated block position $(\hat{r}, \hat{c})$ and the estimated starting position $(\hat{x}, \hat{y})$, we can synchronize them and find the starting position of the watermark area.

3.2 Message Estimation

After detecting the watermark area, messages are estimated from the $\left\lceil \frac{LN}{K} \right\rceil \times K$ blocks of the watermark area. When the size of a cropped image is equal to the segment size $I_W \times I_H$ pixels, only one set of LN bit watermarks is embedded in the rectangular region. However, as shown in Fig. 6, the watermark area is usually split into four pieces. Since the areas appear in a cyclic manner, we can easily reconstruct the area by sorting.

From the estimated watermark $\hat{\omega}^l_\mu$ and spread code ξ^l_μ, the estimated L bit message $\hat{m}_l$ is given by

$$\hat{m}_l = \mathrm{sgn}\left(\frac{1}{N}\sum_{\mu=1}^{N} \xi^l_\mu \hat{\omega}^l_\mu\right), \tag{11}$$

where $\mathrm{sgn}(x)$ stands for the signum function

$$\mathrm{sgn}(x) = \begin{cases} +1 & , \ x \geq 0 \\ -1 & , \ x < 0 \end{cases}. \tag{12}$$

4. Computer Simulations

The proposed method was evaluated by using IHC evaluation criteria ver. 2.0 [4]. In this section, the criteria and results are explained.

4.1 IHC Evaluation Criteria

The test images given by the IHC Committee are shown

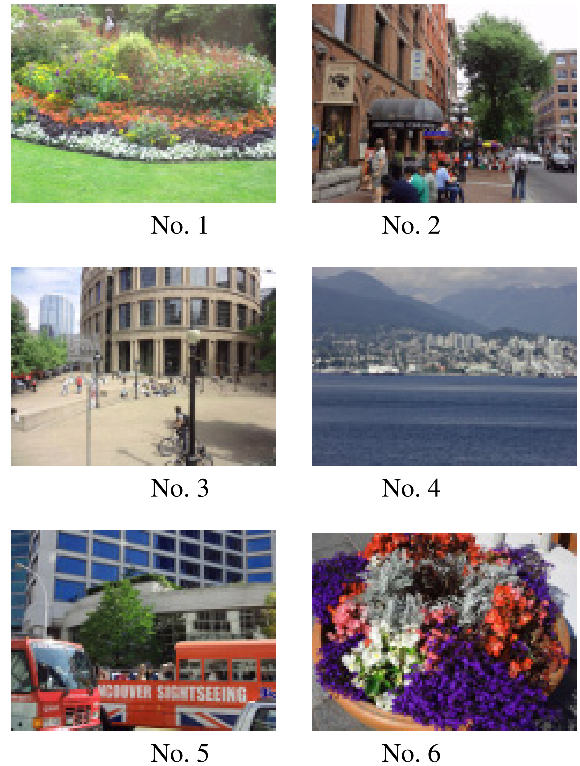

Fig. 7 IHC test images Nos. 1 ~ 6.

from No. 1 to No. 6 in Fig. 7. Each image size is 4608×3456 pixels. A message is generated by using eight ordered maximal length sequences. The message length is $L = 200$ bits. The generator polynomial is given by

$$x^8 + x^4 + x^3 + x^2 + 1, \tag{13}$$

and ten initial values are also given.

The evaluation conditions are as follows.

- Stego-images are compressed twice.
- After the first compression, the file size should be less than 1/15 the original size. After the second compression, the files size should be less than 1/30 the original size. The peak signal to noise ratio (PSNR) should be higher than 30 dB.
- Ten 1920×1080 rectangular regions should be cropped from a stego-image. A 200-bit message is decoded from each region.
- No reference information including the original image can be used in the detection.

For tolerance assessment, the correctness of estimated messages is measured by using the bit error rate (BER). The BER is given by

$$\mathrm{BER} = \frac{1 - M}{2}, \tag{14}$$

$$M = \frac{1}{L}\sum_{l=1}^{L} m_l \hat{m}_l, \tag{15}$$

where M is the overlap between estimated message $\hat{\boldsymbol{m}}$ =

Table 1 Average compression ratio, PSNR, and BER for (1, 1) position in DCT domain.

Test images	Compression ratio [%]		PSNR of Y channel [dB]		PSNR of RGB channels [dB]		BER [%]
	1st coding	2nd coding	1st coding	2nd coding	1st coding	2nd coding	
1	6.5433	3.2625	48.5754	46.6151	48.4679	46.4840	0.000
2	6.4341	3.1975	47.9776	46.5704	47.9592	46.5354	0.000
3	6.5894	3.2917	48.7741	46.9202	48.7691	46.9152	0.000
4	6.6176	3.3045	48.8731	47.5057	48.8731	47.5057	0.000
5	6.6650	3.3238	48.8412	47.9070	48.8303	47.8858	0.000
6	6.5853	3.2550	48.2497	45.1532	48.0909	44.8955	0.000
Average	6.5724	3.2725	48.5449	46.7786	48.4984	46.7036	0.000

Table 2 Average error rate (%) for ten HDTV-size regions after second decompression for (1, 1) position in DCT domain.

Test images	Position									
	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	No. 7	No. 8	No. 9	No. 10
1	0.300	0.000	0.000	0.450	0.350	0.000	0.350	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

$(\hat{m}_1, \hat{m}_2, \cdots, \hat{m}_L)^\top$, $\hat{m}_l \in \{1, -1\}$, and true message $\boldsymbol{m} = (m_1, m_2, \cdots, m_L)^\top$, $m_l \in \{1, -1\}$. For image quality assessment, the image quality of the decompressed image is measured by using the peak signal to noise ratio (PSNR). Let an original image be $\boldsymbol{f}^{\rm org} = \left(f_1^{\rm org}, f_2^{\rm org}, \cdots, f_P^{\rm org}\right)^\top$, which is compressed and decompressed twice. Let a stego-image be $\boldsymbol{f}^{\rm stego} = \left(f_1^{\rm stego}, f_2^{\rm stego}, \cdots, f_P^{\rm stego}\right)^\top$, which is also compressed and decompressed twice. The PSNR of the stego-image for the original image is given by

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right)\ [\mathrm{dB}], \tag{16}$$

where MSE is a mean square error given by

$$\mathrm{MSE} = \frac{1}{P}\sum_{i=1}^{P}\left(f_i^{\rm org} - f_i^{\rm stego}\right)^2, \tag{17}$$

where P is the total number of pixels. The PSNR of the stego-image must be higher than 30 dB.

There are two competition categories: *highest tolerance* and *highest image quality*.

- Highest tolerance
  No error occurs during detection. Those who can achieve the highest compression ratio for the ten rectangular regions in the six test images win the award for highest tolerance.
- Highest image quality
  The BER for each stego-image must be less than or equal to 1.0%. Those who can achieve the highest average PSNR for all images win the award for highest image quality.

4.2 Result of Assessment

We evaluated the proposed method by using the IHC evaluation criteria. The spread code length was $N = 18$, and the width of the watermark area was $K = 72$. In accordance with the criteria, we assumed a rectangular region was 1920×1080 pixels. The quantization step size Δ was decided by using the value of the quantization table in the JPEG file, of which value is given on the basis of the compression ratio. We chose the step size Δ in accordance with the maximum value of the $(1, 1)$ component in the quantization table for all six test images. We calculated PSNRs with both RGB values and luminance signals, Y, in the YUV.

Table 1 shows the average compression ratio, PSNR [dB], and BER [%] after the first and second compression for the $(1, 1)$ position in the DCT domain. The step size was $\Delta = 27$. The same quality factor was used for the original and stego images. For each test image, the BERs were less than 1.0%, and the PSNRs were higher than 30 dB. The maximum value of the PSNR of the Y channel was 47.91 dB for No. 5, and the minimum one was 45.15 dB for No. 6. Table 2 shows the average BER in the cropped rectangular regions after the second compression.

To show that the proposed method can embed other positions in the DCT domain, we show a case with the $(1, 0)$ position. The step size was $\Delta = 25$. Table 3 shows the average compression ratio, PSNR [dB], and BER [%] after the first and second compression for the $(1, 0)$ position in the DCT domain. For each test image, the PSNRs were higher than 30 dB. The maximum value of the PSNR of the Y channel was 48.26 dB for No. 5, and the minimum one was 45.76 dB for No. 6. Table 4 shows the average BER in the cropped regions after the second compression. The BER of region No. 4 in test image 1 was 1.7%, which exceeds 1.0% of the IHC evaluation criteria. However, the PSNRs were better than those of the $(1, 1)$ position in Table 1. Therefore, our method can embed at other positions.

The proposed method was able to detect the watermarks without desynchronization. When $(1, 1)$ position in the DCT domain is used, there were under 1% errors. There-

Table 3 Average compression ratio, PSNR, and BER for (1, 0) position in DCT domain.

Test images	Compression ratio [%]		PSNR of Y channel [dB]		PSNR of RGB channels [dB]		BER [%]
	1st coding	2nd coding	1st coding	2nd coding	1st coding	2nd coding	
1	6.6657	3.3288	49.1619	47.1702	49.0317	47.0140	0.000
2	6.4146	3.2050	48.7327	47.3613	48.7103	47.3256	0.000
3	6.5767	3.2880	49.5529	47.6879	49.5500	47.6814	0.000
4	6.6086	3.2849	49.6512	48.2153	49.6512	48.2153	0.000
5	6.6510	3.3249	49.5765	48.2641	49.5646	48.2386	0.000
6	6.5793	3.2829	48.6910	45.7613	48.5066	45.4997	0.000
Average	6.5826	3.2857	49.2277	47.4100	49.1691	47.3291	0.000

Table 4 Average error rate (%) for ten HDTV-size regions after second decompression for (1, 0) position in DCT domain.

Test images	Position									
	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	No. 7	No. 8	No. 9	No. 10
1	0.300	0.000	0.000	1.700	0.550	0.000	0.550	0.000	0.450	0.000
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.250	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.400	0.000

Table 5 Average PSNR for RGB channels of the proposed method and the DCT-OFDM based method by Hakka *et al.* [7].

Test images	PSNR of RGB channels [dB]	
	Proposed method	DCT-OFDM based method
1	46.4840	40.7333
2	46.5354	41.5566
3	46.9152	43.7490
4	47.5057	45.1215
5	47.8858	44.2024
6	44.8955	39.9132
Average	46.7036	42.5460

Table 6 Average error rate (%) for ten HDTV-size region after second decompression of the DCT-OFDM based method by Hakka *et al.* [7].

Test images	Position									
	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	No. 7	No. 8	No. 9	No. 10
1	0.050	0.050	0.050	0.000	0.000	0.000	0.000	0.150	0.000	0.050
2	0.000	0.050	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.050
3	0.050	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.100	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.100	0.050	0.000	0.000	0.150	0.100	0.150	0.100	0.150	0.100

fore, our method satisfies the condition for the category of highest image quality. These results show that our method has robustness for JPEG compression and clipping.

Let us compare the performance of the proposed method with the DCT-OFDM based method proposed by Hakka *et al.* [7]. Table 5 shows the PSNR for the RGB values of our method and the DCT-OFDM based method [7]. The PSNR with our method was about 4 dB better on average than that with the DCT-OFDM based method. Table 6 shows the BERs (%) for ten HDTV-size areas after the second decompression with our method and the DCT-OFDM based method [7]. Both methods achieved BERs of less than 1.0%. The number of zero BERs with our method was larger than that of the DCT-OFDM based method. From these results, our method had better performance than did the DCT-OFDM based method [7], which won the second watermark competition for high-quality images.

From the viewpoint of decoding times, there were $O(n^2)$ searches to synchronize, where n was the block size. The amount of DCT calculation for each search was $O(n \log n)$. For example, their method needs $256 \times 256 = 65536$ ($n = 256$) searches to detect an embedding position [7]. Since both a twelve-bit watermark and an eight-bit synchronization code are embedded in each block, their method cannot select a smaller block size like $n = 8$. On the one hand, our method needs only $8 \times 8 = 64$ times ($n = 8$). Therefore, our method is much faster than their method. It takes less than 7 seconds for one cropped region

Table 7 Average bit error rate [%] with StirMark 4.0.

Strength	BER [%]		Strength	BER [%]	
	Proposed method	AM-QIM [22]		Proposed method	AM-QIM [22]
JPEG_10	0.000	-	MEDIAN_3	0.000	1.5
JPEG_20	0.000	-	MEDIAN_5	2.530	1.5
JPEG_30	0.000	-	ROT_0.25	0.000	0.0
JPEG_40	0.000	-	ROT_0.5	0.000	0.0
JPEG_50	0.000	1.5	ROT_5	0.000	-
JPEG_60	0.000	-	ROT_45	0.000	-
JPEG_70	0.000	-	CONV_1	1.983	7.0
JPEG_80	0.000	0.0	RESC_50	0.000	-
JPEG_90	0.000	-	RESC_150	0.000	-
JPEG_100	0.000	-	RML_10	0.000	0.0
CROP_42	0.050	-	RML_40	0.000	-
CROP_50	0.000	0.0	RML_70	1.800	0.0
CROP_60	0.000	-	RML_100	0.000	0.0
CROP_70	0.000	-	NOISE_1	0.000	-
CROP_80	0.000	-	NOISE_2	0.000	-
CROP_90	0.000	-	NOISE_3	0.000	-
SS_1	0.000	-	NOISE_4	0.000	-
SS_2	0.000	-	NOISE_5	0.000	-
SS_3	0.000	-			

to synchronize. The decoding times were calculated for ten cropped regions of six IHC images with an Intel Core i7 3930K 3.20-GHz computer with 8 GB of memory. Therefore, our method has practical usefulness.

4.3 Results with StirMark 4.0

We also evaluated the proposed method by using StirMark 4.0 [1]–[3] and compared the method with ST-QIM [21] and AM-QIM [22]. The conditions of the computer simulation were changed to $L = 200$, $N = 60$, and $K = 240$. The quantization step size was $\Delta = 60$, which was the value of the $(1, 1)$ component in the quantization table for quality factor $q = 10$. These parameters were selected in such a way that PSNR was almost same as that with AM-QIM, about 38 dB. Using these parameters, we obtained a stego-image with PSNR = 38.54 dB. To decode from the distorted images, the original size and angle were restored by inverse processing. The processing was executed by using the *convert* command [24].

Table 7 shows the evaluation results of the proposed method and AM-QIM [22] with StirMark 4.0. The symbol "-" stands for no data for evaluation items in AM-QIM. According to the literature [22], the image processing tools used as attacks are JPEG compression, cropping, self similarities, median filtering, rotation, conversion filtering (Gaussian filtering), scaling, line removal, and additive noise. Since the proposed method embeds watermarks into a low frequency in the DCT domain, the results for MEDIAN_5 (median filtering) were not good. However, the BER of NOISE_1 ~ 5 (additive noise) was 0%. And also, in CONV_1 (Gaussian filtering), the BER of our method was about 1.983%, while that of AM-QIM was 7% [22]. Therefore, our method has more robustness for additive noise and Gaussian filtering than does AM-QIM. Moreover, since the BERs against many attacks except for CROP_42, MEDIAN_5, CONV_1, and RML_70 were zero, our method has a high error correcting capability.

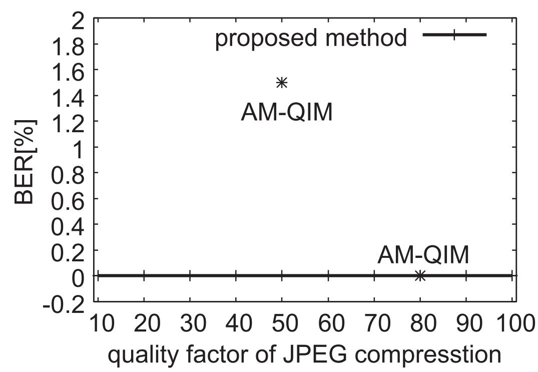

Fig. 8 Bit error rate [%] against JPEG compression with StirMark 4.0.

Figure 8 shows the average BER [%] for the quality factor of the JPEG compression from $q = 10$ to $q = 100$. The average BERs were averaged over all six IHC images and were 0.0%. The BER of ST-QIM was more than 5%, where $q \leq 50$ [21]. The BER of AM-QIM was 1.5% at $q = 50$ [22]. Even if different quality factors are used for the JPEG compression, the proposed method can decode the message correctly. Since the IHC evaluation criteria require a high compression ratio, the proposed method can correctly decode from stego-images with a larger quality factor. From these results, the proposed method had superiority in JPEG compression.

Figure 9 shows the average BER [%] for the clipping ratio from 42% to 90%. The BERs were averaged five times over six images. The clipping ratio was defined as the ratio of the rectangle size to the whole image size. In our method, since we assume that the clipping rectangle is 1920×1080 pixels (41.67%), a clipping ratio of less than 40% is out of the scope of the assumption. Therefore, Fig. 9 shows the ratio from 42%. Both ST-QIM and AM-QIM are not designed

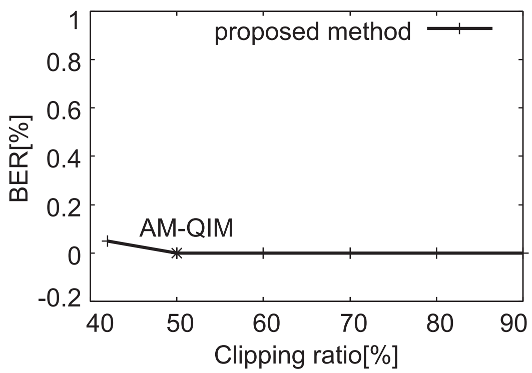

Fig. 9 Bit error rate [%] against clipping with StirMark 4.0.

for cropping attacks. There were few results for the attacks. The BER of AM-QIM was zero at a clipping ratio of 50%. As a result, our method is robust against cropping as well as AM-QIM.

5. Conclusion

In the IHC evaluation criteria, JPEG compression and image cropping are presupposed as attacks on stego-images. JPEG compression causes errors in estimating messages. To correct these errors, messages should be coded to have redundancy. The spread spectrum technique is an error correction technique. In the existing spread spectrum techniques, watermarks are added to coefficients. Under the IHC evaluation criteria, since high-quality stego-images are required, these additive embedding techniques are inappropriate for this purpose. That is, these methods damage image quality, and thus they do not fit the IHC evaluation criteria. We proposed a method in which watermarks are generated by using the spread spectrum technique, and then, they are embedded by QIM. QIM can extract the watermarks without the original image.

Due to image cropping, the embedding position of watermarks becomes unclear. When synchronization code was introduced, the amount of embedded bits increased, and image quality was degraded. Therefore, we do not use any synchronization code but spread codes themselves. To find the position by using the spread codes, correlations between the spread codes and watermark candidates are calculated. The embedding position can be estimated as the position in which the correlation is maximum. With the proposed method, since we use a small spread code length and small block size, it takes a much smaller amount of time to decode messages than do existing methods. Moreover, since no synchronization code is embedded, the total embedding bits can be smaller, and image quality can be higher.

To check that our method works fine, the proposed method was evaluated by using the IHC evaluation criteria. In these criteria, a stego-image is compressed twice. At the first compression, the size of the stego-image is less than one-fifteenth of the original image, and at the second compression, it is less than one-thirtieth. The tolerance assessment is measured by BER. The BER must be less than or equal to 1.0% in the category of highest image quality. The image quality is measured by PSNR. The PSNR of the stego-image must be higher than 30 dB. With our method, the average BER for all images was less than 1.0%, and the average PSNR was 46.70 dB. Therefore, our method achieves a high image quality in accordance with the IHC evaluation criteria. In addition, we also evaluated our method by using StirMark 4.0. The results show that our method is quite robust against not only JPEG compression and clipping but also additive noise and Gaussian filtering.

Acknowledgments

This work was supported by JSPS KAKENHI No. 25330028. The computer simulations were carried out on PC clusters at Yamaguchi University.

References

[1] StirMark benchmark (Ver.4.0), http://www.petitcolas.net/fabien/watermarking/stirmark/
[2] F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, "Attacks on copyright marking systems," Information Hiding, Lecture Notes in Computer Science, vol.1525, pp.218–238, Springer-Verlag, 1998.
[3] F.A.P. Petitcolas, "Watermarking schemes evaluation," IEEE Signal Process. Mag., vol.17, no.5, pp.58–64, 2000.
[4] Information hiding and its criteria for evaluation, http://www.ieice.org/iss/emm/ihc/
[5] M.A. Suhail and M.S. Obaidat, "Digital watermarking-based DCT and JPEG model," IEEE Trans. Instrum. Meas., vol.52, no.5, pp.1640–1647, 2003.
[6] S. Pereira, J.J.K.O. Ruanaidh, F. Deguillaume, G. Csurka, and T. Pun, "Template based recovery of Fourier-based watermarks using log-polar and log-log Maps," IEEE Int. Conf. on Multimedia Computing and Systems, ICMCS '99, vol.1, pp.870–874, 1999.
[7] M. Hakka, M. Kuribayashi, and M. Morii, "DCT-OFDM based watermarking scheme robust against clipping attack," IEICE Technical Report, vol.113, no.291, pp.107–112, 2013 (in Japanese).
[8] Y. Fang, J. Huang, S. Wu, "CDMA-based watermarking resisting to cropping," Proc. 2004 Int'l. Symposium on Circuits and Systems, ISCAS '04, vol.2, pp.25–28, 2004.
[9] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for images, audio and video," IEEE Int. Conf. Image Processing, vol.3, pp.243–246, 1996.
[10] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," IEEE Trans. Image Process., vol.6, no.12, pp.1673–1687, 1997.
[11] H.S. Malvar, D.A.F. Florencio, "Improved spread spectrum: A new modulation technique for robust watermarking," IEEE Trans. Signal Process., vol.51, no.4, pp.898–905, 2003.
[12] T. Fujita, M. Yoshida, and T. Fujiwara, "A new scheme to realize the optimum watermark detection for the additive embedding scheme with the spatial domain," IEICE Trans. Fundamentals, vol.E90-A, no.1, pp.216–225, Jan, 2007.
[13] M. Kuribayashi and M. Morii, "Iterative detection method for CDMA-based fingerprinting scheme," Information Hiding, Lecture Notes in Computer Science, vol.5284, pp.357–371, Springer-Verlag, 2008.
[14] K. Senda and M. Kawamura, "Statistical-mechanical approach for multiple watermarks using spectrum spreading," Information Theoretic Security, Lecture Notes in Computer Science, vol.5973, pp.231–247, Springer-Verlag, 2010.
[15] N. Teranishi and M. Kawamura, "Asynchronous stochastic decoder

for spread spectrum digital watermarking," Proc. Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), pp.49–52, 2011.

[16] N. Teranishi and M. Kawamura, "Performance evaluation for decoder with asynchronous stochastic algorithms and temperature schedules for spread spectrum model," IEICE Trans. Fundamentals (Japanese Edition), vol.J96-A, no.7, pp.452–461, July 2013.

[17] M.H.M. Costa, "Writing on dirty paper," IEEE Trans. Inf. Theory, vol.29, no.3, pp.439–441, 1983.

[18] B. Chen and G.W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," IEEE Trans. Inf. Theory, vol.47, no.4, pp.1423–1443, 2001.

[19] J.J. Eggers, R. Bäuml, R. Tzscchoppe, and B. Girod, "Scalar costa scheme for information embedding," IEEE Trans. Signal Process., vol.51, no.4, pp.1003–1019, 2003.

[20] H. Noda, M. Niimi, and E. Kawaguchi, "High-performance JPEG steganography using quantization index modulation in DCT domain," Pattern Recognition Lett., vol.27, no.5, pp.455–461, 2006.

[21] Y. Jiang, Y. Zhang, W. Pei, and K. Wang, "Adaptive spread transform QIM watermarking algorithm based on improved perceptual models," AEU - International Journal of Electronics and Communications, vol.67, no.8, pp.690–696, 2013.

[22] A. Phadikar, "Multibit quantization index modulation: A high-rate robust data-hiding method," Journal of King Saud University - Computer and Information Sciences, vol.25, no.2, pp.163–171, 2013.

[23] J. Fridrich, M. Goljan, P. Lisoněk, and D. Soukal, "Writing on wet paper," IEEE Trans. Signal Process., vol.53, no.10, pp.3923–3935, 2005.

[24] ImageMagick, http://www.imagemagick.org/

Takahiro Yamamoto received a B.S. degree from Yamaguchi University in 2013. Currently, he is a master's degree student at the Graduate School of Science and Engineering, Yamaguchi University. His research interests include digital watermarking.

Masaki Kawamura received B.E., M.E., and Ph.D. degrees from the University of Tsukuba in 1994, 1996, and 1999, respectively. He joined Yamaguchi University as a research associate in 1999. Currently, he is an associate professor at the same institute. His research interests include associative memory models and information hiding. He is a senior member of IEICE and a member of JNNS, JPS, and IEEE.