Watermarking Method using Concatenated Code for Scaling and Rotation Attacks

Nobuhiro Hirata and Masaki Kawamura

Graduate School of Science and Engineering, Yamaguchi University
1677-1 Yoshida, Yamaguchi-shi, Yamaguchi 753-8512 Japan
kawamura@sci.yamaguchi-u.ac.jp

Abstract. We proposed a watermarking method using a concatenated code and evaluated the method on the basis of IHC evaluation criteria. The criteria include JPEG compression, clipping, scaling, and rotation as attacks. For the robustness of messages, we introduced concatenated code, since it has a high error corrective ability to decode messages against JPEG compression. When a region is cropped from a stego-image, the position of watermarks might be unclear. Therefore, markers or synchronization codes were embedded into the stego-image. Since scaling causes pixel loss, and rotation causes distortion, watermarks were embedded into minified images. Quantization index modulation was used for embedding and extracting the watermarks without the original images. As a result, our method was evaluated on the basis of highest image quality and could achieve an average peak signal-to-noise ratio of 36.250 dB. Moreover, our method was evaluated on the basis of highest tolerance and could achieve an average compression ratio of 2.633% without errors.

Keywords: digital watermarking, concatenated code, BCH code, LDPC code, information hiding criteria

1 Introduction

Digital watermarking techniques are techniques for embedding marks into digital contents such as still images, movies, and music. An image embedded with watermarks is called a stego-image. The stego-image may be degraded by compression, format conversion, clipping, scaling, or rotation. However, it should still be possible to decode watermarks from such a degraded image. To do so, one should embed watermarks strongly or use error correcting codes. However, the image quality of the stego-image should be preserved as much as possible. When watermarks are strongly embedded, visual effects may be worse. When using error correcting codes, the codeword length of embedded information increases in bit-length, possibly resulting in image quality degradation. In other words, there is trade-off between image quality and robustness for watermarks.

Tolerance and image quality assessments are defined by the Information Hiding Criteria (IHC) [1] committee. These criteria define that image quality is measured by peak signal-to-noise ratio (PSNR) and PSNR should be over 30 dB.

The criteria also define attacks on stego-images. The attacks are performed using JPEG compression, clipping, scaling, and rotation. Due to JPEG compression, the watermarks are also damaged. In cropped stego-images, the watermarks are desynchronized. That is, the positions of watermarks become unclear. Geometric attacks such as scaling and rotation may make watermarks undetectable since coordinate axes are changed.

A method using both low density parity check (LDPC) [2–4] and repetition codes against JPEG compression and clipping was proposed [5]. The LDPC code could encode messages by using a low density parity check matrix, and had great capabilities for correcting errors. In their method [5], a watermark was generated from a message by using LDPC code, and it was repeatedly embedded into an image. To decode a message from the distorted image, errors of extracted watermarks could be roughly corrected by majority voting. Moreover, by using LDPC code, almost all errors could be corrected. Due to tolerance against JPEG compression, watermarks were embedded into 2D discrete cosine transform (DCT) coefficients. The watermarks were embedded by using Quantization Index Modulation (QIM)[6], which could extract watermarks without access to an original image. In order to synchronize watermarks, markers or synchronization codes were also embedded into the image.

On the basis of their method [5], we propose a method that has not only JPEG compression and clipping but also tolerance against scaling and rotation. Some pixels are lost in a minified image. If a part of a watermark is on lost pixels, the watermark cannot be extracted correctly. When a stego-image is magnified, errors in extracted watermarks are small. Therefore, we propose a method in which the original image is minified in advance before embedding the watermarks. We call this process pre-reduction.

Rotation of a stego-image causes pixels to become unaligned. Therefore, pixel values change. Since the chance of an alignment error at a nearby rotation center is smaller than one far from the center, a smaller image is better for watermarking. Therefore, pre-reduction is effective. Moreover, we also introduce concatenated code [7, 8] since a lot of errors are induced by scaling and rotation attacks. Concatenated code has great capabilities for correcting errors by using two different error correcting codes. They are used in communication channels. In European digital terrestrial broadcasts [8], a concatenated code with BCH [9, 10] and LDPC codes is in practical use. The BCH code is robust over random errors and can correct within a given number of errors. The LDPC code is a stochastic code and can roughly correct a large number of errors. Therefore, many errors are reduced to a few errors by LDPC code, and then the residual errors are corrected by BCH code.

Let us define the terminologies used in this paper. The meaning of the word 'watermark' in the IHC evaluation criteria [1] includes both the message and the embedded information. The message is the information to be sent, and the embedded information is the encoded message. In this paper, since we use error correcting codes, we distinguish between the message and the embedded information. Moreover, we call the embedded information a watermark. There-

fore, message length, i.e., the amount of watermark information described in the IHC, is 200 bits. Note that we use column vector notation for the watermark and codewords instead of row vector notation, which is usually used in code theory.

This paper is organized as follows. Section 2 describes the proposed method. Results from computer simulations are described in Section 3. We conclude the paper in Section 4.

2 Proposed Method

In the IHC evaluation criteria, the size of an original image is 4608×3456 pixels. Message length, i.e., the amount of watermark information, is 200 bits. Attacks on stego-images are performed using JPEG compression, clipping, scaling, and rotation. Ten HDTV-size areas, i.e., 1920×1080 pixels, are cropped from each stego-image, and then the original message is decoded from the clipping rectangle. The scaling factor and rotation angle are known in the current criteria. Therefore, we will decode messages from restored images by using inverse transformation.

2.1 Embedding Process

Figure 1 shows the encoding and embedding processes. In the beginning, the original image is minified in advance before embedding the watermarks due to tolerance against scaling and rotation. Since the IHC evaluation criteria assumes that scaling ratios are $70, 90, 110$, and 130%, we selected the smallest ratio 70%. Therefore, the Y component of the 70% minified YUV image is divided into 167×93 block segments as shown in Fig. 2. Moreover, each segment is divided into 8×8 pixel blocks. Each block is transformed by using a 2D discrete cosine transform (DCT). Due to JPEG compression, watermarks and markers will be embedded into a low frequency by using Quantization Index Modulation (QIM). Each bit of a watermark is embedded into a fixed position in the DCT domain since there is no information about embedded positions in the cropped regions [11]. We also selected the $(1, 1)$ position in the DCT domain for embedding.

Various synchronization codes or markers are embedded with watermarks in order to synchronize them against a clipping attack [12, 13]. The markers are embedded in the grid pattern as shown in Fig. 2. The value of each marker is one.

After embedding the markers, watermarks are embedded in the watermarked area as shown in Fig. 2. Due to different attacks, watermarks are extracted with errors. Therefore, to decode messages from the extracted watermarks, we introduce a concatenated code with BCH [9, 10] and LDPC [8] codes. As shown in Fig. 3, a message is encoded by BCH code in the outer encoder, and then the encoded message, i.e., the outer codeword, is encoded by LDPC code in the inner encoder. We obtain an inner codeword as a watermark.

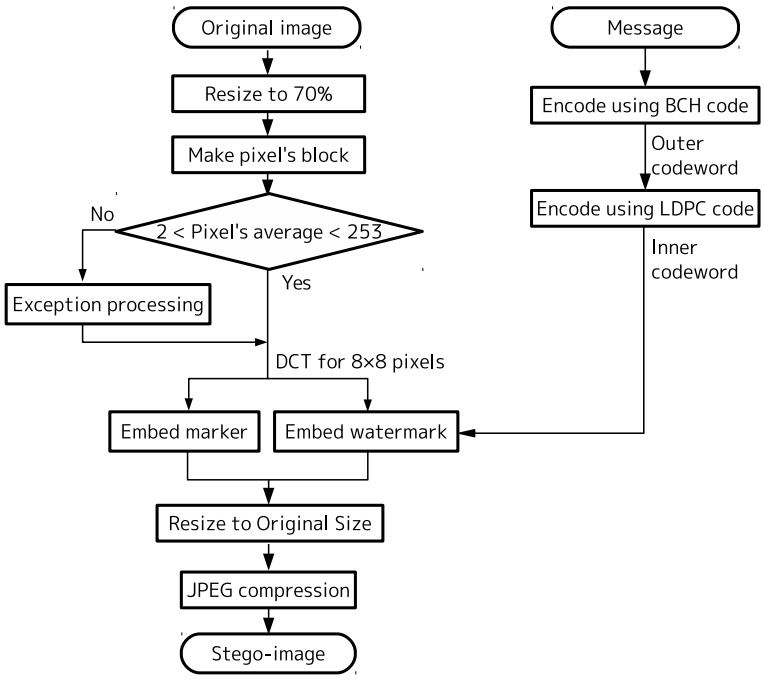

Fig. 1. Encoding and embedding process. Message is encoded by concatenated code with BCH and LDPC codes. Markers and watermarks are embedded into DCT domain of minified image. Image is rescaled to original size and is compressed by JPEG compression.

For more detail, by BCH code, a K bit message $\boldsymbol{\xi}$ is encoded to an outer codeword,

$$\boldsymbol{c}^{\mathrm{out}} = \left(\boldsymbol{\xi}^{\top} \ \left(\boldsymbol{p}^{\mathrm{BCH}}\right)^{\top}\right)^{\top}, \tag{1}$$

where $\boldsymbol{p}^{\mathrm{BCH}}$ is a parity bit in BCH code, $\xi_i \in \{0,1\}, i = 1,2,\cdots,K$, $c_j^{\mathrm{out}} \in \{0,1\}, j = 1,2,\cdots,N_{\mathrm{BCH}}$, $p_j^{\mathrm{BCH}} \in \{0,1\}, j = 1,2,\cdots,N_{\mathrm{BCH}} - K$, and N_{BCH} is the codeword length of $\boldsymbol{c}^{\mathrm{out}}$. By LDPC code, the outer codeword $\boldsymbol{c}^{\mathrm{out}}$ is encoded to an inner codeword,

$$\boldsymbol{c}^{\mathrm{in}} = \left(\left(\boldsymbol{c}^{\mathrm{out}}\right)^{\top} \left(\boldsymbol{p}^{\mathrm{LDPC}}\right)^{\top}\right)^{\top}, \tag{2}$$

where $\boldsymbol{p}^{\mathrm{LDPC}}$ is a parity bit in LDPC code, $c_k^{\mathrm{in}} \in \{0,1\}, k = 1,2,\cdots,N_{\mathrm{LDPC}}$, $p_k^{\mathrm{LDPC}} \in \{0,1\}, k = 1,2,\cdots,N_{\mathrm{LDPC}} - N_{\mathrm{BCH}}$, and N_{LDPC} is the codeword length of $\boldsymbol{c}^{\mathrm{in}}$. The watermark $\boldsymbol{w}$ consists of the inner codeword $\boldsymbol{c}^{\mathrm{in}}$ and check bit $\boldsymbol{s}$, that is, $\boldsymbol{w} = \left[\boldsymbol{s}^{\top} \ \left(\boldsymbol{c}^{\mathrm{in}}\right)^{\top}\right]^{\top}$. The check bit $\boldsymbol{s}$ is used for measuring errors in

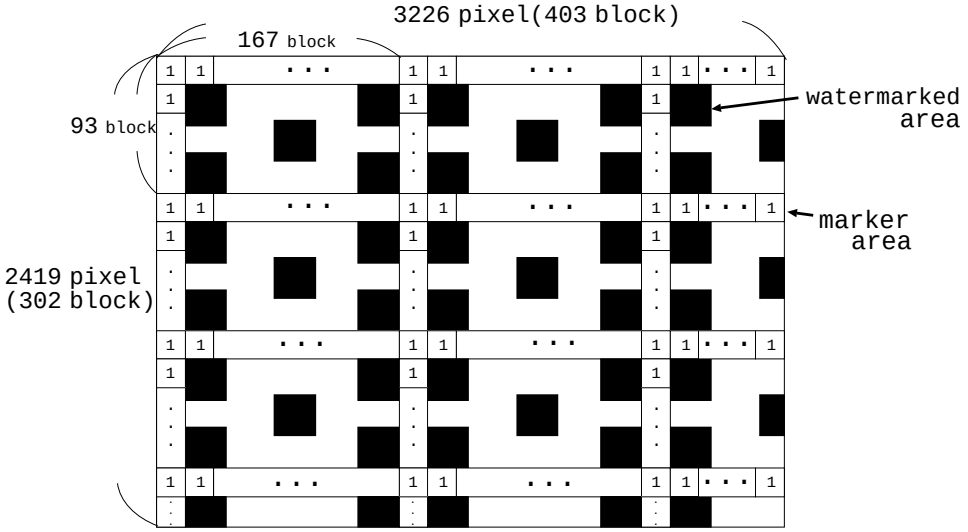

Fig. 2. Layout of watermarked and marker areas in segment within minified image. Segment consists of 167 × 93 blocks. There are five watermarked areas (black) in the segment. Top and right side blocks are marker area. Value of each marker is one. Parts of watermarks are embedded in segments furthest left or at bottom.

extracted watermarks during the decoding process. Let the length of the check bit be B bits. Therefore, $B+N_{\mathrm{LDPC}}$ bit watermark is embedded to watermarked areas. There are five watermarked areas in a segment, as shown in Fig. 2. Each watermarked area is a square of length ℓ on a side, where

$$\ell = \left\lceil \sqrt{B + N_{\mathrm{LDPC}}} \right\rceil, \tag{3}$$

where $\lceil x \rceil$ stands for the ceiling function, which returns the smallest integer greater than x.

QIM is used for embedding watermarks and markers. When a bit of a watermark or marker, $w \in \{0, 1\}$, is embedded, the modified DCT coefficient C' is given by

$$C' = 2\Delta \left(\left\lfloor \frac{C}{2\Delta} - \frac{w}{2} + 0.5 \right\rfloor + \frac{w}{2} \right), \tag{4}$$

where C is the original DCT coefficient and Δ is the quantization step size. The size Δ is shared by both the encoder and decoder. $\lfloor x \rfloor$ stands for the floor function, which returns the largest integer not greater than x.

We note that there is an exception. When the pixel value is near 255, i.e., it is colored white, the pixel values after embedding watermarks or markers might be over 255. Therefore, we introduce an exception in processing for the following

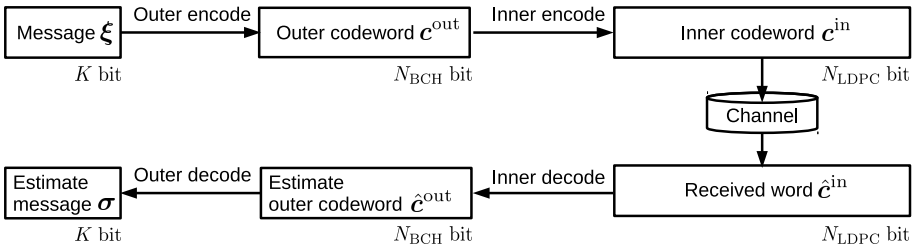

Fig. 3. Flowchart of encoding and decoding process with concatenated code. Message is encoded by both BCH code in outer encoder and LDPC code in inner encoder. The received word is decoded by both LDPC and BCH decoder. Estimated message is decoded from outer codeword.

condition. Let a pixel value in a block be $P_{ij}, i = 0, 1, \cdots, 7, j = 0, 1, \cdots, 7$. When the average over the pixel values in a block,

$$\text{AVE} = \frac{1}{8 \times 8} \sum_{i=0}^{7} \sum_{j=0}^{7} P_{ij}, \tag{5}$$

is near 255 or 0, all pixel values in the block are modified to

$$\tilde{P}_{ij} = \begin{cases} P_{ij} - 5 \text{ , AVE} \geq 253 \\ P_{ij} + 5 \text{ , AVE} \leq 2 \\ P_{ij} \quad \text{ , others} \end{cases} . \tag{6}$$

2.2 Extraction Process

Figure 4 shows the decoding process from a 1920×1080 pixel cropped image. First, the cropped image is resized to 70% of its size. There are 8×8 candidates for watermark areas. The resized image is divided into 8×8 pixel blocks. All blocks are transformed to a frequency domain by 2D DCT. To synchronize the area, embedded markers are detected from the resized image. Since the value of all the markers is one, the position which gives the largest summation of marker candidates in rows and columns will be the marker position. When the marker position is detected, the blocks in the segment are swapped as shown in Fig. 5. The marker row and column are arranged to the top and left, respectively.

After synchronization, watermarks are extracted from the watermark area by QIM. Let the value of a DCT coefficient be $\widehat{C}$. The extracted value of a watermark w is given by

$$w = \left\lfloor \frac{|\widehat{C}|}{\Delta} + 0.5 \right\rfloor \mod 2. \tag{7}$$

There are five watermarked areas in a segment. From each area, $B + N_{\text{LDPC}}$

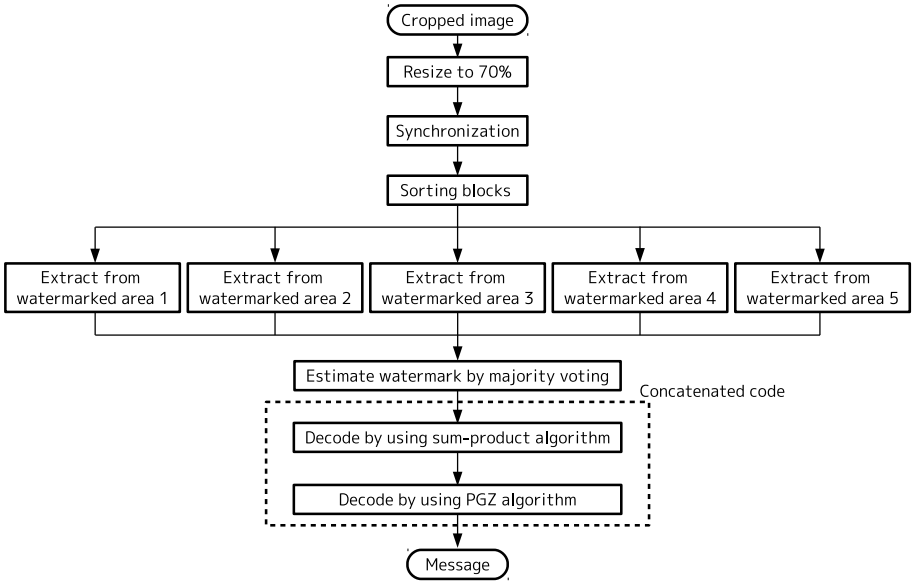

Fig. 4. Decoding process. By extracting marker candidates, marker area is detected from DCT coefficient of minified images. After synchronization, watermarks are extracted from five watermarked areas, and then estimated message is decoded from watermarks.

bit watermark $\widetilde{\boldsymbol{w}}$ is extracted. The μ-th watermark $\widetilde{\boldsymbol{w}}^{\mu} = \left((\widetilde{\boldsymbol{s}}^{\mu})^{\top} \left(\widetilde{\boldsymbol{c}}_{\mu}^{\mathrm{in}} \right)^{\top} \right)^{\top}$ consists of extracted check bit $\widetilde{\boldsymbol{s}}^{\mu}$ and extracted inner codeword $\widetilde{\boldsymbol{c}}_{\mu}^{\mathrm{in}}$. Since there are some errors in the extracted watermarks, the reliability of each watermark, α_{μ}, is calculated from B bit of check bit $\widetilde{\boldsymbol{s}}^{\mu}$. The μ-th reliability for the check bit $\widetilde{\boldsymbol{s}}^{\mu}$ is given by

$$\alpha_{\mu} = \frac{1}{B} \sum_{j=1}^{B} \widetilde{s}_{j}^{\mu}, \ \mu = 1, 2, \cdots, 5. \tag{8}$$

Using the reliability, the estimated inner codeword $\widehat{\boldsymbol{c}}^{\mathrm{in}}$ is calculated by weighted majority voting using the five watermarks $\widetilde{\boldsymbol{c}}_{\mu}^{\mathrm{in}}$. Now, due to weighted majority voting, the extracted inner codeword $\widetilde{c}_{\mu,k}^{\mathrm{in}} \in \{0, 1\}$ is converted to $\widetilde{y}_{k}^{\mu} \in \{1, -1\}, k = 1, 2, \cdots, N_{\mathrm{LDPC}}$. The estimated codeword $\widehat{y}_{k} \in \{1, -1\}$ is given by

$$\widehat{y}_{k} = \mathrm{sgn} \left(\sum_{\mu=1}^{5} \alpha_{\mu} \widetilde{y}_{k}^{\mu} \right), \tag{9}$$

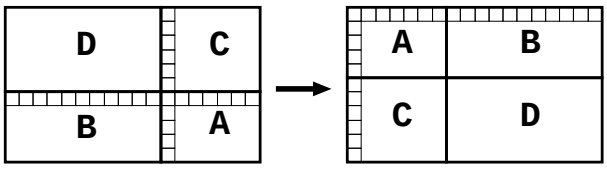

Fig. 5. Sorting blocks. Blocks in the segment are swapped in a manner such that marker row and column are arranged top and left, respectively.

where the function $\mathrm{sgn}(x)$ is defined by

$$\mathrm{sgn}(x) = \begin{cases} +1, x \geq 0 \\ -1, x < 0 \end{cases}, \tag{10}$$

and is then converted to the estimated inner codeword $\widehat{c}_k^{\mathrm{in}} \in \{0, 1\}$. The sum-product algorithm [4] is a stochastic algorithm for LDPC code, and can roughly correct a large number of errors. Some errors in the inner codeword $\widehat{\boldsymbol{c}}^{\mathrm{in}}$ are corrected by using the sum-product algorithm. The first N_{BCH} MSB of it becomes the outer codeword $\widehat{c}_j^{\mathrm{out}} \in \{0, 1\}, j = 1, 2, \cdots, N_{\mathrm{BCH}}$. Next, we apply the PGZ algorithm [14] of BCH code to completely remove residual errors. The estimated message $\sigma_i \in \{0, 1\}, i = 1, 2, \cdots, K$ is decoded from the outer codeword.

3 Computer Simulations

3.1 Contest Flow

We evaluated our proposed method with computer simulations in accordance with IHC evaluation criteria [1]. The message length is $K = 200$ bit. Ten initial values to generate messages and six test images are given by the IHC Committee. Figure 6 shows the IHC standard images. The size of each image is 4608×3456 pixels. The default evaluation procedure is summarized as follows.

1. Generated watermarks are embedded into original images.
2. For image encoding, the first JPEG compression is executed. The file size should be less than 1/15 the original size.
3. For preliminary compression, the second JPEG compression is executed. The file size should be less than 1/25 the original size. To preserve the same compression ratio, the quality factor (QF) used here is stored. The peak signal to noise ratio (PSNR) should be higher than 30 dB. Image quality is also evaluated by MSSIM [15].
4. One of the additional attacks, scaling, rotation, or their combination, is applied to the images. Scaling ratios are $s = \{70, 90, 110, 130\%\}$, rotation angular degrees are $\theta = \{3, 6, 9, 12^\circ\}$, and their combination is $(s, \theta) = \{(90, 3), (90, 9), (110, 3), (110, 9)\}$. They should be checked for the evaluation. These used parameters are known to the decoder.

Fig. 6. IHC standard images

5. The attacked images are compressed by using the same QF as used for the preliminary compression.
6. The attacked images are normalized to the original size and direction by using the parameters s and θ.
7. For each normalized image, ten 1920×1080 rectangular regions are cropped from the image.
8. Watermarks are extracted from a rectangular region, and then the $K = 200$ bit message is estimated from the watermarks. The correctness of the estimated message is measured by the bit error rate (BER).

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

- Highest tolerance
  The bit error rate for the estimated message must be BER $= 0$. Those who can achieve the highest compression ratio for the six 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%, and at worst the BER should be equal to or less than 2.0%. Those who can achieve the highest average PSNR for all images win the award for highest image quality.

3.2 Results

We describe our parameters used for the evaluations. The original message $\boldsymbol{\xi}$ is encoded to the outer codeword $\boldsymbol{c}^{\text{out}}$ by BCH code. The $\tilde{K} = 207$ bit message is

Table 1. Average compression ratio, PSNR, and MSSIM

	Compression ratio [%]		PSNR [dB]		MSSIM	
	1st coding	2nd coding	1st coding	2nd coding	1st coding	2nd coding
Average	6.535	3.967	37.854	36.250	0.954	0.933

Table 2. Average error rate for ten HDTV-size areas with additional attacks (%)

	Position									
	1	2	3	4	5	6	7	8	9	10
No attack	0	0	0	0	0	0	0	0	0	0
Scaling	0	0	0	0	0	0	0	0	0	0
Rotation	0	0	0	0	0	0	0	0	0	0
Combination	0	0	0	0	0	0	0	0	0	0
Average	0	0	0	0	0	0	0	0	0	0

encoded with the outer codeword. Since the original message length is $K = 200$ bit, 7 bits are padded with zero. The codeword length becomes $N_{\mathrm{BCH}} = 255$ bit when the minimal Hamming distance is at least $d = 6$. The outer codeword $\boldsymbol{c}^{\mathrm{out}}$ with the $N_{\mathrm{BCH}} = 255$ bit is encoded with the inner codeword $\boldsymbol{c}^{\mathrm{in}}$ by LDPC code. The inner codeword length becomes $N_{\mathrm{LDPC}} = 1012$ bit. We used a parity-check matrix with a column weight of 3 and a row weight of 4. The length of the check bit is $B = 25$ bit. Therefore, the length of a watermark is 1037 bits, and the length of the watermarked area on a side is $\ell = 33$. The watermarks are embedded into the images by QIM with step size $\Delta = 40$.

We evaluated our method on the basis of highest image quality. Table 1 shows the average compression ratio, PSNR, and MSSIM for the six IHC standard images. The values in the 1st coding stand for values after the first JPEG compression. After the second JPEG compression, the average compression ratio achieved less than 4.0%. For image quality, PSNR was 36.250 dB, which was over the criterion value of 30 dB. Also, MSSIM was 0.933. The BERs for each attack are shown in Table 2. There are ten rectangular regions. The values are the BERs for the ten regions. 'No attack' means that only JPEG compressions are executed twice. No scaling or rotation is applied. 'Scaling' or 'Rotation' mean that either scaling or rotation is applied for the additional attack. 'Combination' means that both scaling and rotation are applied. Our method could achieve zero errors for all attacks.

Next, we evaluated our method on the basis of highest tolerance. Under the conditions in which the bit error rate is BER=0 and image quality PSNR is over 30 dB, stego- images were compressed to be as small as possible. Table 3 shows the average compression ratio, image quality PSNR, and MSSIM for the highest tolerance. Note that PSNR is over 30 dB. The compression ratios for all images are 2.771% (1/35) for No. 1, 3.316% (1/30) for No. 2, 1.800% (1/55) for No. 3, 1.296% (1/75) for No. 4, 3.323% (1/30) for No. 5, and 3.294% (1/30) for No. 6. Our method has robustness for JPEG compression.

Table 3. Average compression ratio, PSNR, and MSSIM for the Highest Tolerance

	Compression ratio[%]		PSNR[dB]		MSSIM	
	1st coding	2nd coding	1st coding	2nd coding	1st coding	2nd coding
Image 1	6.608	2.771	37.523	33.304	0.961	0.907
Image 2	6.492	3.316	36.473	34.683	0.952	0.925
Image 3	6.634	1.800	38.267	34.127	0.955	0.891
Image 4	6.309	1.296	39.666	35.066	0.949	0.870
Image 5	6.554	3.323	38.553	36.805	0.953	0.931
Image 6	6.628	3.294	36.638	33.577	0.956	0.913
Average	6.538	2.633	37.853	34.594	0.954	0.906

4 Conclusion

We proposed a method which achieves a zero bit error rate against scaling, rotation, and their combination. Scaling causes pixel loss, and rotation causes distortion. Therefore, watermarks might be extracted incorrectly due to these attacks. In our method, the original images are minified to 70% of the size of the original ones in advance. Nevertheless, we introduced both concatenated code and majority voting in preparation for the occurrence of errors. For the concatenated code, BCH and LDPC codes are used as outer and inner codes, respectively. The layout for watermark and marker areas in a segment is our original, and it affects the performance of BER. As a result, our method achieved BER = 0 for all attacks, and the average PSNR was 36.250 dB when the compression ratio was 1/25.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 25330028 and was partially supported by the Cooperative Research Project, RIEC in Tohoku University. The computer simulations were carried out on PC clusters at Yamaguchi University.

References

1. Information hiding and its criteria for evaluation, IEICE, http://www.ieice.org/iss/emm/ihc/en/
2. R. G. Gallager, “Low-density parity-check codes,” IRE Trans. on Information Theory, vol. IT-8, no.1, pp. 21–28, 1962
3. D. J. C. MacKay, “Good error-correcting codes based on very Sparse matrices,” IEEE Trans. Inform. Theory, vol. 45, pp.399–431, 1999
4. T. Wadayama, “A coded nodulation scheme based on low density parity check codes,” IEICE Trans. Fundamentals, vol. E84-A, no. 10, pp. 2523–2527, 2001
5. N. Hirata, M. Kawamura, “Digital watermarking method using LDPC code for clipped image,” Proceedings of the 1st international workshop on Information hiding and its criteria for evaluation, IWIHC2014, pp. 25–30, 2014

6. B. Chen, G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," IEEE Trans. Inform. Theory, vol. 47, no. 4, pp. 1423–1443, 2001
7. G. D. Forney, Jr., "Concatenated codes," MIT Press, Cambridge, MA, 1966
8. ETSI EN 302 755, V1.3.1, Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system, 2012
9. A. Hocquenghem, "Codes correcteurs d'Erreurs," Chiffres, vol. 2, pp. 147–156, 1959
10. R. C. Bose, D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," Information and Control, vol. 3, issue 1, pp. 68–79, 1960
11. T. Yamamoto, M. Kawamura, "Method of spread spectrum watermarking using quantization index modulation for cropped images," IEICE Trans. Information and Systems, vol. E98-D, no.7, 2015
12. Y. Fang, J. Huang, S. Wu, "CDMA-based watermarking resisting to cropping," Proc. of 2004 Intl. Symposium on Circuits and Systems, ISCAS'04, vol. 2, pp. 25–28, 2004
13. M. Hakka, M. Kuribayashi, M. Morii, "DCT-OFDM based watermarking scheme robust against clipping attack, "IEICE Technical Report, vol. 113, no. 291, pp. 107–112, 2013 (in Japanese)
14. M. Srinivasan, D. V. Sarwate, "Malfunction in the Peterson-Gorenstein-Zierler decoder," IEEE Trans. Inform. Theory, vol. 40, no. 5, pp. 1649–1653, 1994
15. Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Trans. Image Processing, vol. 13, no. 4, pp. 600–612, 2004