Improved SIFT feature-based watermarking method for IHC ver. 5

Masato Hayashi and Masaki Kawamura
Yamaguchi University, Yamaguchi, Japan
E-mail: m.kawamra@m.ieice.org Tel/Fax: +81-83-933-5701

***Abstract*—The current scale-invariant feature transform (SIFT) feature-based watermarking method proposed by Kawamura and Uchida was evaluated using Information Hiding Criteria (IHC) ver. 5. The marked regions were selected around the SIFT feature points then normalized to a uniform size against scaling attack, and watermarks were embedded into these normalized regions. There are two problems with this method. One is that the marked regions are distorted by normalization. The other is that the feature points are not detectable due to embedding on the points. Therefore, we propose an improved version of this method to solve these problems. We introduce two improvements called as a gradual magnification factor for normalization and concentric square regions for marked regions. The proposed method was evaluated using IHC ver. 5, and the feature detection rate improved and the bit error rate decreased.**

I. Introduction

Robust watermarking methods are necessary for protecting digital content. With image-watermarking methods, one of the requirements for robust watermarking is that embedded watermarks should be extractable from degraded images, which are not only processed by authorized users but also tampered with by illegal attackers. The Information Hiding and its Criteria for Evaluation Committee [1] proposed evaluation standards, called Information Hiding Criteria (IHC). The criteria define three components in the watermarking model of IHC ver. 5: watermarker, attacker, and decoder. A 200-bit message is encoded into the codeword for error correction in the watermarker. A watermark is a bit sequence to be embedded and includes an encoded message or a codeword. The watermarker embeds a watermark into an original image. The image is then compressed using JPEG compression. The compressed marked image is called a stego-image. The image processing to be imposed on images is specified in the attacker. A stego-image is first processed for geometrical attacks, such as scaling and rotation attacks, then clipped to an HDTV-size area (1920 × 1080). It is then compressed again using JPEG compression to save it. The decoder has to extract the watermark from the degraded image without any information about the original image and attacks.

There are two types of attacks: geometrical and non-geometrical. Attacks such that the positions of the pixels in the image can be moved are called geometrical attacks and include rotation, scaling, and clipping. Attacks such that the pixel values in the image can be changed are called non-geometrical attacks and include lossy compression and additive noise. It is effective to embed watermarks in a transform domain against non-geometrical attacks. The discrete cosine transform (DCT) and discrete wavelet transform are usually used for embedding domains. The quantization index modulation (QIM) [2] is also effective for JPEG compression.

The *synchronization* mechanism is required against geometrical attacks due to uncertainty of the marked regions. Synchronization code and a template can be introduced to find the marked regions. The template is a predefined bit sequence and used for template matching [3], [4]. It is robust against rotation and scaling. However, the accuracies of the rotation angle and magnification ratio are not very high. The synchronization code or marker is also a predefined bit sequence to find the marked regions [5], [6] and is robust against clipping. Both the synchronization code and encoded message are embedded as a watermark. The spread code in the spectrum-spread technique can be used for the same purpose [7]. When a stego-image is subjected to geometrical attacks, synchronization is executed to find the marked regions in the attacked image. Since there are no fast searching algorithms, the brute force approach is usually carried out. Therefore, it takes a long time to synchronize.

A distorted watermark is extracted from an attacked image. Therefore, the spectrum-spread technique and error-correction codes are used for error correction of the watermark. With the spectrum-spread technique, a message is spread by the spread code. The spread message becomes a part of the watermark and is robust against errors [7], [8]. The error-correction ability of error-correction codes [9] is better than that of the spread code. Many watermarking methods use error-correction codes [5], [6], [10], [11]. Since this technique and these codes are effective only on specific attacks, the IHC could not be satisfied. Therefore, a combination of them is required.

Both the rotation angle and magnification ratio are blinded in the decoder of IHC ver. 5. Since the attacked image is also clipped, estimation of the angle and ratio is difficult. Their accuracies may be too low to extract the watermark; therefore, a feature-based watermarking method may be promising. The scale-invariant feature transform (SIFT) [12] is robust against scaling attacks, i.e., synchronization for scaling is unnecessary. Kawamura and Uchida [13] proposed a SIFT feature-based watermarking method. The method performs well on the basis of IHC ver. 5, but it does not satisfy the criteria. Therefore, we developed an improved version of this method.

The rest of this paper is organized as follows. In Section II, we summarize IHC ver. 5. In Section III, we discuss related work involving the SIFT feature-based watermarking method. In Section IV, we present our method. In Section V, we explain the computer simulations we conduced that show that our method can detect more feature points than the current method and the bit error rate (BER) can be smaller. We conclude the paper in Section VI.

II. Summary of IHC ver. 5

Since our goal is to satisfy IHC ver. 5, we summarize IHC ver. 5 before explaining watermarking methods. There are two categories of the criteria: "highest image quality (HIQ)" and "highest compression tolerance (HCT)."

IHC ver. 5 defines three components of its watermarking model: a watermarker, attacker, and decoder. The criteria for still images promote developing robust watermarking methods with a large payload. The criteria require that a message $\boldsymbol{m} = (m_1, m_2, \cdots, m_{N_m})^\top$ with a length of $N_m = 200$ bits can be decoded from a tampered stego-image. Therefore, the watermarker should make the message error correctable by encoding error-correction codes or using spread codes. The messages are generated using an M-sequence, and its initial values are defined in the IHC. Six original IHC standard 4608×3456 pixels images are provided by the IHC, and the encoded messages are embedded into the images. The generated images are compressed to JPEG to be distributed (1st compression). The file size should be less than $1/15$ of the original size. The compressed images are called stego-images.

The attacker can conduct geometrical and non-geometrical attacks. The procedure is shown in Fig. 1. Before attacking a stego-image, the Q-value of the JPEG compression is computed in advance to be less than ρ of the original size (ρ will be defined later). Next, scaling, rotation, and their combinations can be done to the images. The scaling ratios are $s \in \{80, 90, 110, 120\%\}$, and the degrees of angular rotation are $\theta \in \{3, 5, 7, 10°\}$. The combinations of scaling and rotation attacks are $(s, \theta) \in \{(80, 9), (90, 7), (110, 5), (120, 3)\}$. After geometrical attacks, a transformed image is clipped to an HDTV-sized (1920×1080-pixels) area at four specified coordinates. Each clipped area is compressed again using the same previously computed Q-value (2nd compression).

The decoder should extract a watermark from the degraded image without any information about the original image and the attacks. If a message is encoded, it can be decoded from the degraded watermark. The accuracy of a decoded message is measured using the BER between a given message $\boldsymbol{m}$ and decoded message $\hat{\boldsymbol{m}} = (\hat{m}_1, \hat{m}_2, \cdots, \hat{m}_{N_m})^\top$, which is defined by

$$\mathrm{BER} = \frac{1}{N_m} \sum_{i=1}^{N_m} m_i \oplus \hat{m}_i, \tag{1}$$

where $\oplus$ stands for exclusive OR (XOR). The image quality is measured using the peak signal-to-noise ratio (PSNR) and mean structural similarity (MSSIM) [14]. The PSNR and MSSIM are calculated for the luminance signal of the stego-image after the 1st compression. The PSNR of the stego-image should be more than 30 dB.

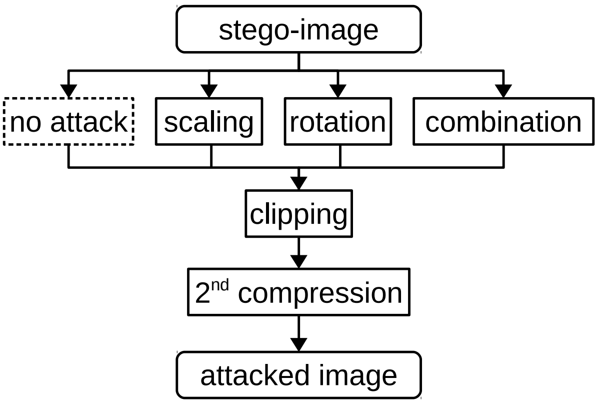

Fig. 1. Attacker model

For the HIQ category, the worst BER should be less than or equal to 2%, and the average BER should be less than 1%. After the 2nd compression, the file size should be less than $\rho = 1/25$. The method providing the highest PSNR under these conditions is superior. In the HCT category, the BERs for all decoded messages should be equal to zero, and the PSNR of the stego-images should be more than 30 dB. The method providing the smallest compression ratio ρ at the 2nd compression is superior.

III. Related Work

Kawamura and Uchida [13] proposed a SIFT feature-based watermarking method on the basis of the IHC. Watermarks are embedded into marked regions around SIFT feature points. Many SIFT feature points can be extracted from an image [12]. The advantage of a SIFT feature detector is that the scale parameter σ by the detector is almost proportional to the scaling factor. No other feature detectors, e.g., KAZE [15] and AKAZE [16], have this characteristic. Therefore, SIFT features are effective for watermarking.

A. *marked regions*

The marked regions with Kawamura and Uchida's method [13] are selected as follows. The SIFT feature detector is applied to an original image, then many SIFT feature points and corresponding scale parameters σ_i are detected, as shown in Fig. 2. Since scale parameters are too small to construct marked regions, a magnification factor d is introduced. That is, circular patches of $d\sigma_i$-pixel radius are generated then the bounding squares are selected as candidates of marked regions. Note that the bounding squares are squares of $2d\sigma_i$ pixel sides. With Kawamura and Uchida's method [13], $d = 7$ is used. Furthermore, when the image is magnified or shrunk, squares that are too small might disappear and those that are too large might overlap. Therefore, a selection of scale parameters is introduced [17]. The feature points in the range of $\sigma_L \leq \sigma \leq \sigma_U$ remain as candidates and others are removed. The values of $\sigma_L = 4$ and $\sigma_U = 10$ are used in previous studies [13], [17]. Even if the range of the parameters is restricted, the marked regions might still overlap. When the bounding squares overlap, the square that has the largest $2d\sigma_i$ remains as the marked region.

Watermarks are embedded in the marked regions, which are bounding squares of $2d\sigma^p$ pixel sides, where $p = 1, 2, \cdots, P$. We assume there are P marked regions. Each marked region is normalized to a square of $h = 96$ pixel sides in preparation for a scaling attack. From $2d\sigma^p = h$, the squares of $\sigma^p < 6.86$ pixel sides are magnified to 96×96 pixel normalized regions. After embedding the watermarks in the normalized regions, these regions are shrunk to the original size, and squares of $\sigma^p > 6.86$ pixel sides are shrunk to the normalized regions, then after embedding the watermarks, the regions are magnified to the original size.

B. message encoding and embedding

There may be many errors in extracted watermarks due to attacks. To correct such errors, the low-density parity-check (LDPC) code [9] is introduced. A message $\boldsymbol{m} = (m_1, m_2, \cdots, m_{N_m})^\top$, $m_i \in \{0, 1\}$ of length N_m bits is encoded to a codeword $\boldsymbol{M} = (M_1, M_2, \cdots, M_{N_M})^\top$ of length N_M bits. Moreover, the feature points in which no watermark is embedded may be incorrectly extracted from a degraded image due to distortion. Therefore, a check bit $\boldsymbol{c}$ is introduced. A check bit of length N_c is provided in advance and shared between the watermarker and decoder, or it can be open to the public. The $\boldsymbol{c} = (1, 1, \cdots, 1)^\top$ is then set. The check bit can be used for not only detecting the existence of watermarks but also measuring the amount of errors. Finally, the watermark $\boldsymbol{w}$ consists of codeword $\boldsymbol{M}$ and check bit $\boldsymbol{c}$ and is given by

$$w_i = \begin{cases} c_i & , \quad 1 \leq i \leq N_c \\ M_{i-N_c} & , \quad N_c < i \leq N_c + N_M \end{cases} . \tag{2}$$

A normalized region of $h = 96$ pixel sides is divided into 32×32 pixel blocks. Each block is transformed using the 2D DCT. Since there are nine blocks, an N_B-bit watermark is embedded in the DCT coefficients of each block, where N_B is given by

$$N_B = \left\lceil \frac{32 \times 32}{h \times h} (N_c + N_M) \right\rceil, \tag{3}$$

where $\lceil x \rceil$ stands for the ceiling function. With Kawamura and Uchida's method [13], a watermark of $N_B = 43$ bits is embedded in each block since the codeword length is $N_M = 300$ bits and the length of the check bit is $N_c = 87$ bits. The DCT coefficients are sorted in a line by zig-zag scan order. The watermarks are embedded from the 14-th to $(14 + N_B)$-th coefficients by using the QIM [2]. Since a stego-image will be clipped, the same watermarks are embedded in all marked regions. After embedding, nine blocks are inversely transformed and combined as a normalized stego-region. The stego-region is put back to its original size, and the entire image is compressed using JPEG compression to be less than $1/15$ of the original size (1st compression). The stego-image will then be attacked by the attacker.

C. extraction and decoding

The degraded image is received by the decoder. In the case of IHC ver. 5, since the scaling ratio is in the range of $0.8 \leq s \leq 1.2$, the SIFT feature points with the scale parameter in the range of $0.8\sigma_L \leq \sigma \leq 1.2\sigma_U$ are selected in the decoder. Note that no feature points are removed even if they overlap. All feature points in the range are candidates. We assume that there are $\hat{P}$ feature points. The normalized regions can be constructed in a similar manner of the embedding process. Since the scale parameter σ is almost proportional to the scaling ratio, the same normalized regions can be obtained.

When only a scaling attack is conducted, watermarks can be extracted from the normalized regions without any operations by using the QIM. However, when a rotation attack can be conducted, the watermarks cannot be extracted due to asynchronization by the rotation; thus, estimation of the rotation angle is required. Let a candidate for the p-th watermark rotated by a θ-degree angle be $\widetilde{\boldsymbol{w}}^p(\theta)$. It consist of a check bit and a codeword, i.e., $\widetilde{\boldsymbol{w}}^p(\theta) = \left(\widetilde{\boldsymbol{c}}^p(\theta), \widetilde{\boldsymbol{M}}^p(\theta)\right)$. Check bit $\boldsymbol{c}$ can be used for this purpose. The matching ratio for $\boldsymbol{c}$ is defined by

$$R^p(\theta) = \frac{1}{N_c} \sum_{i=1}^{N_c} c_i \oplus \widetilde{c}_i^p(\theta). \tag{4}$$

The estimated degree of the angle, $\hat{\theta}^p$, can be given by angle θ, which gives the maximum value of the matching ratio $R^p(\theta)$, i.e.,

$$\hat{\theta}^p = \arg \max_{0 \leq \theta \leq 90^\circ} R^p(\theta). \tag{5}$$

Accordingly, rotation-and-scaling synchronization is carried out. Since the estimated angle $\hat{\theta}^p$ is fixed, we abbreviate $\widetilde{\boldsymbol{w}}^p(\hat{\theta}^p)$ to $\widetilde{\boldsymbol{w}}^p$ and so on.

After synchronization, a codeword $\widehat{\boldsymbol{M}}$ is estimated from the P candidates $\widetilde{\boldsymbol{M}}^p$. There are spurious candidates in which no watermark is embedded, and the matching ratio R^p for a spurious candidate may be small. Even if there are correct candidates in which the watermark is embedded, they may be distorted by attacks. Therefore, these candidates are not used as they are. A weighted majority voting (WMV) algorithm [5], [6] is introduced to reject inappropriate candidates. The codeword $\widehat{\boldsymbol{M}}$ is estimated using the WMV algorithm, i.e.,

$$\widehat{M}_i = \Theta\left(\sum_{p=1}^{P} \alpha(R^p) \left(\widetilde{M}_i^p - 0.5\right)\right), \tag{6}$$

where the step function $\Theta(x)$ is defined by

$$\Theta(x) = \begin{cases} 1, & (x \geq 0) \\ 0, & (x < 0) \end{cases} , \tag{7}$$

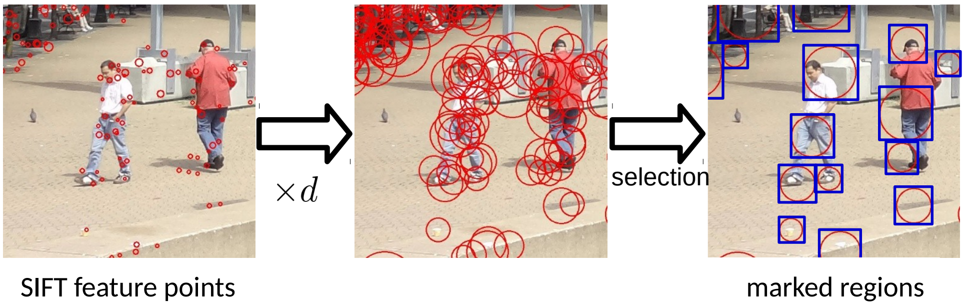

Fig. 2. Selecting SIFT feature points

and the weight function $\alpha(x)$ is defined by

$$\alpha(x) = \begin{cases} 0, & (x < T) \\ \tanh(\beta(x - T)), & (T \leq x) \end{cases}, \tag{8}$$

where T is the threshold and β is the weight coefficient. With Kawamura and Uchida's method [13], $T = 0.7$ and $\beta = 7$. The estimated message $\hat{\boldsymbol{m}} = (\hat{m}_1, \hat{m}_2, \hat{m}_3....., \hat{m}_{N_m})^\top$ can be calculated using the sum-product algorithm [19].

IV. Proposed Method

Kawamura and Uchida's method [13] uses the SIFT feature detector [12] to resist rotation and scaling attacks, then the marked regions are normalized. There are two problems with this method. (1) When a marked region is small, it is magnified to the normalized region then the region is shrunk to the original size after embedding. The watermark in the marked region is distorted by the embedding process. However, since the magnification after embedding does not affect watermarks, the distortion in the large marked region does not matter. Therefore, we introduce a gradual magnification factor d for normalization. (2) Watermarks are embedded in all nine blocks of the normalized region. That is, parts of the watermarks are embedded in the center block, which includes the SIFT feature point. Since the pixels around the feature points may end up changing, the same feature points may not be detected again in the decoder. Therefore, we introduce concentric square regions. In other words, no watermark is embedded in the center block.

By the introduction of this factor and these regions, it is expected that the extraction rate (ER) of the marked regions can be improved and smaller errors will occur. The embedding procedure is shown in Fig. 3. In this section, the difference between the proposed method and Kawamura and Uchida's [13] is described.

A. gradual magnification factor

The magnification factor d with Kawamura and Uchida's method [13] is constant, $d = 7$, as described in III. Therefore, small marked regions are still smaller than normalized ones. The shrinkage after watermarking damages the watermarks in the regions. We introduce the gradual magnification factor d to avoid this situation, which is defined by

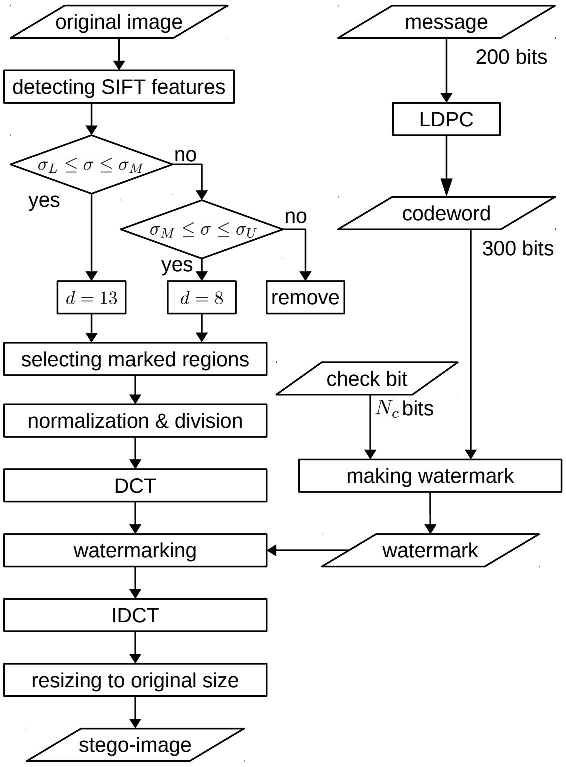

Fig. 3. Embedding procedure

$$d = \begin{cases} 13 & , \ \sigma_L \leq \sigma \leq \sigma_M \\ 8 & , \ \sigma_M \leq \sigma \leq \sigma_U \end{cases}, \tag{9}$$

where $\sigma_M = 7$. Note that σ_M must be larger than 6.86 to avoid shrinkage, as described in III. If the scale parameter σ is smaller than σ_M, a large magnification factor d will be selected. Therefore, the bounding square of $2d\sigma$ pixel sides can be larger than its normalized region of 96 pixel sides. We also select a slightly larger value, $d = 8$, for the case of $\sigma_M \leq \sigma \leq \sigma_U$. After that, watermarks are embedded in the normalized regions by using the QIM, as described in III-B.

The magnification factor $d = 13$ is large enough, but when a larger value, $d > 13$ is selected, the marked regions

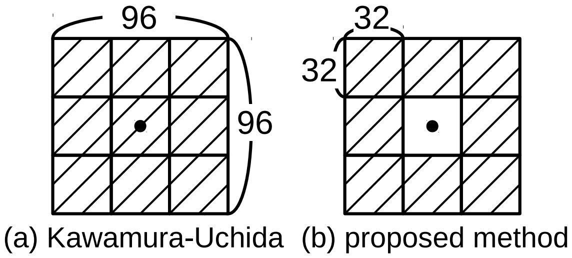

Fig. 4. Shape of embeddable blocks

would be larger and might overlap. Therefore, fewer marked regions would remain, i.e., using the large magnification factor degrades BER.

In the extraction process, two different sized marked regions should be considered. The decoder has no information about the original scale parameters σ_i and scaling ratio s by the attacker. However, the attacker keeps the IHC. Therefore, almost twice as many candidates of the marked regions are extracted as those in the embedding process.

As the scaling ratio s is assumed to be $0.8 \leq s \leq 1.2$, the SIFT feature points with the scale parameters in the range of $0.8\sigma_L \leq \sigma \leq 1.2\sigma_U$ are selected. The bounding squares of $2d\sigma_i$ pixel sides are extracted as candidate marked regions. Note that two candidates are constructed from one feature point due to the gradual magnification factor d. No candidates are removed, even if they overlap. The candidate regions are normalized to 96×96 pixels. After that, the watermarks are detected and decoded, as described in III-C.

B. concentric square-shaped regions

The 96×96 pixel normalized regions are divided into nine blocks of 32 pixel sides. Since the center block includes the SIFT feature point, embedding in the block can be avoided. We introduce concentric square regions. That is, no watermark is embedded in the center block. The shape of embeddable blocks is shown in Fig. 4.

Let us reconsider the watermark length. The length of a message $\boldsymbol{m}$ is $N_m = 200$ bits and is encoded to a codeword $\boldsymbol{M}$ of length $N_M = 300$ bits by the LDPC code, and the length of the check bit is $N_c = 87$ bits. The total length of a watermark is $N_c + N_M = 387$ bits [13]. Therefore, $N_B = 43$ bits of the watermark are embedded into each block. With the proposed method, there are only eight embeddable blocks in a normalized region. It is not advisable to reduce the codeword length since error-correction ability will decrease. On the other hand, the watermark length should not be longer due to image quality. Therefore, we reduce the length of the check bit to $N_c = 44$ bits to maintain image quality. That is, the total length of a watermark is $N_c + N_M = 344$ bits, and $N_B = 43$ bits of the watermark are embedded into each block.

V. Computer Simulations

We now show the effectiveness of the proposed method. By using a large magnification factor d, marked regions may overlap. This may result in a decrease in the number of the marked regions. However, shrinkage can be avoided after watermarking, and less degradation of the watermark is possible. Moreover, the length of the check bit is short due to the concentric square region, which may result in a decrease in the accuracy of the matching ratio (4) and affecting the accuracy of the WMV algorithm (6). On the other hand, due to the concentric square region, the SIFT feature points remain unchanged. This can improve the ER of the feature points in stego-images. Therefore, the WMV algorithm may be effective. We discuss these points by computer simulations.

We evaluated our method by using the ER, BER, and PSNR. Now let us define the ER. Since there are four clipped areas in an image, this rate is defined for each area. Therefore, we define the ER as

$$ER = \frac{P_{stego}}{P_{orig}} \times 100 \ [\%], \tag{10}$$

where P_{orig} is the number of the marked regions extracted from a clipped area of an original image. Note that the marked regions do not overlap in the watermarker, as described in III-A, and the selected marked regions remain. The notation P_{stego} is the number of candidates of the marked regions extracted from a clipped area of a stego-image. Even if they overlap, all remain. Since a stego-image degrades due to geometrical and non-geometrical attacks, the candidates with a small matching ratio, i.e., $R^p(\theta) < T$, are removed, and those with a large matching ratio are treated as marked regions. They may also be incorrect candidates; therefore, P_{stego} may be larger than P_{orig}. Since ten different messages are generated, P_{stego} is averaged over ten messages.

We consider three versions of the proposed method. The first one involves using only the concentric square regions and is called sub-method (S1). The second one involves using only the gradual magnification factor d and is called sub-method (S2). The third one is our main method and involves using both this factor and those regions. It is called the proposed method (PM). We abbreviate Kawamura and Uchida's method [13] as KU. Tables I–IV show the average ERs for these methods. As mentioned above, there are four clipped areas in an image, and P_{orig} and P_{stego} are averaged over all four clipped areas. The attack parameters, i.e., scaling ratio, rotation angle, and clipping size, are described in II. The term 'no attacks' in the tables means that the stego-image was not attacked by any geometrical attacks. There are six IHC standard images. We found that ER for S1 was larger than that for KU, whereas ER for S2 was smaller than that for KU. Furthermore, ER for PM was the largest. Therefore, the combination of both this factor and those regions is effective.

A. BER and PSNR

Next, we evaluated our method on the basis of IHC ver. 5. In accordance with the HCT category in the IHC, the BERs for three of the four clipped areas must be zero. In other words, one can be discounted. Therefore, the best three BERs were used for the evaluation. In the HIQ category, the average BERs within 1.0% can be acceptable.

TABLE I
ERs for KU [%]

	no	Scaling (%)				Rotation (°)				Combination (s, θ)			
	attacks	80	90	110	120	3	5	7	10	(80,9)	(90,7)	(110,5)	(120,3)
Image 1	48.3	16.3	24.0	29.3	31.0	24.9	26.5	24.2	22.9	7.3	11.3	18.3	21.5
Image 2	40.9	20.3	22.8	24.6	23.8	29.6	25.0	25.6	21.6	9.9	12.5	16.6	21.8
Image 3	32.6	20.2	20.5	21.8	25.6	25.2	22.2	24.6	21.5	11.6	12.6	18.0	24.2
Image 4	38.7	21.0	22.9	26.1	27.8	30.8	29.0	31.8	29.7	12.6	21.2	24.8	27.1
Image 5	37.7	20.9	22.1	25.9	26.9	27.8	26.5	24.5	21.8	12.1	14.7	20.6	22.5
Image 6	40.1	15.3	16.9	18.5	19.6	18.3	17.2	17.6	13.7	6.4	7.9	10.1	13.4

TABLE II
ERs for S1 [%]

	no	Scaling (%)				Rotation (°)				Combination (s, θ)			
	attacks	80	90	110	120	3	5	7	10	(80,9)	(90,7)	(110,5)	(120,3)
Image 1	70.0	27.3	35.2	47.9	51.9	41.5	44.5	42.5	37.5	15.4	22.2	35.4	41.5
Image 2	65.5	34.0	36.4	41.7	42.4	40.9	38.6	38.7	35.7	19.7	23.9	30.0	35.7
Image 3	57.5	31.4	34.9	46.7	47.3	47.4	39.2	39.7	39.1	24.5	27.9	38.8	48.0
Image 4	57.2	34.1	30.5	32.8	41.4	47.3	42.9	44.1	43.4	23.1	30.1	36.3	38.3
Image 5	57.1	28.8	32.2	35.1	38.9	41.1	40.8	43.1	34.6	20.6	24.7	28.0	34.9
Image 6	61.1	28.4	30.6	36.7	38.3	33.1	35.4	34.2	29.4	16.4	20.8	24.8	29.6

TABLE III
ERs for S2 [%]

	no	Scaling (%)				Rotation (°)				Combination (s, θ)			
	attacks	80	90	110	120	3	5	7	10	(80,9)	(90,7)	(110,5)	(120,3)
Image 1	21.5	18.8	19.0	19.6	20.4	20.1	19.8	19.7	19.1	18.7	18.5	17.6	21.5
Image 2	14.6	12.7	11.8	13.9	15.0	15.2	14.5	13.6	11.6	12.3	10.8	11.5	13.7
Image 3	9.1	7.9	8.2	9.2	9.7	8.7	9.5	9.8	8.1	8.7	8.5	10.5	9.6
Image 4	13.9	12.7	10.9	11.8	12.7	11.4	10.5	13.2	13.0	14.1	12.0	11.6	11.8
Image 5	11.7	10.4	8.9	8.8	10.6	14.2	12.9	13.7	10.2	9.7	10.4	10.3	12.8
Image 6	15.1	14.0	12.1	12.1	13.0	14.2	14.4	16.6	13.4	14.5	12.9	10.7	13.2

TABLE IV
ERs for PM [%]

	no	Scaling (%)				Rotation (°)				Combination (s, θ)			
	attacks	80	90	110	120	3	5	7	10	(80,9)	(90,7)	(110,5)	(120,3)
Image 1	98.2	78.6	79.7	79.8	74.6	80.9	71.6	81.1	78.4	77.4	75.6	73.1	71.2
Image 2	77.6	57.3	52.0	64.7	75.9	63.1	56.6	48.6	55.1	49.1	46.4	45.1	74.7
Image 3	61.4	47.8	43.1	58.0	65.3	53.7	53.8	57.1	54.1	43.7	42.1	58.1	60.8
Image 4	70.5	57.3	50.7	44.8	53.2	49.1	51.0	59.7	63.5	62.6	49.4	43.5	53.5
Image 5	66.7	49.7	42.3	59.9	64.1	69.3	66.5	64.8	61.4	53.0	56.0	62.0	64.9
Image 6	91.3	71.9	66.8	78.3	89.6	75.2	71.6	80.7	82.0	72.2	70.3	68.9	80.8

Tables V–VIII show the average BERs (%) for attacks. The compression ratio was less than $1/25$ of the original size for the second compression. KU [13] did not satisfy IHC ver. 5. Many of the BERs for combination attacks were over 1.0%. The BERs for S1 and S2 remained large, i.e., Neither the gradual magnification factor d nor the concentric square regions could achieve a BER of zero. However, PM could achieve BERs of zero for scaling or rotation attacks, and smaller BERs for combination attacks. Strictly speaking, PM could not yet satisfy the HCT category. However, the BERs improved and most, except in two cases, could be zero. The BERs for the worst cases were larger than 2%. Therefore, PM could not also satisfy the HIQ category.

Tables IX–XII show the results for compression ratio (CR) and image quality. The CR must be under $1/15 = 6.67\%$ for the first compression. The image quality was measured using the PSNR and MSSIM. These values were calculated for the luminance signal of the stego-image. All PSNRs should be over 30 dB. As a result, there was no negative effect on image quality with both the gradual magnification factor d and concentric square regions.

VI. Conclusions

Kawamura and Uchida's method [13] was a promising method for satisfying IHC ver. 5. To improve the ER of the SIFT feature points, we introduced the gradual magnification factor and concentric square regions. The ER for each was not so good; however, that for the proposed method using both was significantly large.

We evaluated our method on the basis of IHC ver. 5. No watermarks were embedded into the center block with concentric square regions, and the size of the marked regions became lager with the gradual magnification factor. Therefore, the image quality was almost the same as that with Kawamura

TABLE V
AVERAGE BERS FOR KU [%]

	no	Scaling (%)				Rotation (°)				Combination (s, θ)			
	attacks	80	90	110	120	3	5	7	10	(80,9)	(90,7)	(110,5)	(120,3)
Image 1	0	0	0	0	0	0	0	0	0	2.083	0.583	0.517	0
Image 2	0	0	0	0	0	0	0	0	0	1.000	0	1.167	0.517
Image 3	0	0	0.300	0	0.383	0.233	0.650	0	0	1.467	1.617	4.817	1.583
Image 4	0	0	0	0	0	0	0	0	0	1.450	2.383	1.133	1.083
Image 5	0	0	0	0	0	0	0	0	0	0.517	1.500	0	0.133
Image 6	0	0	0	0.067	0	0	0	0	0	3.167	3.983	2.667	3.667

TABLE VI
AVERAGE BERS FOR S1 [%]

	no	Scaling (%)				Rotation (°)				Combination (s, θ)			
	attacks	80	90	110	120	3	5	7	10	(80,9)	(90,7)	(110,5)	(120,3)
Image 1	0	0.027	0	0.617	0	2.533	1.417	1.300	0	5.833	4.033	4.383	3.117
Image 2	0	0	0	0	0	0	0	0	0	3.900	2.267	1.767	3.833
Image 3	0	0	0	0	0	0	0.117	0.700	0	3.300	2.183	4.783	0.550
Image 4	0	0	0	0	0	0	0	0	0	1.233	1.650	0.633	0.883
Image 5	0	00	0	0	0	0	0	0	0	1.533	1.000	1.183	0.167
Image 6	0	0.217	0	0	0.567	0	0.083	0.383	0.067	6.083	5.883	3.567	7.850

TABLE VII
AVERAGE BERS FOR S2 [%]

	no	Scaling (%)				Rotation (°)				Combination (s, θ)			
	attacks	80	90	110	120	3	5	7	10	(80,9)	(90,7)	(110,5)	(120,3)
Image 1	0	0	0	0	2.000	1.000	0	0.333	0	0.433	0.067	0.383	0.183
Image 2	0.183	0.700	0.900	2.917	3.000	0.700	1.517	3.000	1.683	2.900	2.233	2.433	4.367
Image 3	1.800	5.417	1.550	3.633	4.633	6.433	4.917	3.183	4.583	6.783	4.867	7.367	11.33
Image 4	2.750	1.150	4.500	4.367	5.583	6.283	6.933	5.733	6.317	4.117	3.700	7.000	8.450
Image 5	1.433	1.817	2.467	5.117	3.783	3.933	3.017	2.883	2.317	4.600	3.267	6.967	7.600
Image 6	0.167	0.433	0.633	3.717	2.233	1.017	0.980	0.683	0.583	0.933	1.617	3.650	4.517

TABLE VIII
AVERAGE BERS FOR PM [%]

	no	Scaling (%)				Rotation (°)				Combination (s, θ)			
	attacks	80	90	110	120	3	5	7	10	(80,9)	(90,7)	(110,5)	(120,3)
Image 1	0	0	0	0	0	0	0	0	0	0	0	0	0
Image 2	0	0	0	0	0	0	0	0	0	0	0	0	0
Image 3	0	0	0	0	0	0	0	0	0	0	0	0	0.083
Image 4	0	0	0	0	0	0	0	0	0	0	0	0	0
Image 5	0	0	0	0	0	0	0	0	0	0	0	0	0
Image 6	0	0	0	0	0	0	0	0	0	0	0	0.367	0

and Uchida's method. The BERs for only two cases were not zero. Although the average BERs were less than 1%, the worst BER was over 2%. Therefore, the proposed method could not satisfy IHC ver. 5. There was an unsuitable area to extract the SIFT feature points in some images. Since there were few feature points in this area, error correction did not work well, even when some error-correction codes and the WMV algorithm were introduced. However, most of the BERs for the proposed method could be zero.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number JP16K00156. The computer simulations were carried out on PC clusters at Yamaguchi University.

REFERENCES

[1] The Information Hiding Criteria (IHC) Committee, IEICE, http://www.ieice.org/iss/emm/ihc/index.php (accessed April 10, 2018)

[2] 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

[3] S. Pereira, T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Trans. on Image Processing*, vol. 9, no. 6, pp. 1123-1129, 2000

[4] A. Pramila, A. Keskinarkaus, T. Seppnen, "Multiple domain watermarking for print-scan and JPEG resilient data hiding," *International Workshop on Digital Watermarking*, Springer Berlin Heidelberg, pp. 279-293, 2007

[5] N. Hirata, M. Kawamura, "Watermarking method using concatenated code for scaling and rotation attacks," *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2015)*, pp. 259-270, 2015

[6] H. Hirata, T. Nozaki, M. Kawamura, "Image watermarking method satisfying IHC by using PEG LDPC code," *IEICE Trans. Inf. and Syst.*, vol. E100-D, no. 1, pp. 13-23, 2017

[7] T. Yamamoto, M. Kawamura, "Method of spread spectrum watermarking using quantization index modulation for cropped images," *IEICE Trans. Inf. & Syst.*, vol. E98-D, no. 7, pp. 1306-1315, 2015

[8] I. J. Cox, J. Kilian, F. T. Leighton, T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*,

TABLE IX
AVERAGE CRS, PSNRS, AND MSSIMS FOR KU

	CR [%]	PSNR [dB]	MSSIM
Image 1	6.616	33.6	0.927
Image 2	6.609	34.5	0.932
Image 3	6.593	36.2	0.952
Image 4	6.593	37.1	0.957
Image 5	6.529	35.4	0.936
Image 6	6.613	33.8	0.923
Average	6.592	35.1	0.938

TABLE X
AVERAGE CRS, PSNRS, AND MSSIMS FOR S1

	CR [%]	PSNR [dB]	MSSIM
Image 1	6.610	34.2	0.932
Image 2	6.604	35.0	0.937
Image 3	6.559	36.7	0.955
Image 4	6.515	37.8	0.961
Image 5	6.574	36.0	0.940
Image 6	6.613	34.4	0.930
Average	6.579	35.7	0.943

TABLE XI
AVERAGE CRS, PSNRS, AND MSSIMS FOR S2

	CR [%]	PSNR [dB]	MSSIM
Image 1	6.622	33.2	0.922
Image 2	6.630	33.3	0.918
Image 3	6.559	34.5	0.934
Image 4	6.564	35.9	0.945
Image 5	6.544	34.3	0.923
Image 6	6.595	33.0	0.916
Average	6.586	34.0	0.926

TABLE XII
AVERAGE CRS, PSNRS, AND MSSIMS FOR PM

	CR [%]	PSNR [dB]	MSSIM
Image 1	6.633	33.7	0.928
Image 2	6.598	33.8	0.923
Image 3	6.528	35.0	0.939
Image 4	6.568	36.5	0.950
Image 5	6.508	34.9	0.929
Image 6	6.639	33.4	0.922
Average	6.579	34.5	0.932

vol.6, no.12, pp.1673-1687, 1997

[9] R. G. Gallager, “Low-density parity-check codes,” *IRE Trans. on Information Theory*, vol. IT-8, no.1, pp. 21-28, 1962

[10] X. Kang, J. Huang, Y. Q. Shi, Y. Lin, “A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 776—786, 2003

[11] I. Usman, A. Khan, “BCH coding and intelligent watermark embedding: Employing both frequency and strength selection,” *Applied Soft Computing Journal*, vol. 10, issue. 1, pp. 332-343, 2010

[12] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International J. Compute Vision*, vol. 60, no. 2, pp. 91-110, 2004

[13] M. Kawamura, K. Uchida, “SIFT feature-based watermarking method aimed at achieving IHC ver.5,” *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2017)*, pp. 381-389, 2017

[14] 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

[15] P. F. Alcantarilla, A. Bartoli, A. J. Davison, “KAZE features,” *Eur. Conf. on Computer Vision (ECCV)*, 2012

[16] P. F. Alcantarilla, J. Nuevo, A. Bartoli, “Fast explicit diffusion for accelerated features in nonlinear scale spaces,” *British Machine Vision Conf. (BMVC)*, 2013

[17] Y. Yu, H. Ling, F. Zou, Z. Lu, L. Wang, “Robust localized image watermarking based on invariant regions,” *Digital Signal Processing*, vol. 22, no. 1, pp. 170-180, 2012

[18] X. Y. Hu, E. Eleftheriou, D. M. Arnold, “Regular and irregular progressive edge-growth tanner graphs,” *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 386-398, 2005

[19] T. Wadayama, “A coded modulation scheme based on low density parity check codes,” *IEICE Trans. Fundamentals*,vol. E84-A, no. 10, pp. 2523-2527, 2001