IEICE
TRANSACTIONS

on Information and Systems

DOI:10.1587/transinf.2023EDP7222

Publicized:2024/05/22

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

PAPER

Watermarking Method with Scaling Rate Estimation Using Pilot Signal

Rinka KAWANO[†a)], ***Student Member and*** **Masaki KAWAMURA**[†b)], ***Fellow***

SUMMARY Watermarking methods require robustness against various attacks. Conventional watermarking methods use error-correcting codes or spread spectrum to correct watermarking errors. Errors can also be reduced by embedding the watermark into the frequency domain and by using SIFT feature points. If the type and strength of the attack can be estimated, the errors can be further reduced. There are several types of attacks, such as scaling, rotation, and cropping, and it is necessary to aim for robustness against all of them. Focusing on the scaling tolerance of watermarks, we propose a watermarking method using SIFT feature points and DFT, and introduce a pilot signal. The proposed method estimates the scaling rate using the pilot signal in the form of a grid. When a stego-image is scaled, the grid interval of the pilot signal also changes, and the scaling rate can be estimated from the amount of change. The accuracy of estimating the scaling rate by the proposed method was evaluated in terms of the relative error of the scaling rate. The results show that the proposed method could reduce errors in the watermark by using the estimated scaling rate.

key words: *watermarking method, SIFT feature points, discrete Fourier transform, pilot signal*

1. Introduction

Watermarking is a technique for invisibly embedding other information into digital content such as still images. The embedded information is called a watermark, and the image embedded with the watermark is called a stego-image. Possible applications of watermarking include the prevention of unauthorized use of images and digital content management. Because watermarks are invisible, they are unlikely to be deleted. However, many images used in social media are processed by scaling, rotating, cropping, and other manipulations. Moreover, they are compressed when the images are saved. The watermark may be lost due to these image manipulations. Therefore, robust watermarking methods are required for various image processing techniques. Since these processes degrade the watermark, they are regarded as attacks on the watermark. The Information Hiding and its Criteria (IHC) Committee [1] has established a criteria for evaluating the robustness of watermarking methods, providing image standards and defining the type and magnitude of attacks. Our goal is to develop watermarking methods that meet such evaluation criteria.

Attacks that change the position of pixels, such as image scaling, rotation, and clipping, are called geometric attacks, while attacks that change pixel values, such as JPEG compression and noise addition, are called non-geometric attacks. Therefore, it is necessary to consider attack tolerance depending on the type of attack. First, the embedding domain is important with respect to attack tolerance. The easiest method is to embed the watermark directly into the pixel value. However, this domain is vulnerable to attacks, which cause the watermark to disappear. The alternative is to embed the watermark in the frequency domain, such as by discrete cosine transform (DCT), discrete Fourier transform (DFT), or wavelet transform [2]–[4]. In these domains, the watermark is less perceptible and robust even if the image is degraded by non-geometric attacks such as compression. In particular, the watermarking method using scale-invariant feature transform (SIFT) features [5] is robust against geometrical attacks. In this method, watermarks are embedded around SIFT feature points [6]–[9].

Even if the embedding domain is improved, the watermark can still be degraded by attacks. Thus, redundancy must be added to correct errors in the watermark, which can be done by watermarking either using error-correcting codes in the message [10] or by using a spread spectrum [11], [12]. These methods are effective against non-geometric attacks but not against geometric attacks, so countermeasures against geometric attacks must be further investigated.

Redundancy cannot improve robustness against geometric attacks because the location of the embedded watermark cannot be determined. If the type and strength of the attack could be estimated, it would be possible to extract the watermark more accurately. Thus, we introduce the communication channel model framework for such estimation. In this model, a degraded message is transmitted through a communication channel to the receiver. The strength of communication channel noise and other parameters can be estimated as the hyperparameters, and errors in the message can be further corrected. One method of estimating the communication channel is to use a pilot signal [13]. This method estimates the degradation of the communication channel by transmitting a signal, i.e., the pilot signal, which is different from the message. Here, we model the watermarking method on the communication channel. The embedding and extraction of the watermark correspond to the sender and receiver of the signal, and the attack can be regarded as a communication channel. Therefore, if a pilot signal can be introduced into the watermarking method to estimate the attack, errors in the watermark can be reduced.

Su *et al.*'s watermarking method [14] uses a pilot sig-

[†]Graduate School of Sciences and Technology for Innovation, Yamaguchi University
1677–1 Yoshida, Yamaguchi-shi, Yamaguchi 753–8512 Japan
a) E-mail: rinka1005@m.ieice.org
b) E-mail: m.kawamura@m.ieice.org

nal to detect the embedding position of the watermark. SIFT feature points are detected from the image, and the pilot signal around the points is detected. The pilot signal is not used to correct errors in the watermark. In Rodríguez *et al.* ' s method [15], the same watermarks are embedded as pilot signals at equal intervals on the image. The pilot signal is detected by calculating the autocorrelation of the watermarks. The scaling rate can then be obtained by solving an optimization problem for the distortion of the pilot signal. The watermarks can be obtained by inverting the stego-image using the estimated scaling rate. Rodríguez *et al.* [15] used 256 × 256-pixel images in their experiments. The method must compute the autocorrelation of the image-size watermark for any geometric transformations and estimate the optimal parameters. Solving this optimization problem is generally computationally time consuming, and the larger the image, the longer it takes to solve it. Our aim is to satisfy the aforementioned criteria [1]. The size of the IHC standard images is 4608×3456 pixels, so it would be computationally expensive to apply Rodriguez *et al.*'s method to these images. Their method estimates the message by inverse transforming the image using the estimated parameters, and the message is degraded by the inverse transform. Therefore, a new technique is required to estimate the type and strength of attacks. However, it is difficult to quickly and efficiently estimate an attack from the many type that exist. Thus, we focus only on scaling attacks and propose a method for estimating the scaling rate.

In this paper, the scaling rate is estimated using a grid-shaped pilot signal. Chapter 2 describes the watermarking method using SIFT feature points and DFT, and chapter 3 describes the method for estimating the scaling rate using the pilot signal. Chapter 4 presents the results of computer simulations.

2. Proposed method

Hayashi and Kawamura's blind watermarking method [7] extracts SIFT feature points [5] from the original image and performs DCT on the region around the feature points to embed watermarks. As with the method of Su *et al.* [14], this method normalizes the embedding regions during embedding so that the watermarks can be extracted even if the image is rescaled. The drawback of these methods is that the larger the difference between the embedding region and the normalized region on one side length, the more the watermark may be degraded. To overcome this drawback, the proposed method uses a DFT domain instead of a DCT domain, which is robust to geometrical transformations.

The parameters of the proposed method are set as follows with reference to IHC. The host images are IHC standard images (4608 × 3456 pixels). As we are aiming to satisfy IHC [1], the attack on the stego-image is assumed to be scaled by $0.7 - 1.3$ and cropped. The watermark estimation is performed on the scaled stego-image, which is cropped to FHD size (1920 × 1080 pixels) with the center coordinates $(x \pm 700, y \pm 500)$ at four locations. Figure 1 shows the cropping positions for each image.

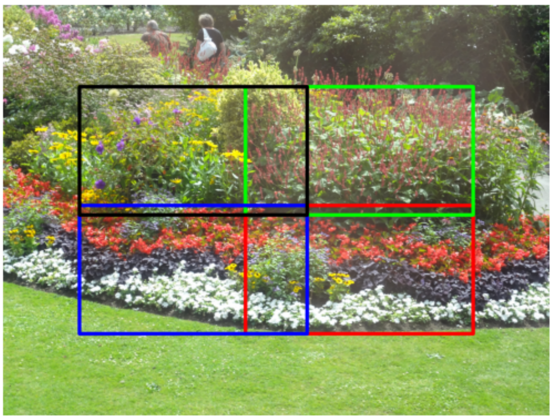


Fig. 1 Cropping positions. The four colored rectangles are the cropped images.

2.1 Properties of DFT

When performing DFT on the image $Y(x, y)$, which is $h \times h$ pixels, the DFT coefficients $F(k_1, k_2)$ are given by

$$F(k_1, k_2) = \mathcal{F}[Y(x, y)] \tag{1}$$

$$= \frac{1}{h} \sum_{x=0}^{h-1} \sum_{y=0}^{h-1} Y(x, y)\, e^{-2\pi j \frac{k_1 x + k_2 y}{h}}, \tag{2}$$

where j denotes the imaginary unit and $\mathcal{F}$ represents the discrete Fourier transform. When image Y is scaled by a scaling rate μ [12], [16], the DFT coefficients are known to satisfy the following equation,

$$\mathcal{F}[Y(\mu x, \mu y)] = \frac{1}{\mu} F\left(\frac{1}{\mu} k_1, \frac{1}{\mu} k_2\right). \tag{3}$$

Thus, the DFT coefficients of μ times magnified images can be detected at the same location as the DFT coefficients of the original image. Other transforms, such as DCT, do not have this invariance and require inverse scaling the scaled image to its original size. Normalizing the embedding area enables watermarks to be extracted without inverse scaling. However, since normalization is equivalent to a scaling attack, it may degrade the image quality and watermarks. Therefore, the proposed method embeds the watermark in the DFT region which does not require inverse scaling.

2.2 Embedding process

As shown in Figure 2, the embedding procedure of the proposed method is as follows: (1) The watermark consists of an encoded message and check bits. (2) SIFT feature points are extracted from the original image. (3) An embedding region is selected around the feature points using a scaling rate. (4) A spectrum is computed from the DFT coefficients. (5) The watermark is embedded in the amplitude spectrum. (6) A stego-image is generated by inverse DFT.

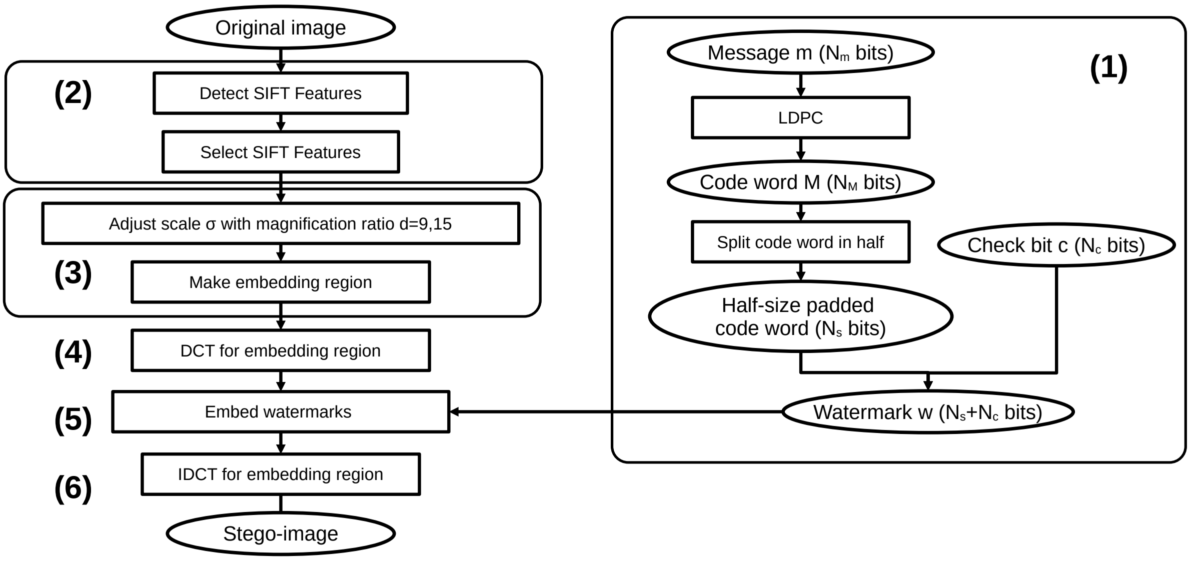


Fig. 2 Embedding process of proposed method

2.2.1 Composition of watermarks

Let $\boldsymbol{m} = (m_1, m_2, ..., m_{N_m})$, $m_i \in \{0, 1\}$ be a message with length $N_m = 200$ bits. If the stego-image is attacked, the message may be extracted incorrectly. Therefore, the message is encoded into a 297-bit code word $\boldsymbol{M}$ using a regular low-density parity-check (LDPC) code to correct the error. If the entire code word is embedded into one embedding region, the image quality in that region will degrade significantly. To avoid this degradation, the code word $\boldsymbol{M}$ is divided into two parts which are each then embedded into different embedding regions. As described in Section 2.2.3 and Table 1, 150 bits of watermark can be embedded in one embedding region. Therefore, the code word is padded so that $N_M = 300$ bits and is divided into two parts which are $N_s = 150$ bits each. Let $\boldsymbol{M}^H$ and $\boldsymbol{M}^T$ denote the upper bits (including MSB) and lower bits (including LSB) of the code word, respectively. To identify the upper and lower sides, two orthogonal $N_c = 30$ bit check bits $\boldsymbol{c}^H$ and $\boldsymbol{c}^T$ are embedded with each code word, where $c_i^H \in \{0, 1\}$ and $c_i^T \in \{0, 1\}$. Finally, the watermarks $\boldsymbol{w}^H$ and $\boldsymbol{w}^T$ consist of the code words $\boldsymbol{M}^H, \boldsymbol{M}^T$ and the check bits $\boldsymbol{c}^H, \boldsymbol{c}^T$, given by

$$\boldsymbol{w}^H = \left(M_1^H, M_2^H, \cdots, M_{N_s}^H, c_1^H, c_2^H, \cdots, c_{N_c}^H\right), \quad (4)$$

$$\boldsymbol{w}^T = \left(M_1^T, M_2^T, \cdots, M_{N_s}^T, c_1^T, c_2^T, \cdots, c_{N_c}^T\right). \quad (5)$$

2.2.2 Embedding regions

To obtain robust embedding regions against geometric transformations, the SIFT feature detector is applied to the original image. The coordinates and scale of feature points are obtained as SIFT features. After rescaling, feature points with a small scale may disappear due to scaling, while those with a large scale tend to have varying coordinates. Therefore, we limit the scale σ of feature points that are tolerant to geometric transformation. That is, the range of the scale is $4 \leq \sigma \leq 10$ [17].

The embedding region of the watermark is the bounding rectangle of a circle whose center point is the feature point and whose radius is proportional to the scale. A magnification ratio d is introduced to embed a watermark with a large bit length. Thus, the radius r is given by $r = d\sigma$. The embedding regions generated from each feature point may overlap with each other. Therefore, if watermarks are embedded in all embedding regions, the watermarks may be corrupted due to overlap. To avoid this, the feature points are further reduced using collision detection, which is performed by comparing the feature points in the order of their difference of Gaussian (DoG) output [6]. If the regions overlap, the feature point with the smaller DoG output is deleted. Finally, two types of watermarks $\boldsymbol{w}^H$ and $\boldsymbol{w}^T$ are embedded in the embedding regions of length $h = 2d\sigma$ in roughly equal numbers.

Here, let us consider the value of the magnification ratio d. The diameter of the embedding region, which will be discussed in the next section, must be at least 83 pixels. It is known empirically that if the block size is larger than the embedding region, watermarking errors will be reduced. The range of the scale is $4 \leq \sigma \leq 10$. When σ is equal to 4, i.e., the diameter is 8 pixels, the magnification d must be at least 11. Meanwhile, when $\sigma = 10$, a magnification ratio larger than 11 would make the size of the embedding region too large to ensure a large number of regions, which degrades performance. Therefore, a different magnification ratio d is introduced depending on the scale σ [7]. In this paper, the magnification ratio d is given by

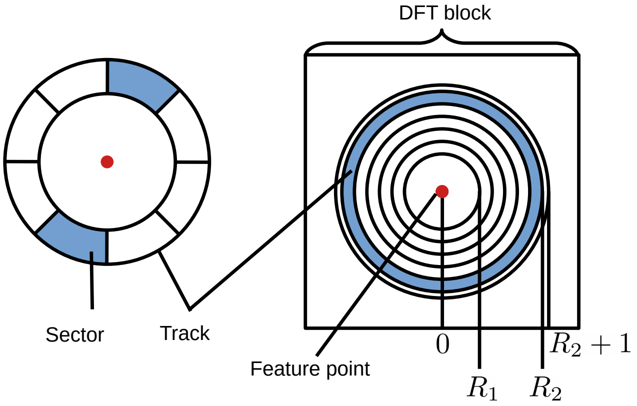


Fig. 3 Tracks and sectors

$$d = \begin{cases} 15 & (4 \leq \sigma \leq 6) \\ 9 & (6 < \sigma \leq 10) \end{cases} . \tag{6}$$

2.2.3 Embedding watermarks

The watermark is embedded in discrete Fourier transform (DFT) coefficients. The DFT coefficients when DFT is performed on the region $Y(x, y)$ are defined as $F(k_1, k_2)$. The amplitude spectrum $|F|$ and phase spectrum $\angle F$ are calculated from the DFT coefficients $F\ (k_1, k_2)$. The embedding region converted to an amplitude spectrum is referred to as a DFT block. In each DFT block, a pair of watermarks and check bits of $N_s + N_c = 180$ bits are embedded in a circular form by using quantization index modulation (QIM) [18].

As shown in Figure 3, the embedding region consists of a watermark region and a check-bit track. The former is a region enclosed by concentric circles of radii R_1 and R_2, and the latter is a concentric circle of radius R_2 to width 1. The watermark region is divided into $t = 5$ tracks in the radial direction. Each track is divided into $2s$ sectors in the angular direction.

The watermark $\boldsymbol{w}$ is embedded in each sector one bit at a time, starting from the inner track. All amplitude spectrums $|F(k_1, k_2)|$ in each sector are embedded with the same watermark w_i using QIM. The embedded amplitude spectrum is given by

$$|F'(k_1, k_2)| = 2\Delta_w \left(\left\lfloor \frac{|F(k_1, k_2)|}{2\Delta_w} - \frac{w_i}{2} + 0.5 \right\rfloor + \frac{w_i}{2} \right), \tag{7}$$

where $\lfloor x \rfloor$ is the floor function and Δ_w is the embedding strength. Let $F^*(k_1, k_2)$ be a complex number conjugate to $F(k_1, k_2)$. Note that the following symmetry

$$F(k_1, k_2) = F^*(h - k_1, h - k_2), \tag{8}$$

exists for the DFT coefficients. Thus, when embedding a watermark in $F(k_1, k_2)$, the same watermark must also be embedded in $F(h - k_1, h - k_2)$. This results in s bits of watermark per track.

Table 1 Sectors per track

Track	1	2	3	4	5
Radius (R_1-R_2)	15-20	20-25	25-30	30-35	35-40
sectors (s)	10	30	30	40	40
strength (Δ_w)	30	20	20	15	15

After embedding the watermark, the amplitude spectrum of the DFT block is converted to DFT coefficients. Let $|F'(k_1, k_2)|$ be the amplitude spectrum after embedding; the real part $\mathrm{Re}\, F'(k_1, k_2)$ and the imaginary part $\mathrm{Im}\, F'(k_1, k_2)$ of the embedded DFT coefficients are given by

$$\mathrm{Re}\, F'(k_1, k_2) = |F'(k_1, k_2)| \cos \angle F(k_1, k_2), \tag{9}$$

$$\mathrm{Im}\, F'(x, y) = |F'(k_1, k_2)| \sin \angle F(k_1, k_2). \tag{10}$$

By performing an inverse DFT on this region, the embedding process is completed.

The radii R_1 and R_2 of the embedding region affect the performance of the proposed method. In our method, the radii are set to $R_1 = 15, R_2 = 40$ by trial and error. Since the inner tracks have a shorter circumference, the number of embedded bits s is different for each track. In addition, the embedding strength of QIM, Δ_w, is different for each track. Table 1 shows the number of embedded bits s and embedding strength Δ_w for each track used in this method.

2.3 Extraction procedure

As shown in Figure 4, the extraction procedure of the proposed method is as follows: (1) Feature points are detected from the stego-image using the SIFT detector. (2) Extraction regions are generated. (3) Watermarks are extracted by using DFT and QIM. (4) Error correction is performed on the watermark, and the estimated message is determined.

2.3.1 Extraction regions and watermark candidates

Feature points are detected from the stego-image using the SIFT feature detector. First, we reduce the number of feature points whose scale σ is in the range of $2.8 \leq \sigma \leq 13$. Collision detection is not performed; instead, two extraction regions are selected for each feature point using two different magnification ratios (6). The extraction region is a rectangle with one side $h = 2d\sigma$ pixels centered on the feature point. When the stego-image is scaled, the size of the scale σ changes depending on the scaling rate. Thus, the same region as the embedded region can be obtained even if the stego-image is scaled. DFT is performed on the $h \times h$-pixel region to obtain the amplitude spectrum $|F|$. Let $\hat{P}$ be the number of feature points extracted. Watermarks $\hat{\boldsymbol{w}}$ are extracted from $\hat{P}$ extracted regions.

After scaling a stego-image, the values of the DFT coefficients are scaled by a scaling rate with respect to the values before embedding. Therefore, it is necessary to divide the DFT coefficients by the scaling rate. If the estimated scaling rate $\hat{\mu}$ can be obtained by the pilot signal, the watermark will be extracted accurately by dividing by the factor. The

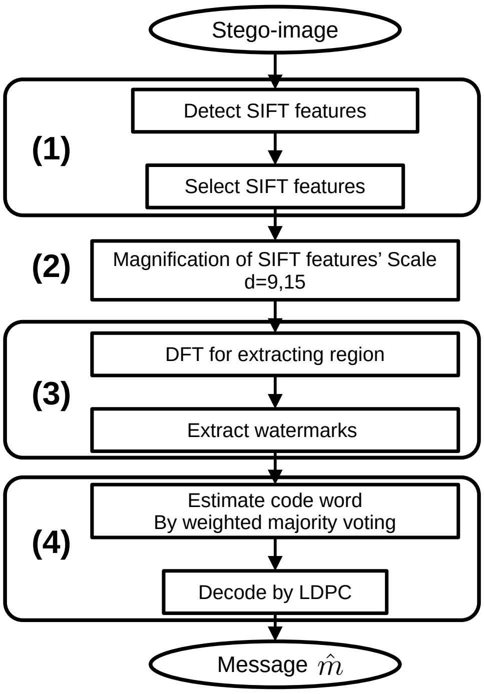

Stego-image
(1)
Detect SIFT features
Select SIFT features
(2)
Magnification of SIFT features' Scale d=9,15
(3)
DFT for extracting region
Extract watermarks
(4)
Estimate code word By weighted majority voting
Decode by LDPC
Message $\hat{m}$

Fig. 4 Extraction process of the proposed method

method of estimating the scaling rate will be explained in Section 3.

Since the radial direction of the DFT coefficients is invariant to scaling, watermarks can be extracted from the watermark and check-bit regions. Watermark candidates $\hat{w}_i$ are extracted from the amplitude spectrum $|\hat{F}(k_1, k_2)|$ with embedding strength Δ_w by

$$\hat{w}_i = \left\lfloor \frac{|\hat{F}(k_1, k_2)|}{\Delta_w} + 0.5 \right\rfloor \bmod 2. \tag{11}$$

Since the same watermark is embedded in the i-th sector, watermark candidate $\hat{w}_i \in \{0, 1\}$ can be calculated from a majority voting of their values [7]. The p-th watermark candidate $\hat{\boldsymbol{w}}^p$ is then split into a code word candidate $\hat{\boldsymbol{M}}^p$ and a check-bit candidate $\hat{\boldsymbol{c}}^p$. It is not possible to determine whether the extracted region contains a watermark from the code word candidates alone. Moreover, even if the watermark is included, it is impossible to determine whether the upper side $\boldsymbol{M}^H$ or the lower side $\boldsymbol{M}^T$ is contained. We determine this using the coincidence between the check bits $\boldsymbol{c}^H, \boldsymbol{c}^T$ and the check-bit candidate $\hat{\boldsymbol{c}}^p$,

$$r^{\lambda,p} = 1 - \frac{1}{N_c} \sum_{i}^{N_c} c_i^{\lambda} \oplus \hat{c}_i^{p}, \ \lambda \in \{H, T\}. \tag{12}$$

From the coincidence rate $r^{\lambda,p}$, the upper or lower side can be determined by

$$\hat{\lambda}^p = \arg \max_{\lambda \in \{H,T\}} r^{\lambda,p}. \tag{13}$$

Accordingly, the P pairs of code words and check bits can be determined as $\left(\hat{\boldsymbol{M}}^{H,p}, \hat{\boldsymbol{c}}^{H,p}\right)$ or $\left(\hat{\boldsymbol{M}}^{T,p}, \hat{\boldsymbol{c}}^{T,p}\right)$. We defined a set $\left\{\hat{\lambda}^p = H | p = 1, 2, \cdots, P\right\}$ as Λ^H and another a set $\left\{\hat{\lambda}^p = T | p = 1, 2, \cdots, P\right\}$ as Λ^T.

2.3.2 Estimation of code words

The P pairs of code words and check bits can be obtained by the coincidence rate. However, these pairs are degraded by scaling. Therefore, an estimated code word is computed by weighted majority voting [7]. The i-th estimated code words $\hat{M}_i^H, \hat{M}_i^T$ are calculated by

$$\hat{M}_i^H = \Theta\left(\sum_{\lambda \in \Lambda^H} \alpha\left(r^{\lambda,p}\right) \left(\hat{M}_i^{\lambda,p} - 0.5\right) \right), \tag{14}$$

$$\hat{M}_i^T = \Theta\left(\sum_{\lambda \in \Lambda^T} \alpha\left(r^{\lambda,p}\right) \left(\hat{M}_i^{\lambda,p} - 0.5\right) \right), \tag{15}$$

where the step function Θ is defined by

$$\Theta(x) = \begin{cases} 1 & (x \geq 0) \\ 0 & (x < 0) \end{cases}, \tag{16}$$

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

$$\alpha(x) = \begin{cases} 0.0 & (x < \beta) \\ \tanh(\gamma x - \beta) & (x \leq x) \end{cases}, \tag{17}$$

where β is the threshold and γ is the weight coefficient [19]. Here, let $\beta = 0.49$ and $\gamma = 7$, respectively. Thus, the full estimated code word is $\hat{\boldsymbol{M}} = \left(\hat{\boldsymbol{M}}^H, \hat{\boldsymbol{M}}^T\right)$, and the estimated message $\hat{\boldsymbol{m}}$ is obtained using the Noisy GDBF decoder [20].

3. Estimation of scaling rate using pilot signal

We propose a method of embedding a pilot signal to estimate the geometric attack. Since it is difficult to estimate all attacks simultaneously, the scaling rate is estimated.

3.1 Embedding the pilot signal

We also need to determine the (a) embedding domain, (b) embedding method, and (c) signal shape for the pilot signal.

(a) Embedding domain

To avoid affecting the watermark, the pilot signal is embedded in a different color space from the watermark. A color image is decomposed into its YUV components. The watermark is embedded in the Y component, which is the luminance value, and the pilot signal is embedded in the V or U component in order not to affect the watermark. In the following description, the U component is used. Let the size of the U-component image be $L_w \times L_h$ pixels, and the pixel value at coordinate (i, j) is represented by $U(i, j)$.

(b) Embedding method

The QIM [18] described in 2.2.3 can be used to embed the

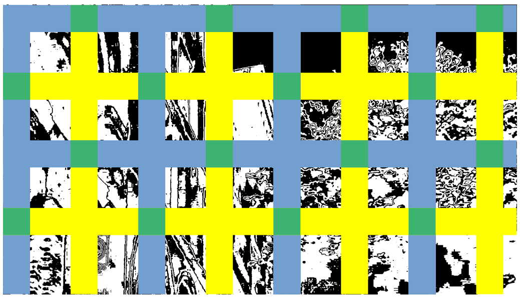


Fig. 5 Shape of pilot signal

pilot signal instead of the watermark. Note that the pilot signal $p \in \{0, 1\}$ is embedded in the pixel value $U(i, j)$. Let Δ_p be the step width for the pilot signal. Here, $\Delta_p = 10$.

(c) Signal shape

Figure 5 shows the position of the embedded pilot signal. The blue and yellow parts are embedded with the pilot signal values 1 and 0, respectively. The intersection of the two colors painted in green is embedded with the values 1 and 0 alternately. The remaining parts of the image are left unchanged. Figure 5 shows the grid interval and grid width, i.e., $G = 50$ and $B = 5$ pixels, respectively. In general, template matching [21] is used to find a particular pattern in a scaled image. However, this search is very expensive and time consuming. The proposed method can easily estimate the scaling rate by using grid patterns. The grid interval is the secret key for estimating the scaling rate. This means that even if an attacker knows that the pilot signal is embedded in the image, he cannot estimate the scaling rate without knowing the grid interval used during embedding.

3.2 Estimation of scaling rate

When a stego-image is scaled, the pilot signal can be applied to estimate the scaling rate. Since the pilot signal is a grid, the grid interval changes in proportion to the scaling rate. Therefore, the scaling rate can be estimated by calculating the grid interval obtained from the scaled stego-image. Since the same process is applied to the vertical and horizontal directions, we only discuss the vertical direction here. The calculated scaling rate will be used to extract the watermark, and the procedure for automatically obtaining the scaling rate is as follows.

First, the pilot signal $\hat{p}(i, j)$ is extracted from the pixel values $\hat{U}(i, j)$ of the U-component image of the stego-image by using QIM. The vertical sum of the signal $\hat{p}(i, j)$ is then computed. The sum of the j-th column is given by

$$S_j = \sum_{i=1}^{L'_h} p(i, j), \ j = 1, 2, 3, \cdots, L'_w, \tag{18}$$

where L'_w and L'_h are the width and height of the scaled stego-image, respectively. Let $\boldsymbol{S} = \{S_0, S_1, ..., S_{M'-1}\}$ be the series of sums. Second, as shown in Figure 6 (a), the series $\boldsymbol{S}$ has peaks at both 1s and 0s in the pilot signal because the values of 1s and 0s are embedded in the grid. Therefore, the scaling rate can be calculated from the grid intervals $\widehat{G}$ detected from the series $\boldsymbol{S}$. The autocorrelation of the series $\boldsymbol{S}$ is calculated to obtain the grid interval $\widehat{G}$. The autocorrelation function $A(\tau), \tau = 0, 1, \cdots, M' - 1$ at lag τ is represented by

$$A(\tau) = \frac{1}{(M' - 1 - \tau)\sigma_s^2} \sum_{i=1}^{M'-1-\tau} \left(S_i - \bar{S}\right)\left(S_{i+\tau} - \bar{S}\right), \tag{19}$$

where $\bar{S}$ and σ_s^2 are the mean and variance of the series $\boldsymbol{S}$, respectively. The autocorrelation function with negative lag τ is defined as

$$A(\tau) = A(-\tau), \ \tau < 0, \tag{20}$$

since the autocorrelation function is an even function. The range of the autocorrelation function $A(\tau)$ is defined as $\tau = -M'+1, -M'+2, \cdots, -1, 0, 1, \cdots, M'-1$. Figure 6 (b) shows the autocorrelation of the series $\boldsymbol{S}$. The grid interval can be calculated from the periodic peaks of the autocorrelations.

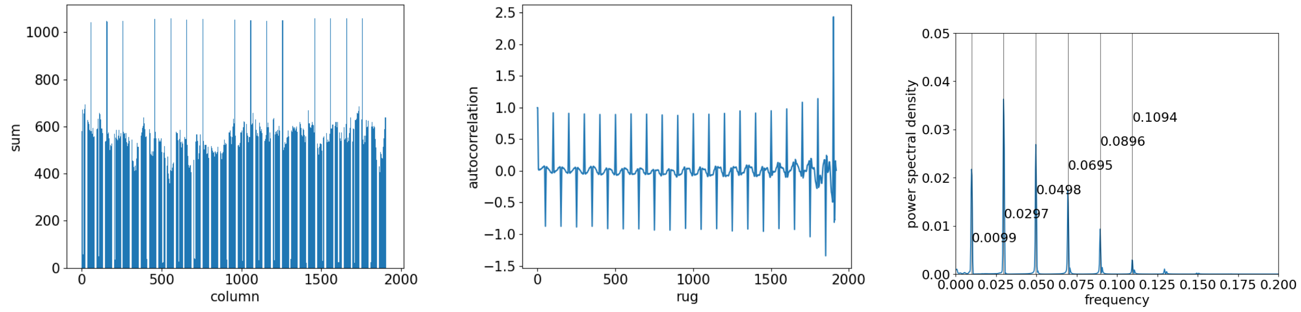
Third, the period of the autocorrelation can be calculated by performing a DFT of the autocorrelation coefficients. The period is obtained by calculating the inverse of the frequency detected from the DFT power spectrum. Figure 6 (c) shows the DFT power spectrum versus frequency. If attacks such as scaling or compression are applied to the stego-image, periods other than one of the pilot signal may be detected. Furthermore, since the data length is also detected as a frequency, the DFT is performed after applying a window function to the autocorrelation. The proposed method uses a flat-top window function $W(x), 0 \leq x \leq 1$, denoted by

$$\begin{aligned} W(x) &= 1 - 1.93\cos(2\pi x) + 1.29\cos(4\pi x) \\ &\quad -0.388\cos(6\pi)x + 0.032\cos(8\pi x). \end{aligned} \tag{21}$$

The window function makes it difficult to detect the data length as a period, which makes it easier to detect the period of the pilot signal. The position of the peak of the flat-top window function and the lag $\tau = 0$ of the autocorrelation function are expanded so that they coincide. The autocorrelation function $A'(\tau)$ multiplied by the window function is denoted by

$$A'(\tau) = A(\tau) W\left(\frac{\tau + M' - 1}{2(M' - 1)}\right). \tag{22}$$

Fourth, the peak of the autocorrelation will appear at every odd multiple of the grid interval $\widehat{G}$. That is, frequencies $3f_0, 5f_0, \cdots$ which are odd multiples of the frequency $f_0 = 1/\widehat{G}$ will also be detected. Hence, if a set of frequencies $f_0, 3f_0, 5f_0, \cdots$ that are odd multiples of f_0 can be detected,



(a) Series of sums $\boldsymbol{S}$

(b) Autocorrelation of series

(c) DFT power spectrum versus frequency

Fig. 6 Procedure for estimating scaling rate

this frequency f_0 can be considered as the pilot signal. The grid interval $\widehat{G}$ is then calculated by

$$\widehat{G} = \frac{1}{f_0}. \tag{23}$$

The pilot signal is only embedded in the pixel values on the grid. Therefore, values extracted from non-embedding regions are noise with respect to the pilot signal. In addition, the pilot signal may be degraded when the image is attacked. As a result, there may be more than one set of frequencies, or none may be found. If more than one set of frequencies exists, the strongest frequency f_0 is regarded as the pilot signal frequency.

Finally, the estimated scaling rate $\hat{\mu}$ can be calculated by

$$\hat{\mu} = \frac{\widehat{G}}{G}. \tag{24}$$

If the aspect ratio has changed due to an attack, two different scaling rates can be obtained. Otherwise, the average of the two scaling rates is used as the estimated scaling rate.

4. Computer simulation

The proposed method finds the estimated scaling rate from the pilot signal embedded in the U component and then extracts the watermark embedded in the DFT domain of the Y component using the estimated scaling rate. The proposed method was built using the SIFT detector in OpenCV version 4.4.0. As mentioned in section 2, the parameters of the proposed method were determined with reference to IHC. However, to assess the performance, our method was evaluated under the following conditions. The attack on the stego-image is assumed to be scaled by wider scaling rate than IHC and cropped. The scaled stego-images were cropped in HDTV (1280 × 720 pixels) and VGA (640 × 480 pixels) sizes as well as in FHD size. First, we evaluate how accurately the scaling rate is estimated from the pilot signal in terms of relative error. Next, the estimated scaling rate is used to estimate the watermark. The bit error rate (BER) of the extracted watermark and the peak signal-to-noise ratio (PSNR) of the stego-image are evaluated. Finally, we compare the resulting BER and PSNR with those of the previous study using SIFT feature points and DFT [2].

4.1 Evaluation of estimated scaling rate

We evaluate the estimated scaling rate in terms of relative error. Let μ be the true scaling rate and $\hat{\mu}$ be the estimated scaling rate. The relative error R is given by

$$R = \left| 1 - \frac{\hat{\mu}}{\mu} \right|. \tag{25}$$

The attack on the stego-image is assumed to be scaled between 0.1 and 3.0 and cropped at FHD size. The scaling rate was estimated from the vertical direction (1080 pixels). Figure 7 shows the relative error R of the estimated scaling rate $\hat{\mu}$ at each scaling rate μ $(0.1 \leq \mu \leq 3.0)$. There are twenty four images cropped from four locations per image after scaling six IHC images. The blue and orange lines represent the mean and median, respectively, and the box-and-whisker plot represent the quartiles. The figure shows that when the image was scaled down, the estimation was successful with a small relative error. However, when the image was scaled up, the relative error exceeded 0.1 at scaling rates greater than 1.6x. Furthermore, the relative error was clearly larger in more cases when the scaling rate exceeded 2.0x. The number of gridlines of the pilot signal in the cropped image was important for detection. Many gridlines could be detected from scaled-down images, whereas fewer gridlines could be detected from scaled-up images. As the grid interval was 100 pixels, attack estimation was found to be difficult from images containing roughly 5.5 gridlines or less.

4.2 Evaluation of estimated watermark

We evaluate the proposed method using the BER of the message. Let $\boldsymbol{m}$ be the true message and $\hat{\boldsymbol{m}}$ be the extracted message. The BER is given by

$$\mathrm{BER} = \frac{1}{N_m} \sum_{i=1}^{N_m} m_i \oplus \hat{m}_i. \tag{26}$$

We compare the results with scaling rate estimation (SRE), without SRE, and with a known scaling rate (KSR).

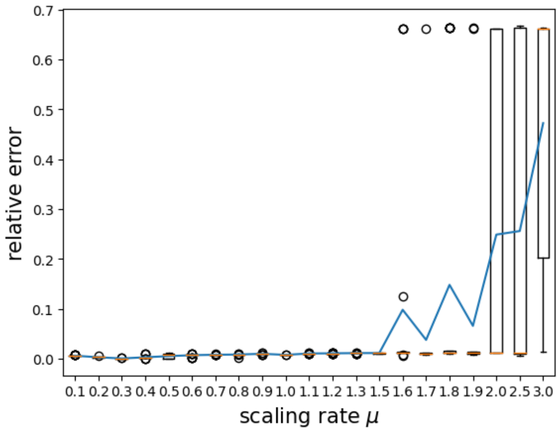


Fig. 7 Relative error per scaling rate when the scaling rate is $0.1 - 3.0$

Figure 8 shows the average BER for each case. The stego-images were scaled by $0.7 - 1.3$ and cropped at FHD size. The horizontal and vertical axes represent the scaling rate and BER, respectively. The blue and orange lines are the mean and median of the BER, respectively. As shown in these figures, the results of the KSR and SRE methods are similar, and the median is equal to zero for all scaling rates for both methods. In contrast, the BER of the method without SRE was larger than that of the others. Thus, the proposed scaling rate estimation is shown to be effective and the estimated value can be applied to reduce message errors.

Next, we evaluated the BERs when the stego-images were scaled by between 0.5x and 1.5x and cropped to FHD, HDTV and VGA sizes. Figure 9 shows the average BER for each cropping size. The horizontal and vertical axes represent the scaling rate and BER, respectively. The blue, orange, and green lines represent the average BER for KSR, without SRE, and SRE. Figure 9 (a) shows the results for the FHD size. Assuming a scaling rate between 0.7 and 1.3, the parameters were set so that feature points could be extracted. Therefore, the BERs were larger than 0.1 for scaling rates below 0.6 and above 1.3. When the scaling rate was smaller than 0.7, the embedding regions became smaller and the watermarks were more difficult to extract. On the other hand, when the scaling rate was greater than 1.3, the number of feature points in the cropped images became fewer and the error rate became larger. Figures 9 (b) and (c) show the results for HDTV and VGA sizes, respectively. The BERs increased when the cropped image size was small because the number of feature points became fewer.

4.3 Evaluation of image quality

Image quality is degraded due to the embedding of the pilot signal. Therefore, the image quality must be evaluated using PSNR. Table 2 shows the average PSNRs of six stego-images for when only the watermark is embedded and when both the watermark and the pilot signal are embedded. The image quality was reduced by about 3.4 dB when embedding the pilot signal. Because the watermarks in both cases were embedded with the same embedding strength, the image quality degradation was caused by the pilot signal.

Table 2 Image quality (PSNR)

	PSNR [dB]
only watermarks	38.3
both watermark and pilot signal	34.9

For the conventional methods that do not estimate the scaling rate, the embedding strength Δ_w of the QIM must be large in order to extract the watermark accurately. For comparison, the embedding strength of the proposed method is the same as that of the method without the pilot signal. However, the proposed method can accurately extract the watermark by estimating the scaling rate. Therefore, it may be possible to extract the watermark even if the embedding strength Δ_w is small.

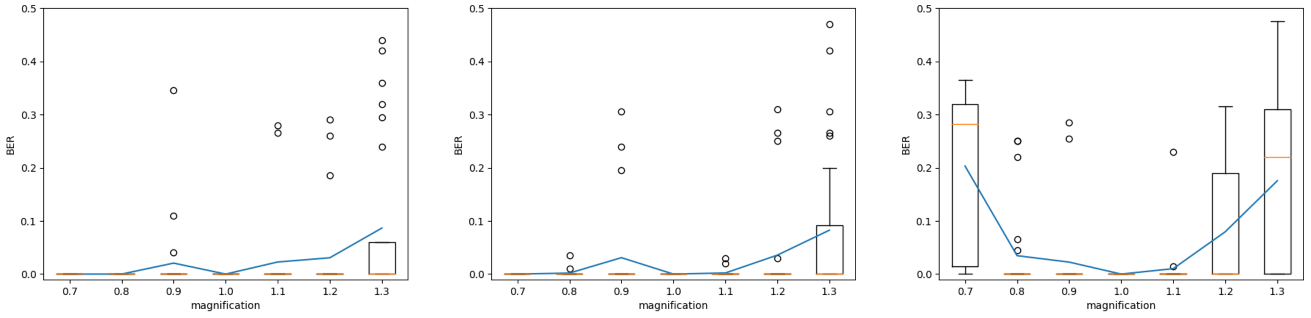
4.4 Comparison with previous study

Luo *et al.*'s method [2] is similar to the proposed method in that DFT is performed on the regions around the SIFT feature points and a watermark is embedded in the DFT coefficients. However, their method requires saving descriptors of the feature points where the watermark is embedded. In other words, their method is not a completely blind watermarking. In addition, their method only embeds a 1-bit message indicating whether or not the watermark is embedded. The proposed method can embed a 200-bit message into the image. In evaluating the image quality, we compare the two methods in terms of the length of the watermark to be embedded, not the message length.

We compared both methods in terms of watermark length, average PSNR, and BER. Table 3 shows the length of the watermark, PSNR, and BER for each method. The image quality varies depending on the length of the watermarks. The BER for each method is calculated as the average of the results obtained when the stego-image is scaled at scaling rates $\mu = 0.9, 1.1$. Here we compare the length of the watermarks embedded in an embedding region generated at a feature point. In Luo *et al.*'s method, the watermark length to be embedded in one embedding region is 128 bits, whereas in the proposed method, the length is 180 bits. Since the image quality degrades as the watermark length increases, Luo *et al.*'s method yields a higher image quality than the proposed method. However, the proposed method demonstrated a lower BER for both scaling rates, indicating its effectiveness over Luo *et al.*'s method.

5. Conclusion

Conventional watermarking methods have accomplished robustness against attacks by error correction and normalization. Since non-geometric attacks degrade the watermark, the errors in the watermark can be corrected by adding redundancy. However, redundancy is not effective against geometric attacks. The type and strength of geometric attacks

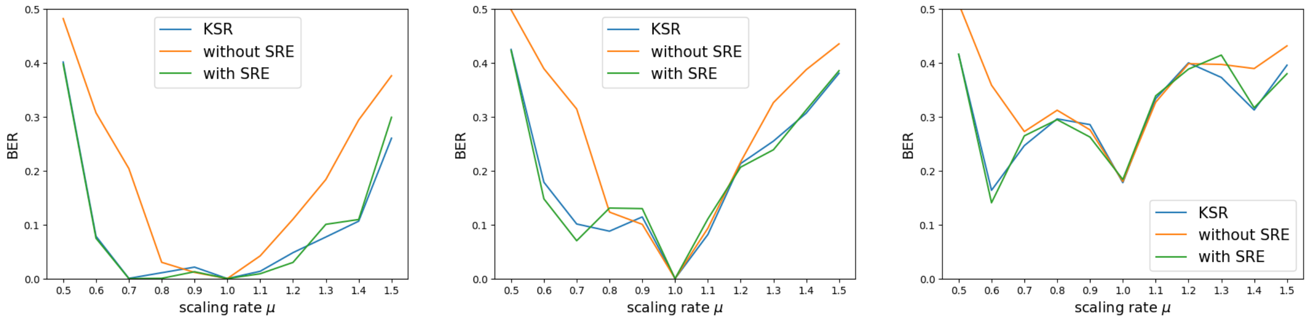


(a) BER of message with KSR

(b) BER of message with SRE

(c) BER of message without SRE

Fig. 8 BER of watermarks for each case when stego-images are scaled by 0.7 – 1.3 and cropped at FHD size.



(a) BER when image size is FHD

(b) BER when image size is HDTV

(c) BER when image size is VGA

Fig. 9 BER of watermarks for each method

Table 3 Comparison of proposed method with Luo *et al.*'s method

	watermark length	PSNR	BER	
scaling rate μ			0.9	1.1
Luo *et al.*	128	more than 42	0.060	0.103
only watermarks	180	38.3	0.023	0.010
both watermark and pilot signal	180	34.9	0.031	0.002

need to be estimated to extract the watermark accurately.

The proposed method estimates the scaling rate of the stego-image by embedding a grid-shaped pilot signal into the images. At the same time, the watermark is embedded by performing DFT on the embedding region around the SIFT feature points. Since the value of the DFT coefficient is proportional to the scaling rate, the watermark can be correctly estimated from the DFT coefficient by using the estimated scaling rate. The image quality of the proposed method was degraded by embedding the pilot signal. However, since the scaling rate could be estimated almost correctly, the proposed method achieved a lower BER than the previous study.

In this paper, we have shown that the proposed watermarking scheme that estimates the geometric attack with the pilot signal and estimates the non-geometric attack with redundancy, e.g. error correction codes, was effective. In the future, we would like to attempt to estimate geometric transformations. However, the issue of overwriting the watermark is an open problem.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP20K11973. Part of this work was carried out under the Cooperative Research Project Program, R04/B09 Research on Multifunctional Multimedia Production, of the RIEC, Tohoku University.

References

[1] Information Hiding and its Criteria for evaluation, https://www.ieice.org/iss/emm/ihc/en/image/image.php (20th February, 2023 access)

[2] H. Luo, X. Sun, H. Yang, Z. Xia, " A Robust Image Watermarking Based on Image Restoration Using SIFT," Radioengineering, vol.20, no.2, pp. 525-532, 2011

[3] L. Li, B. Guo, J. Pan, "Feature-Based Image Watermarking Resisting Geometric Attacks, " 2008 3rd International Conference on Innovative Computing Information and Control, Dalian, Liaoning, p. 18, 2008

[4] M. Beguma, J. Ferdushb, M. S. Uddinc, " A Hybrid robust watermarking system based on discrete cosine transform, discrete wavelet

transform, and singular value decomposition, ” J. King Saud Univ Sci., vol. 34, Issue 8, part B, pp. 5856-5867, 2022

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

[6] M. Kawamura, K. Uchida, “ SIFT feature-based watermarking method aimed at achieving IHC ver.5, ” IIH-MSP, pp.381-389, 2017

[7] M. Hayashi, M. Kawamura, “ Improved SIFT feature-based watermarking method for IHC ver. 5, ” Proc. of APSIPA ASC 2018, 2018

[8] V. Solachidis, I. Pitas, “Circularly Symmetric Watermark Embedding in 2-D DFT Domain, ” IEEE Trans Image Process, vol. 10, no.11, pp.1741-1752, 2001

[9] J. S. Seo, C. D. Yoo, “ Localized image watermarking based on feature points of scale-space representation, ” Pattern Recognition, vol. 37, Issue 7, pp. 1365-1375, 2004

[10] H. Kang, K. Iwamura, “ Watermarking based on the difference of discrete cosine transform coefficients and an error-correcting code, ” Proc. of IWIHC, pp. 9-17, 2014

[11] I. J. Cox, J. Kilian, F. T. Leighton and T. Shamoon, “ Secure spread spectrum watermarking for multimedia, ” in IEEE Transactions on Image Processing, vol. 6, no. 12, pp. 1673-1687,1997

[12] J. J.K.O. Ruanaidh, T. Pun, “Rotation, scale and translation invariant spread spectrum digital image watermarking, ” Signal Process., vol. 66, Issue 3, pp. 303-317, 1998

[13] L. Tong, B. M. Sadler and M. Dong, “ Pilot-assisted wireless transmissions: general model, design criteria, and signal processing, ” IEEE Signal Process Mag., vol. 21, no. 6, pp. 12-25, 2004,

[14] P. C. Su, Y. C. Chang and C. Y. Wu, “ Geometrically Resilient Digital Image Watermarking by Using Interest Point Extraction and Extended Pilot Signals, ” IEEE Trans. Inf. Forensics Secur., vol. 8, no. 12, pp. 1897-1908, 2013,

[15] M. Á. Rodríguez, F. P. González, “ Analysis of pilot-based synchronization algorithms for watermarking of still images, ” Signal Process. Image Commun., vol. 17, Issue 8, pp. 611-633, 2002

[16] M. C. Hernandez, F. G. Ugalde, M. N. Miyatake, H. P. Meana, “Robust Object-Based Watermarking Using SURF Feature Matching and DFT Domain, ” Radioengineering, vol. 22, no. 4, pp.1057-1071, 2013

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

[18] B. Chen, 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] M. Kawamura, K. Uchida, “ SIFT feature-based watermarking method aimed at achieving IHC ver.5, ” Intelligent Information Hiding and Multimedia Signal Processing, pp. 381-389, 2017

[20] G. Sundararajan, C. Winstead and E. Boutillon, “Noisy Gradient Descent Bit-Flip Decoding for LDPC Codes, ” IEEE Trans. Commun., vol. 62, no. 10, pp. 3385-3400, 2014

[21] S. Pereira, T. Pun, “ Fast Robust Templete Matching for Afine Resistant Image Watermarks, ” Proc. 3rd Int. Information Hiding Workshop, pp.207-218, 1999

Rinka KAWANO received a B.S. and M.S. from Yamaguchi University in 2021 and 2023, respectively. Currently, she is a doctoral student at the Graduate School of Science and Engineering, Yamaguchi University. Her research interests include digital watermarking. She is a student member of IEICE.

Masaki KAWAMURA received a B.E., M.E., and Ph.D. from the University of Tsukuba in 1994, 1996, and 1999. He joined Yamaguchi University as a research associate in 1999, where he is currently a professor. His research interests include optimization, associative memory models, and information hiding. He is a fellow of IEICE and a member of JNNS, JPS, and IEEE.