Multimedia Tools and Applications (2020) 79:11297–11323
https://doi.org/10.1007/s11042-019-08367-6

Enhancing the robustness of image watermarking against cropping attacks with dual watermarks

Ching-Sheng Hsu[1] · Shu-Fen Tu[2]

Received: 10 February 2019 / Revised: 30 September 2019 / Accepted: 9 October 2019 /
Published online: 18 December 2019

Abstract
In this study, a QR-based digital watermarking scheme that can use color images is proposed. The main purpose of this method is to enhance robustness against cropping attacks. To achieve this aim, each bit of the robust watermark has four copies, which are hidden in different image blocks. The embedding rule is designed based on the sinusoidal function, and the wavelength of the sinusoidal function controls the trade-off between imperceptibility and robustness. The four copies of the watermark bit may be inconsistent if the watermarked image undergoes cropping attacks. Therefore, after the four copies of the watermark bit are extracted, the actual value of the watermark bit is judged based on the result of the tampering detection. Experimental results indicate that improving robustness with the help of tampering detection results effectively assists in watermark extraction. In addition, the method is superior to other methods in terms of invisibility, robustness, and embedding payload.

Keywords Image watermarking · Robust watermark · QR decomposition · Fragile watermark

1 Introduction

The transmission of digital images through the Internet has become more convenient and frequent. However, digital messages are susceptible to malicious modification or illegal copying. Malicious modifications may cause misinterpretation if a message receiver cannot verify an image's integrity. A false medical image may cause an erroneous diagnosis, or a falsified personal photo may damage a person's reputation [1]. Illegal copying may cause copyright disputes among individuals. Therefore, the integrity verification and copyright protection of digital images are crucial research concerns.

✉ Shu-Fen Tu
dsf3@ulive.pccu.edu.tw

[1] Department of Information Management, Ming Chuan University, No. 5, Deming Rd., Guishan District, Taoyuan City 333, Taiwan

[2] Department of Information Management, Chinese Culture University, No. 55, Huagang Rd., Shihlin District, Taipei City 11114, Taiwan

Digital watermarking is a common method for protecting copyright or verifying the integrity of digital images [2]. The process of digital watermarking comprises two stages, namely watermark embedding and watermark extraction. For copyright protection, a watermark, which is a logo or binary stream, is embedded into a host image according to an embedding rule in the first stage. In the second stage, the watermark is extracted and used to examining whether copyright piracy has taken place. Because a person may attempt to alter the watermarked image to erase the watermark, the watermark must be robust against attacks. If the watermarked image undergoes modifications, the extracted watermark may be different from the original watermark but still recognizable. To detect tampering, the watermark representing an image feature is embedded into the host image in the first stage. In the second stage, the watermark is extracted and compared with the image feature to verify integrity and determine the tampered area. Because malicious alterations may be small to avoid being easily noticed, the watermark must be sensitive to any alteration. Regardless of the use of a robust or fragile watermarking scheme, the original host image or original watermark should not be used when extracting the watermark; otherwise, the practicability of the method is reduced. A watermarking scheme without the host image to extract the watermark is called a blind scheme.

The watermark operates in a spatial [3–5] or frequency domain [6–8]. In a spatial domain scheme, pixel values of the host image are modified to embed the watermark, whereas in a frequency domain scheme, frequency coefficients of the host image are modified to embed the watermark. The most common methods used to transform the host image from the spatial domain to the frequency domain are discrete Fourier transform (DFT) and discrete wavelet transform (DWT). Recently, some studies have investigated another domain that is derived from matrix factorization by using methods such as singular value decomposition (SVD), QR decomposition, and LU decomposition. A matrix factorization method is used to decompose the host image into matrices, and the watermark is embedded into one of these matrices. Watermarking schemes for copyright protection generally embed the watermark into the frequency or matrix domain because these domains are more robust against attacks than the spatial domain. By contrast, the watermarking scheme for tampering detection generally embeds the watermark into the spatial domain, particularly in the least significant bit planes.

Robustness can be enhanced by increasing the energy of the watermark, such as embedding the watermark in low-frequency bands or high-order bits of the image. Such domains generally remain invariant if the watermarked image undergoes image processing, such as contrast adjusting, brightening, darkening, and JPEG compression. However, the trade-off between the robustness and imperceptibility must be considered. Moreover, if the watermarked image undergoes cropping attacks, some parts of the watermark may be removed. Therefore, the extracted watermark may be unrecognizable or insufficient for validating copyright. Robustness against cropping attacks is crucial because most malicious attacks are cropping attacks. A cropping attack removes a part of the image and replaces it with white or black color. However, an attacker may replace the cropped part with a meaningful object to disguise malicious behavior. The replacement of the cropped part with a black-and-white or meaningful object generates the same underlying manipulation; that is, some pixels of the image are changed. Cropping attacks can be prevented using two methods, namely the uniform distribution of the watermark into the image [9] or embedding multiple copies of the watermark [10]. Creating multiple backups is a favorable strategy. When a watermarked image is cropped, some copies may be removed, and some may survive. Therefore, the probability of successfully extracting the watermark becomes higher. However, this strategy has one notable drawback. When the watermarked image is cropped, each copy of the same watermark bit may be inconsistent. Because the original watermark image is not available, determining which copy is the authentic one is difficult. In light of increasing the probability of prediction, the watermark bit can be determined according to a majority copy or by a high-priority copy.

As we have mentioned above, cropping attacks are usually malicious; therefore, robustness against cropping attacks is more important than that against common image processing. In this study, a digital image watermarking scheme with dual watermarks was proposed to enhance robustness against cropping attacks. Robust and fragile watermarks were simultaneously embedded into the image. The robust watermark operates in an R matrix of the image decomposed using QR factorization. Each watermark bit is embedded four times, and the four copies are effectively distributed in the host image. To avoid the overflow problem caused by watermark embedding, the allowed ranges of the elements of the R matrix were analyzed. Therefore, it is ensured that the pixel value of the image is within the legal range when the watermarked R matrix is multiplied by the original Q matrix to restore the image. When the watermark bit is being extracted, determining appropriate copies for extraction is essential if all copies are not the same. The prominent feature of the proposed scheme involves using the fragile watermark to determine unaltered carriers of the copies and the extracted watermark bit according to the copies obtained from these carriers. Through this approach, the appropriate rate of the recovered watermark bits and robustness against cropping attacks can be increased. For security reasons, the keys used in this study were encrypted using asymmetric encryption before transmitting them to a receiver.

The remainder of this paper is organized as follows: In Section 2, some methods for robust watermarking using matrix decomposition were reviewed, along with studies related to dual digital watermarking schemes. In Section 3, the proposed scheme is described in detail. Section 4 presents experimental results and analysis. Finally, conclusions are provided in Section 5.

2 Related studies

Until now, three matrix decomposition methods have been used in watermarking schemes: QR, SVD, and LU decomposition. These matrix decomposition methods factorize a matrix into several matrices, and the original matrix can be returned by multiplying these factorized matrices. These three matrix decompositions, without explaining the mathematical theories behind them, can be described as follows: LU decomposition factorizes a $u \times u$ matrix A into a $u \times u$ lower triangular matrix L and a $u \times u$ upper triangular matrix U, with $A = L \times U$. QR decomposition factorizes a $u \times v$ matrix A into a $u \times u$ orthogonal matrix Q and a $u \times v$ upper triangular matrix R, with $A = Q \times R$. SVD decomposition factorizes a $u \times v$ matrix A into a $u \times u$ orthogonal matrix U, a $u \times v$ diagonal matrix Σ, and a $v \times v$ orthogonal matrix V, with $A = U \times \Sigma \times V^T$.

Wang et al. [11] proposed an LU-based watermarking scheme. In this scheme, both the host and watermark images are in full color. The host image is transformed into the frequency domain through 1-level DWT. Subsequently, low–high (LH) and high–low (HL) bands are divided into 4 × 4 blocks, and each block is factorized through LU decomposition. The watermark bits are embedded into the element in the upper right corner of each matrix U. The experimental results indicate that robustness against cropping attacks reduced with an increased crop area. Moreover, Su et al. [12] proposed a blind color image watermarking scheme based on LU decomposition. The host image is divided into 4 × 4 blocks, and each block is factorized into matrices L and U using LU decomposition. The watermark bit is embedded blockwise by adjusting the relationship between the sizes of two elements of matrix L. The embedding capacity of these two methods is 0.3, which is not particularly satisfactory. Liu et al. [13] transformed the host image into the frequency domain through 1-level DWT, but the working band was low–low (LL). The LL band is factorized through SVD decomposition, and the watermark is embedded into matrix Σ. The embedding capacity of the scheme is considerably satisfactory, but the original image is required to assist in extracting the watermark.

QR decomposition is frequently used in image processing [14] and digital watermarking schemes [9, 15]. QR decomposition is more accurate than LU decomposition for least square problems. Furthermore, LU factorization can only be applied to square matrices when QR decomposition can be applied to rectangular and square matrices. QR decomposition has lower computational complexity and a less severe false positive detection problem than SVD [16, 17]. A QR-decomposed matrix has a favorable value for robust watermarking. If columns of matrix A are correlated with each other, the absolute values of the elements in the first row of the R matrix are probably higher than those of the other rows. Because the adjacent pixels of the image are generally highly correlated, the first row elements of the R matrix exhibit favorable properties related to robustness and hiding capacity [15, 17]. Naderahmadian and Hosseini-Khayat [18] proposed a rapid watermarking scheme based on QR decomposition in the wavelet domain. The host image is a grayscale image and the watermark image is binary. However, the robustness of this scheme against cropping attacks is not satisfactory. Su et al. [15] embedded the watermark bit into the element in the upper right corner of each R matrix because they believed that only modifying this element has the lowest impact on the original image. Su et al. [9] and Su et al. [16] observed that the values of the second row first column element and third row first column element of the Q matrix are highly correlated. As a result, they designed a watermark embedding rule based on the relationship between these two values. However, the modification of the elements in the Q matrix affects the visual quality of the host image. Therefore, Su et al. [16] performed a compensatory operation on the elements in the first row of the R matrix. Although the design principle of the watermark hiding rules of Su et al. is the same as that of Su et al. [9, 15, 16], compensation for the R matrix is not given special mention in these papers. The host image and watermark of the aforementioned methods of Su et al. are all color images. A common problem is that the hiding capacity is not strong, and robustness against cropping attacks is not appropriate. When the cropping ratio of the image is 25%, the method of Su et al. [9] can ensure 90% similarity between the extracted watermark and original watermark. However, when the cropping ratio reaches 50%, none of these methods achieves a similarity of more than 90%.

In terms of the merit of QR decomposition for robust watermarking schemes, a QR-based robust watermarking scheme was proposed in this study, and the working domain was the R matrix. Changing the values of the decomposition matrix elements must be cautious because the restored pixel values may be out of range. Therefore, most QR-based digital watermarking schemes cannot completely utilize the elements of the R matrix. Consequently, such methods may have a low data-hiding capacity. Therefore, this study proposed a method with a modifiable range of element values of the R matrix. All elements of the first row of the R matrix were used to embed the watermark. Considering the necessity of robustness against cropping attacks, this study proposed two strategies corresponding to cropping attacks. The first strategy is to dispersedly embed multiple copies of the robust watermark into the image. The second strategy is to simultaneously embed the fragile watermark and use it to enhance the extracted robust watermark. Limited studies have proposed schemes to embed both robust and fragile watermarks. However, these studies have not proposed the concept of using the fragile watermark to strengthen robustness, and these methods have not been strengthened specifically for cropping attacks. Schlauweg et al. [19] proposed a semifragile watermarking scheme using a quantization index modulation technique. They later proposed a digital watermarking scheme with dual watermarks. The robust watermark was embedded according to the geometric structure of objects to attain favorable robustness against lossy compression [20]. The fragile watermark (i.e., the image feature) was embedded through their previous work [19]. Furthermore, Shivani et al. [21] proposed a dual watermarking scheme where the robust watermark was embedded in DWT coefficients, and the image features were embedded in the first least significant bitplane. In their experimental results, Shivani et al. validated robustness against the image processing of histogram equalization, Gaussian, and JPEG compression. However, Schlauweg et al. [20] and Shivani et al. [21] did not focus on robustness against cropping attacks.

3 Proposed scheme

In the proposed scheme, the robust and fragile watermarks are embedded into the host image. Therefore, the proposed scheme has two functions: tampering detection and copyright protection. During the embedding process, the robust watermark was repeatedly embedded to enhance its robustness. Because the watermarked image may be tampered with, all copies may be different from each other. The utilization of the fragile watermark to assist in determining an appropriate copy is the most distinguishing feature of the proposed method. This feature can further improve robustness during the extraction process. The proposed scheme was first employed for a grayscale host image and binary watermark image. The extension of the proposed method to color host and watermark images was later studied.

3.1 Embedding process

Figure 1 presents the embedding process. In Fig. 1, the host image H is a grayscale image of size $M \times N$ (M rows and N columns), and the robust watermark image RW is a binary image of size $m \times n$, where $m = \lfloor M/4 \rfloor$ and $n = \lfloor N/4 \rfloor$. The host image H is factorized through QR decomposition, and the robust watermark RW is embedded in the R matrix of H. The R' matrix with RW is composed with the Q matrix to return to pixel values of the watermarked image H'. The feature FW of H' is then extracted and embedded in the least significant bitplane of H' to generate the final watermarked image H''.

3.1.1 Robust watermark embedding

The following describes the hiding rules of a single watermark bit and the distribution of all watermark bits in the entire image.

Initially, H is divided into 4×4 nonoverlapping blocks, each of which is factorized into the Q and R matrices through QR decomposition. The watermark is hidden in the first four elements of each R matrix. “Hidden” refers to changing the element value such that the state of the element value can represent the watermark bit. In this study, a sine function was used to present the state of the element values. For any element value x_0, watermark bit 1 and bit 0 were represented using $f(x_0) \geq 0$ and $f(x_0) < 0$, respectively, and function f is defined as follows:

$$f(x) = sin\left(x.K.\frac{\pi}{180}\right) \tag{1}$$

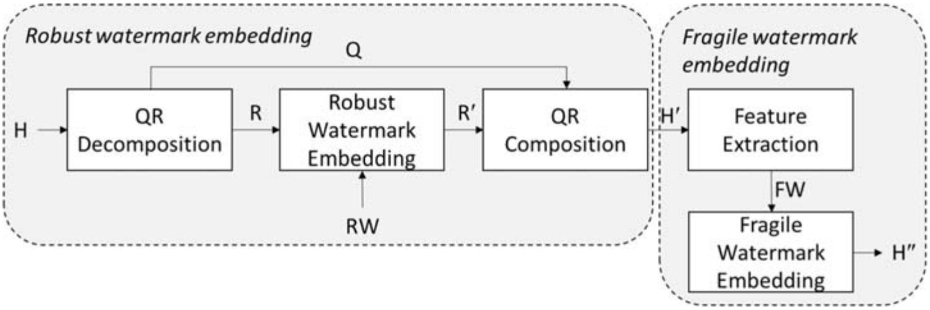

Robust watermark embedding
Q
H
QR Decomposition
R
Robust Watermark Embedding
R′
QR Composition
RW
H′
Fragile watermark embedding
Feature Extraction
FW
Fragile Watermark Embedding
H″

Fig. 1 Illustration of the embedding process

where $K \geq 0$ is a real number. The wavelength λ of the function f is $(360/K)$, and thus, parameter K is a wavelength controller.

When hiding the watermark bit, the state of the element value may be inconsistent with it. When this happens, the element value must be adjusted such that its state conforms to the watermark bit to be hidden. Let w, x_0, and x' be the watermark bit, element used to embed w, and adjusted value of x_0, respectively. If $w = 0$ and x_0 is in the negative half cycle, or if $w = 1$ and x_0 is in the positive half cycle, then the value of x' must be the midpoint of the same half cycle. If $w = 1$ and x_0 is in the negative half cycle, or if $w = 0$ and x_0 is in the positive half cycle, then the value of x' has two candidates: the midpoint x_l of the left adjacent half cycle and the midpoint x_r of the right adjacent half cycle. Let d_l and d_r be the relative distance between x_0 and x_l and between x_0 and x_r, respectively. If $d_r \leq d_l$, x' is set to x_r; otherwise, x' is set to x_l. In summary, the rules for element value adjustment are as follows:

Case 1: $w = 1$ and $f(x_0) \geq 0$

$$x^{'} = x_0 - r + \frac{\lambda}{4} \tag{2}$$

Case 2: $w = 0$ and $f(x_0) < 0$

$$x^{'} = x_0 - r + \frac{3\lambda}{4} \tag{3}$$

Case 3: $w = 1$ and $f(x_0) < 0$

$$x^{'} = x_0 - r + \frac{\lambda}{4} + \lambda . \left\lfloor \frac{4r}{3\lambda} \right\rfloor \tag{4}$$

Case 4: $w = 0$ and $f(x_0) \geq 0$

$$x^{'} = x_0 - r - \frac{\lambda}{4} - \lambda . \left\lfloor \frac{4r}{\lambda} \right\rfloor \tag{5}$$

The symbol r in the aforementioned rules is the remainder of x divided by λ and is defined as follows:

$$r = x - \lambda . \left\lfloor \frac{x}{\lambda} \right\rfloor \tag{6}$$

The distribution of watermark bits in the image was then studied. As mentioned previously, image H is divided into 4 × 4 nonoverlapping blocks. Each image block A can be viewed as a 4 × 4 matrix and is factorized into the Q and R matrix through QR decomposition. For security reasons, all image blocks are reordered according to a pseudo-random number generator before the watermark is embedded. Let R_i and w_i represent the R matrix of the rearranged ith image block and the ith watermark bit, where $i = 0..(q - 1)$ and $q = \lfloor M/4 \rfloor \times \lfloor N/4 \rfloor$. To improve robustness, each watermark bit is duplicated four times. The jth copy of w_i is embedded into the element $R_{(i+j)\%q}[0, j]$, where $j = 0..3$. To elucidate the watermark distribution rule of the proposed scheme, Fig. 2 is presented as an example. Let five watermark bits (w_0 to w_4) be embedded into five R matrices (R_0 to R_4). Figure 2 indicates that four copies of w_0 and w_1 are embedded into $R_0[0, 0]$, $R_1[0, 1]$, $R_2[0, 2]$, and $R_3[0, 3]$ and into $R_1[0, 0]$, $R_2[0, 1]$, $R_3[0, 2]$, and $R_4[0, 3]$, respectively. The distribution of the four copies of the other three watermark bits, w_2, w_3, and w_4, are evident in Fig. 2.

After modifying the value of the R matrix, it is multiplied by the Q matrix and is returned to the image block. As mentioned previously, the returned value may exceed the legal range of grayscale image pixel values, which is 0–255. Therefore, it is necessary to confirm that the adjusted value x' does not cause the restored pixel value to exceed the allowable range. If x' is not between the upper and lower bounds, the value must be adjusted to the closest boundary before returning to the pixel value. For any element value $x_0 \in R[0, j]$, the adjusted value x' should be within the lower and upper bounds as follows:

$$lower\_bound = \max_{i \in \{0,1,2,3\}} (LB[i]) \tag{7}$$

and

$$upper\_bound = \max_{i \in \{0,1,2,3\}} (UB[i]) \tag{8}$$

where

$$LB[i] = \begin{cases} -\infty & if\ Q[i,0] = 0, \\ min\left(\left(R[0,j] - \dfrac{A[i,j]}{Q[i,0]}\right), \left(R[0,j] + \left(255 - \dfrac{A[i,j]}{Q[i,0]}\right)\right)\right) & otherwise. \end{cases} \tag{9}$$

and

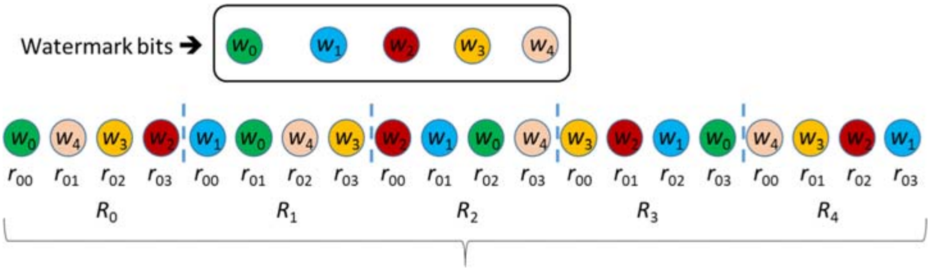

Watermark bits
w_0 w_1 w_2 w_3 w_4
w_0 w_4 w_3 w_2 w_1 w_0 w_4 w_3 w_2 w_1 w_0 w_4 w_3 w_2 w_1 w_0 w_4 w_3 w_2 w_1
r_{00} r_{01} r_{02} r_{03} r_{00} r_{01} r_{02} r_{03} r_{00} r_{01} r_{02} r_{03} r_{00} r_{01} r_{02} r_{03} r_{00} r_{01} r_{02} r_{03}
R_0 R_1 R_2 R_3 R_4

Fig. 2 Distribution of the robust watermark

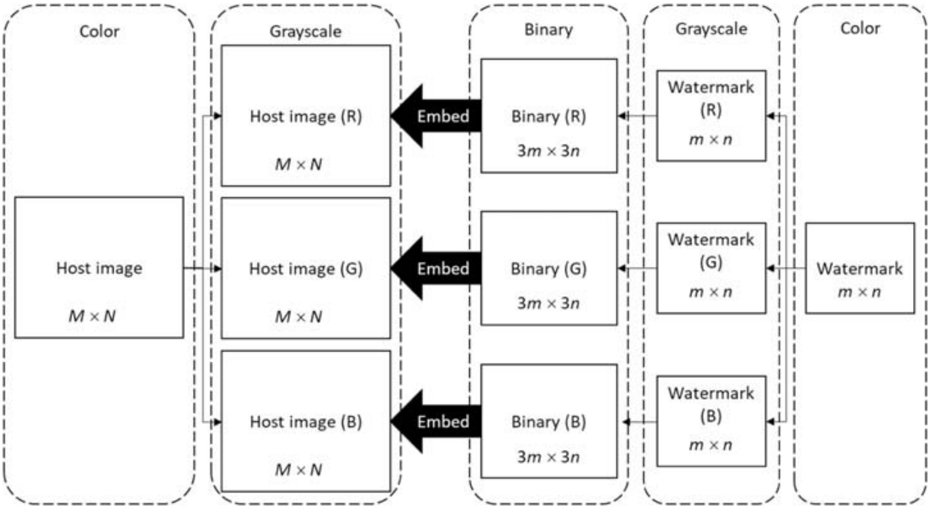


Fig. 3 The conversion process of watermark embedding with color images

$$UB[i] = \begin{cases} \infty & if\ Q[i,0] = 0, \\ max\left(\left(R[0,j]-\frac{A[i,j]}{Q[i,0]}\right), \left(R[0,j] + \left(255-\frac{A[i,j]}{Q[i,0]}\right)\right)\right) & otherwise. \end{cases} \quad (10)$$

3.1.2 Fragile watermark embedding

Similar to the robust watermark embedding stage, image H′ is first divided into 4 × 4 nonoverlapping blocks. Feature extraction and embedding are performed blockwise. In this section, the hidden approach of fragile watermarking for a single 4 × 4 block is explained.

Let (x, y) and p_i denote the coordinate of the block in H' and the pixel of the image block, respectively, where i = 0..15. A 256-bit message digest MD of the image block was first generated as follows:

$$MD = \mathrm{SHA256}(x\|y\|SK\|p_0 \gg 1\|p_1 \gg 1\|\ldots\|p_{15} \gg 1), \quad (11)$$

where SHA256 is a type of Secure Hash Algorithm 2, “>>” is a bitwise right shift operator, “||” is a concatenation operator, and SK is a secret key. MD is then split into 16-bit strings S_0, S_1, …, S_{15}, and each string is defined as follows:

$$S_i = (b_{i\times 16}, b_{i\times 16+1}, b_{i\times 16+2}, \ldots, b_{i\times 16+15})_2, \quad (12)$$

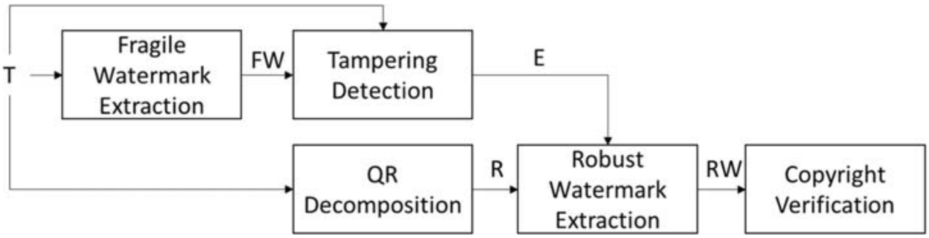


Fig. 4 Illustration of the extraction process

where $b_i \in \{0,1\}$ and i =0..15. The feature of the block, (i.e., the fragile watermark *FW*) is generated as follows:

$$FW = S_0 \oplus S_1 \oplus \ldots \oplus S_{15} = (w_0, w_1, w_2, \ldots, w_{15})_2, \tag{13}$$

where '$\oplus$' is a bitwise XOR operator and $w_i \in \{0,1\}$. Finally, each pixel p_i of the image block is modified to p_i' according to the following formula:

$$p_i^{'} = ((p_i \gg 1) \ll 1) + w_i, \tag{14}$$

where i = 0..15 and "<<" is a bitwise left shift operator. When all the blocks are processed according to the aforementioned method, the final watermarked image H'' can be obtained.

The entire watermark embedding algorithm is as follows:

```
**Algorithm** Embed($SK, K, H, RW$):
***Input:*** secret key $SK$, wavelength controller $K$, host image $H$, and robust watermark $RW$
***Output:*** the watermarked image $H''$ corresponding to $H$
$H' \leftarrow$ EmbedRobustWatermark($SK$, $K$, $H$, $RW$)
$H'' \leftarrow$ EmbedFragileWatermark($H'$)
**Return** $H''$
```

```
**Algorithm** EmbedRobustWatermark($SK, K, H, RW$):
***Input:*** secret key $SK$, wavelength controller $K$, host image $H$, and robust watermark $RW$
***Output:*** the watermarked image $H'$ corresponding to $H$
Let $M$ and $N$ be the height and width of $H$.
Let $m$ and $n$ be the height and width of $RW$.
Let $H'$ be the gray-scale watermarked image with the same size of $H$.
Let $q = \lfloor M/4 \rfloor \times \lfloor N/4 \rfloor$ be the number of blocks of $H$.
Let *shuffled* be a list of $q$ shuffled block numbers of $H$ according to a pseudo-random-number-generator
    seeded with $SK$.
Let *aqr* be a list of objects storing $A$, $Q$ and $R$ matrices (where $A = Q \times R$) generated from $H$ and rearranged
    according to *shuffled*.
**For** $y \leftarrow 0$ to $m - 1$ **Do**
    **For** $x \leftarrow 0$ to $n - 1$ **Do**
        $p \leftarrow y \times n + x$
        $watermark \leftarrow RW[y, x]$;
        **For** $i \leftarrow 0$ to 3 **Do**
            $A \leftarrow aqr[(p + i) \bmod q].A$
            $Q \leftarrow aqr[(p + i) \bmod q].Q$
            $R \leftarrow aqr[(p + i) \bmod q].R$.
        Use $A$, $Q$, $R$, $K$ *and* Eq.(2) to Eq.(10) to embed *watermark* in $R[0, i]$.
Use each newly generated $R$ matrix to revise the corresponding $A$ matrix in *aqr*.
Use *shuffled* and the new version of *aqr* to construct the watermarked image $H'$.
**Return** $H'$
```

```
Algorithm EmbedFragileWatermark($SK$, $H$):
Input: secret key $SK$ and host image $H$
Output: the watermarked image $H'$ corresponding to $H$
Let $M$ and $N$ be the height and width of $H$.
Let $bx = \lfloor N/4 \rfloor$ be the number of blocks each row of $H$.
Let $by = \lfloor M/4 \rfloor$ be the number of blocks each column of $H$.
Let $H'$ be the gray-scale watermarked image with the same size of $H$.
For $y \leftarrow 0$ to $by - 1$ Do
    For $x \leftarrow 0$ to $bx - 1$ Do
     Let $B$ be a $4 \times 4$ matrix representing a $4 \times 4$ image block of $H$.
        For $r \leftarrow 0$ to 3 Do
            For $c \leftarrow 0$ to 3 Do
                $B[r, c] \leftarrow H[y\times4 + r, x\times4 + c]$
     $w \leftarrow$ GenerateFragileWatermark($B$, $x\times4$, $y\times4$, $SK$)
     Embed $w$ into $B$ according Eq.(14) and use $B$ to construct the corresponding block of the watermarked
       image $H'$.
Return $H'$
```

```
Algorithm GenerateFragileWatermark($B$, $x$, $y$, $SK$):
Input: a $4 \times 4$ image block $B$, $B$'s X-axis and Y-axis coordinates, and secret key $SK$
Output: a list of the 16-bit authentication message $w$
$msg \leftarrow SK \,||\, x \,||\, y$
For $i \leftarrow 0$ to 3 Do
    For $j \leftarrow 0$ to 3 Do
        $b \leftarrow \lfloor B[i, j]/2 \rfloor$
        $msg \leftarrow msg \,||\, b$
Let $msgDigest$ be the 256-bit SHA-256 message digest corresponding to $msg$.
Divide $msgDigest$ into the eight 16-bit components $S_0$, $S_1$, …, and $S_{15}$.
$w \leftarrow S_0 \oplus S_1 \oplus S_2 \oplus \ldots \oplus S_{15}$
Return $w$
```

3.1.3 Extension for color images

If the host image and robust watermark are both color images, we propose a method of converting image formats so that the method proposed in this study can process color images. Figure 3 illustrates the conversion process. If the size of the color host image is $M \times N$ (M rows and N columns), the size of the color watermark is $m \times n$, where $m = \lfloor \lfloor M/4 \rfloor /3 \rfloor$ and $n = \lfloor \lfloor N/4 \rfloor /3 \rfloor$. Both the color host image and watermark image are decomposed into three single-tone images in a red–green–blue (RGB) model. The pixel of the decomposed image is an 8-bit value, which represents the intensity of the tone. Therefore, these decomposed images can be treated as grayscale images. The three grayscale watermark images are binarized. Fuller explanation of binarization will be presented in the next paragraph. Let us return to our main subject. After the color host image is decomposed and the watermark is binarized, the scenario is simplified into embedding a binary watermark into a grayscale host image. Therefore, these three binary watermarks can be embedded into the three monotone images of the host image by using the method presented in Section 3.1.1. The three monotone watermarked host images are separately watermarked with an authentication message according to the method presented in Section 3.1.2. Finally, the three monotone watermarked images are combined based on the RGB model to restore the color watermarked image.

We now return to the method of binarization which we postponed in the above paragraph. For the grayscale watermark, each pixel is converted into an 8-bit binary value, and each binary value is then transformed into a 3 × 3 binary block. For example, assume that the pixel value of the top left corner of the grayscale watermark is 74, whose binary value is $(01001010)_2$. The content of the corresponding 3 × 3 binary block is then represented as follows:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

When each pixel is sequentially transformed into a 3 × 3 binary block, a binary image can be obtained. This binary image is not a normal image. The position of the bottom right corner of the 3 × 3 block is not used. The value of this position is fixed to 1. For future research, this location will be used to hide other information.

3.2 Extracting process

Figure 4 illustrates the process of watermark extraction. The fragile watermark is first extracted from the test image T to detect tampering area. The output of tampering detection is error map E, which is a Boolean matrix that records the falsified blocks. After the test image is decomposed through QR factorization, the robust watermark can be extracted from the R matrix with the error map E as an input. The robust watermark can help us verify the copyright of the image. The method of tampering detection and copyright verification are thoroughly explained as below.

3.2.1 Tampering detection

Let T denote the test image of $M \times N$ pixels and E denote the error map, which is a Boolean matrix of the size of $\lfloor M/4 \rfloor \times \lfloor N/4 \rfloor$. The test image T is first divided into 4 × 4 nonoverlapping blocks. For each block, the feature is generated in the same way as shown in section 3.1.2 and then compared with the fragile watermark extracted from the least significant bitplane of the block. If the comparison result indicates that they are inconsistent, this block is considered to be tampered with, and the Boolean value ‘*true*’ is recorded in error map *E*. Otherwise, record the Boolean value ‘*false*’ in error map *E*. The detailed steps are listed as follows, and the output is error map *E*, which can display the falsified area.

```
Algorithm GenerateErrorMap($T$, $SK$):
Input: test image $T$ and secret key $SK$
Output: the error map $E$ corresponding to $T$
Let $M$ and $N$ be the height and width of $T$, respectively.
Let $bx = \lfloor N/4 \rfloor$ be the number of blocks each row of $T$.
Let $by = \lfloor M/4 \rfloor$ be the number of blocks each column of $T$.
Let $E$ be a $by \times bx$ matrix representing the error map.
For $y \leftarrow 0$ to $by - 1$ Do
    For $x \leftarrow 0$ to $bx - 1$ Do
        Let $A$ be a $4 \times 4$ matrix.
        For $row \leftarrow 0$ to 3 Do
            For $col \leftarrow 0$ to 3 Do
         $A[row, col] = T[y\times4 + row, x\times4 + col]$
     $w \leftarrow GenerateFragileWatermark(A, x\times4, y\times4, SK)$
        $ew \leftarrow ExtractFragileWatermark(A)$
        If $w = ew$ Then
            $E[y, x] \leftarrow false$    // untampered
        Else
            $E[y, x] \leftarrow true$    // tampered
Return $E$

Algorithm ExtractFragileWatermark($B$):
Input: a $4 \times 4$ image block $B$
Output: a list of the 16-bit authentication message $w$
Let $w$ be a list of the 16-bit authentica-tion message.
For $row \leftarrow 0$ to 3 Do
    For $col \leftarrow 0$ to 3 Do
        $w[row \times 4 + col] \leftarrow A[row, col] \bmod 2$
Return $w$
```

3.2.2 Copyright verification

In the previous section, each watermark bit has four copies hidden in the first row of the R matrix; thus, the image is first factorized through QR decomposition to retrieve these copies. For any element value x_0 of the first row of R, the embedded bit w is decoded as follows:

$$w = \begin{cases} 1, & f(x_0) \geq 0 \\ 0, & otherwise. \end{cases} \tag{15}$$

According to the robust watermark embedding process mentioned in Section 3.1.1, multiple copies of the watermark bit are obtained from different blocks. If each copy is the same, the value of the watermark bit can be determined. However, the image may be altered, which results in inconsistencies in each copy of the watermark. As a result, determining whether the embedded watermark bit is 0 or 1 is difficult. A feasible solution is to adopt a majority decision, that is to say that the watermark bit value depends on the most frequently occurring value of the four copies. For example, if the value of most copies is 1, the value of the watermark bit is 1, and vice versa. However, adopting a majority decision has several limitations. If the number of copies is an even number, the majority decision may not be determined in some cases. In addition, the majority is not necessarily correct if most of the blocks where the copy is located are destroyed. In this study, the result of tampering detection (i.e., error map E) is used to assist in determining the watermark bit. Before deciding the value of the watermark bit, the result of tampering detection is used to judge whether the blocks are altered. If all copies are consistent and are hosted in untampered blocks, then these copies can accurately determine the value of the watermark bit. If these copies are inconsistent, the value of the watermark bit is determined using the copies extracted from the blocks that have not

Table 1 Experimental result for 64 color test images under cropping attacks

CR	PSNR	NC	FNR	FPR
0.05	20.9046	1.0000	0.0000	0.0057
0.10	17.7831	0.9999	0.0000	0.0042
0.15	15.8701	0.9999	0.0000	0.0037
0.20	14.5821	0.9997	0.0000	0.0090
0.25	13.5403	0.9992	0.0000	0.0003
0.30	12.7334	0.9983	0.0000	0.0004
0.35	12.0651	0.9967	0.0000	0.0076
0.40	11.4696	0.9936	0.0000	0.0212
0.45	10.9337	0.9909	0.0000	0.0055
0.50	10.4523	0.9811	0.0000	0.0344
0.55	10.0455	0.9735	0.0000	0.0334
0.60	9.6668	0.9623	0.0000	0.0317
0.65	9.3262	0.9422	0.0000	0.0377
0.70	8.9989	0.9134	0.0000	0.0460
0.75	8.7014	0.8772	0.0000	0.0707
0.80	8.4364	0.8417	0.0000	0.0557
0.85	8.1611	0.7848	0.0000	0.0064
0.90	7.9221	0.6770	0.0000	0.1172
0.95	7.6762	0.4562	0.0000	0.3941
1.00	7.4577	0.0000	0.0000	0.4531

been altered. If none of the four copies are from the untampered blocks, then the watermark bits are determined using these four copies according to the majority voting principle. It is reasonable to assume that copies extracted from untampered blocks are consistent. However, if the falsification of blocks is misjudged, that is, those blocks have been tampered with, but the opposite was concluded in the process, then these copies may be inconsistent. When such situations occur, this study also adopts majority voting principle to determine the value of the watermark bit. The aforementioned misjudgment is termed a false negative. The false negative may not be serious because the probability of its occurrence is low, and the probability is analyzed in a subsequent section. The robust watermark extraction algorithm is as follows:

```
**Algorithm** ExtractWatermarkByErrorMap($T$, $E$, $SK$, $m$, $n$):
***Input:***   Test image $T$, error map $E$, secret key $SK$, watermark height $m$, and watermark width $n$
***Output:***  The extracted watermark image $RW$ of size $m \times n$ from $T$
 Let $M$, $N$ be the height and width of $T$, respectively.
 Let $RW$ be an $m \times n$ binary watermark image.
 Let $q = \lfloor M/4 \rfloor \times \lfloor N/4 \rfloor$ be the number of blocks of $T$.
 Let $shuffled$ be a list of $q$ shuffled block numbers of $T$ according to a pseudo-random-number-generator
     seeded with $SK$.
**For** $y \leftarrow 0$ to $m - 1$ **Do**
    **For** $x \leftarrow 0$ to $n - 1$ **Do**
        $p \leftarrow y \times n + x$
    Let $R_i$ be the $R$ matrix corresponding to $shuffled[(p + i) \bmod q]$ for     $i = 0..3$.
    Let $w_i$ be the decoded watermark according to $R_i[0, i]$ for $i = 0..3$.
        $watermark \leftarrow 0$
        $all\_tampered \leftarrow$ true
    Let $U$ be a list containing watermark bits decoded from untampered blocks.
        **For** $i \leftarrow 0$ to 3 **Do**
            $ex \leftarrow shuffled[(p + i) \bmod q] \bmod n$
            $ey \leftarrow \lfloor shuffled[(p + i) \bmod q]/n \rfloor$
            **If** $E[ey, ex] = false$ **Then**
                $all\_tampered \leftarrow false$
                $U.Add(w_i)$
        **If** NOT $all\_tampered$ **Then**
      **If** $Average(U)$ >= 0.5 **Then** $watermark \leftarrow 1$
        **Else**
      **If** $(w_0 + w_1 + w_2 + w_3)/4$ >= 0.5 **Then** $watermark \leftarrow 1$
        $RW[y, x] \leftarrow watermark$
**Return** $RW$
```

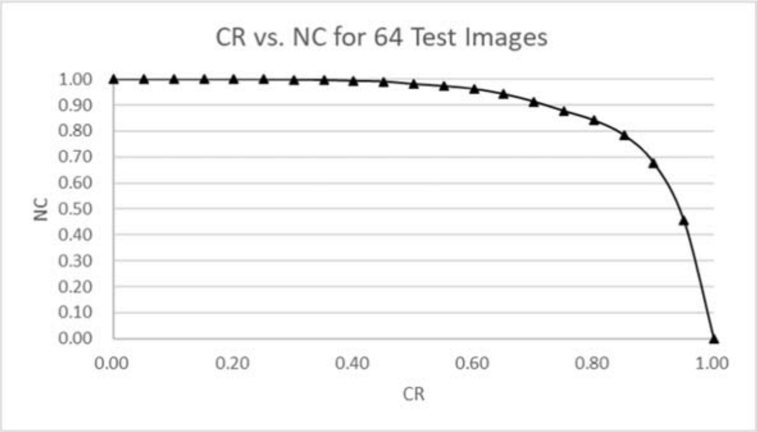


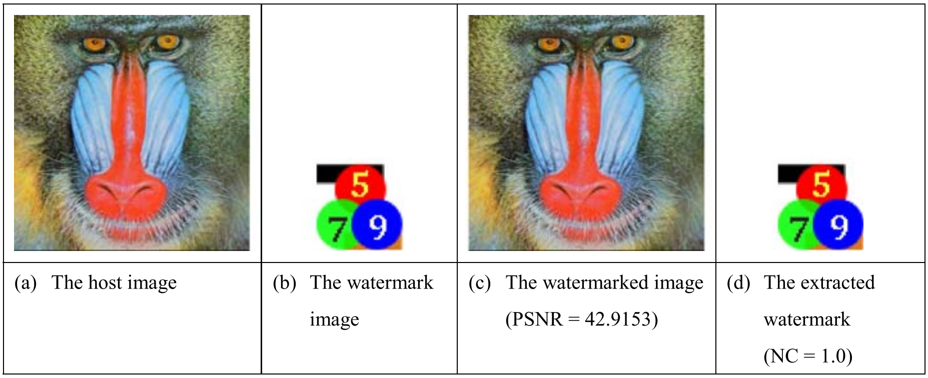
Fig. 5 Line chart of CR versus NC

3.2.3 Extension for color images

When the test image is a color image, it is decomposed into three monotone images based on the RGB model. Each of the three images uses the method proposed in Section 3.2.1 to generate their respective error maps. Then, the method in Section 3.2.2 is used, with respective error maps, to extract the robust watermark from the three monotone images. The three extracted watermarks of size ($3\ m \times 3n$) are binary and meaningless images. Each binary watermark is then restored to a grayscale image by reversing the binarization mentioned in Section 3.1.3. Finally, the three grayscale watermark images based on the RGB model are combined to obtain the color watermark image.

4 Experimental results and analysis

This study primarily focuses on color images. The size of the watermark image depends on that of the host image. According to Section 3.1.3, if the height and width of the host image are



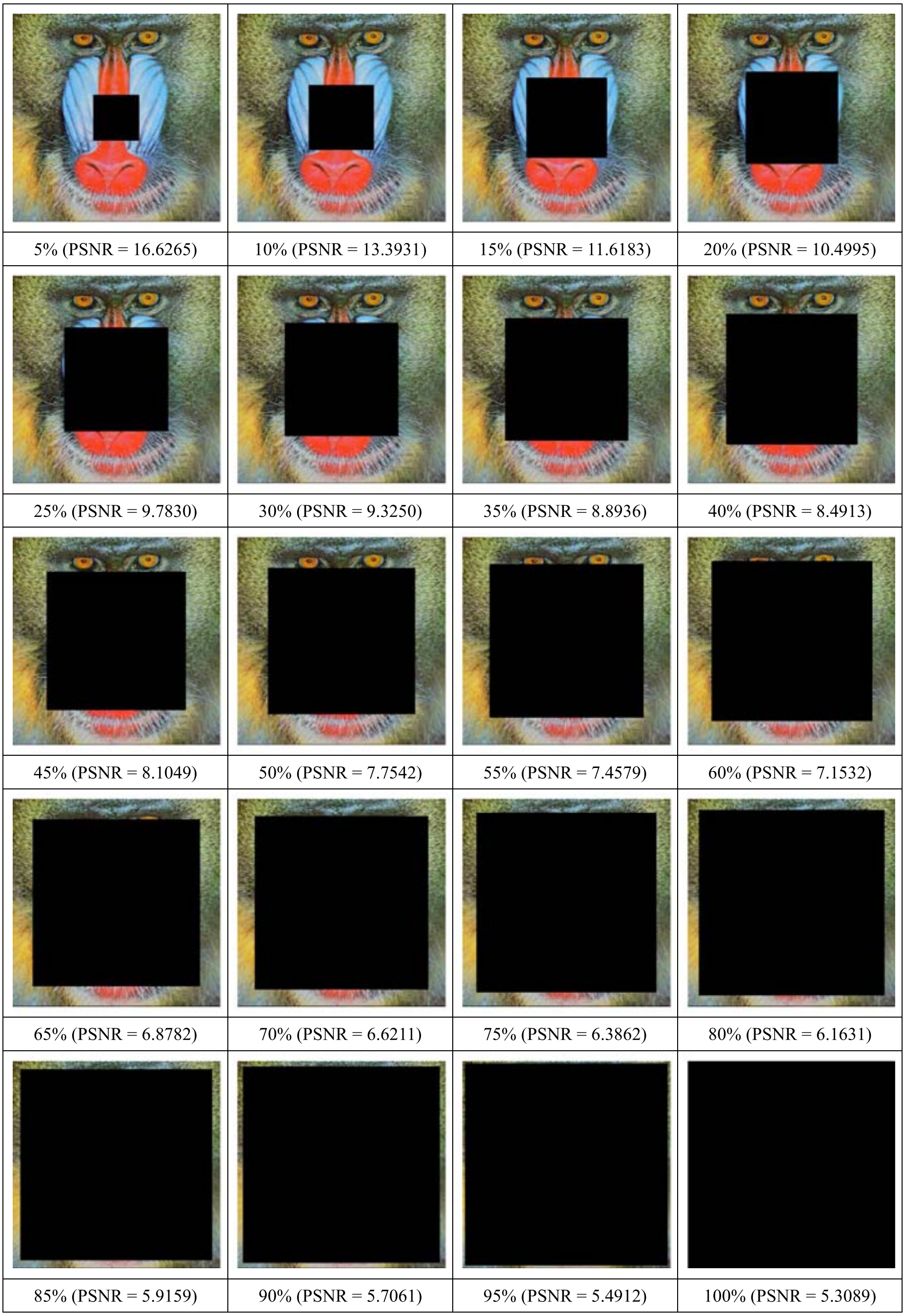
(a) The host image

(b) The watermark image

(c) The watermarked image (PSNR = 42.9153)

(d) The extracted watermark (NC = 1.0)

Fig. 6 The experimental results. **a** The host image **b** The watermark image **c** The watermarked image (PSNR = 42.9153) **d** The extracted watermark (NC = 1.0)



5% (PSNR = 16.6265)

10% (PSNR = 13.3931)

15% (PSNR = 11.6183)

20% (PSNR = 10.4995)

25% (PSNR = 9.7830)

30% (PSNR = 9.3250)

35% (PSNR = 8.8936)

40% (PSNR = 8.4913)

45% (PSNR = 8.1049)

50% (PSNR = 7.7542)

55% (PSNR = 7.4579)

60% (PSNR = 7.1532)

65% (PSNR = 6.8782)

70% (PSNR = 6.6211)

75% (PSNR = 6.3862)

80% (PSNR = 6.1631)

85% (PSNR = 5.9159)

90% (PSNR = 5.7061)

95% (PSNR = 5.4912)

100% (PSNR = 5.3089)

Fig. 7 Cropping attacks to the watermarked image with different cropping ratios

M and N, respectively, the height and width of the watermark image are $\lfloor \lfloor M/4 \rfloor /3 \rfloor$ and $\lfloor \lfloor N/4 \rfloor / 3 \rfloor$, respectively. In our experiments, the host image is a color image with 512 × 512 pixels. Therefore, the size of the watermark is 42 × 42 pixels. The wavelength controller K is 31.

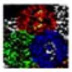
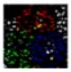
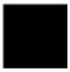
5% (NC = 1.0)	10% (NC = 1.0)	15% (NC = 1.0)	20% (NC = 0.9998)
25% (NC = 0.9994)	30% (NC = 0.9986)	35% (NC = 0.9971)	40% (NC = 0.9940)
45% (NC = 0.9913)	50% (NC = 0.9817)	55% (NC = 0.9741)	60% (NC = 0.9630)
65% (NC = 0.9429)	70% (NC = 0.9142)	75% (NC = 0.8779)	80% (NC = 0.8423)
			
85% (NC = 0.7855)	90% (NC = 0.6776)	95% (NC = 0.4568)	100% (NC = 0.0)

Fig. 8 Extracted watermark images under various cropping attacks

The most crucial requirements for robust watermarking schemes are imperceptibility and robustness. Imperceptibility means that the watermarked image must be visually indistinguishable from the original host image, and robustness denotes that the extracted watermark should exhibit some similarity with the original watermark. Generally, robust watermarking schemes calculate the peak signal to noise ratio (PSNR) of the watermarked image against the host image to measure imperceptibility. The higher the PSNR is, the better the imperceptibility is [22]. Robustness is generally measured through normalized correlation (NC) of the extracted watermark against the original one. The NC value ranges from 0 to 1, where 1 and 0 indicate that the two images are exactly identical and completely different, respectively [23]. As regards the tampering detection performance, the following performance evaluation measurements are defined. Let *FN*, *FP*, *TN*, and *TP* be the number of false negative pixels, false positive pixels, true negative pixels, and true positive pixels, respectively. The tampering ratio (*TR*), false

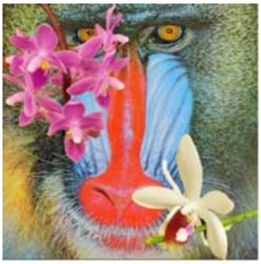
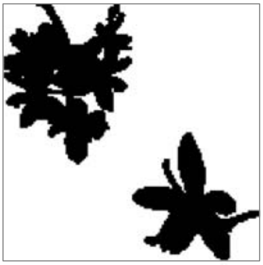
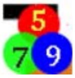
		
(a) Collage attack (TR = 0.25, PSNR = 15.0195)	(b) Authentication result (FNR = 0.0, FPR = 0.0340)	(c) Extracted watermark (NC = 0.9994)

Fig. 9 Collage attack. **a** Collage attack (TR = 0.25, PSNR = 15.0195) **b** Authentication result (FNR = 0.0, FPR = 0.0340) **c** Extracted watermark (NC = 0.9994)

negative rate (*FNR*), and false positive rate (*FPR*) used to measure tampering detection accuracy are respectively defined as follows [23, 24]:

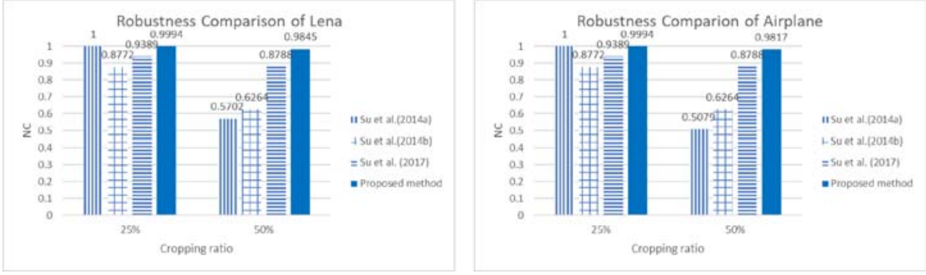
$$TR = \frac{FN + TP}{M \times N}. \tag{16}$$

$$FNR = \frac{FN}{FN + TP}. \tag{17}$$

$$FPR = \frac{FP}{FP + TN}. \tag{18}$$

4.1 Experimental results

In this study, 64 color images from CVG-UGR image database were used as the host images [25]. The average PSNR value of 64 images after embedding watermarks is 42.9153. Various attacks with different cropping ratios were conducted on the 64 images, and the average experimental data are listed in Table 1. In Table 1, CR denotes the cropping ratio of the watermarked image. The PSNR indicates the difference between the cropped watermarked image and original watermarked image. The NC represents the similarity between the original watermark and the watermark extracted from the cropped image. The FNR and FPR represent the results of the tampering detection of the watermarked image. Columns 2 to 5 represent the average of the PSNR, NC, FNR, and FPR for 64 images corresponding to different cropping ratios. Figure 5 shows a line chart with NC values corresponding to the cropping ratio. Table 1 and Fig. 5 indicate that when the cropping ratio reaches 70%, the extracted watermark is still 90% similar to the original watermark. When the cropping ratio is up to 80%, the similarity between the extracted watermark and original watermark can still reach 80% or higher. These data show that the method proposed in this study exhibits satisfactory robustness against cropping attacks. To save space, only the results of the attacks on Bamboo image are presented here. Figure 6a and b are the host image and watermark image, respectively. The result of the robust and fragile watermark embedding processes is shown in Fig. 6c, and the PSNR value is 42.9153. When the PSNR value is higher than 30, the human eye cannot distinguish differences from the original image. Figure 6d is the watermark extracted from Fig. 6c, and the NC value is 1.0, which indicates that the extracted watermark is exactly the same as the original watermark. Figure 7 shows the simulated cropping attacks with various cropping ratios. Figure 8 shows the watermarks extracted from the respective attacked images of Fig. 7 and their NC values. When the cropping ratio is as high as 70%, the extracted watermark still exhibits a 90% similarity to the original watermark. In general, people who steal images may modify the image to avoid copyright disputes. The modification is a type of collage attack, which is to add other objects to the image. Figure 9a is the result of a simulated collage attack on Fig. 6a, and the object added on Fig. 6a is an image made by us. Figure 9b is the authentication result based on the error map, and Fig. 9c is the extracted robust watermark. The NC value shows that the proposed method exhibits satisfactory robustness against collage attacks.



(a) (b)

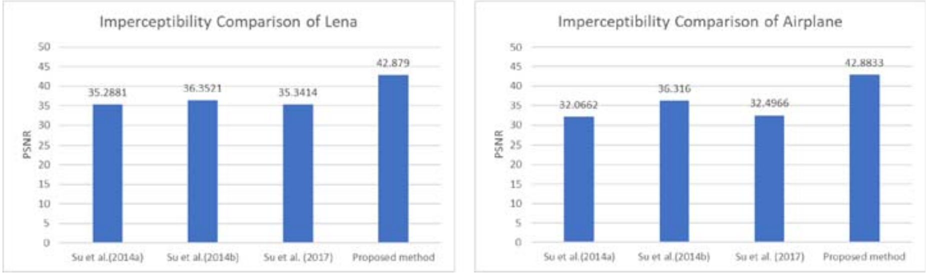
Fig. 10 Comparison of the robustness against cropping attacks

For fairness, the comparisons in the Figs. 10 and 11 are based on the experimental results of Lena and Airplane as the host images. Figure 10 shows the NC values obtained using the proposed method and other methods at cropping ratios of 25% and 50%. For the cropping ratio of 25%, our proposed method and those proposed by Su et al. [9, 16] have an NC value of more than 90%. When the cropping ratio is expanded to 50%, only the NC value obtained using this method can reach more than 90%, and the NC value of Su et al. [9] is less than 10% of the NC value of the proposed method. Regardless of whether the cropping ratio is 25% or 50%, the NC value of the proposed method is close to 1, which indicates that the extracted watermark is highly similar to the original watermark. Figure 11 compares the PSNR value of the proposed method and that of other methods and indicates that the invisibility of the proposed method is superior to that of other methods. It seems reasonable to conclude that the proposed method does not lead to unsatisfactory imperceptibility because of improvements in robustness.

4.2 Analysis and discussions

4.2.1 Trade-off of imperceptibility and robustness

The trade-off between imperceptibility and robustness of a robust watermarking scheme is generally controlled by a parameter, and in this study, the wavelength controller *K* plays this role. Researchers usually pre-test different parameter values to select appropriate one. This study also performed pretests. The experiments were performed with *K* values from 1 to 150,



(a) (b)

Fig. 11 Comparison of the imperceptibility of methods

and TRs from 0% to 95% to analyze the effect of *K* values on PSNR and NC values. Because of the limited length of the article, only some of the experimental data are presented in Table 2, and complete experimental data is drawn as a line chart shown as Fig. 12. Figure 12a shows the effect of different *K* values on imperceptibility. A larger *K* value can increase the PSNR value of the watermarked image, which indicates improved imperceptibility. However, as

Table 2 Pretest results for parameter *K*

K	PSNR	NC values for cropping attacks with various cropping ratios					
		0%	10%	25%	50%	75%	95%
1	11.9078	1.0000	0.9999	0.9988	0.9773	0.8673	0.4291
2	18.1774	1.0000	0.9999	0.9988	0.9773	0.8668	0.4253
3	20.5748	1.0000	0.9999	0.9988	0.9770	0.8668	0.4270
4	22.6352	1.0000	0.9999	0.9988	0.9770	0.8666	0.4269
5	25.1022	1.0000	0.9999	0.9988	0.9770	0.8671	0.4263
6	27.4009	1.0000	0.9999	0.9988	0.9772	0.8671	0.4263
7	29.1543	1.0000	0.9999	0.9988	0.9772	0.8669	0.4264
8	30.4047	1.0000	0.9999	0.9988	0.9773	0.8670	0.4259
9	31.0332	1.0000	0.9999	0.9988	0.9772	0.8670	0.4255
10	30.8475	1.0000	0.9999	0.9988	0.9772	0.8670	0.4274
11	32.0797	1.0000	0.9999	0.9988	0.9772	0.8670	0.4276
12	33.4853	1.0000	0.9999	0.9988	0.9772	0.8668	0.4266
13	34.6117	1.0000	0.9999	0.9985	0.9765	0.8654	0.4228
14	35.5148	1.0000	0.9999	0.9983	0.9756	0.8640	0.4182
15	35.5722	1.0000	0.9999	0.9985	0.9762	0.8646	0.4207
16	34.8812	1.0000	0.9999	0.9988	0.9760	0.8642	0.4189
17	35.8887	1.0000	0.9999	0.9986	0.9756	0.8639	0.4196
18	37.1376	1.0000	0.9999	0.9982	0.9753	0.8629	0.4173
19	37.4639	1.0000	0.9999	0.9983	0.9754	0.8641	0.4193
20	38.5466	1.0000	0.9999	0.9983	0.9751	0.8632	0.4178
21	38.592	1.0000	0.9999	0.9981	0.9754	0.8637	0.4190
22	37.3319	1.0000	0.9999	0.9986	0.9754	0.8631	0.4171
23	38.4463	1.0000	0.9999	0.9982	0.9748	0.8622	0.4164
24	38.7155	1.0000	0.9999	0.9982	0.9744	0.8619	0.4141
25	40.0597	1.0000	0.9999	0.9981	0.9743	0.8618	0.4144
26	40.2836	1.0000	0.9999	0.9981	0.9740	0.8616	0.4138
27	41.0351	1.0000	0.9999	0.9981	0.9741	0.8618	0.4144
28	39.3377	1.0000	0.9999	0.9981	0.9741	0.8618	0.4139
29	40.5286	1.0000	0.9999	0.9981	0.9745	0.8621	0.4145
30	41.055	1.0000	0.9999	0.9985	0.9749	0.8625	0.4152
31	41.1817	1.0000	0.9999	0.9987	0.9772	0.8666	0.4276
32	42.3439	1.0000	0.9999	0.9988	0.9767	0.8663	0.4258
33	42.4519	1.0000	0.9999	0.9987	0.9766	0.8659	0.4234
34	41.2865	0.9999	0.9997	0.9985	0.9767	0.8663	0.4254
35	41.4864	0.9991	0.9988	0.9968	0.9743	0.8634	0.4254
36	42.0736	0.9998	0.9994	0.9974	0.9745	0.8644	0.4249
37	43.4895	0.9999	0.9999	0.9982	0.9753	0.8637	0.4247
38	43.5068	1.0000	0.9994	0.9975	0.9743	0.8635	0.4225
39	43.4414	0.9996	0.9995	0.9975	0.9735	0.8625	0.4224
40	41.9417	0.9984	0.9981	0.9954	0.9695	0.8593	0.4203
41	42.5212	0.9982	0.9970	0.9926	0.9668	0.8558	0.4197
42	44.0222	0.9993	0.9987	0.9956	0.9698	0.8581	0.4225
43	44.0731	0.9997	0.9994	0.9961	0.9703	0.8601	0.4212
44	44.1596	0.9996	0.9987	0.9948	0.9693	0.8595	0.4208
45	44.0644	0.9982	0.9972	0.9930	0.9664	0.8561	0.4190

Fig. 12b indicates, larger K values may cause the extracted watermark not to be complete. The incompleteness results from that the embedded fragile watermark may destroy the previously embedded robust watermark. Observing the NC value of a 0% cropping ratio column of Table 2, we can see that the watermark can be completely extracted without destruction if the K value is less than 33. Figures 12c–f are NC values corresponding to different K values under various cropping ratios. When the K value is higher than 40, the NC value has a relatively obvious downward trend. It follows from what has been observed that the K value should not exceed 33 to maintain the integrity of the watermark and should not less than 25 to ensure the PSNR value to be greater than 40. View in this light, this study set the value of K at 31.

4.2.2 Embedding payload

In theory, the embedding payload of the proposed scheme can be derived as follows:

$$\frac{\frac{{}^{M}/_{4}}{3}\times\frac{{}^{N}/_{4}}{3}\times 24}{M\times N\times 3}=\frac{1}{18}\cong 0.05556. \tag{19}$$

Figure 13 shows the comparison of the embedding payload using different methods. This figure indicates that the embedding payload of the proposed method is 1.78 times higher than that of the other methods.

4.2.3 Utility of tampering detection

Fragile watermarks are generally used for tampering detection. In this study, fragile watermarking has another role, which is to enhance the ability of the robust watermark against cropping attacks. To show whether the fragile watermark actually functions, Fig. 14 shows robustness against cropping attacks when robust watermarks are extracted with and without the assistance of fragile watermark. Extracting the robust watermark without the fragile watermark means that the value of the bit is determined only according to the mode of all the copies of the watermark bit and without the assistance of the error map. For the attack of various cropping ratios, the NC value of the robust watermark without the fragile watermark and that with the fragile watermark is drawn as a line graph (Fig. 14). This figure indicates that when an assistant of the fragile watermark is available, the robustness of the extracted watermark improves. In particular, when the cropping ratio is large, the fragile watermark can be more effective. It is one of the crucial contributions of the proposed method in enhancing robustness against cropping attacks with the results of tampering detection. The results presented in Fig. 14 confirm the merit of this study.

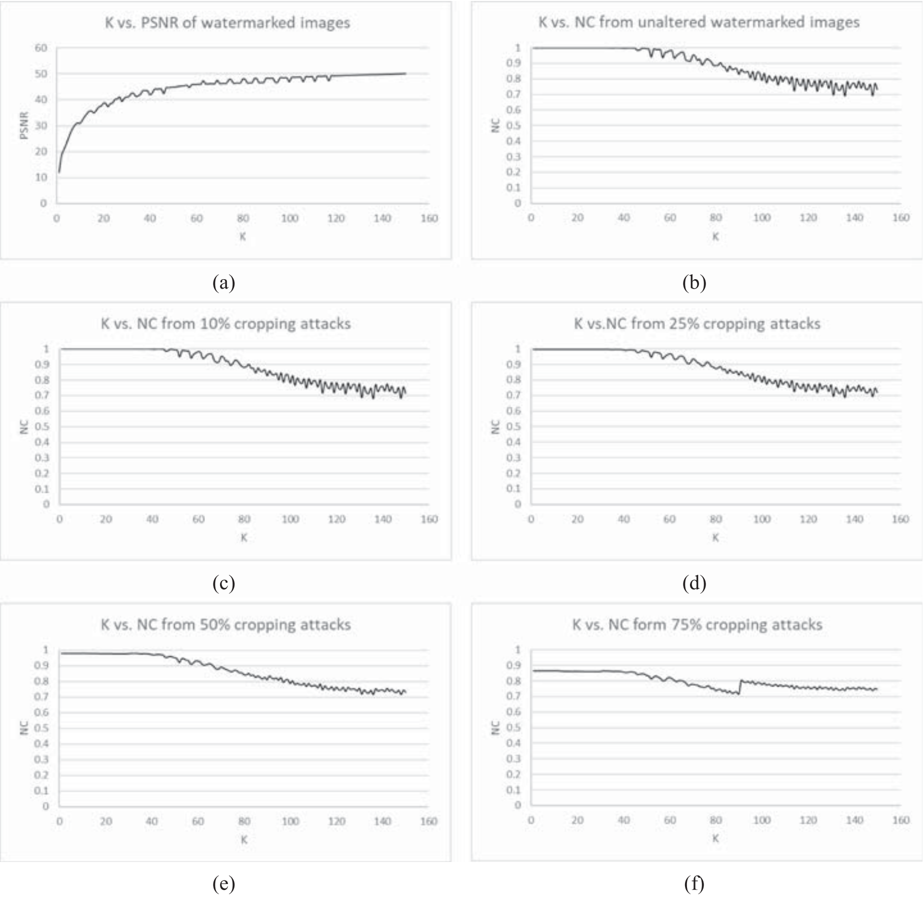
According to Section 3.2, if the blocks where the four copies of the watermark bit are located are evaluated to not have been tampered with, these copies should be consistent. If that evaluation is a false negative, then these copies may be inconsistent. However, the probability of a false negative is extremely low, which can also be validated from the data in Table 1. Under different cropping ratios, the average FNR values of 64 test images approaches 0. In fact, the theoretical value of the FNR of the proposed method is 2^{-16}.

4.2.4 Time complexity

The embedding procedure consists of the “EmbedRobustWatermark” and “EmbedFragileWatermark” algorithms. In the “EmbedRobustWatermark” algorithm, $m \times n$ blocks of the host image are shuffled using the Fisher-Yates shuffle algorithm in time $T_{\text{shuffle}} = \mathrm{O}(mn)$. Then, the host image is factorized by QR decomposition in a block-wise manner, and the time complexity is $T_{\text{qr}} = \mathrm{O}(mns^3)$ since the time complexity for facotrizing a $s \times s$ matrix is $O(s^3)$. Next, the three-layer for-loops for altering the R matrices takes $T_{\text{alter}} = \mathrm{O}(mn)$ since the inner for-loop can be done in $O(1)$ time. Finally, the inverted QR decomposition also takes T_{qr} time. Therefore, the time complexity for the “EmbedRobustWatermark” algorithm is given by.

$$\begin{aligned} T_{\text{rwembed}} &= T_{\text{shuffle}} + T_{\text{qr}} + T_{\text{alter}} + T_{\text{qr}} = \mathrm{O}(mn) + \mathrm{O}\left(mns^3\right) + \mathrm{O}(mn) + \mathrm{O}\left(mns^3\right) \\ &= O\left(mns^3\right). \end{aligned}$$

Note that the matrix is with fixed 4-by-4 dimensions, so effective T_{rwembed} will be $O(mn)$. Regarding the “EmbedFragileWatermark” algorithm, the time complexity is $T_{\text{fwembed}} = O(mn)$ because performing SHA-256 and hiding the authentication message



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 12 Pretest results for parameter K

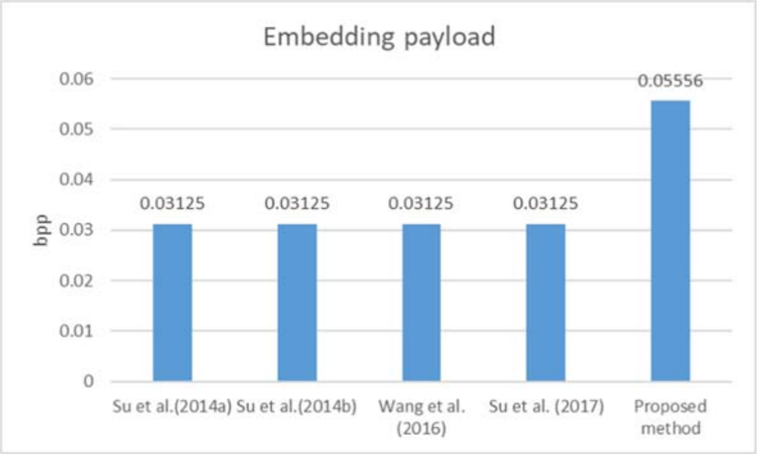


Fig. 13 Comparison of embedding payload of different methods

for a 4 × 4 block can be done in $O(1)$ time. In conclusion, the time complexity for the embedding procedure is given by.

$$T_{\text{embed}} = T_{\text{rwembed}} + T_{\text{fwembed}} = O(mn) + O(mn) = O(mn).$$

Because the watermark extraction is the reversion operation of the watermark embedding, thus the time complexity of watermark extraction is also $O(mn)$.

4.2.5 Robustness against common image processing

When the watermarked image is altered by malicious attacks, some copies of the watermark bit may survive because malicious attacks usually do not change the entire image. However, if the image is modified by some common image processing, all pixel values will be changed, and thus, the elements of the R matrix are all modified. Also, the four copies of the watermark may

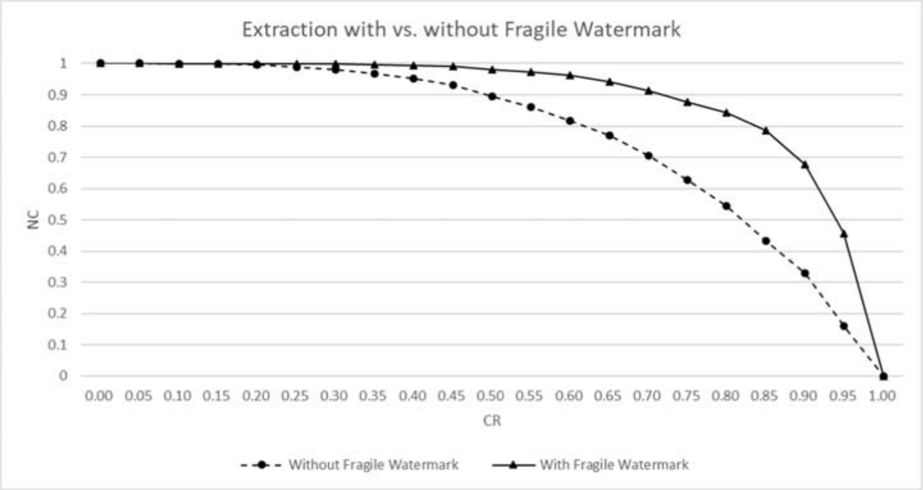
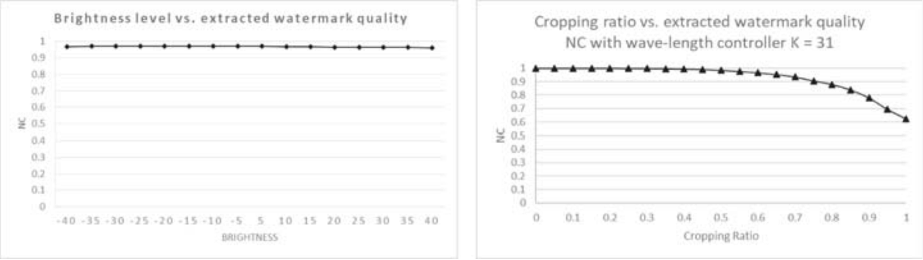


Fig. 14 Effect of fragile watermarking on improving robustness



(a) (b)

Fig. 15 Robustness against brightness adjustment and cropping attacks

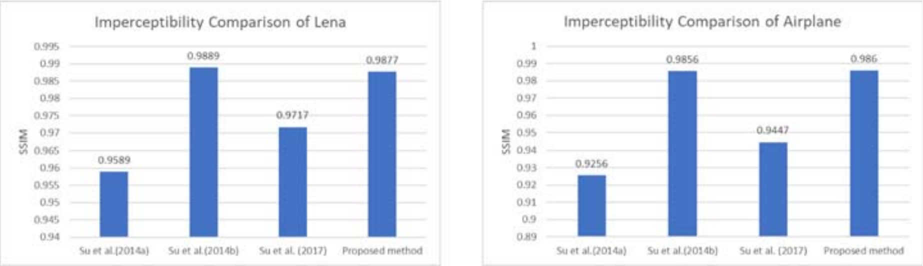
have been changed. One possible way to cope with this situation is to add an extra step to embed each watermark bit again. At this extra step, each watermark bit is not carried by a single element but carried by the relationship between two elements of the R matrix. Note that the extra step is taken after all copies of the watermark are embedded into the R matrices and before the pixel values are recovered. Let a and b denote the $R_i[0,0]$ and $R_i[0,1]$ elements of the R matrix, respectively. For the watermark bit w_i, a is set to a' as follows:

$$a' = b + [(w_i - 0.5) \times 2] \times [1 - (f(a) \oplus f(b)) \times 0.5] \times \lambda, \tag{20}$$

where ‘$\oplus$’ is a bitwise XOR operator and $f(\cdot)$ is the sine function as Eq. 1. It is still necessary to make adjustment if the modified value a' is out of bounds according to Eqs. 7 and 8. At the stage of watermark extraction, the result of tampering detection (i.e., error map E) is used to assist in determining the watermark bit. The watermark bit value is determined by the copies embedded in untampered blocks as described in section 3.2.2. However, when none of the four copies are from the untampered blocks, the watermark bit w_i is determined in the following manner.

$$w_i = \begin{cases} 1, & R_i[0,0] \geq R_i[0,1] \\ 0, & \quad otherwise. \end{cases} \tag{21}$$

To examine the performance of the extra step mentioned above, we also use the 64 color images from CVG-UGR image database as the host images and Fig. 6b as the watermark. We simulated different brightness adjustments to the watermarked image, and Fig. 15a is the line chart with average NC values of the extracted watermark corresponding to various brightness



(a) (b)

Fig. 16 Comparison of the imperceptibility based on SSIM

level. This diagram shows that the NC values against different brightness adjustments are all above 0.95, which indicates that the quality of the extracted watermark is satisfactory. Figure 15b is the line chart with average NC values of the extracted watermark corresponding to various cropping ratios. This diagram helps to prove that the proposed extra embedding step does not reduce the ability to resist cropping attacks. The average NC value can still reach 90% or higher when the cropping ratio is up to 75%. However, the imperceptibility is not as good as the previous experiment in section 4.1. The average PSNR value of 64 watermarked images is 32.3211. Therefore, how to balance the robustness against image processing and imperceptibility of the watermarked image is our future research topic.

4.2.6 Other measurement of imperceptibility

Most studies use PSNR to measure imperceptibility of the watermarking scheme, but some use structural similarity (SSIM) index [9, 15, 16]. The SSIM index is designed based on human visual system, and hence measures the perceptual difference between two images [26]. The perceptual quality is believed to be good if the value of SSIM is above 0.9 [27]. Figure 16 shows the comparisons of the imperceptibility based on SSIM. This figure indicates that our method achieves satisfactory perceptual quality, and the SSIM of the proposed method is almost the best of these methods.

4.2.7 Other discussions

This study used a sine function to design the embedding rule. The advantage of using the sine function is that it can process real numbers, and the value of the *R* matrix includes real values. In addition, the sinusoidal function has two distinct states, namely positive and negative cycles, which correspond to 0 and 1 of the watermark bit. In addition, because we embedded both the robust and fragile watermarks, the additional benefit of this study is that the proposed method can have two functions: tampering detection and copyright protection. Moreover, the proposed method can extract the robust watermark without using the original host image and original watermark; thus, the method exhibits improved practicability.

This study used a secret key to scramble the image block before embedding the robust watermark. Therefore, even if the attacker knows the algorithm of this study but does not have this secret key, they cannot restore the order of the blocks and acquire the watermark. Therefore, it is possible to prevent an attacker from modifying the embedded watermark content to claim the copyright of the image. In addition, this study includes a secret key when generating the image authentication message. Therefore, the attacker cannot deduce the authentication message according to the algorithm of the study and cannot deliberately change the content of the verification message to invalidate the tampering detection. In summary, this method satisfies the Kirchhoff's principle: all algorithms must be public, and only the key is secret. Kirchhoff's criterion is frequently used to evaluate the usefulness of cryptographic methods [9, 15]. If the secret key must be passed over a public network to a specific person, asymmetric cryptography can be used to protect the security of the secret key during transmission. The secret key can be encrypted with the public key of the sender before transmitting it. After receiving the specific key, the receiver can use his/her private key to decrypt the content of the secret key.

5 Conclusions

This study proposed a QR-based digital watermarking method for improving robustness against cropping attacks. Our approach has the following merits: (1) The watermark embedding rule is designed based on a sine function that can handle real values of the *R* matrix; (2) each watermark bit has multiple copies spread across different blocks, so that the survival probability of the watermark bit may increase; (3) robustness against cropping attacks is enhanced by using fragile watermark; (4) the modifiable ranges of elements of the matrix *R* are analyzed; therefore, the hidden space of the *R* matrix is effectively used to improve the hiding capacity further; (5) the method can process the color host image and watermark, and the watermark can be extracted without the original host image and original watermark; and (6) the proposed method has a dual function: copyright verification and tampering detection. The experimental results of the proposed method are superior to those of the other methods in terms of imperceptibility, robustness against cropping attacks, and embedding payload. The main purpose of this study is the enhance the robustness against cropping attacks; therefore, we do not pay much attention to the robustness against common image processing. In this study, parameter *K* balanced imperceptibility and robustness. The same *K* value was used for the four elements of the *R* matrix. In fact, setting different *K* values for different hidden locations may increase imperceptibility and robustness. Therefore, in future research, the evolutionary algorithm will be used to determine the optimal *K* value for different hidden positions.

References

1. Hsu CS, Tu SF (2010) Probability-based tampering detection scheme for digital images. Opt Commun 283(9):1737–1743
2. Shrivastava G, Pandey A, Sharma K (2013) Steganography and its technique: technical overview. Lecture Notes Electr Eng 150:615–620
3. Parah SA, Sheikh JA, Loan NA, Bhat GM (2017) A robust and computationally efficient digital watermarking technique using inter block pixel differencing. In: Multimedia forensics and security. Springer, Cham, pp 223–252
4. Parah SA, Sheikh JA, Assad UI, Bhat GM (2017) Realisation and robustness evaluation of a blind spatial domain watermarking technique. Int J Electron 104(4):659–672
5. Parah SA, Sheikh JA, Akhoon JA, Loan NA (2018) Electronic health record hiding in images for smart city applications: a computationally efficient and reversible information hiding technique for secure communication. Futur Gener Comput Syst. https://doi.org/10.1016/j.future.2018.02.023
6. Parah SA, Sheikh JA, Dey N, Bhat GM (2017) Realization of a new robust and secure watermarking technique using DC coefficient modification in pixel domain and chaotic encryption. J Glob Inf Manag 25(4):80–102
7. Hurrah NN, Loan NA, Parah SA, Sheikh JA (2017) A transform domain based robust color image watermarking scheme for single and dual attacks. In: Proceedings of ICIIP, Waknaghat, Shimla, H.P., India, Dec 2017, pp 1–5
8. Parah SA, Loan NA, Shah AA, Sheikh JA, Bhat GM (2018) A new secure and robust watermarking technique based on logistic map and modification of DC coefficient. Nonlinear Dynam 93(4):1933–1951
9. Su Q, Wang G, Zhang X, Lv G, Chen B (2017) An improved color image watermarking algorithm based on QR decomposition. Multimed Tools Appl 76(1):707–729
10. Hsu CS, Tu SF (2018) Digital watermarking scheme enhancing the robustness against cropping attack. Lecture Notes Electr Eng 464:143–152
11. Wang D, Yang F, Zhang H (2016) Blind color image watermarking based on DWT and LU decomposition. J Inf Process Syst 12(4):765–778
12. Su Q, Wang G, Zhang X, Lv G, Chen B (2018) A new algorithm of blind color image watermarking based on LU decomposition. Multidim Syst Sign Process 29(3):1055–1074

13. Liu Y, Tang S, Liu R, Zhang L, Ma Z (2018) Secure and robust digital image watermarking scheme using logistic and RSA encryption. Expert Syst Appl 97:95–105
14. Gao J, Fan L, Xu L (2012) Solving the face recognition problem using QR factorization. WSEAS Trans Math 11(8):712–721
15. Su Q, Niu Y, Wang G, Jia S, Yue J (2014b) Color image blind watermarking scheme based on QR decomposition. Signal Process 94:219–235
16. Su O, Niu Y, Zou H, Zhao Y, Yao T (2014a) A blind double color image watermarking algorithm based on QR decomposition. Multimed Tools Appl 72(1):987–1009
17. Subhedar MS, Manbkar VH (2016) Image steganography using redundant discrete wavelet transform and QR factorization. Comput Electr Eng 54:406–422
18. Naderahmadian Y, Hosseini-Khayat S (2010) Fast watermarking based on QR decomposition in wavelet domain. In: Proceedings of IIH-MSP, Darmstadt, Germany, Oct. 2010, pp 127–130
19. Schlauweg M, Palfner T, Profrock D, Muller E (2005) The Achilles' heel of JPEG-based image authentication. In: Proceedings of IASTED CNIS, Phoenix, USA, Nov 2005, pp 1–6
20. Schlauweg M, Profrock D, Zeibich B, Muller E (2006) Dual watermarking for protection of rightful ownership and secure image authentication. In: Proceedings of MCPS, Santa Barbara, California, USA, Oct 2006, pp 59–66
21. Shivani S, Singh P, Agarwal S (2017) A dual watermarking scheme for ownership verification and pixel level authentication. In: Proceedings of ICCAE, Sydney, Australia, Feb 2017, pp 131–135
22. Dube S, Sharma K (2019) Hybrid approach to enhance contrast of image for forensic investigation using segmented histogram. Int J Adv Intell Paradig 13(1/2):43–66
23. Katzenbeisser S, Petitcolas FAP (2000) Introduction to watermarking techniques. In: Information hiding techniques for steganography and digital watermarking, Artech House, 2000, pp 97–120
24. Cox IJ, Miller ML, Bloom JA (2002) Analyzing errors. In: Digital watermarking. Morgan Kaufmann Publishers, pp 157–199
25. University of Granada. (2012). Computer Vision Group. CVG-UGR Image Database. (2012, Oct. 22). Available http://decsai.ugr.es/cvg/dbimagenes/c512.php
26. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similairty. IEEE T Image Process 13(4):600–612
27. Wang Y, Cao Y, Zhao X, Zhou L (2017) A prediction mode-based information hiding approach for H. 264/AVC videos minimizing the impacts on rate-distortion optimization. In: Proceedings of IWDW, Magdeburg, Germany, Aug 2017, pp 163–176

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.