electronics

Article

Reversible Image Fragile Watermarking with Dual Tampering Detection

Cai Zhan [1], Lu Leng [1,*], Chin-Chen Chang [2,*] and Ji-Hwei Horng [3]

[1] Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, Nanchang Hangkong University, Nanchang 330063, China
[2] Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan
[3] Department of Electrical Engineering, National Quemoy University, Kinmen 89250, Taiwan; horng@email.nqu.edu.tw
* Correspondence: leng@nchu.edu.cn (L.L.); alan3c@gmail.com (C.-C.C.); Tel.: +86-791-8645-3251 (L.L.)

Abstract: The verification of image integrity has attracted increasing attention. Irreversible algorithms embed fragile watermarks into cover images to verify their integrity, but they are not reversible due to unrecoverable loss. In this paper, a new dual tampering detection scheme for reversible image fragile watermarking is proposed. The insect matrix reversible embedding algorithm is used to embed the watermark into the cover image. The cover image can be fully recovered when the dual-fragile-watermarked images are not tampered with. This study adopts two recovery schemes and adaptively chooses the most appropriate scheme to recover tampered data according to the square errors between the tampered data and the recovered data of two watermarked images. Tampering coincidence may occur when a large region of the fragile-watermarked image is tampered with, and the recovery information corresponding to the tampered pixels may be missing. The tampering coincidence problem is solved using image-rendering techniques. The experimental results show that the PSNR value of the watermarked image obtained using our scheme can reach 46.37 dB, and the SSIM value is 0.9942. In addition, high-accuracy tampering detection is achieved.

Keywords: fragile watermark; reversible embedding; Arnold transform; insect matrix

check for updates

Citation: Zhan, C.; Leng, L.; Chang, C.-C.; Horng, J.-H. Reversible Image Fragile Watermarking with Dual Tampering Detection. *Electronics* **2024**, *13*, 1884. https://doi.org/10.3390/electronics13101884

Academic Editor: Frederic Ros

Received: 10 April 2024
Revised: 3 May 2024
Accepted: 9 May 2024
Published: 11 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Both data hiding [1,2] and message encryption [3,4] can be used to transmit secret information in secure communication [5]. The confusion in encrypted messages easily catches attackers' attention. Data hiding, which embeds secret information into the cover image, solves this problem [6]; however, the images can be easily modified by unauthorized persons. Fragile watermarking can verify image integrity and authenticity. However, the existing methods of verifying image integrity and authenticity have the following shortcoming: when an image has been identified as having been tampered with, its content is not trusted, and the sender must resend the watermarked image. Retransmission causes extra overhead and time delay. The identification of tampered regions and the recovery of tampered data can effectively improve the reliability of the recovered images.

Fridrich and Jiri [7] proposed a fragile watermarking scheme to detect and recover tampered data. Many researchers have continued work in this field, and their methods mainly focus on the generation of recovery information (RI), verification information (VI), and the location of the embedded watermark.

The generation of fragile watermarks determines the accuracy of tampered region recovery. Popular technologies divide the image into several blocks, and each block generates RI. Lee [8], Chang [9], and Sinhal et al. [10] used the average pixel value in each block as the RI of this block. Su et al. [11] generated 12-digit verification codes as the VI and the 20-digit RI from each block as the watermark. Singh et al. [12] used the

quantization matrix values of the discrete cosine transform for each block to generate 10-bit RI. Tohidi et al. [13] used a new compression strategy to generate the RI for each block. Shehab and Abdulaziz [14] performed singular value decomposition on each block and used the transform information of the block as the RI. Hemida et al. [15] used block truncation coding to divide blocks into smooth blocks and rough blocks. Coding block content generates RI by assigning fewer bits to the smooth blocks and more bits to the rough blocks. Lin et al. [16] used absolute moment block truncation coding (AMBTC) based on the Huffman code (HC) to generate watermarks. Aminuddin et al. [17] divided the cover image into non-overlapping blocks of 5×5 pixels and non-overlapping blocks of 2×2. The hash code of each 5×5 block was the VI, while the pixel mean of each 2×2 block was the RI.

Fragile watermarks in the spatial domain can more accurately detect tampering. Su et al. [18] used the turtle shell algorithm to embed self-recovery information and checked the code of each block in the pixel values of other blocks. Renklier et al. [19] proposed a spatial-domain self-embedding watermarking method based on Sudoku puzzles.

Fragile watermarks in the frequency domain can enhance robustness. Molina-Garcia et al. [20] introduced a frequency-domain digital watermarking method based on Daubechies discrete wavelet transformation (DWT), halftone, and quantization index modulation. The scheme can resist JPEG compression with a quality factor above 75. HagHighi et al. [21] introduced an image tampering detection and recovery scheme based on the wavelet transform and genetic algorithm. Rhayma et al. [22] embedded a fragile watermark into the approximation sub-band of a DWT-transformed image.

In addition, some researchers improved the recovery effect by setting up multiple recovery watermarks. For example, Molina-Garcia et al. [23] used a bilateral filtering algorithm and patch algorithm to ensure the high quality of the recovered image. Al-Otum [24] and Haghighi et al. [25] proposed double watermarks for tampering detection and recovery and embedded the watermark of a block in other blocks. Li et al. [26] proposed a four-layer tampering detection algorithm to achieve high identification accuracy for the tampered region. Sinhal et al. [27] embedded both a robust watermark and a fragile watermark in the cover image.

In addition, some reversible data-hiding algorithms have been used to improve the embedding capacity, employing (7,4) Hamming code [28], the difference between pixels [29], and multiple-image embedding [30].

Although these schemes have achieved good results in image integrity, image tampering detection, and recovery, they still have defects.

(1) Irreversible watermark-embedding algorithms cause the unrecoverable loss of the original information of the cover image. However, they have a limited embedding capacity. The size of fragile watermarks is often large, so there is a contradictory relationship between them.

(2) When the watermarked image tampering rate reaches more than 50%, the above schemes exhibit an unsatisfactory recovery effect due to tampering coincidence.

Image-rendering techniques [23] have been widely used in some tasks, such as restoring damaged paintings and photographs and removing defective objects [31]. The aim is to repair damaged images with authentic content [32]. Image-rendering techniques typically derive from texture synthesis, deep generative modeling [33], and coherence between adjacent pixels [34]. Image-rendering technology can solve the problem of tampering coincidence satisfactorily.

In this paper, a new reversible image fragile watermarking scheme with dual tampering detection is proposed. The contributions of this paper are as follows:

(1) Dual tampering detection is used to verify image integrity. The insect matrix reversible embedding algorithm is used to fully recover the cover image when neither fragile-watermarked image is tampered with.

(2) If tampering has occurred, two recovery schemes are adopted, and an appropriate recovery scheme is adaptively chosen in this paper to recover the tampered data according

to the square errors between the tampered data and the recovered data of two fragile-watermarked images, so the quality of the recovered images is high. The other untampered regions can still be fully recovered.

(3) Image-rendering technology is launched to solve the problem of tampering coincidence when it occurs.

Compared with other SOTA schemes, the scheme proposed in this paper obtains a higher PSNR of the watermarked image. Our scheme also obtains a higher PSNR of the recovered image under the same tampering rate.

The rest of this paper is organized as follows: Section 2 introduces the related technologies. Section 3 describes the proposed methodology. Section 4 introduces the experimental results. Finally, Section 5 summarizes this research and prospects for future work.

2. Preliminaries

2.1. Arnold Algorithm for Scrambling

The Arnold transform [35] is a popular scrambling algorithm. It is a chaotic mapping method that repeatedly folds and stretches the transform in a limited area. The algorithm is generated using Equation (1):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & P \\ q & p \bullet q+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} mod(N_A) \quad (1)$$

where p and q are two positive constants, and N_A is the side length of the image matrix. The inverse Arnold transform can be implemented as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} p \bullet q+1 & -P \\ -q & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} mod(N_A) \quad (2)$$

The Arnold transform is used to embed the RI of each block in other blocks so that the RI can be extracted for recovery when the image is tampered with.

2.2. Insect Matrix for Embedding Algorithm

Liu and Chang [36] proposed a data embedding algorithm using a turtle shell reference matrix, as shown in Figure 1. This algorithm can embed a 3-bit secret per two pixels, as specified in Algorithm 1.

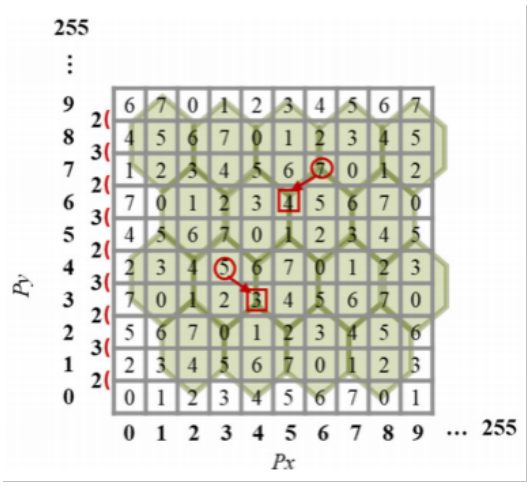
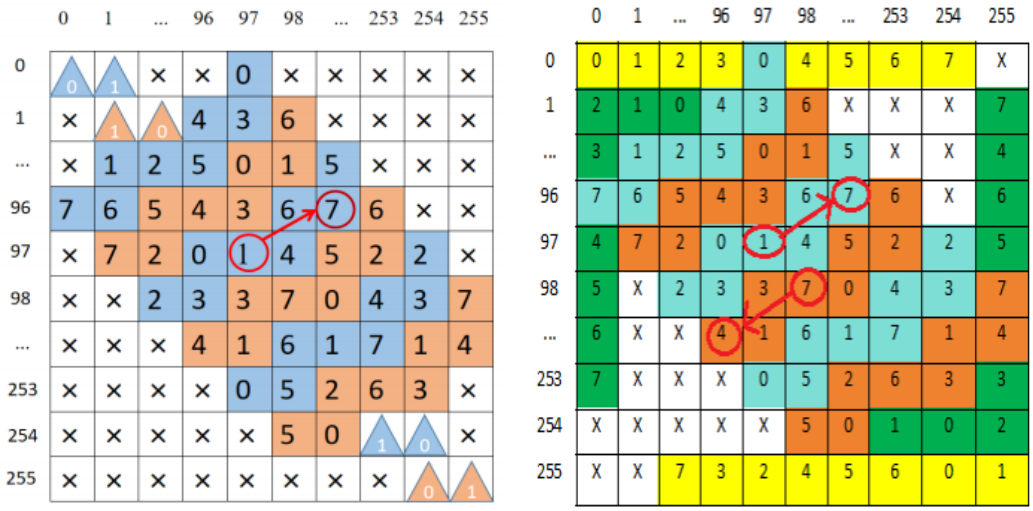


Figure 1. Turtle shell matrix.

Algorithm 1: Turtle shell embedding algorithm.

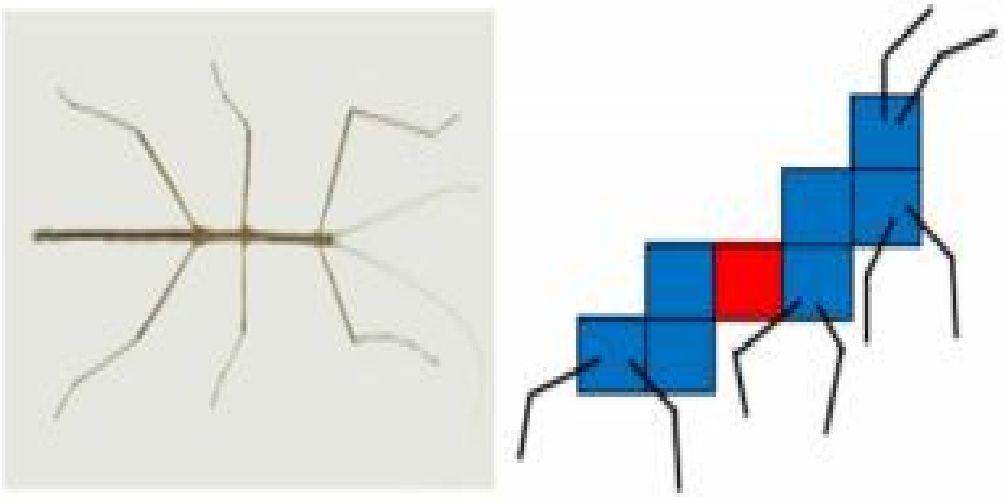
Original pixel: (6, 7).
Secret information: '100'.
1. Find the pixel position in the matrix: (6, 7) → 7.
2. Make the turtle shell matrix: With (6, 7) as the vertex, make three turtle shells.
3. Convert secret information to an octal number: '100'→ 4.
4. Find the value of 4 in the turtle shell and the position closest to the pixel pair (6, 7): (5, 6).
5. Output the final pixel pair: (5, 6).

The turtle shell embedding algorithm cannot fully recover the cover image after extracting the secret information. Gao et al. [37] designed a query matrix method based on the turtle shell matrix, as shown in Figure 2a. It can extract secret information and fully recover the cover image. It is dubbed an insect matrix because its shape is like an insect, as shown in Figure 3. It uses a single image to generate a double-watermarked image for watermark embeddings, which can greatly increase the embedding capacity.



(**a**) (**b**)

Figure 2. Insect matrix. (**a**) Original insect matrix; (**b**) improved insect matrix.



(**a**) (**b**)

Figure 3. Insect shape. (**a**) Insect; (**b**) insect grid.

As shown in Figure 2, the eight grids of each color are composed of 0–7, and each grid corresponds to 3-bit secret information. If the pixel value is 0, 1, 254, or 255, only 1-bit secret information is embedded. In this paper, the empty grids are supplemented to be more suitable for fragile watermark embeddings, as shown in Figure 2b. When the pixel value is 0, 1, 254, or 255, 3-bit information can still be embedded, as shown in the yellow and green areas.

The main embedding process of the insect matrix is depicted in Figure 4. The octal '7' and '4' are embedded into pixel pairs (97, 98).

First, the pixel pair (97, 98) is divided into two pairs of pixels, namely, (97, 97) and (98, 98), with the embeddings '7' and '4', respectively.

Second, in the insect matrix, (97, 97) is taken as the center to make the insect torso.

Third, the point with a value of '7' in the made insect torso, (96, 99), is identified.

Fourth, the same step embeds '4' into the pixel pair (98, 98) and then obtains (98, 97).

Finally, the pixel pair (96, 99) and (98, 97) are recombined to obtain the pixel pair (96, 98) and (99, 97) with the embedded secret information.

In the extraction stage, (96, 98) and (99, 97) are combined to obtain (96, 99) and (98, 97). The corresponding values of (96, 97) and (99, 97) in the insect matrix are searched for, and the secret messages '7' and '4' are obtained. The original pixels (97, 98) are obtained by searching for the center points (97, 97) and (98, 98) of insects with (96, 99) and (98, 97) as the torso, respectively.

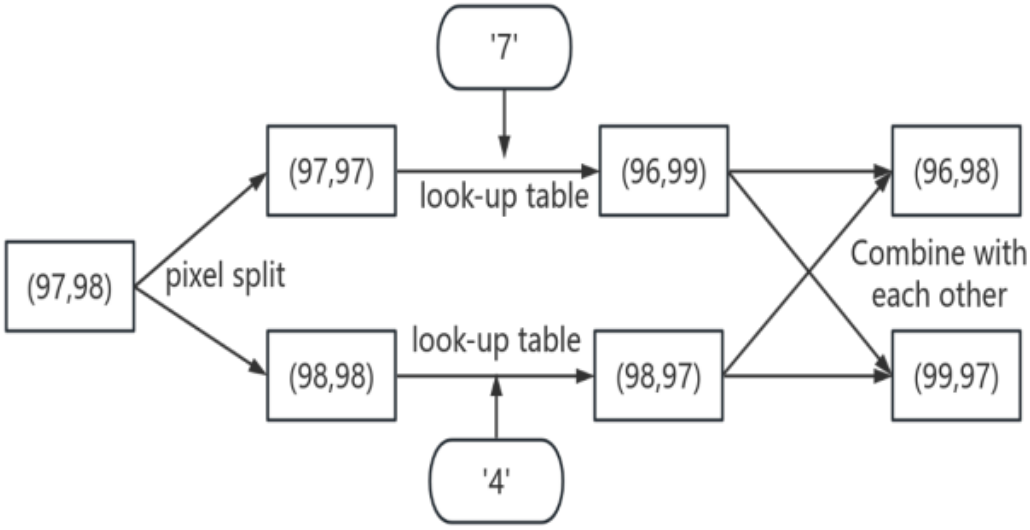


Figure 4. Flow chart of insect matrix embedding algorithm.

3. Proposed Method

The scheme proposed in this paper includes four parts: watermark generation, watermark embedding, tampering detection, and tampering recovery.

Figure 5a shows the framework of the proposed watermark embeddings. The image is divided into two subgraphs for cross set C and dot set D. Each subgraph is divided into 4×4 non-overlapping blocks. The size of all blocks in this paper is 4×4. Each block generates verification information (VI), i.e., hashing values for tampering detection, and recovery information (RI) for tampering recovery. The insect matrix embedding algorithm is used to embed the VI and RI to obtain two matrices, C′ and D′. C′ is divided into C1 and C2; D′ is divided into D1 and D2. C1 and D1 are combined to obtain Watermarked Image 1; C2 and D2 are combined to obtain Watermarked Image 2. The extraction process is the inverse process.

The watermark extraction shown in Figure 5b is the inverse process. Watermarked Image 1 is divided into C1 and C2; Watermarked Image 2 is divided into D1 and D2. C1 and D1 are combined to obtain C′; C2 and D2 are combined to obtain D′. The insect matrix

is used to extract VI′ and RI′ and recover C from C′. VI*, RI*, and D are obtained from D′. RI-C and VI-C are generated from C, while RI-D and VI-D are generated from D, which are compared with RI′, VI′, RI*, and VI* for tampering judgment.

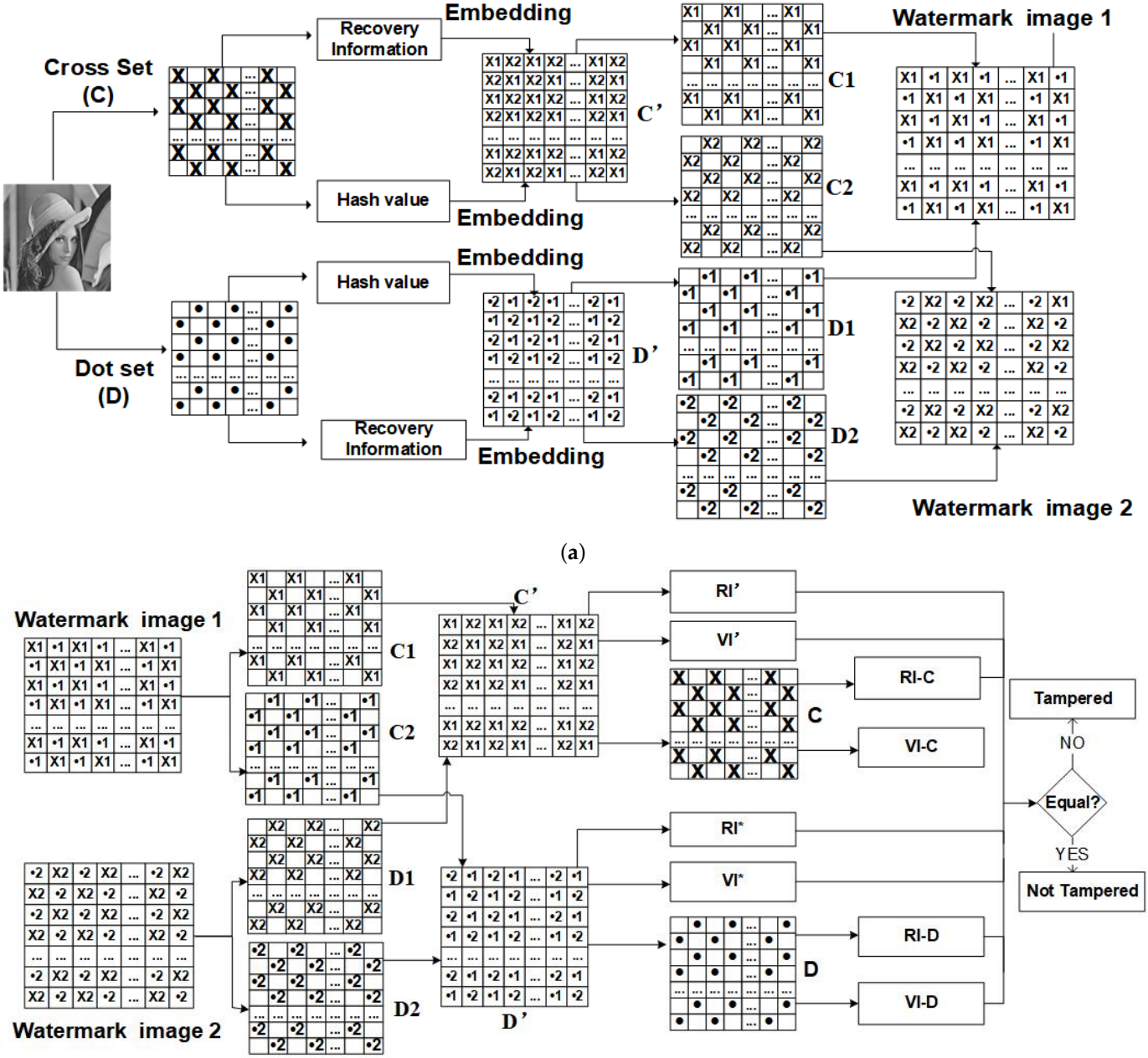


Figure 5. Framework of fragile-watermarked information embedding and extraction. (**a**) Watermark embedding stage; (**b**) watermark extraction stage.

3.1. Watermark Generation

The fragile watermark consists of VI and RI. The watermark generation steps are as follows:

Step 1: The cover image is divided into two parts: C and D. The watermark generation process for C and D is the same. Taking C as an example, it is divided into 4 × 4 non-overlapping blocks.

Step 2: The hash value is calculated for each block. Each block has an 8-bit hash value as the VI for tampering detection.

Step 3: Each block contains 8 pixel values, which are divided into two groups by positional order. The two means of the pixel values in the two groups are used as the RI and are expressed in a 16-bit code.

Step 4: There are a total of n non-overlapping blocks, and each block has a piece of RI and a piece of VI. Thus, there are n pieces of RI and n pieces of VI in total. The n pieces of RI are scrambled using the Arnold algorithm and then combined with the n pieces of VI to obtain n pieces of watermarks of 16 bits each. The n pieces of VI are not scrambled.

3.2. Watermark Embeddings

The watermark-embedding process for D is the same as that for C. C is taken as an example here. C is divided into non-overlapping 4×4 blocks, each containing 8 pixels. The insect matrix scheme is used for the watermark embeddings. Each pixel embeds a 3-bit watermark. Each block has 8 pixels, so 24-bit watermarks can be embedded. The embedding process is described in Section 2.2.

C′ is obtained after the watermark-embedding process and is then divided into C1 and C2. A similar process is conducted on D. C1 and D1 are combined to obtain Watermarked Image 1. C2 and D2 are combined to obtain Watermarked Image 2. The RI generated from one block is embedded in other blocks via the Arnold algorithm. The VI generated from one block is embedded in this current block.

3.3. Tampering Detection

This paper proposes a dual tampering detection scheme that takes advantage of both VI and RI to improve the tampering detection accuracy. The tampering detection steps are as follows:

Step 1: Watermarked Image 1 and Watermarked Image 2 are divided into C1, D1, C2, and D2. C1 and C2 are combined to form C′; D1 and D2 are combined to form D′.

Step 2: Taking C1 and C2 as examples, the detection process for D1 and D2 is the same. The insect matrix is used to extract the watermark and recover C.

Step 3: C is divided into non-overlapping blocks. The VI and RI for each block are extracted.

Step 4: The extracted VI is used for the first tampering detection.

Step 5: The RI is extracted from other blocks and re-ordered using the inverse Arnold algorithm. The extracted RI is compared with the RI generated from the current block for the second tampering detection.

The second tampering detection is specified in Algorithm 2, which judges whether Block 1 is tampered with. The RI of Block 2 is embedded in Block 1. The RI of Block 1 is embedded in Block 3.

Algorithm 2: Second tampering detection process.

Input: Block1, Block2, Block3.
Output: Is Block1 tampered with?
1. Extract the recovery information RI2 from Block1.
2. Recalculate the recovery information RI2′ from Block2.
3. Extract the recovery information RI1 from Block3.
4. Recalculate the recovery information RI1′ from Block1.
5. Determine whether Block1 is tampered with:
   If RI2 = RI2′ and RI1 = RI1′: Output Block1 is not tampered with.
   Else: Output Block1 is tampered with.

The proposed dual tampering detection scheme has a high tampering detection accuracy, but it has the following drawback: if one watermarked image, Watermarked Image 1 or Watermarked Image 2, is tampered with, the other watermarked image is also judged to be tampered with, even if it is not. In other words, the proposed scheme cannot judge which watermarked image is tampered with if only one watermarked image is tampered with.

This paper proposes a supplemental judgment to solve this problem. Equation (3) is used to calculate the square error (SE1 and SE2 for the Watermarked Images 1 and 2, respectively) in the tampered region between the watermarked image and its counterpart RI.

$$SE = \frac{1}{100 \times H \times W} \sum_{i=1}^{H \times W} (P_i - P_{wi})^2 \quad (3)$$

$H \times W$ is the size of the tampered region, and P_i and P_{wi} represent the i-th pixel of a recovered block using the extracted RI and the corresponding block in the watermarked image.

If the RI is lost, the RI of the adjacent blocks is used instead. A threshold T is set to determine whether the tampering occurred in Watermarked Image 1 or Watermarked Image 2.

If SE1 $\geq$ T and SE2 $\geq$ T: Watermarked Image 1 and Watermarked Image 2 are both tampered with.

Else, if SE1 $\geq$ SE2: Watermarked Image 1 is tampered with.

Else: Watermarked Image 2 is tampered with.

3.4. Tampering Recovery

Case 1: Watermarked Image 1 is tampered with, and Watermarked Image 2 is not tampered with.

Case 2: Watermarked Image 2 is tampered with, and Watermarked Image 1 is not tampered with.

Case 3: Watermarked Images 1 and 2 are both tampered with.

For Case 1: Watermarked Image 2 corresponding to the block is used as the RI to recover the data tampered with in Watermarked Image 1.

For Case 2: Watermarked Image 1 corresponding to the block is used as the RI to recover the data tampered with in Watermarked Image 2.

For Case 3: The RI in both Watermarked Images 1 and 2 corresponding to the tampered region is used to recover the tampered data.

In the extreme case, Watermarked Images 1 and 2 are both tampered with, and both of their RI is destroyed. In this case, an image-rendering technique is launched for recovery. As shown in Figure 6, the blue pixels are tampered with, and the white pixels are not tampered with. The four estimated $value_i$s of P are calculated via Equation (4). V_{ij} is the untampered pixel value adjacent to P. Equation (5) calculates the square distance between two pixel points. (x, y) represents the position of the pixel. Finally, the mean of the four $value_i$s is determined, and the tampered pixels are replaced.

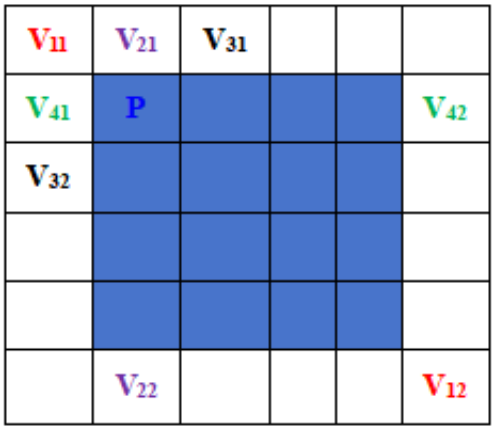


Figure 6. Tampered pixel P and its adjacent untampered pixels in eight directions.

$$\boldsymbol{value_i = \frac{V_{i1} \times S_1^2 + V_{i2} \times S_2^2}{S_1^2 + S_2^2}} \quad (4)$$

$$S_j^2 = (x_v - x_p)^2 + (y_v - y_p)^2 \tag{5}$$

4. Experiments

Various tampering attacks, including content deletion, collage attack, text addition, and content blur, were tested.

4.1. Evaluation Criteria

●Peak signal-to-noise ratio (PSNR): The PSNR is used for image quality evaluation based on the differences between the watermarked image and the cover image. The higher the PSNR value, the better the quality of the watermarked image.

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{6}$$

$$PSNR = 10log_{10}(\frac{MAX^2}{MSE}) \tag{7}$$

I represents the cover image. K represents the watermarked image. $m \times n$ represents the image size. (i, j) represents the pixel position. MAX represents the maximum upper limit of the pixel. In the gray image, $MAX = 255$.

●Structural similarity (SSIM): SSIM is used to measure the similarity between the watermarked image and the cover image. It is related to the perception of the human visual system. The SSIM ranges between 0 and 1. The larger the SSIM value, the better the quality of the watermarked image.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + g_1)(2\sigma_{xy} + g_2)}{(\mu_x^2 + \mu_y^2 + g_1)(\sigma_x^2 + \sigma_y^2 + g_2)} \tag{8}$$

μ_x and μ_y represent the pixel averages of the two images. σ_x and σ_y represent the standard deviations of the two images. σ_{xy} represents the covariance of the two images. g_1 and g_2 are two constants to avoid a denominator of 0.

●Tampering detection rate

True positive (TP): The number of blocks that are actually tampered with and correctly detected.

False positive (FP): The number of blocks that are not tampered with but erroneously detected.

False negative (FN): The number of blocks that are actually tampered with but erroneously undetected.

True negative (TN): The number of blocks that are not tampered with and correctly undetected.

Recall and ***precision*** are used as two tampering detection rates:

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

4.2. Threshold for Supplemental Judgment

A suitable threshold T must be set for better tampering detection and recovery. Figure 7 shows the relationship between the recovered image PSNR and threshold T at different tampering rates. The PSNR decreases with the increment in the tampering rate. The PSNR between the recovered image and the cover image first increases and then decreases or remains unchanged with the increment in the threshold T. This is because it is difficult to judge whether the tampering occurs in Watermarked Image 1 or Watermarked Image 2 when T is too small or too large. When T = 60, the PSNR is satisfactory.

Figure 8 shows the tampering detection results under a collage attack at T = 60. The tampering detection is correct and accurate.

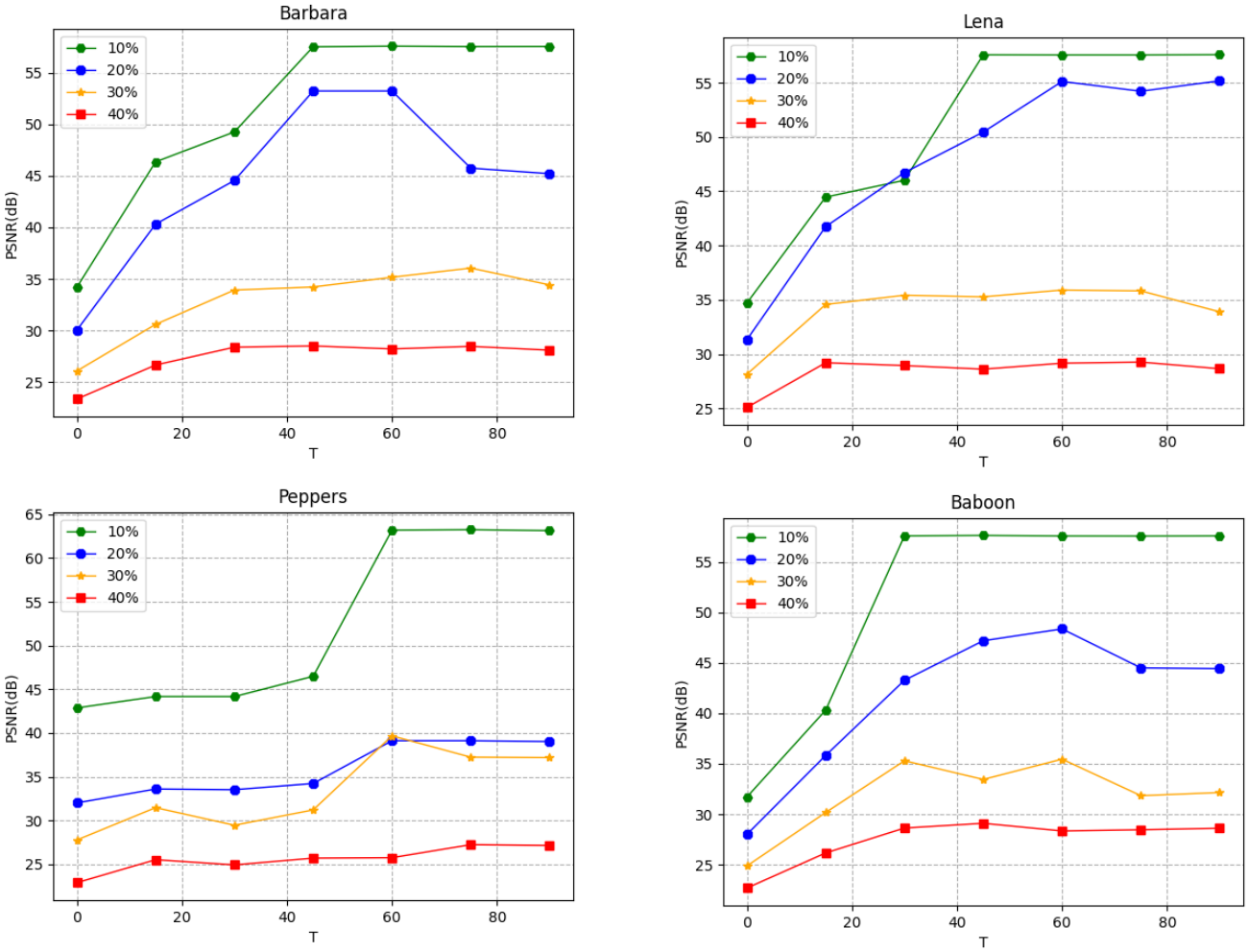


Figure 7. PSNR between recovered image and cover image with threshold T under different tampering rates.

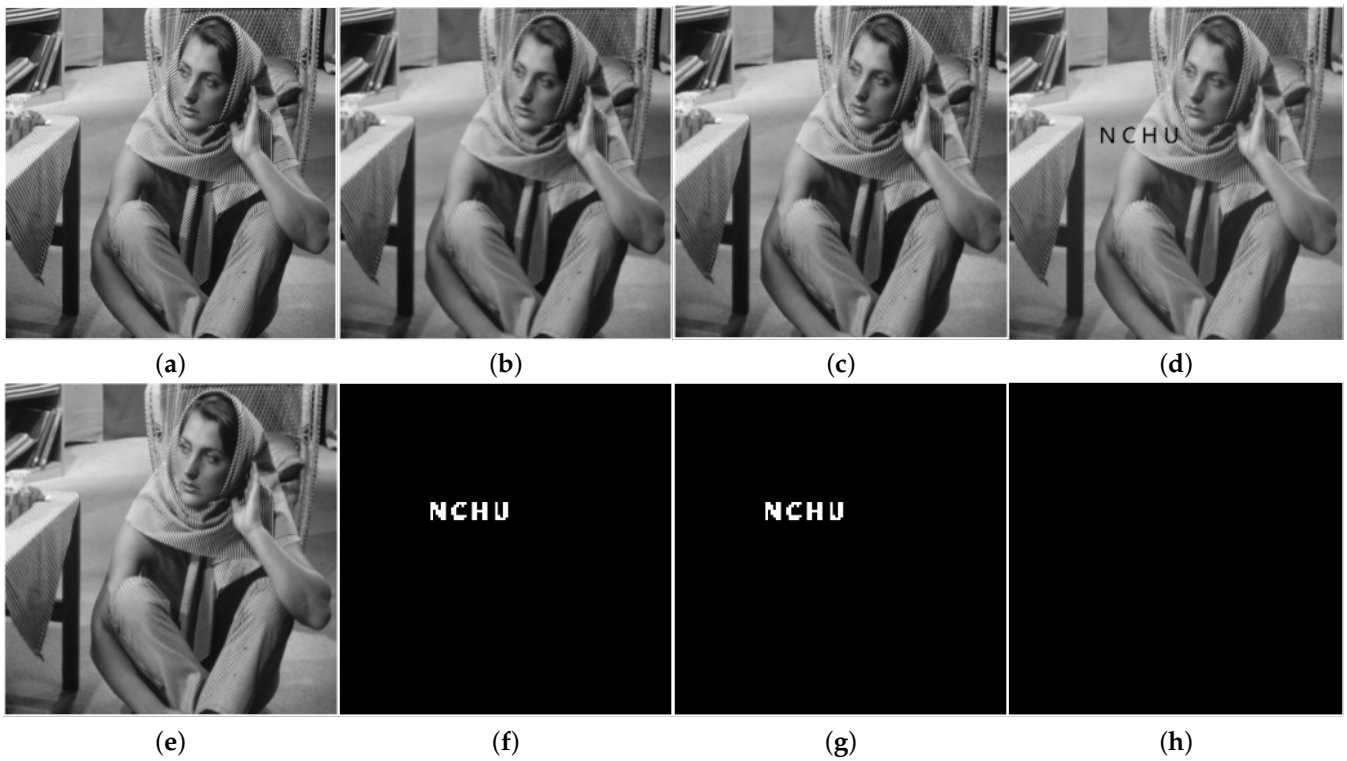


Figure 8. (**a**) The cover image. (**b**) Watermarked Image 1. (**c**) Watermarked Image 2. (**d**) Watermarked Image 1 after tampering. (**e**) Watermarked Image 2 that is not tampered with. (**f**) The detected tampered region. (**g**) The detected tampered region in Watermarked Image 1. (**h**) The detected tampered region in Watermarked Image 2.

4.3. Tampering Detection and Recovery

This paper proposes a dual tampering detection scheme that takes advantage of both VI and RI. Figure 9 shows the tampering detection and recovery results when only tampering with Watermarked Image 1 and those when simultaneously tampering with Watermarked Images 1 and 2 at T = 60.

Figure 10 shows the tampering detection results and recovered images at tampering rates of 5%, 10%, 30%, and 50%. When the tampering rate is small, the tampering detection and recovery results are both satisfactory. However, as the tampering rate increases, tampering coincidence probably occurs and degrades the results, but the main image content can still be recovered. In general, our method outperforms the existing SOTA methods.

Tables 1 and 2 compare our method and some state-of-the-art methods in terms of the PSNR and SSIM between the watermarked image and the cover image. Our method can obtain higher PSNR and SSIM values. The insect matrix used in our method can embed 3 bits of watermark information per pixel, surpassing the existing methods in embedding capacity.

Table 3 compares the tampering detection accuracy at different tampering rates. Our method uses VI and RI for dual tampering detection in each block, resulting in higher tampering detection accuracy compared to the single tampering detection approach of the other methods.

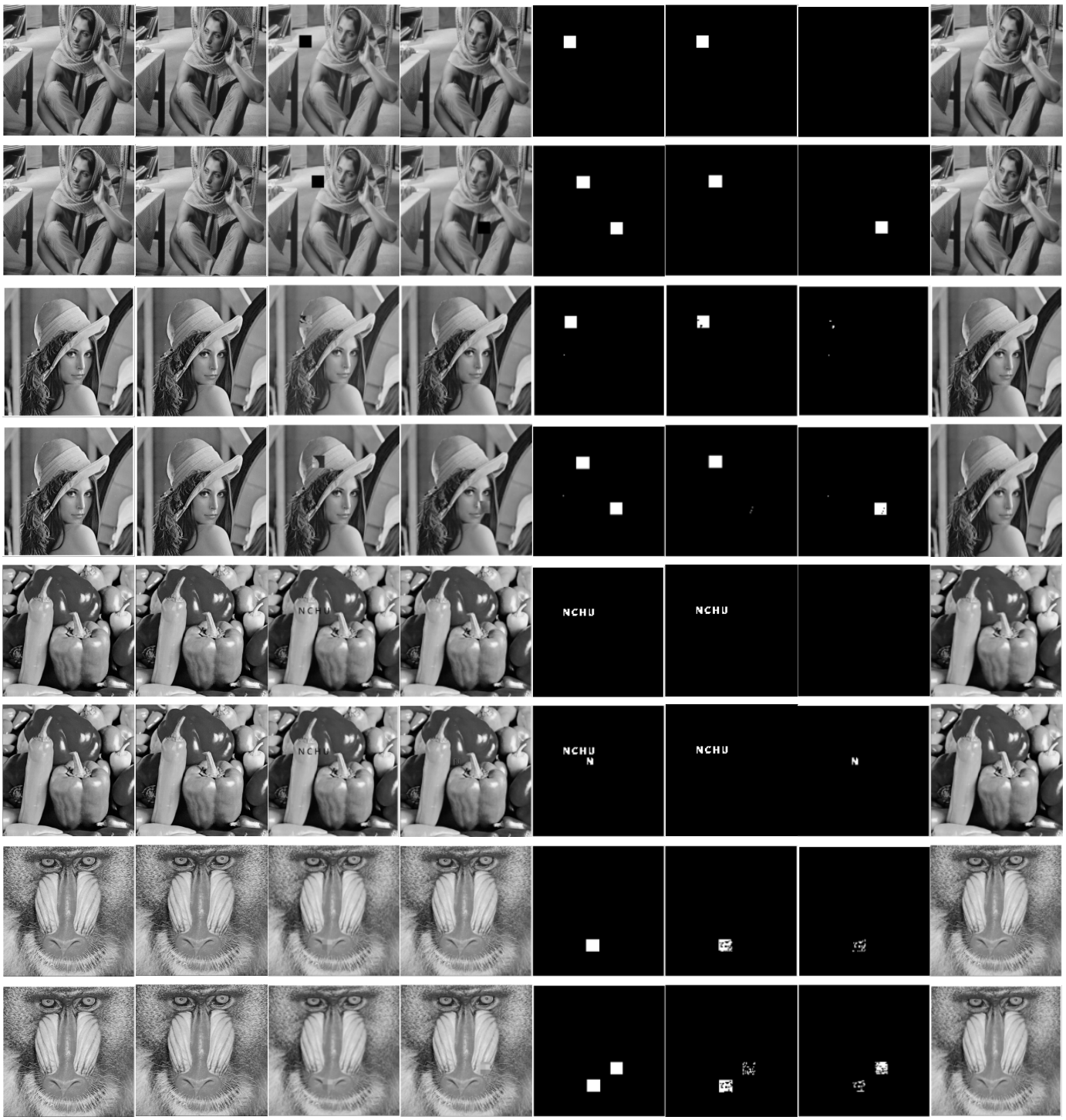


Figure 9. The 1st and 2nd columns are Watermarked Images 1 and 2, respectively. The 3rd and 4th columns are tampered Images 1 and 2, respectively. The 5th column presents the tampering detection results. The 6th and 7th columns present the tampering detection results for Watermarked Images 1 and 2, respectively. The 8th column shows the recovered image.

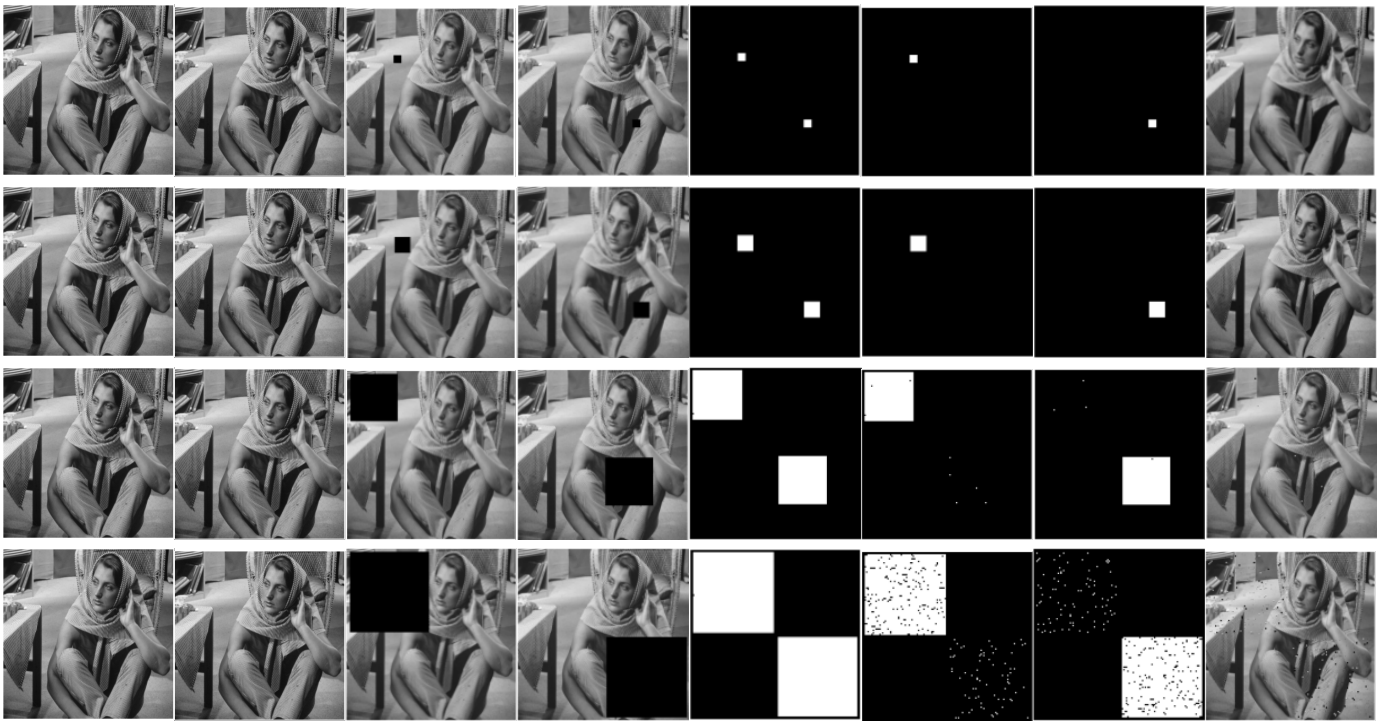


Figure 10. The 1st and 2nd columns are Watermarked Images 1 and 2, respectively. The 3rd and 4th columns are the tampered Watermarked Images 1 and 2, respectively. The 5th column shows the tampering detection results. The 6th and 7th columns show the tampering detection results for Watermarked Images 1 and 2, respectively. The 8th column shows the recovered image.

Table 1. PSNR comparison.

Method		PSNR (dB)				
		Barbara	**Lena**	**Peppers**	**Baboon**	**Average**
[13]		44.17	44.11	44.1	44.29	44.17
[17]		45.69	45.71	45.66	45.7	45.69
[19]		44.32	44.27	44.39	44.22	44.24
[25]		44.16	44.17	44.16	44.15	44.16
[26]		40.71	40.72	40.71	40.7	40.71
[27]		41.16	41.17	41.01	40.2	40.89
Proposed	Watermark Image 1	**46.37**	**46.36**	**46.35**	**46.39**	**46.37**
	Watermark Image 2	**46.36**	**46.36**	**46.36**	**46.38**	**46.37**

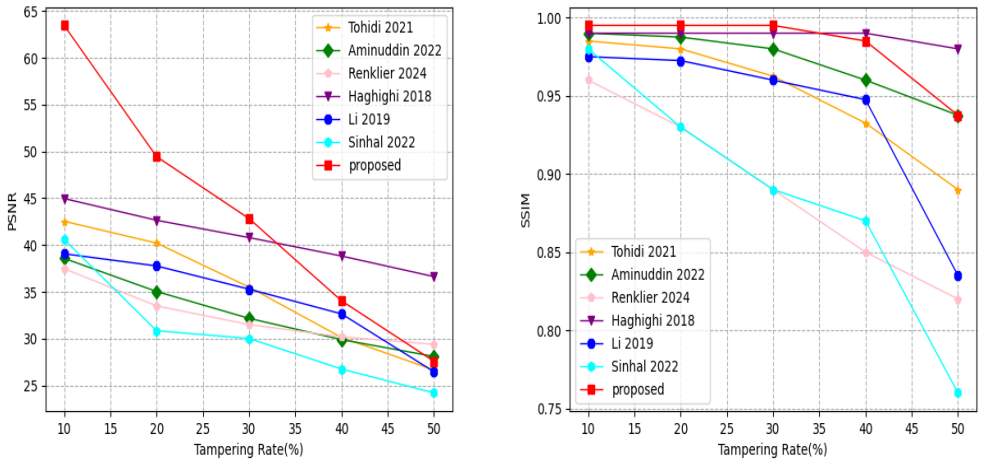
Figure 11 shows the PSNR and SSIM comparison results between recovered images and cover images at different tampering rates compared to the methods of Tohidi [13], Aminuddin [17], Renklier [19], Haghighi [25], Li [26], Sinhal [27]. When the tampering rate is low, our method obtains higher PSNR and SSIM values. With the increase in the tampering rate, the tampering coincident area increases, degrading the accuracy of determining whether tampering occurs in Watermark Image 1 or Watermark Image 2. This degradation further impacts the PSNR and SSIM of the recovered images. It is worth emphasizing that our method can accurately recover the original cover image when no tampering occurs, a capability lacking in existing methods.

Table 2. SSIM comparison.

Method		SSIM				
		Barbara	**Lena**	**Peppers**	**Baboon**	**Average**
[13]		0.9867	0.9821	0.9827	0.9945	0.9865
[17]		**0.9967**	**0.9993**	0.9889	0.9900	0.9937
[19]		0.9878	0.9814	0.9881	0.9922	0.9874
[25]		0.9851	0.9835	0.9817	0.9937	0.9860
[26]		0.9785	0.9687	0.9722	0.9900	0.9774
[27]		0.9734	0.9667	0.9650	0.9823	0.9719
Proposed	Watermark Image 1	0.9947	0.9963	**0.9893**	**0.9966**	**0.9942**
	Watermark Image 2	0.9947	0.9963	**0.9893**	**0.9966**	**0.9942**

Table 3. Comparison of tampering detection accuracy at different tampering rates.

Method		Tampering Rate				
		10%	**20%**	**30%**	**40%**	**50%**
[13]	Recall	99.50%	99.25%	99.44%	98.75%	99.12%
	Precision	**100%**	**100%**	**100%**	**100%**	**100%**
[17]	Recall	**100%**	**100%**	**100%**	**100%**	**100%**
	Precision	99.50%	99.18%	99.25%	**100%**	**100%**
[25]	Recall	99.95%	**100%**	**100%**	**100%**	99.99%
	Precision	**100%**	**100%**	**100%**	**100%**	**100%**
[26]	Recall	99.83%	99.95%	99.99%	**100%**	**100%**
	Precision	**100%**	**100%**	**100%**	**100%**	**100%**
[27]	Recall	99.88%	99.88%	99.99%	99.95%	**100%**
	Precision	**100%**	**100%**	**100%**	**100%**	**100%**
Proposed	Recall	**100%**	99.65%	**100%**	**100%**	**100%**
	Precision	**100%**	**100%**	**100%**	**100%**	**100%**



(**a**) (**b**)

Figure 11. Comparison of PSNR and SSIM between recovered images and cover images. (**a**) PSNR comparison results; (**b**) SSIM comparison results [13,17,19,25–27].

5. Conclusions and Future Work

This paper proposes a reversible fragile-watermarked image scheme with dual tampering detection. The insect matrix is used to embed the watermark in the cover image to obtain two watermarked images. The cover image can be completely recovered when tampering does not occur. When a watermarked image is tampered with, high-quality recovered images can be obtained using the effective recovery method proposed in this paper. Compared with existing methods, the dual tampering detection method adopted in this paper obtains a higher tampering accuracy and higher PSNR for the watermarked image. Although our method currently exhibits satisfactory performance, its effectiveness degrades as the tampering rate increases, especially when it exceeds 50%. We aim to enhance our method for handling larger-scale tampering in the future.

Author Contributions: Each author discussed the details of the manuscript. C.Z. designed and wrote the manuscript. C.Z. implemented the proposed technique and provided the experimental results. L.L. reviewed and revised the article. L.L., C.-C.C. and J.-H.H. drafted and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (61866028) and the Technology Innovation Guidance Program Project (Special Project of Technology Cooperation, Science and Technology Department of Jiangxi Province) (20212BDH81003).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy issues.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Panchikkil, S.; Vegesana, S.P.; Manikandan , V.M.; Donta, P.K.; Maddikunta, P.K.; Gadekallu, T.R. An ensemble learning approach for reversible data hiding in encrypted images with fibonacci transform. *Electronics* **2023**, *12*, 450. [CrossRef]
2. Chi, H.X.; Chang, C.C.; Wang, X.; Lin, C.C. Hiding and extracting important information in encrypted images by using the transformation of all possible permutations and VQ codebook. *Electronics* **2022**, *11*, 3475. [CrossRef]
3. Zhou, N.R.; Tong, L.J.; Zou, W.P. Multi-image encryption scheme with quaternion discrete fractional Tchebyshev moment transform and cross-coupling operation. *Signal Process.* **2023**, *211*, 109107. [CrossRef]
4. Zhou, N.R.; Hu, L.L.; Huang, Z.W.; Wang, M.M.; Luo, G.S. Novel multiple color images encryption and decryption scheme based on a bit-level extension algorithm. *Expert Syst. Appl.* **2024**, *238*, 122052. [CrossRef]
5. Serrano, D.; Ruíz, J.F.; Muñoz, A.; Maña, A.; Armenteros, A.; Crespo, B.G. Development of applications based on security patterns. In Proceedings of the 2009 Second International Conference on Dependability, Athens, Greece, 18–23 June 2009.
6. Feng, Q.; Leng, L.; Chang, C.C.; Horng, J.H.; Wu, M. Reversible data hiding in encrypted images with extended parametric binary tree labeling. *Appl. Sci.* **2023**, *13*, 2458. [CrossRef]
7. Fridrich, J. Protection of digital images using self-embedding. In Proceedings of the Symposium on Content Security and Data Hiding in Digital Media, Newark, NJ, USA, 9–12 May 1999; New Jersey Institute of Technology: Newark, NJ, USA, 1999.
8. Lee, T.Y.; Lin, S.D. Dual watermark for image tamper detection and recovery. *Pattern Recognit.* **2008**, *41*, 3497–3506. [CrossRef]
9. Chang, Y.F.; Tai, W.L. A block-based watermarking scheme for image tamper detection and self-recovery. *Opto-Electron. Rev.* **2013**, *21*, 182–190. [CrossRef]
10. Sinhal, R.; Ansari, I.A.; Ahn, C.W. Blind image watermarking for localization and restoration of color images. *IEEE Access* **2020**, *8*, 200157–200169. [CrossRef]
11. Su, G.-D.; Chang, C.-C.; Lin, C.-C. High-precision authentication scheme based on matrix encoding for AMBTC-compressed images. *Symmetry* **2019**, *11*, 996. [CrossRef]
12. Singh, D.; Singh, S.K. Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability. *J. Vis. Commun. Image Represent.* **2016**, *38*, 775–789. [CrossRef]
13. Tohidi, F.; Paul, M.; Hooshmandasl, M.R. Detection and recovery of higher tampered images using novel feature and compression strategy. *IEEE Access* **2021**, *9*, 57510–57528. [CrossRef]
14. Shehab, A.; Elhoseny, M.; Muhammad, K.; Sangaiah, A.K.; Yang, P.; Huang, H.; Hou, G. Secure and robust fragile watermarking scheme for medical images. *IEEE Access* **2018**, *6*, 10269–10278. [CrossRef]
15. Hemida, O.; He, H. A self-recovery watermarking scheme based on block truncation coding and quantum chaos map. *Multimed. Tools Appl.* **2020**, *79*, 18695–18725. [CrossRef]
16. Lin, C.-C.; He, S.-L.; Chang, C.-C. Pixel P Air-Wise Fragile Image Watermarking Based on HC-Based Absolute Moment Block Truncation Coding. *Electronics* **2021**, *10*, 690. [CrossRef]

17. Aminuddin, A.; Ernawan, F. AuSR1: Authentication and self-recovery using a new image inpainting technique with LSB shifting in fragile image watermarking. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 5822–5840. [CrossRef]
18. Su, G.D.; Chang, C.C.; Lin, C.C. Effective self-recovery and tampering localization fragile watermarking for medical images. *IEEE Access* **2020**, *8*, 160840–160857. [CrossRef]
19. Renklier, A.; Öztürk, S. Image authentication and recovery: Sudoku puzzle and MD5 hash algorithm based self-embedding fragile image watermarking method. *Multimed. Tools Appl.* **2024**, *83*, 13929–13951. [CrossRef]
20. Molina-Garcia, J.; Reyes-Reyes, R.; Ponomaryov, V.; Cruz-Ramos, C. Watermarking algorithm for authentication and self-recovery of tampered images using DWT. In Proceedings of the 2016 9th International Kharkiv Symposium on Physics and Engineering of Microwaves, Millimeter and Submillimeter Waves (MSMW), Kharkiv, Ukraine, 20–24 June 2016.
21. Haghighi, B.B.; Taherinia, A.H.; Mohajerzadeh, A.H. TRLG: Fragile blind quad watermarking for image tamper detection and recovery by providing compact digests with optimized quality using LWT and GA. *Inf. Sci.* **2019**, *486*, 204–230. [CrossRef]
22. Rhayma, H.; Makhloufi, A.; Hamam, H.; Hamida, A.B. Semi-fragile self-recovery watermarking scheme based on data representation through combination. *Multimed. Tools Appl.* **2019**, *78*, 14067–14089. [CrossRef]
23. Molina-Garcia, J.; Garcia-Salgado, B.P.; Ponomaryov, V.; Reyes-Reyes, R.; Sadovnychiy, S.; Cruz-Ramos, C. An effective fragile watermarking scheme for color image tampering detection and self-recovery. *Signal Process. Image Commun.* **2020**, *81*, 115725. [CrossRef]
24. Al-Otum, H.M.; Ellubani, A.A. Secure and effective color image tampering detection and self restoration using a dual watermarking approach☆. *Optik* **2022**, *262*, 169280. [CrossRef]
25. Haghighi, B.B.; Taherinia, A.H.; Harati, A. TRLH: Fragile and blind dual watermarking for image tamper detection and self-recovery based on lifting wavelet transform and halftoning technique. *J. Vis. Commun. Image Represent.* **2018**, *50*, 49–64. [CrossRef]
26. Li, Y.; Song, W.; Zhao, X.; Wang, J.; Zhao, L. A novel image tamper detection and self-recovery algorithm based on watermarking and chaotic system. *Mathematics* **2019**, *7*, 955. [CrossRef]
27. Sinhal, R.; Ansari, I.A. Multipurpose image watermarking: Ownership check, tamper detection and self-recovery. *Circuits Syst. Signal Process.* **2022**, *41*, 3199–3221. [CrossRef]
28. Kang, H.; Leng, L.; Chang, C.C. Overlapped (7, 4) hamming code for large-capacity and low-loss data hiding. *Multimed. Tools Appl.* **2023**, *82*, 30345–30374. [CrossRef]
29. Huang, C.T.; Shongwe, N.S.; Weng, C.Y. Enhanced Embedding Capacity for Data Hiding Approach Based on Pixel Value Differencing and Pixel Shifting Technology. *Electronics* **2023**, *12*, 1200. [CrossRef]
30. Kang, H.-Y.; Leng, L.; Kim, B.-G. Data hiding of multicompressed images based on Shamir threshold sharing. *Appl. Sci.* **2022**, *12*, 9629. [CrossRef]
31. Qureshi, M.A.; Deriche, M.; Beghdadi, A.; Amin, A. A critical survey of state-of-the-art image inpainting quality assessment metrics. *J. Vis. Commun. Image Represent.* **2017**, *49*, 177–191. [CrossRef]
32. Wang, N.; Zhang, Y.-P.; Zhang, L.-F. Dynamic selection network for image inpainting. *IEEE Trans. Image Process.* **2021**, *30*, 1784–1798. [CrossRef] [PubMed]
33. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. Generative image inpainting with contextual attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
34. Bugeau, A.; Bertalmío, M.; Caselles, V.; Sapiro, G. A comprehensive framework for image inpainting. *IEEE Trans. Image Process.* **2010**, *19*, 2634–2645. [CrossRef] [PubMed]
35. Wu, L.; Zhang, J.; Deng, W.; He, D. Arnold transformation algorithm and anti-Arnold transformation algorithm. In Proceedings of the 2009 First International Conference on Information Science and Engineering, Nanjing, China, 26–28 December 2009.
36. Liu, Y.; Chang, C.C. A turtle shell-based visual secret sharing scheme with reversibility and authentication. *Multimed. Tools Appl.* **2018**, *77*, 25295–25310. [CrossRef]
37. Gao, K.; Horng, J.H.; Liu, Y.; Chang, C.C. A reversible secret image sharing scheme based on stick insect matrix. *IEEE Access* **2020**, *8*, 130405–130416. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.