Hindawi
Security and Communication Networks
Volume 2022, Article ID 2699802, 14 pages
https://doi.org/10.1155/2022/2699802

Research Article

New Framework of Self-Embedding Fragile Watermarking Based on Reference Sharing Mechanism

Lin Huang,[1] **Zhaoyang Xiang**,[1] **Jian Li**,[2] **Heng Yao**,[1] **and Chuan Qin**[1]

[1]*School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China*
[2]*School of Cyber Security, Qilu University of Technology (Shandong Academy of Sciences), Shandong Provincial Key Laboratory of Computer Networks, Jinan 250353, China*

Correspondence should be addressed to Jian Li; ljian20@gmail.com

Received 30 April 2022; Accepted 8 June 2022; Published 25 June 2022

Academic Editor: Hanzhou Wu

We proposed in this paper a new self-embedding framework based on a reference sharing mechanism. The framework has high flexibility; it can not only estimate the optimal recovered image quality based on a given tampering rate but also estimate the largest tampering rate that the framework can resist based on the given peak signal-to-noise ratio (PSNR) of the recovered image. When the tampering rate is given, we first calculate the largest number of character bits and then allocate an appropriate number of character bits according to the complexity of the image block to achieve the optimal recovered image quality. When the PSNR of the recovered image is given, the number of character bits is minimized by satisfying the corresponding constraints to achieve the largest tolerable tampering rate. Experimental results show the flexibility, effectiveness, and superiority of the proposed scheme compared with some reported schemes.

1. Introduction

With the rapid development of digital signal communication multimedia tools, we enjoy a mass of information from every aspect of life. At the same time, it brings us the convenience of communication, and it is also associated with information security issues. The multimedia information may suffer tampering or manipulation through transmission. Therefore, authentication for the integrity and authenticity of multimedia data is vital in communication [1–3]. Researching image authentication technology and content protection is a hot topic currently. Many techniques are applied to verify the authenticity and integrity of images, such as reversible data hiding [4–6], perceptual image hashing [7], and fragile digital watermarking [8–35]. This paper mainly studies digital fragile watermarking. According to various functions, fragile watermarking could be divided into two categories: tampering localization schemes [8–14] and self-embedding schemes [15–35]. The former is used to detect and locate the tampered regions of an image, while the latter is designed to recover the image content to its original state, in addition to identifying the tampered regions.

In [8], Walton proposed the first fragile watermarking scheme for tampering authentication. This scheme calculates the checksum of randomly selected pixels as a watermark and embeds it in the least significant bit (LSB). To resist vector quantization (VQ) attack, Celik et al. [9] proposed a hierarchical watermarking for secure image authentication by a hierarchical structure. The VQ attack is resisted by the high-level signature shared in the low level. Zhang and Wang proposed a statistical scheme of fragile watermarking to locate tampered regions with pixelwise accuracy [10]. The watermark data of this scheme consisted of tailor-made authentication data for each pixel and some additional test data that can be used to reveal the exact pattern of the tampered contents. To improve the ability to detect the tampered regions with equal modification in brightness, instead of embedding block-independent authentication code (AC) used in other methods, Hong et al. [11] embedded the hash value of block features to avoid the tampered regions that are undetectable in prior works. However, in many real applications, just detecting tampering is not enough, and it is highly desirable to recover the original content from the tampered regions. Therefore, many researchers have investigated ways of recovering the original content after tampering has been detected.

Fridrich and Goljan [15] developed a fragile watermarking scheme with self-recovery capability and proposed the first self-embedding watermarking, which encoded the discrete cosine transform (DCT) coefficients of each block and embedded them into other blocks. When the tampered blocks were detected, the tampered blocks could be recovered by the extracted bits from intact blocks. In [16], Zhu et al. proposed to use the exclusive-or (XOR) between a pseudorandom sequence and a polar sequence of DCT coefficients as a watermark. In this method, data representing the principal content in a region are always hidden in a different region within the image. If both regions are tampered with, the restoration will fail. This problem is called the tampering coincidence problem. To solve this problem, an effective dual watermark scheme was proposed in [17]. In this scheme, there are two copies of the watermark for each nonoverlapping block in the image, thereby providing a second chance for block recovery in case one copy is destroyed. In [18], a fragile watermarking with error-free restoration capability was proposed, which can achieve lossless recovery through the combination of ingenious watermark design and difference expansion algorithm. But a necessary condition of the perfect image restoration is that the proportion of tampered content must be less than 3.2%.

Zhang et al. [19] proposed a self-embedding fragile watermarking scheme based on a reference sharing mechanism. The watermark in the tampered area could be recovered accurately under a certain tampering rate, and it will be introduced in Section 2. Based on the reference sharing mechanism, an adaptive scheme is proposed in [20], which has two embedding modes, overlapping-free embedding and overlapping embedding. What is more, it takes adaptive selection between the most significant bit (MSB) and LSB layers for embedding according to different tampering rates and results in a better quality of image recovery. In 2018, Qin et al. proposed a self-recovery scheme based on a nonuniform reference sharing mechanism [21]. An optimal iterative block truncation coding (OIBTC) algorithm is used to generate recovery bits, including binary patterns and reconstruction levels. A nonuniform sharing mechanism is used to interleave these recovery bits. The recovered image quality is in the range [31, 40] dB under the tamper ratio of less than 50%. Since the traditional manner of concealing image content within the image is inflexible and fragile to diverse digital attacks, that is, image cropping and JPEG compression, to address this issue, a novel self-embedding algorithm based on deep learning was proposed by Ying et al., which is an image tamper resilient generative scheme [22] by jointly training a U-Net backboned encoder, a tamper localization network, and a decoder for image recovery.

Self-embedding based on a reference sharing mechanism is effective in solving the tampering coincidence problem; however, existing algorithms can only achieve the fixed recovered quality under different tampering rates, and it is not possible to achieve more flexible watermark embedding operations according to user customization. In some application scenarios, the user only needs an acceptable recovery quality when the tampering is large, and at the same time, the user wishes to estimate the largest tolerable tampering rate when specifying the recovery quality. To this end, we propose a new framework of self-embedding fragile watermarking based on the reference sharing mechanism in this paper. The main contributions of this work are summarized as follows:

(1) Our framework sufficiently considers the user preference and customization for tampering recovery; that is, the optimal recovered quality and the largest tolerable tampering rate can be achieved under the given conditions by the proposed scheme.

(2) When the possible tampering rate is given, we can estimate the optimal visual quality of the recovered image by designing an optimization algorithm to obtain the largest number of character bits. Detailedly, a different number of character bits generated from different methods are adaptively allocated to corresponding image blocks according to the block complexity, while the successful tampering recovery under the given tampering rate can also be guaranteed effectively.

(3) When the lowest requirement of PSNR value for the recovered image is decided, we can achieve the largest tolerable tampering rate by minimizing the total number of character bits for tampering recovery.

The remainder of this paper is organized as follows. Section 2 describes the baseline of the reference sharing mechanism and the motivation of our work. Our scheme and framework are proposed in Section 3. Theoretical analysis is given in Section 4. Section 5 presents experimental results and comparisons. Finally, we conclude this paper in Section 6.

2. Motivation

In this Section, the basic idea of reference sharing mechanism for image tampering recovery is first introduced, and then its limitations and the motivations of our work are presented subsequently.

2.1. Reference Sharing Mechanism. As proposed in [19], for the original image I sized $N_1 \times N_2$, 5 MSB layers of all $N = N_1 \times N_2$ pixels are collected as character bits for the reference bits generation, and 3 LSB layers are set to zeros for embedding watermark bits including authentication bits and reference bits.

Detailedly, all the $5N$ character bits from 5 MSB layers are first permuted by a secret key and divided into $M = 5\,N/L$ subsets, each of which containing L bits, that is, $c_{m,1}, \ldots, c_{m,L}$, and for each subset, $L/2$ reference bits can be calculated by equation

$$\begin{bmatrix} r_{m,1} \\ r_{m,2} \\ \vdots \\ r_{m,L/2} \end{bmatrix} = \mathbf{A}_m \times \begin{bmatrix} c_{m,1} \\ c_{m,2} \\ \vdots \\ c_{m,L} \end{bmatrix}, \tag{1}$$

where $\mathbf{A}_m$ is the pseudorandom $L/2 \times L$ binary matrices derived by the secret key and the arithmetic in equation (1) is modulo-2. Then, all generated $5N/2$ reference bits are permuted and divided into $N/64$ groups, each of which contains 160 bits. The original image is divided into $N/64$ blocks sized 8×8, and the 160 reference bits in each group correspond to each block. In addition, the 32 hash-based authentication bits for each 8×8 block are generated by feeding the 5 MSBs of the block and the 160 corresponding reference bits into the hash function. According to the secret key, the 160 reference bits and 32 authentication bits are permuted to form the 192 watermark bits, which are embedded into each block by LSB replacement. Thus, after all blocks are processed with the above steps, the watermarked image is obtained.

After receiving a suspicious image, the integrity of each block can be judged by comparing the equality of the extracted authentication bits and the recalculated ones. Clearly, if they are different, the corresponding block is judged as tampered; otherwise, as intact. The character bits of tampered blocks can be recovered by the reference bits extracted from intact blocks as follows:

$$\begin{bmatrix} r_{m,e(1)} \\ r_{m,e(2)} \\ \vdots \\ r_{m,e(v)} \end{bmatrix} = \mathbf{A}_m^{(E)} \times \begin{bmatrix} c_{m,1} \\ c_{m,2} \\ \vdots \\ c_{m,L} \end{bmatrix}, \quad m = 1, 2, \ldots, M, \tag{2}$$

where $r_{m,e(1)}, r_{m,e(2)}, \ldots, r_{m,e(v)}$ are the reference bits extracted from the intact blocks, the number v may be less than $L/2$ due to tampering, and $\mathbf{A}_m^{(E)}$ is the matrix composed of the rows taken from $\mathbf{A}_m$ corresponding to the rows of extracted reference bits. Then, according to whether the character bits come from tampered or intact blocks, equation (2) can be reformulated as

$$\begin{bmatrix} r_{m,e(1)} \\ r_{m,e(2)} \\ \vdots \\ r_{m,e(v)} \end{bmatrix} - A_m^{(E,R)} \times C^R = A_m^{(E,T)} \times C^T, \tag{3}$$

where $\mathbf{C}^T$ is a column vector consisting of the n_T character bits from tampered blocks, $\mathbf{C}^R$ is a column vector consisting of the $L - n_T$ character bits from intact blocks, $\mathbf{A}_m^{(E,T)}$ and $\mathbf{A}_m^{(E,R)}$ are the matrices composed of the columns taken from $\mathbf{A}_m^{(E)}$ corresponding to the bits in $\mathbf{C}^T$ and $\mathbf{C}^R$, respectively, and the size of $\mathbf{A}_m^{(E,T)}$ is $v \times n_T$. Thus, the n_T unknown character bits can be obtained by solving the v equations in a binary system, and the tampered blocks can be recovered if and only if equation (3) has a unique solution.

In [19], the authors also considered the probability of equation (3) with a unique solution. That is, if and only if the n_T columns of $\mathbf{A}_m^{(E,T)}$ are linearly independent, a unique solution to equation (3) exists. Here, the probability $q(x, y)$ of columns being linearly dependent on a random binary matrix sized $x \times y$ can be calculated as

$$q(x,y) = \begin{cases} \dfrac{1}{2^x}, & y = 1, \\ q(x, y-1) + [1 - q(x, y-1)] \times \dfrac{2^{y-1}}{2^x}, & 1 < y \leq x, \\ 1, & y > x. \end{cases} \tag{4}$$

Assume the tampering rate is γ; thus, the number v of reference bits extracted from intact blocks conforms to a binomial distribution:

$$P_v(x) = C_{1/2\times L}^x \cdot (1-\gamma)^x \cdot \gamma^{1/2\times L - x}, \quad x = 0, 1, 2, \ldots, \frac{1}{2} \times L. \tag{5}$$

The number n_T of unknown character bits from tampered blocks also conforms to a binomial distribution:

$$P_{n_T}(y) = C_L^y \cdot (1-\gamma)^{L-y} \cdot \gamma^y, \quad y = 0, 1, 2, \ldots, L. \tag{6}$$

The probability of the $n_T$columns in $\mathbf{A}_m^{(E,T)}$ being linearly independent can be calculated as

$$P_f = \sum_{x=0}^{1/2\times L} \sum_{y=0}^{L} \left\{ P_v(x) \cdot P_{n_T}(y) \cdot [1 - q(x,y)] \right\}. \tag{7}$$

Thus, the probability of unknown character bits in all M subsets, that is, 5 MSBs, being recovered is

$$P_R(\gamma, L, N) = P_f^M. \tag{8}$$

Specifically, for an original image sized $N = 512 \times 512$, when $L = 512$, the probability P_R can be equal to 1 under the tampering rate γ not greater than 24%.

2.2. Motivation of Our Work.

Based on the above analysis, we find an important parameter that expands the L character bits into several reference bits in each subset, and the parameter is called the expansion coefficient ε in our work; that is, ε represents the expansion coefficient that is the ratio between the numbers of reference bits and character bits. The expansion coefficient ε is fixed in [19]; that is, $\varepsilon = 1/2$. However, we find that the probability PR of successful tampering recovery is closely related not only to the tampering rate γ, the number of character bits L in each subset, and the image size N but also to the expansion coefficient ε.

To ensure the successful tampering recovery, $P_R(\gamma, L, N, \varepsilon) = 1$ is used as the necessary condition of the objective function in our work.

In [19], the number of character bits and expansion coefficient in each subset are fixed, which leads to the largest tolerable tampering rate equaling 24% constantly under an original image sized $N = 512 \times 512$. Even when the actual tampering rate is smaller than 24%, the PSNR of the recovered image is fixed as 40.7 dB. That is to say, this scheme is inflexible towards the tolerable tampering rate and recovered image quality, which is not applicable to the scenarios of larger tampering rate and customized recovered quality; that is, when a larger tampering rate or recovery quality is given, the user hopes to obtain the optimal recovery quality or the largest tolerable tampering rate under the corresponding conditions, but the algorithm in [19] cannot realize this function. Considering that the computational complexity and PSNR of 40.7 dB have achieved a good visual effect, our work will mainly study the problem of optimizing the recovered image quality under large tampering rates and estimating the largest tolerable tampering under the given PSNR is less than 40.7 dB. To this end, the motivation of our work is to achieve the flexible self-embedding based on a general reference sharing mechanism, which includes two main aspects in the following:

(1) When the largest requirement of tolerable tampering rate is given, how to achieve the best visual quality of the recovered image? From the analysis in Section 2.1, we can see that the quality of the recovered image is almost proportional to the total number of character bits and can also be verified from Table 1. However, the largest possible number of reference bits is decided by the number t of LSB layers used for embedding; that is, when the tampering rate is given and the tampered image is required to be recovered, the largest possible number of character bits is fixed. Therefore, we first calculate the largest number of character bits allowed and then allocate an appropriate number of character bits according to the complexity of the image block. Then, the problem of achieving the highest PSNR of the recovered image under the given tampering rate can be transformed into two optimization problems: (a) Maximize the number of character bits. (b) For blocks of different complexity, allocate an appropriate number of character bits for blocks to make the quality of the recovered image better.

(2) When the lowest requirement for the visual quality of the recovered image is given, how to achieve the largest tolerable tampering rate? First, satisfy the given recovered image quality, that is, PSNR, with the least number of character bits. And then calculate the expansion coefficient ε according to the number of reference bits and the least number of character bits. Finally, the largest tolerable tampering rate can be estimated according to equations (2)–(8). Thus, the problem of achieving the largest tolerable tampering rate under the given lowest PSNR requirement for the recovered image can be transformed into an optimization problem: minimize the number of character bits subject to satisfying the required recovered quality to achieve the largest tolerable tampering rate.

Table 1: Details of the 15 character bits generation methods.

Index	$F_i^{(M)}(\cdot)$	Character bits number per block	E_i
1	5MSB	320	5.50
2	DCT level 1	120	49.53
3	DCT level 2	112	63.41
4	DCT level 3	98	88.79
5	DCT level 4	80	127.61
6	DWT LL_1	128	176.33
7	DCT level 5	60	180.21
8	VQ level 1	40	187.65
9	VQ level 2	36	204.90
10	VQ level 3	32	227.34
11	DCT level 6	40	255.10
12	VQ level 4	28	255.56
13	DCT level 7	22	376.95
14	DCT level 8	8	638.20
15	AMBTC	80	1185.87

3. Proposed Framework

In order to solve the two above-mentioned problems in Section 2.2, we propose a general framework of self-embedding fragile watermarking based on a reference sharing mechanism, and the overall framework of our scheme is shown in Figure 1. Since the framework designed in this paper is based on blocks, the original image **I** is first divided into N_b nonoverlapping blocks of size $g \times g$, and two schemes are introduced: (1) Given the possible tampering rate, a scheme to optimize the recovery quality is designed to obtain the optimal recovered quality image. (2) Given the PSNR of the recovered image, the largest tampering rate estimation scheme is designed to obtain the largest tolerable tampering rate. More details will be described in the next two subsections.

3.1. The Optimal Recovered Quality Given the Largest Requirement of Tolerable Tampering Rate.

The problem of achieving the highest PSNR of the recovered image under the given tampering rate can be transformed into two optimization problems: (a) Maximize the total number of character bits under the condition of satisfying the given possible tampering rate. (b) Allocate an appropriate number of character bits for blocks with different complexity. For two reasons, one is the quality of the recovered image proportional to the total number of character bits, and the other is the recovery difficulty of regions with different complexities, which is usually different in the process of self-embedding recovery. Therefore, we collect many character bits generation methods to construct a library, which have different resilience to blocks of different complexity. The appropriate method from the library is selected to achieve the best recovery quality

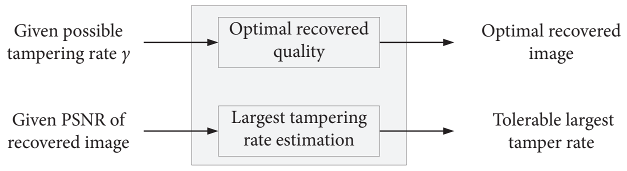

Given possible tampering rate γ
Optimal recovered quality
Optimal recovered image
Given PSNR of recovered image
Largest tampering rate estimation
Tolerable largest tamper rate

Figure 1: The overall framework of our scheme.

for blocks of different complexity. Finally, the entire optimization process includes three optimization objectives: the optimization problem for reference indicator bits generation, the optimization problem for character bits generation, and the optimization problem for recovered image quality. The optimization process is shown in Figure 2.

Firstly, in order to achieve the optimal quality of the recovered image, many character bits generation methods are collected to generate character bits and solve the minimum number problem of reference indicator bits. Secondly, determine the largest number of image content character bits that can guarantee successful recovery under the given largest tampering rate γ. Thirdly, calculate the average mean squared error (MSE) of the recovered image quality for each character bits generation method, which will be used to initially allocate the number of blocks for the character bits generation method. Then recalculate a new MSE according to the number of allocated blocks and the complexity of the blocks to obtain the optimal recovered quality of the image. Finally, watermark bits are obtained, which contain two parts: (1) reference bits consisting of image content reference bits and corresponding reference indicator bits and (2) hash-based authentication bits used to verify the authenticity of each block. After watermark bits are embedded into the original image, the watermarked image can be produced.

3.1.1. Reference Indicator Bits Generation. For image blocks with different complexity, different character bits generation methods have different recovery capabilities. To achieve better recovery quality, we collect a variety of typical character bits generation methods and denote it as $F_i^{(M)}(\cdot)$. Obviously, during the recovery process, it is necessary to identify which $F_i^{(M)}(\cdot)$ is used for each block. Thus, the indicator bits of the character bits should be recoverable. Under this requirement, we should minimize the reference indicator bits to reserve more space for content reference bit embedding while guaranteeing indicator bits to resist the possible largest tampering attacks.

The number of indicator bits used to mark each block can be calculated according to the number of $F_i^{(M)}(\cdot)$ collected in the framework; see equations (9) and (10).

$$s = \left\lceil \log_2\left(n^{(M)}\right) \right\rceil, \tag{9}$$

$$nc^{(1)} = N_b \times s, \tag{10}$$

where $n^{(M)}$ is the number of $F_i^{(M)}(\cdot)$ and N_b is the number of image blocks. As a result, there are totally $nc^{(1)}$ indicator bits, which are then interleaved to $nr^{(1)}$ reference indicator bits through the reference sharing mechanism. In fact, this interleaving process is realized by the expansion coefficient ε, which can be rewritten as $\varepsilon = f_P^{\#}(\gamma, P_R = 1, L, N)$ according to equations (4)–(8). Detailedly, during bit interleaving, the $nc^{(1)}$ bits are divided into $M^{(1)}$ groups, and each group contains $L^{(1)}$ bits. The $nr^{(1)}$ reference indicator bits are embedded into the t LSB layers of the original image $\mathbf{I}$ with $nr^{(2)}$ content reference bits; since the largest tolerable tampering rate of reference indicator bits is set to 80%, it is possible to recover even if the reference indicator bits suffer a larger tamper rate than the given tampering rate.

In order to reduce the computational complexity and ensure the recoverability of indicator bits, the tampering rate that indicator bits can resist is set to 80%; that is, when the given tampering rate is less than 80%, the relevant parameters, that is, the number of reference indicator bits and the expansion coefficient, $M^{(1)}$ and $L^{(1)}$, of the indicator bits do not need to be recalculated. When the tampering rate $\gamma = 80\%$, the least number of reference indicator bits $nr^{(1)}$ can be calculated as follows:

$$\begin{gathered} M_{\mathrm{op}}^{(1)} = \arg\min_{M^{(1)}} \left\{ M^{(1)} \times L^{(1)} \times \varepsilon^{(1)} \right\}, \\ \text{s.t. } L^{(1)} = \left\lceil \frac{nc^{(1)}}{M_{\mathrm{op}}^{(1)}} \right\rceil, \varepsilon^{(1)} = f_P^{\#}\left(\gamma = 80\%, P_R = 1, L^{(1)}, N\right), \end{gathered} \tag{11}$$

where $M^{(1)} \in [M_{T1}{}^{(1)}, M_{T2}{}^{(1)}]$ and $\varepsilon^{(1)}$ is the expansion coefficient of indicator bits.

3.1.2. Character Bits Generation. In order to achieve the best recovery quality, the total number of character bits, $nc^{(2)} = L^{(2)} \times M^{(2)}$, should be as large as possible. By combining the expansion coefficient calculation method derived from equations (4)–(8), that is, $\varepsilon^{(2)} = f_P^{\#}(\gamma, P_R = 1, L^{(2)}, N)$, and the largest possible number of reference bits possible, that is, $nr^{(\mathrm{lar})} = [(g^2 \times t - n_a) \times N_b - nr^{(1)}]$, n_a is the number of authentication bits per block. Finally, the largest number of character bits can be calculated by adjusting the variable $M^{(2)}$ and two constraints as follows:

$$\begin{gathered} M_{\mathrm{op}}^{(2)} = \arg\max_{M^{(2)}} \left\{ L^{(2)} \times M^{(2)} \right\}, \\ \text{s.t. } \varepsilon^{(2)} = f_P^{\#}\left(\gamma, P_R = 1, L^{(2)}, N\right), \varepsilon^{(2)} \times L^{(2)} \le \left\lceil \frac{nr^{(lar)}}{M_{\mathrm{op}}^{(2)}} \right\rceil, \end{gathered} \tag{12}$$

where $M^{(2)} \in [M_{T1}{}^{(2)}, M_{T2}{}^{(2)}]$.

3.1.3. Optimization of Recovered Image Quality. After the total number of character bits is calculated, in order to improve the quality of the recovered image, an appropriate $F_i^{(M)}(\cdot)$ for each block should be decided to obtain character bits of each block. In this work, PSNR is utilized to evaluate the visual quality of the recovered image with respect to the original image; see equation

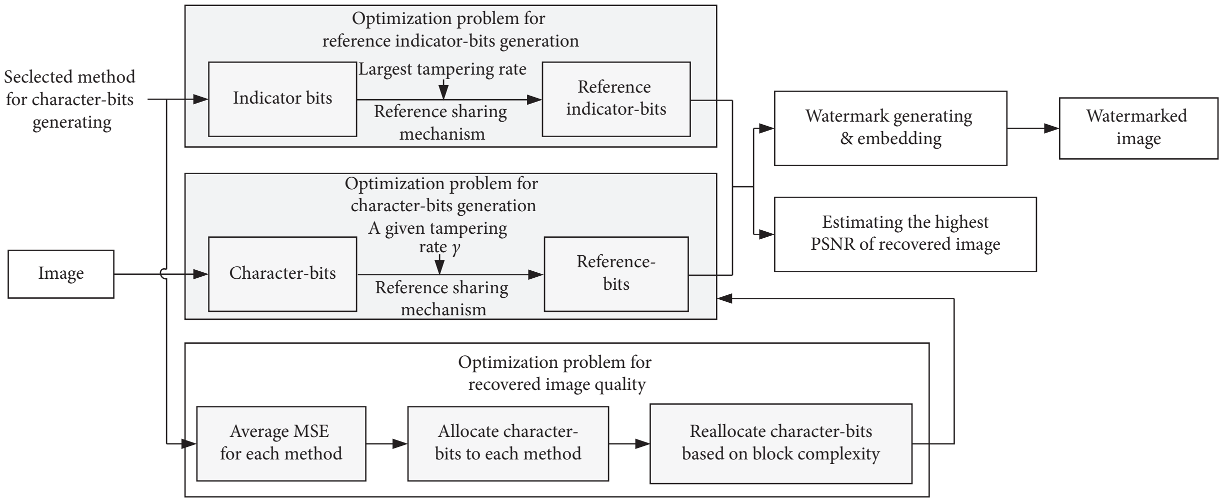

Seclected method for character-bits generating
Image
Optimization problem for reference indicator-bits generation
Indicator bits
Largest tampering rate
Reference sharing mechanism
Reference indicator-bits
Optimization problem for character-bits generation
A given tampering rate γ
Character-bits
Reference sharing mechanism
Reference-bits
Watermark generating & embedding
Watermarked image
Estimating the highest PSNR of recovered image
Optimization problem for recovered image quality
Average MSE for each method
Allocate character-bits to each method
Reallocate character-bits based on block complexity

Figure 2: A framework of the estimation for the optimal recovered image quality.

$$\begin{cases} \mathrm{PSNR} = 10 \cdot \log\left(\dfrac{255^2}{\mathrm{MSE}}\right), \\ \mathrm{MSE} = \dfrac{1}{N_1 \times N_2} \displaystyle\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} \| I(i,j) - R(i,j) \|^2, \end{cases} \tag{13}$$

where $\mathbf{I}(i,j)$ and $\mathbf{R}(i,j)$ denote the pixels in the original image and the corresponding recovered image, respectively. We take MSE minimization as the optimization objective to obtain the best recovered image quality. The average MSE E_i of each whole image in image databases is first used to initially allocate the number of blocks corresponding to the selected $F_i^{(M)}(\cdot)$. The optimization formula is as follows:

$$\begin{aligned} &\min \frac{1}{N_b} \sum_{i=1}^{n^{(M)}} \mathrm{nb}_i^{(M)} \times E_i, \\ &s.t. \sum_{i=1}^{n^{(M)}} \mathrm{nc}_i^{(M)} \times \mathrm{nb}_i^{(M)} = \mathrm{nc}^{(2)}, \sum_{i=1}^{n^{(M)}} \mathrm{nb}_i^{(M)} = N_b, \end{aligned} \tag{14}$$

where $nc_i^{(M)}$ is the number of character bits generated by the selected $F_i^{(M)}(\cdot)$ and $nb_i^{(M)}$ is the number of blocks corresponding to the selected $F_i^{(M)}(\cdot)$.

The number of blocks corresponding to the selected $F_i^{(M)}(\cdot)$ can be obtained according to the optimization of equation (14). Since the difficulty of recovering blocks with different complexities is different, in our framework, the complexity of the blocks is sorted, and the block with higher complexities are allocated to the $F_i^{(M)}(\cdot)$ with lower MSE to obtain a better quality of the recovered image. Detailedly, a complexity measurement algorithm is first proposed to sort the complexities of all blocks. Then, the better average MSE $E_i^{(M)}$ of each block can be calculated based on the specific $F_i^{(M)}(\cdot)$ and the corresponding allocated blocks; that is, $E_i^{(M)}$ is the average MSE of the recovery quality for each block corresponding to specific $F_i^{(M)}(\cdot)$. Unlike the previous E_i, which is only used to allocate the number of blocks of the corresponding $F_i^{(M)}(\cdot)$, $E_i^{(M)}$ is used to estimate the block corresponding to the appropriate complexity of the specific $F_i^{(M)}(\cdot)$ to achieve better recovery quality. Finally, the optimal MSE of the recovered image can be estimated as follows:

$$E_e = \frac{1}{N_b} \sum_{i=1}^{n^{(M)}} \mathrm{nb}_i^{(M)} \times E_i^{(M)}. \tag{15}$$

3.2. The Largest Tolerable Tampering Rate Given the Lowest Requirement of Recovered Quality. If the user wants to achieve a larger tolerable tampering rate, the larger extension coefficient should be used; thereby, less number of character bits are used; see equations (2)–(8). Thus, how to achieve the largest tolerable tampering rate under the given lowest requirement of recovered image quality can be transformed into the problem: how to achieve the given recovered quality (PSNR) with the least number of character bits.

As shown in Figure 3, PSNR is first converted to MSE, and the corresponding character bits of $F_i^{(M)}(\cdot)$ are allocated according to the optimization equation (16) to obtain the least character bits. Second, the largest tolerable tampering rate can be calculated by optimization equation (17). The reference indicator bits and indicator bits are calculated in Section 3.1. It is worth noting that since the largest tolerated tampering rate needs to be estimated, the total number of reference bits takes the maximum value; that is, $nr = (g^2 \times t - n_a) \times N_b$.

3.2.1. Minimization of the Number of Character Bits. In order to solve the least character bits that satisfy the conditions of a given PSNR, the average MSE E_i is first used to calculate the appropriate number blocks to $F_i^{(M)}(\cdot)$ to satisfy a given PSNR. And then, the least number of character bits can be calculated according to the number blocks of $F_i^{(M)}(\cdot)$ and its corresponding character bits number; the solution process is as follows:

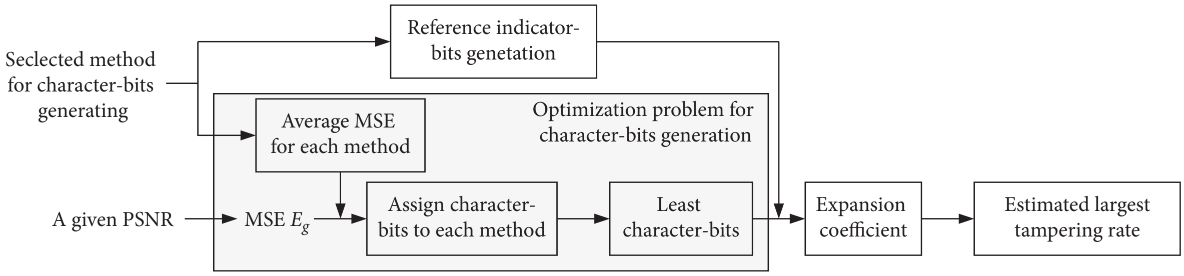


Figure 3: A framework of the estimation for the largest tolerable tampering rate.

$$\begin{gathered}\mathrm{nc}^{(\min)} = \min \sum_{i=1}^{n^{(M)}} \mathrm{nb}_i^{(M)} \times \mathrm{nc}_i^{(M)}, \\ \text{s.t. } \frac{1}{N_b} \sum_{i=1}^{n^{(M)}} E_i \times \mathrm{nb}_i^{(M)} = E_g, \sum_{i=1}^{n^{(M)}} \mathrm{nb}_i^{(M)} = N_b.\end{gathered} \tag{16}$$

3.2.2. Calculation of the Largest Tolerable Tampering Rate. After the least number of character bits to satisfy a given PSNR has been obtained by equation (16), the largest coefficient ε can be calculated; that is, $\varepsilon = nr/nc^{(\min)}$, where nr is the largest possible number of reference bits. In addition, the calculation method of the tolerable tampering rate can be derived according to equations (2)–(8); that is, $\gamma = f_P^*(P_R = 1, L, N, \varepsilon)$. Finally, according to the two constraints and the tampering rate calculation method, the largest tolerable tampering rate γ_e can be obtained by adjusting the variable M as follows:

$$\begin{gathered}M_{\mathrm{op}} = \arg\max_{M} \{ f_P^*(P_R = 1, L, N, \varepsilon) \}, \\ \text{s.t. } \varepsilon = \frac{nr}{\mathrm{nc}^{(\min)}}, L = \left\lceil \frac{nc^{(\min)}}{M_{\mathrm{op}}} \right\rceil,\end{gathered} \tag{17}$$

where $M \in [M_{T1}, M_{T2}]$.

4. Theoretical Analysis

In this section, we conduct the theoretical analysis of the performance of the proposed scheme from three aspects: (1) relevant introduction of character bits generation methods and image block complexity measurement algorithm, (2) optimal quality estimation of the recovered image under the given tampering rate, and (3) largest tolerable tampering rate estimation under the given PSNR. Note that the experimental results in the theoretical analysis are estimated according to the appropriate number of character bits obtained from the optimization formulas in Section 3 and are reasonably allocated to the selected $F_i^{(M)}(\cdot)$ for reconstructing the image without real watermark embedding and tampering recovery operation on specific images.

4.1. Character Bits Generation Methods. Suppose that the original image $\mathbf{I}$ is sized 512 × 512, which is divided into 8 × 8 nonoverlapping blocks, and the 3 LSB layers are utilized for watermark embedding. That is, $N_1 = N_2 = 512$, $t = 3$, $g = 8$, and $N_b = 4096$, and all subsequent experiments are based on these parameters. As mentioned in Section 3, an image block complexity measurement algorithm and $F_i^{(M)}(\cdot)$ are used to improve the quality of the recovered image.

Here, we define the block complexity measurement algorithm as $F^{(C)}(\cdot)$. Denote the divided blocks as $\mathbf{B}^{(i)}$, $i = 1, \ldots, N_b$, and each pixel in the block can be represented as $B_j^{(i)}$, $j = 1, \ldots, g^2$. The minimum values of the pixel in each block $B^{(i)}$ are denoted as

$$B_{\min}^{(i)} = \min\left\{B_j^{(i)}, j = 1, 2, \ldots, g^2\right\}, \quad i = 1, 2, \ldots, N_b. \tag{18}$$

Then, the average difference value of each block can be calculated as

$$D_i = \frac{1}{g^2 - 1} \sum_{j=1}^{g^2} \left(B_j^{(i)} - B_{\min}^{(i)}\right), \quad i = 1, 2, \ldots, N_b, \tag{19}$$

where D_i is the average difference and regarded as the complexity degree of each block $B^{(i)}$. Thus, all the N_b blocks can be sorted in the descending order of D_i (i.e., from rough to smooth), and we denote the sorted complexity degrees D_i for all N_b blocks as the set D_s.

In our work, 15 kinds of $F_i^{(M)}(\cdot)$ were collected for experimental analysis. For simplicity, 2 kinds of $F_i^{(M)}(\cdot)$ are used to combine, that is, $n^{(M)} = 2$, to generate character bits in our work, a total of 105 combinations. The 15 kinds of $F_i^{(M)}(\cdot)$ are derived from 5 self-embedding algorithms, of which the character bits generation methods of the 5 kinds of self-embedding algorithms are briefly described as follows:

(1) *MSB-Based Algorithm.* Five MSBs are collected as the character bits of each pixel, and a block has a total of 320 character bits [19].

(2) *DCT-Based Algorithm.* The quantized DCT coefficients of each image block are collected as the character bits. According to the DCT coefficients in [30], we construct 8 different levels of character bits generation by setting different numbers for different coefficients; that is, 8 kinds of character bits generation methods are designed.

(3) *DWT-Based Algorithm.* The low-frequency subband in level 1, LL1, of DWT coefficients is selected, and 128 character bits of a block will be generated.

(4) *AMBTC-Based Algorithm.* AMBTC [32] for each block is used, and 80 character bits will be generated.

(5) *VQ-Based Algorithm*. Different codebooks of VQ [31] are used in this paper. Four different character bits generation methods will be constructed according to 4 different codebooks corresponding to 1024, 512, 256, and 128 codewords, respectively; that is, 4 kinds of character bits generation methods are designed.

Through testing on the UCID database, the statistical average of MSE between the reconstructed image and original image for each $F_i^{(M)}(\cdot)$ described above can be estimated as listed in Table 1, which sorts all the 15 methods in the descending order of the estimated MSE. The number of generated character bits for one 8×8 block corresponding to $F_i^{(M)}(\cdot)$ is also given in Table 1.

As for the authentication bits generation method $F^{(a)}(\cdot)$, the cryptographic MD5 hash function is utilized. Thus, each 8×8 block can produce 32 authentication bits for tampering detection; that is, $n_a = 32$. As a result, in 3 LSB layers of each block, the remaining space for accommodating reference bits is $g^2 \times t - n_a = 160$.

4.2. Optimal Quality Estimation of Recovered Image. As mentioned in Section 4.1, 2 kinds of $F_i^{(M)}(\cdot)$ are actually applied; that is, $n^{(M)} = 2$; thus, s and $nc^{(1)}$ can be calculated by equation (10); that is, $s = 1$ and $nc^{(1)} = 4096$. And then, according to the optimization equation (11), the reference indicator bits can be calculated, that is, $nr^{(1)} = 29568$, when the tempering rate $\gamma = 80\%$. And then, the optimal recovered image quality under the condition of the given tampering rate can be estimated by equations (12)–(15). As shown in Table 2, the combination of optimal $F_i^{(M)}(\cdot)$ and the optimal recovered image quality is calculated. To further explain the optimization problem in Section 3.1.3, a concrete example will be introduced. Suppose the given tampering rate is $\gamma = 50\%$. Firstly, according to the optimization equation (12) and the average MSE value of each algorithm in Table 1, the number of blocks allocated to the optimal combination can be calculated, $nb_1^{(M)} = 2884$ and $nb_2^{(M)} = 1212$. $F_2^{(M)}(\cdot)$ and $F_{13}^{(M)}(\cdot)$ will be selected to generate character bits. Secondly, the blocks are sorted in descending order according to the complexity calculation algorithm $F^{(C)}(\cdot)$. Therefore, the first 2884/4096 blocks and the last 1212/4096 blocks corresponding to $F_2^{(M)}(\cdot)$ and $F_{13}^{(M)}(\cdot)$, respectively, will be used to generate character bits and estimate better MSE; that is, $E_1^{(M)} = 66.19$ dB and $E_2^{(M)} = 31.72$ dB. Finally, the optimal PSNR will be estimated; that is, $\mathrm{PSNR}_e = 30.65$ dB. As shown in Table 2, more relevant optimization results are given. The relationship between the tampering rate and the estimated optimal PSNR_e is given in Figure 4.

4.3. Estimation of the Largest Tolerable Tampering Rate. As described in Section 3.2, the given PSNR of the recovered image is denoted as PSNR_g, and then PSNR_g is converted to MSE, denoting it as E_g. Firstly, the least number of character bits that satisfies the given PSNR_g is calculated by the optimization equation (16). Secondly, the largest tolerable tampering rate is estimated based on the calculated least number of character bits and the optimization equation (17). As shown in Table 3, more relevant optimization results are given.

We can know from [19] that the highest PSNR is 40.7 dB in our instance with the largest tampering rate being 24%. The relationship between the estimated largest tolerable tampering rate γ_e and given PSNR_g is shown in Figure 5.

5. Experimental Results and Comparisons

In this paper, all experiments were implemented on a computer with a 3.70 GHz Intel i9 processor, 32.0 G memory, and Windows 10 operating system, and the programming environment was Matlab R2020b. The relevant parameter settings are shown in Section 4.1. As shown in Figure 6, the image size used in the following experiments is 512×512.

5.1. Results of Our Framework. By observing the embedding process, the 5 MSBs of each pixel in the image keep unchanged during the embedding process, and only 3 LSBs are used to embed the watermark. Therefore, the PSNR of watermarked image can be obtained by calculating the change of 3 LSBs. The calculation process is as follows:

$$\left\{ E_D = \sum_{g_o=0}^{7} \sum_{g_w=0}^{7} \frac{(g_w - g_o)^2}{64} = 10.5, \mathrm{PSNR}_w \approx 10 \cdot \log_{10}\left(\frac{255^2}{E_D}\right) = 37.9\,\mathrm{dB}, \right. \tag{20}$$

where g_o and g_w are the decimal values of the three original LSBs and three new LSBs of a pixel, respectively. Since the new LSBs are produced in a pseudorandom manner, the distribution of g_w is approximately uniform. It is assumed that the original distribution of the data in the three LSB layers is also uniform.

In the experiment of this paper, only two kinds of recovered image quality are calculated. PSNR_r and PSNR_b are the PSNR of the recovered image compared with the whole original image and the recovered parts of the recovered image compared with the corresponding part in the original image, respectively. As shown in Tables 4–6, standard test images and 100 images in UCID are used for random tampering experiments under different tampering rate conditions. The average in Tables 4 and 5 represents the average value of the experiments by using 100 images in UCID; due to the inconsistency between the complexity of the UCID images and the standard images, the experimental results are slightly different. Compared with the estimated PSNR_e in Section 4.1, the PSNR_r in the experiment is slightly different. Because the UCID database is used in the estimation process, the image complexity in the database is inconsistent with the 6 images used in the test. At the same time, the estimated PSNR_e is calculated by the original image and the reconstructed image using the number of character bits calculated in equations (9)–(15) and the corresponding $F_i^{(M)}(\cdot)$, without the real watermark embedding and tampering recovery process.

As shown in Figure 7, the tampered image is on the left, and the recovered image is on the right. A recovered image

Table 2: Statistic and estimated results under different given γ.

γ	25%	30%	40%	50%	60%	70%	80%
(nc, L, M)	(1223050, 610, 2005)	(932325, 465, 2005)	(569420, 284, 2005)	(372807, 207, 1801)	(224316, 124, 1809)	(135900, 90, 1510)	(68105, 53, 1285)
Index of $F_i^{(M)}(\cdot)$	(1, 2)	(1, 2)	(1, 2)	(2, 13)	(2, 9)	(8, 14)	(12, 14)
$(nc_1^{(M)}, nc_2^{(M)})$	(320, 120)	(320, 120)	(320, 120)	(120, 22)	(120, 36)	(40, 8)	(28, 8)
(E_1, E_2)	(5.50, 49.53)	(5.50, 49.53)	(5.50, 49.53)	(49.53, 376.95)	(49.53, 204.90)	(187.65, 638.20)	(255.56, 638.20)
$(nb_1^{(M)}, nb_2^{(M)})$	(3657, 439)	(2204, 1892)	(389, 3707)	(2884, 1212)	(915, 3181)	(3222, 874)	(1766, 2330)
$(E_1^{(M)}, E_2^{(M)})$	(5.50, 8.02)	(5.50, 16.19)	(5.50, 42.04)	(66.19, 31.72)	(107.35, 122.59)	(219.74, 27.96)	(448.32, 138.24)
MSE_e	5.77	10.44	38.57	55.99	119.19	178.82	271.93
PSNR_e	40.52	37.95	32.27	30.65	27.37	25.61	23.79

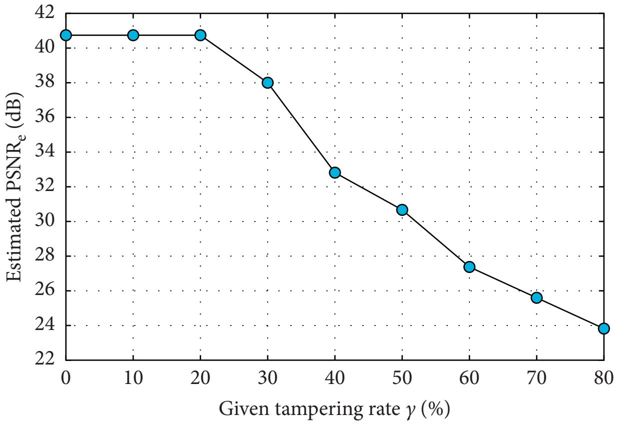


Figure 4: Estimated PSNR_e under the given tampering rates γ.

Table 3: Statistic and estimated results under different given PSNR_g.

PSNR_g (dB)	E_g	Index of $F_i^{(M)}(\cdot)$	(E_1, E_2)	$(nb_1^{(M)}, nb_2^{(M)})$	$nc^{(\min)}$	γ_e (%)
38.00	10.31	(1, 2)	(5.50, 49.53)	(3648, 448)	1221120	25
37.00	12.97	(1, 2)	(5.50, 49.53)	(3400, 696)	1171520	26
36.00	16.33	(1, 2)	(5.50, 49.53)	(3088, 1008)	1109120	27
35.00	20.56	(1, 2)	(5.50, 49.53)	(2694, 1402)	1030320	28
34.00	25.89	(1, 2)	(5.50, 49.53)	(2199, 1877)	931320	30
33.00	32.59	(1, 2)	(5.50, 49.53)	(1575, 2521)	806520	33
32.00	41.03	(1, 2)	(5.50, 49.53)	(790, 3306)	649520	37
31.00	51.65	(2, 8)	(49.53, 187.65)	(4033, 63)	486480	44
30.00	60.03	(2, 8)	(49.53, 187.65)	(3636, 460)	454720	45
29.00	81.86	(2, 8)	(49.53, 187.65)	(3137, 959)	414800	47
28.00	103.06	(2, 8)	(49.53, 187.65)	(2508, 1588)	364480	50
27.00	129.74	(2, 8)	(49.53, 187.65)	(1717, 2379)	301200	54
26.00	163.34	(3, 8)	(63.41, 187.65)	(801, 3295)	221512	60
25.00	205.63	(9, 10)	(204.90, 227.34)	(3963, 133)	146924	68
24.00	258.87	(12, 14)	(255.56, 638.20)	(4060, 36)	113968	73
23.00	325.90	(12, 14)	(255.56, 638.20)	(3343, 753)	99628	75
22.00	410.28	(12, 14)	(255.56, 638.20)	(2439, 1657)	81548	78
21.50	460.34	(12, 14)	(255.56, 638.20)	(1903, 2193)	70828	80
21.00	516.51	(12, 14)	(255.56, 638.20)	(1302, 2794)	58808	82

with better visual quality can be obtained by using the proposed scheme. For intentional tampering, we take image *Lake* in Figure 6(d) as an example with tampering rates set to be 30%, 60%, and 80%. The results are shown in Figure 8. They are the watermarked image, the tampered image, the tampering detection result, and the recovered image from

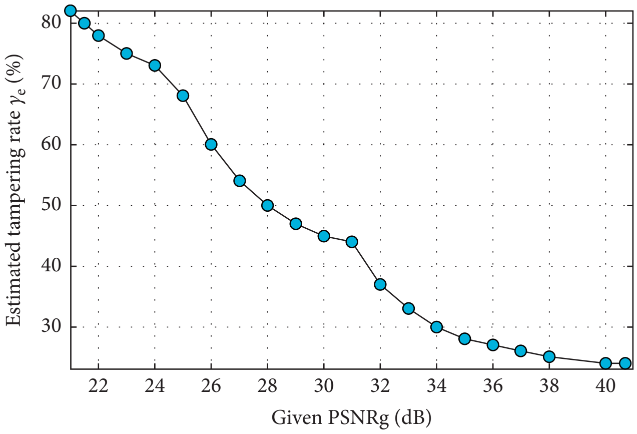


Figure 5: Estimated largest tolerable tampering rate γ_e under the given PSNR_g.

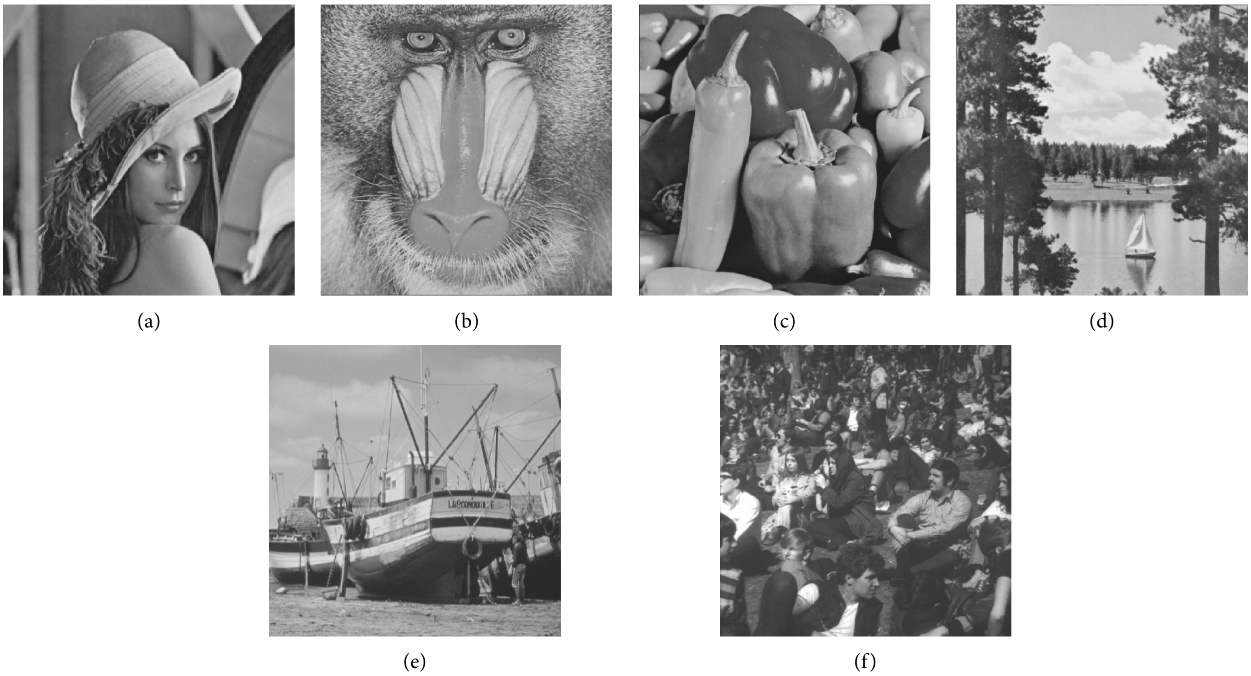


Figure 6: Standard test images. (a) *Lena*. (b) *Baboon*. (c) *Peppers*. (d) *Lake*. (e) *Boat*. (f) *Crowd*.

left to right. Additionally, the detected tampered parts are marked with white in the results.

5.2. Comparison with State-of-the-Art Schemes. In Table 6, we compare the MSB-based self-embedding watermarking algorithms. It can be observed from Table 6 that the proposed scheme can achieve more robustness against tampering rate and comparable quality of recovered image quality than the reported schemes under the different tampering rates. Furthermore, 80% is not the upper limit of the tampering rate but is set by us considering the recovery quality. A larger tampering rate can still be achieved in our framework by sacrificing recovery quality.

Table 4: PSNR$_r$ of experimental images under different given γ (dB).

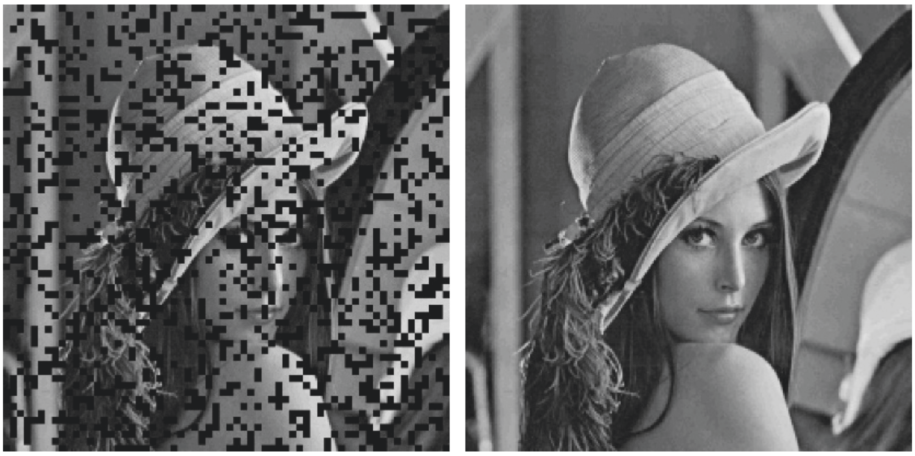
γ	25%	30%	40%	50%	60%	70%	80%
Lena	36.39	36.06	34.70	33.95	32.92	31.38	29.78
Baboon	36.36	34.82	30.82	29.06	26.79	25.11	22.69
Peppers	36.36	35.90	34.62	33.86	32.14	30.51	28.57
Lake	36.36	35.66	33.57	32.72	30.75	28.70	26.59
Boat	36.36	36.10	34.55	33.79	31.99	30.19	28.05
Crowd	36.21	35.81	34.47	33.62	31.55	29.50	26.90
Average	36.34	35.81	33.91	32.55	29.49	27.08	25.04

Table 5: PSNR$_b$ of experimental images under different given γ (dB).

γ	25%	30%	40%	50%	60%	70%	80%
Lena	39.46	36.99	33.52	32.71	31.74	30.34	29.04
Baboon	39.20	33.26	27.77	26.55	24.80	23.68	21.77
Peppers	39.47	36.50	33.41	32.58	30.77	29.38	27.77
Lake	39.45	35.59	31.60	30.97	29.13	27.42	25.73
Boat	39.44	37.20	33.26	32.45	30.58	29.03	27.24
Crowd	39.69	36.71	33.30	32.32	30.10	28.29	26.06
Average	39.56	36.42	32.38	30.98	27.83	25.77	24.78

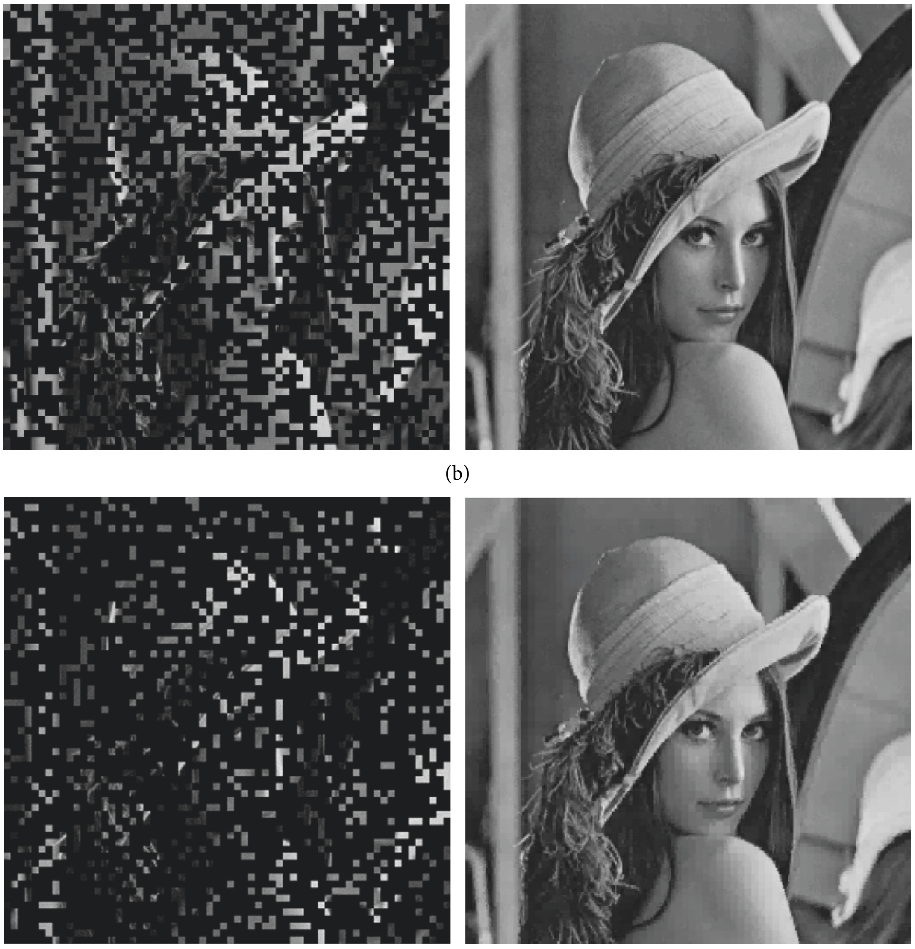
Table 6: Performance comparisons with MSB-based schemes.

Watermarking scheme	PSNR of watermarked image (dB)	PSNR of recovered content (dB)	Condition of successful restoration
Scheme 1 in [19]	38	40.7	Tampering rate ≤24%
Scheme 2 in [19]	38	[22, 40]	Tampering rate ≤66%
Scheme in [33]	38	[31, 42]	Tampering rate ≤50%
Proposed scheme	38	[22, 41]	Tampering rate ≤80%



(a)

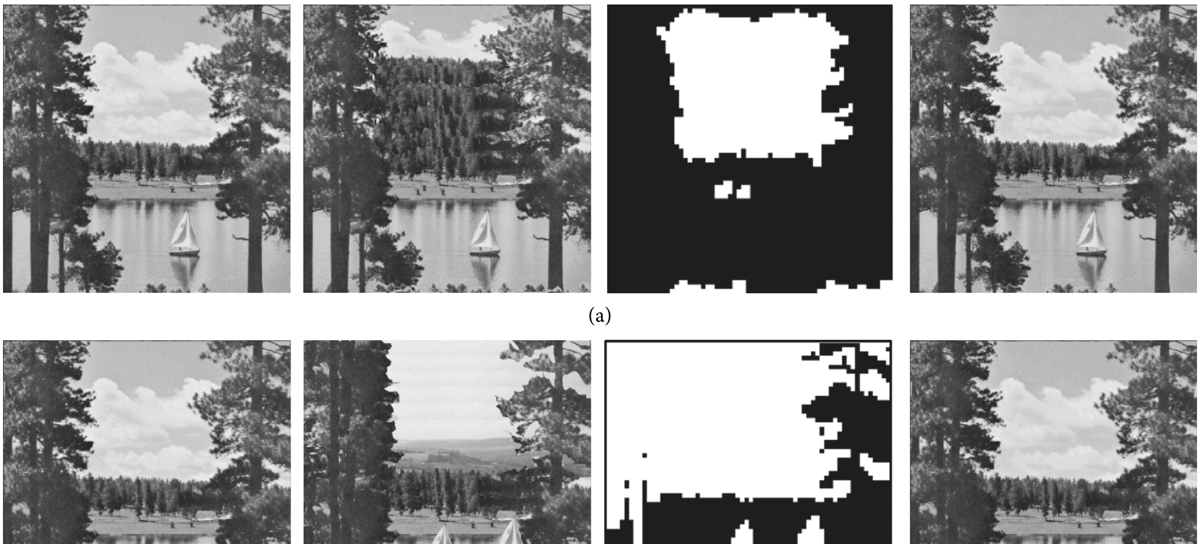
Figure 7: Continued.



(b)

(c)

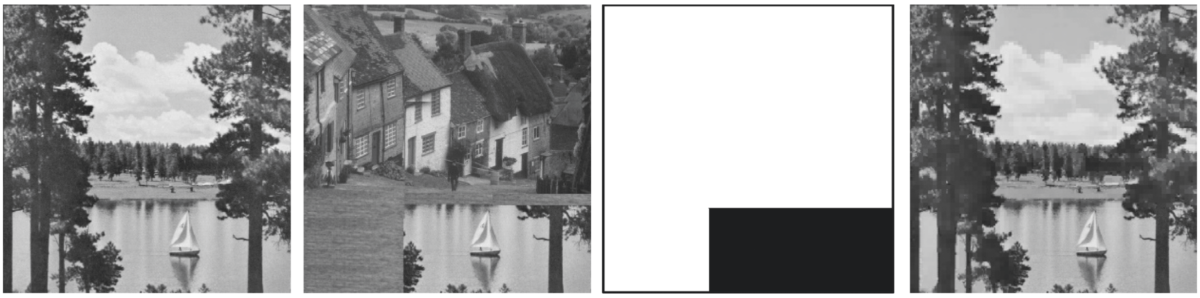
Figure 7: Random tampering and recovered results of *Lena* with different γ. (a) $\gamma = 30\%$, $\text{PSNR}_\text{r} = 30.06$ dB, and $\text{PSNR}_b = 36.99$ dB. (b) $\gamma = 60\%$, $\text{PSNR}_r = 32.92$ dB, and $\text{PSNR}_b = 31.74$ dB. (c) $\gamma = 80\%$, $\text{PSNR}_r = 29.78$ dB, and $\text{PSNR}_b = 29.04$ dB.



(a)

(b)

Figure 8: Continued.



(c)

Figure 8: Intentional tampering and recovered results of *Lake* with different γ. (a) $\gamma = 30\%$, PSNR$_r = 35.79$ dB, and PSNR$_b = 36.00$ dB. (b) $\gamma = 60\%$, PSNR$_r = 31.22$ dB, and PSNR$_b = 29.66$ dB. (c) $\gamma = 80\%$, PSNR$_r = 26.21$ dB, and PSNR$_b = 25.39$ dB.

Table 7: Comparison of PSNR of recovered content with respect to original content under different tampering rates with DCT-based schemes.

Tampering rate (%)	*Lena*			*Baboon*		
	[34]	[35]	Proposed (dB)	[34]	[35]	Proposed (dB)
25	39.50 dB	39.69	39.46	36.06 dB	31.24	39.20
50	35.79 dB	29.82	32.71	33.16 dB	26.35	26.55
80	—	24.18	29.04	—	21.89	21.77

Furthermore, we take *Lena* and *Baboon* as examples to compare the recovered quality with other schemes under the conditions of some tampering rate. In Table 7, we compare the self-embedding watermarking algorithms based on DCT coefficients. Considering the tampering rate that the scheme can tolerate and the quality of the recovered image, the scheme proposed in this paper can achieve a better trade-off.

6. Conclusions

In this paper, we proposed a new self-embedding framework based on reference sharing mechanism. Different from the reported schemes that the PSNR of the recovered image can only be calculated by completing the entire embedding process, the framework of the proposed scheme can be categorized into estimating the highest PSNR of the recovered image and estimating the largest tolerable tampering rate when the tampering rate and the PSNR of the recovered image are given, respectively, because the number of character bits that are used will influence the quality of the recovered image and the ability to resist the tampering rate. In this paper, the problem of estimating the highest PSNR of the recovered image is first transformed into the problem of calculating the largest number of character bits and then reallocating the character bits to each block according to the complexity of the image block to achieve the best recovery quality. The problem of estimating the largest tolerable tampering rate is transformed into the problem of calculating the least number of character bits, and then the largest tolerable tampering rate can be obtained. In addition, the experimental results show the flexibility and effectiveness of the proposed scheme.

Data Availability

The image datasets used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China, under Grants 62172280, U20B2051, and 62172281, the Natural Science Foundation of Shanghai, under Grant 21ZR1444600, the Natural Science Foundation of Shandong, under Grant ZR2020MF054, the STCSM Capability Construction Project for Shanghai Municipal Universities, under Grant 20060502300, the Shandong Provincial Natural Science Foundation (ZR2019BF017), and Major Scientific and Technological Innovation Projects of Shandong Province (2019JZZY010127, 2019JZZY010132, and 2019JZZY010201).

References

[1] J.-C. Lee, C.-P. Chang, and W.-K. Chen, “Detection of copy-move image forgery using histogram of orientated gradients,” *Information Sciences*, vol. 321, pp. 250–262, 2015.

[2] J. Jian Li, X. L. Xiaolong Li, B. Bin Yang, and X. M. Xingming Sun, “Segmentation-based image copy-move forgery detection scheme,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, 2015.

[3] F. Zou, Y. Chen, J. Song, K. Zhou, Y. Yang, and N. Sebe, “Compact image fingerprint via multiple kernel hashing,” *IEEE Transactions on Multimedia*, vol. 17, no. 7, pp. 1006–1018, 2015.

[4] X. Wang, X. Wang, B. Ma, Q. Li, and Y.-Q. Shi, “High precision error prediction algorithm based on ridge regression predictor for reversible data hiding,” *IEEE Signal Processing Letters*, vol. 28, pp. 1125–1129, 2021.

[5] B. Ma and Y. Q. Shi, “A reversible data hiding scheme based on code division multiplexing,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1914–1927, 2016.

[6] C. Qin, W. Zhang, F. Cao, X. Zhang, and C.-C. Chang, “Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection,” *Signal Processing*, vol. 153, pp. 109–122, 2018.

[7] C. Qin, E. Liu, G. Feng, and X. Zhang, “Perceptual image hashing for content authentication based on convolutional neural network with multiple constraints,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4523–4537, 2021.

[8] S. Walton, “Image authentication for a slippery new age,” *Dr. Dobb’s Journal*, vol. 20, no. 4, pp. 18–26, 1995.

[9] M. Utku Celik, G. Sharma, E. Saber, and A. Murat Tekalp, “Hierarchical watermarking for secure image authentication with localization,” *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 585–595, 2002.

[10] X. Zhang and S. Wang, “Statistical fragile watermarking capable of locating individual tampered pixels,” *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 727–730, 2007.

[11] W. Hong, M. Chen, and T. S. Chen, “An efficient reversible image authentication method using improved PVO and LSB substitution techniques,” *Signal Processing: Image Communication*, vol. 58, pp. 111–122, 2017.

[12] X. L. Liu, C. C. Lin, and S. M. Yuan, “Blind dual watermarking for color images’ authentication and copyright protection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 5, pp. 1047–1055, 2016.

[13] E. Gul and S. Ozturk, “A novel hash function based fragile watermarking method for image integrity,” *Multimedia Tools and Applications*, vol. 78, pp. 1–18, 2019.

[14] N. R. Neena Raj and R. Shreelekshmi, “Security analysis of hash based fragile watermarking scheme for image integrity,” *International Conference on Intelligent Computing, Instrumentation and Control Technologies*, vol. 1, pp. 651–654, 2019.

[15] J. Fridrich and M. Goljan, “Images with self-correcting capabilities,” in *Proceedings of the IEEE International Conference on Image Processing*, pp. 792–796, IEEE, Kobe, Japan, October 1999.

[16] X. Zhu, A. T. S. Ho, and P. Marziliano, “A new semi-fragile image watermarking with robust tampering restoration using irregular sampling,” *Signal Processing: Image Communication*, vol. 22, no. 5, pp. 515–528, 2007.

[17] T.-Y. Lee and S. D. Lin, “Dual watermark for image tamper detection and recovery,” *Pattern Recognition*, vol. 41, no. 11, pp. 3497–3506, 2008.

[18] X. Zhang and S. Wang, “Fragile watermarking with error-free restoration capability,” *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1490–1499, 2008.

[19] X. Zhang, S. Wang, Z. Qian, and G. Feng, “Reference sharing mechanism for watermark self-embedding,” *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 485–495, 2011.

[20] C. Qin, H. Wang, X. Zhang, and X. Sun, “Self-embedding fragile watermarking based on reference-data interleaving and adaptive selection of embedding mode,” *Information Sciences*, vol. 373, pp. 233–250, 2016.

[21] C. Qin, P. Ji, C.-C. Chang, J. Dong, and X. Sun, “Non-uniform watermark sharing based on optimal iterative BTC for image tampering recovery,” *IEEE MultiMedia*, vol. 25, no. 3, pp. 36–48, 2018.

[22] Q. C. Ying, Z. X. Qian, H. Zhou, H. S. Xu, X. P. Zhang, and S. Y. Li, “From image to imuge: immunized image generation,” in *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3565–3573, Association for Computing Machinery, New York, NY, U S A, October 2021.

[23] L. Huang, D. Kuang, C. Li, Y. Zhuang, S. Duan, and X. Zhou, “A self-embedding secure fragile watermarking scheme with high quality recovery,” *Journal of Visual Communication and Image Representation*, vol. 83, Article ID 103437, 2022.

[24] M. Jana, B. Jana, and S. Joardar, “Local feature based self-embedding fragile watermarking scheme for tampered detection and recovery utilizing AMBTC with fuzzy logic,” *Journal of King Saud University-Computer and Information Sciences*, 2022.

[25] M. Swain and D. Swain, “An effective watermarking technique using BTC and SVD for image authentication and quality recovery,” *Integration*, vol. 83, pp. 12–23, 2022.

[26] Z. N. You, Y. Liu, and T. G. Gao, “A lossless self-recovery watermarking scheme with JPEG-LS compression,” *Journal of Information Security and Applications*, vol. 58, Article ID 102733, 2021.

[27] X. Yuan, X. Li, and T. Liu, “Gauss–Jordan elimination-based image tampering detection and self-recovery,” *Signal Processing: Image Communication*, vol. 90, Article ID 116038, 2021.

[28] J. Molina-Garcia, B. P. Garcia-Salgado, V. Ponomaryov, R. Reyes-Reyes, S. Sadovnychiy, and C. Cruz-Ramos, “An effective fragile watermarking scheme for color image tampering detection and self-recovery,” *Signal Processing: Image Communication*, vol. 81, Article ID 115725, 2020.

[29] W.-L. Tai and Z.-J. Liao, “Image self-recovery with watermark self-embedding,” *Signal Processing: Image Communication*, vol. 65, pp. 11–25, 2018.

[30] Z. Qian, G. Feng, X. Zhang, and S. Wang, “Image self-embedding with high-quality restoration capability,” *Digital Signal Processing*, vol. 21, no. 2, pp. 278–286, 2011.

[31] C. Qin, C.-C. Chang, and K.-N. Chen, “Adaptive self-recovery for tampered images based on VQ indexing and inpainting,” *Signal Processing*, vol. 93, no. 4, pp. 933–946, 2013.

[32] C.-C. Tsou, Y.-C. Hu, and C.-C. Chang, “Efficient optimal pixel grouping schemes for AMBTC,” *The Imaging Science Journal*, vol. 56, no. 4, pp. 217–231, 2008.

[33] P. Korus and A. Dziech, “Efficient method for content reconstruction with self-embedding,” *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 1134–1147, 2013.

[34] D. Singh and S. K. Singh, “DCT based efficient fragile watermarking scheme for image authentication and restoration,” *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 953–977, 2017.

[35] X. Zhang, Y. Xiao, and Z. Zhao, “Self-embedding fragile watermarking based on DCT and fast fractal coding,” *Multimedia Tools and Applications*, vol. 74, no. 15, pp. 5767–5786, 2015.