Hindawi
Security and Communication Networks
Volume 2018, Article ID 2516324, 12 pages
https://doi.org/10.1155/2018/2516324

Research Article

Reference Sharing Mechanism-Based Self-Embedding Watermarking Scheme with Deterministic Content Reconstruction

Dongmei Niu ,[1,2] Hongxia Wang ,[1] Minquan Cheng,[3] and Canghong Shi[1]

[1] *Southwest Jiaotong University, Chengdu 610031, China*
[2] *Southwest University of Science and Technology, Mianyang 621010, China*
[3] *Guangxi Normal University, Guilin 541004, China*

Correspondence should be addressed to Hongxia Wang; hxwang@swjtu.edu.cn

Received 29 September 2017; Revised 19 December 2017; Accepted 9 January 2018; Published 11 February 2018

Academic Editor: Xinpeng Zhang

This paper presents a reference sharing mechanism-based self-embedding watermarking scheme. The host image is embedded with watermark bits including the reference data for content recovery and the authentication data for tampering location. The special encoding matrix derived from the generator matrix of selected systematic Maximum Distance Separable (MDS) code is adopted. The reference data is generated by encoding all the representative data of the original image blocks. On the receiver side, the tampered image blocks can be located by the authentication data. The reference data embedded in one image block can be shared by all the image blocks to restore the tampered content. The tampering coincidence problem can be avoided at the extreme. The maximal tampering rate is deduced theoretically. Experimental results show that, as long as the tampering rate is less than the maximal tampering rate, the content recovery is deterministic. The quality of recovered content does not decrease with the maximal tampering rate.

1. Introduction

With the rapid development of information science and computer network techniques, digital images can be easily copied, altered, and spread over the network. The problems of copyright protection, authentication, and integrity identification of digital images [1, 2] are still a focus of multimedia information security research. Self-embedding watermarking technique is proposed [3] to detect the tampered image areas and recover approximately the tampered content. In most self-embedding watermarking schemes, the original image will be partitioned into nonoverlapping blocks. In addition to the authentication data for detecting the tampered image blocks, the reference data for recovering the tampered image blocks is embedded in the redundant space of the image. The watermark data is composed of the reference data and the authentication data. The performance of self-embedding schemes is commonly evaluated by the peak signal to noise ratio (PSNR) between the original image and the watermarked image, PSNR between the recovered image and the watermarked image, and a bound on the allowed amount of modifications, that is, the maximal tampering rate.

In some self-embedding schemes, the reference data is the representative information of the original image blocks such as the prime DCT (Discrete Cosine Transform) coefficients, the MSB (Most Significant Bit) bits of the block pixels or the vector quantization values. The reference data of an image block is usually embedded into another different image block according to the predetermined rule. In [3], the primary DCT coefficients of an image block were quantized and binary encoded, and the resulting bit string was inserted into the LSB of an offset block. In [4], the reference data is also the quantized and encoded DCT coefficients; the embedding position is determined by a block-mapping sequence. As the analysis in [5], this embedding method will result in the problems of the tampering coincidence and watermark waste. The first problem means that when both the image block and the image block containing its representative information are tampered or lost, the content recovery will fail. The second problem means that when both of them are reserved, the watermark data embedded in will be useless. To alleviate the problem of the tampering coincidence, in some other schemes the representative information for one image block will be duplicated and embedded in the image for many times. As in [6], for each image block, there are two copies of its representative information hidden in two different blocks that provide the second chance for block recovery in case one copy is destroyed. In [7], the representative information of the image block is the VQ index. Compared with [6], more copies of the representative information are embedded. The same processing method appear in [8–11]. By using this method the probability of tampering coincidence is reduced, but the cost of watermark waste increased.

In some other schemes, coding technique is introduced. The reference information is not the representative data of the original image blocks. The reference data is generated by encoding the representative information [12–23]. In [12, 13], the reference sharing mechanism for watermark self-embedding is proposed. The reference information of one image block will be distributed over many blocks. The 5 MSB bits of the image pixels are pseudo-randomly permuted and divided into groups. Each group is multiplied by a pseudo-random binary matrix to generate the reference data. All the reference bits are then pseudo-randomly permuted and divided into groups, which will be embedded in the image blocks. Scrambling and coding make the reference bits to be embedded in an image block derived from the MSB bits of many different image blocks and shared by these blocks for content restoration. When the number of tampered image blocks is not more than a certain threshold, there is always some part of the reference available. So, the tampering coincidence problem is avoided. This thought is also reflected in the other schemes [14–19]. In [17], the reference sharing mechanism for watermark self-embedding is extended. The numbers of the MSB layers to generate the reference bits are flexible and the numbers of LSB layers to accommodate watermark bits are variable. The relationship for the overall performance of self-embedding scheme, the embedding modes that are used, and the ranges of tampering rates are presented. In [21], the erasure channel is taken as the natural model of the self-embedding problem, and the random linear fountain (RLF) code is used to encode the representative information of all image blocks to generate the reference data. The reference bits to be embedded in an image block will be shared by all the image blocks for content restoration; the tampering coincidence problem can be avoided at the extreme. For that reason, with the same rate of reference information per image block, the proposed approach in [21] allows for working with higher tampering rates than other self-embedding schemes. In these classic reference sharing mechanism-based self-embedding schemes, the binary random matrix is used as the encoding matrix. The tampered image blocks will be restored with probability, but not deterministically. In our scheme, we use the special matrix as the encoding matrix to construct the reference sharing mechanism-based self-embedding scheme. The reference information to be embedded in an image block can be shared by all the image blocks. So the tampering coincidence problem can be avoided at the extreme. As long as the tampering rate is not larger than the maximal tampering rate, the representative data of the tampered image blocks can be recovered deterministically.

The remaining part of the paper is organized as follows. Section 2 reviews the reference data generation method of the prior reference sharing mechanism-based schemes. The detailed procedure of the proposed self-embedding watermarking scheme is presented in Section 3. Section 4 analyzes the bound on the maximal tampering rate of the proposed scheme. The experimental evaluation and comparisons with the existing schemes are presented in Section 5.

2. Related Prior Research

The method of the reference sharing for self-embedding schemes is proposed and described in detail in [12, 13]. This technique is also applied in some other schemes [17–19, 21, 22]. But in these schemes the encoding matrices are the binary random matrices. In [13] the original image is an 8-bit gray-level image. The 5 MSB of each pixel are collected and permuted based on the secret key and then divided into M subsets, each of which containing L bits. For each subset, the reference data generation is performed by

$$\left(r_{m,1}, r_{m,2}, \ldots, r_{m,L/2}\right)^T = \mathbf{R}_m \left(c_{m,1}, c_{m,2}, \ldots, c_{m,L}\right)^T, \quad m = 1, 2, \ldots, M, \tag{1}$$

where $\mathbf{R}_m$ is the encoding matrix, which is a binary random matrix sized $L/2 \times L$, and $(c_{m,1}, c_{m,2}, \ldots, c_{m,L})$ is the mth subset of the 5 MSB. The generated reference data will be stored as part of the watermark in the 3 LSB planes of the image block. The M systems of linear equations in (1) establish a link between the reference data and the 5 MSB of the original image. Scrambling and coding make the reference bits embedded in an image block derived from the MSB bits of many different image blocks and shared by these blocks for content restoration.

After the tampering detection procedure, all the image blocks of the watermarked image will be marked as either "tampered" or "reserved." The ratio between the number of tampered image blocks and the number of all blocks is called the tampering rate, which will be denoted as α. The maximal tampering rate is denoted as $\alpha_{\max}$, which is the upper bound of the tampering rate the scheme can tolerate. Recollect the 5 MSB of each pixel while marking the MSB of the tampered blocks as the unknowns. Separate the reference bits from the watermark while marking the reference bits of the tampered blocks as the unknowns. Reconstruct the M systems of linear equations in (1). For each equation, the invalid equations that the reference bit is unknown are removed:

$$\left(r_{m,e(1)}, r_{m,e(2)}, \ldots, r_{m,e(v)}\right)^T = \mathbf{R}_m^{(E)} \left(c_{m,1}, c_{m,2}, \ldots, c_{m,L}\right)^T, \quad m = 1, 2, \ldots, M, \tag{2}$$

where $(r_{m,e(1)}, r_{m,e(2)}, \ldots, r_{m,e(v)})^T$ and $\mathbf{R}_m^{(E)}$ are the constant vector and the coefficient matrix after removing the invalid equations. Then reformulate the equations for the standard system of equations as follows:

$$\left(r_{m,e(1)}, r_{m,e(2)}, \dots, r_{m,e(v)}\right)^T - \mathbf{R}_m^{(E,R)}\mathbf{C}_R = \mathbf{R}_m^{(E,T)}\mathbf{C}_T, \qquad m = 1, 2, \dots, M, \tag{3}$$

where $\mathbf{R}_m^{(E,T)}$ is the coefficient matrix of the standard system of equations. If the rank of the coefficient matrix $\mathbf{R}_m^{(E,T)}$ is equal to the number of unknowns, the system of equations in (3) will have the unique solution. In other words, the necessary and sufficient condition for the solution of (3) is that, for any submatrix of the binary random matrix, if the number of rows of it is greater than the number of columns, the submatrix is full column rank. However, this condition can only be satisfied with the probability because both the matrix $\mathbf{R}_m$ and the tampering are random. The probability is dependent upon the tampering rate, image size, and system parameter L, which has been deduced in [13].

In brief, using the binary random matrix as the encoding matrix, the procedure of the encoding and decoding will be simple. However, because of the randomness of the encoding matrix and tampering, the tampered image blocks will be restored with probability, but not deterministically. According to the knowledge of coding theory [24], we knew that if $[\mathbf{I} \mid \mathbf{A}]$ is the generator matrix of the systematic MDS code, any square submatrix of $\mathbf{A}$ will be nonsingular. This is equivalent to saying that, for any submatrix of $\mathbf{A}$, if the number of rows of it is greater than the number of columns, the submatrix is full column rank. Due to the excellent properties of the matrix $\mathbf{A}$, we use it in this paper to construct the reference sharing mechanism-based self-embedding watermarking schemes. We select the appropriate matrix $\mathbf{A}$ to generate the reference data by encoding the representative data of all image blocks based on the matrix $\mathbf{A}$. By this way, the generated reference information embedded in an image block will be shared by all the image blocks. Based on this spreading mechanism, our method can be immune to the tampering coincidence and the reference waste. Moreover, after locating the tampered image blocks by the embedded authentication data, as long as the tampering rate is larger than the maximal tampering rate, the restoration will be deterministic due to the use of the matrix $\mathbf{A}$. The tampered image blocks can be reconstructed by using the recovered representative data.

3. Proposed Self-Embedding Watermarking Scheme

Similar to the common self-embedding schemes, the proposed watermarking scheme includes the following two parts: the first one is the watermark generation and embedding and the second one is the tampering detection and content recovery. The detailed process will be described in the following section.

3.1. Watermark Generation and Embedding. Watermark generation and embedding procedure can be divided into four phases: the first one is the representative data generation, the second is the reference data generation, the third one is the authentication data generation, and the last one is watermarking embedding.

3.1.1. Representative Data Generation

Step 1. Divide the original image $\mathbf{I}$ into K nonoverlapping blocks. They are marked as the first, the second, and so on and the Kth block by the Zig-Zag order.

Step 2. Collect the representative data of each image block. The representative data can be the compressed data of image block, for example, the prime DCT coefficients and the MSB bits of the block pixels. There are K representative data blocks in total, which are denoted as $(\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_K)$.

Step 3. Calculate the ratio R. R is the ratio of the length of the representative data block to the size of the redundant space used to embed the reference data in one image block. In our scheme, we suppose R is an integer or $1/R$ is an integer.

For example, suppose the original image is divided into 9 blocks sized 8 × 8 pixels. The redundant space of an image block is the 3 LSB of all pixels in the image block. If we use 160 bits to store the reference data block data and we use the prime DCT coefficients to represent the image block and quantified and encoded the DCT coefficients to 80 bits, the calculated ratio R is 1/2. If the 5 MSB of all pixels in an image block are extracted as the representative data of this image block, the length of the representative data block is 320 bits and the calculated ratio R is 2.

3.1.2. Reference Data Generation

Step 1. Encoding the representative data of image blocks to generate the reference data $(\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_K)$, there are two cases needed to be considered.

Case 1. The ration R is less than or equal to 1. Encode the K representative data blocks in the following way:

$$\begin{aligned}&(\mathbf{C}_{11}, \mathbf{C}_{12}, \dots, \mathbf{C}_{1,1/R}, \mathbf{C}_{21}, \mathbf{C}_{22}, \dots, \mathbf{C}_{2,1/R}, \dots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \dots,\\ &\quad \mathbf{C}_{K,1/R}) = (\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_K)\, \mathbf{A}_{K\times K/R},\end{aligned} \tag{4}$$

where $\mathbf{A}$ is the K rows and K/R columns matrix and $[\mathbf{I} \mid \mathbf{A}]$ is the generator matrix of the systematic $((1/R + 1)K, K)$-MDS code over the finite field. The calculating is finished over the finite field. For this purpose, $\mathbf{D}_i$ $(i = 1, \dots, K)$ will be transformed to an n-dimensional column vector in the finite field. For example, $\mathbf{D}_{11}$ is transformed to $(d_{11}, d_{21}, \dots, d_{n1})^T$. So, (4) can be rewrote as

$$\begin{aligned}&(\mathbf{C}_{11}, \mathbf{C}_{12}, \dots, \mathbf{C}_{1,1/R}, \mathbf{C}_{21}, \mathbf{C}_{22}, \dots, \mathbf{C}_{2,1/R}, \dots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \dots,\\ &\quad \mathbf{C}_{K,1/R}) = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1,K} \\ d_{21} & d_{22} & \cdots & d_{2,K} \\ \vdots & \vdots & \cdots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{n,K} \end{bmatrix} \mathbf{A}_{K\times K/R}.\end{aligned} \tag{5}$$

So, $\mathbf{C}_{ij}$ $(i = 1, \ldots, K,\ j = 1, \ldots, 1/R)$ are n-dimensional column vectors in the finite field. Let

$$\mathbf{C}_i = (\mathbf{C}_{i1}, \mathbf{C}_{i2}, \ldots, \mathbf{C}_{i,1/R}) \quad i = 1, \ldots, K. \tag{6}$$

The reference data $(\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K)$ in the finite field is generated.

Case 2. The ration R is greater than 1. First, $\mathbf{D}_i$ $(i = 1, 2, \ldots, K)$ are divided into R smaller blocks $\mathbf{D}_{i1}, \mathbf{D}_{i2}, \ldots, \mathbf{D}_{iR}$. The length of each smaller block will be equal to the size of the redundant space used to embed the reference data block in an image block. Then generate the K data blocks in the following way:

$$\begin{aligned}(\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K) = (&\mathbf{D}_{11}, \mathbf{D}_{12}, \ldots, \mathbf{D}_{1,R}, \mathbf{D}_{21}, \mathbf{D}_{22}, \ldots, \mathbf{D}_{2,R}, \\ &\ldots, \mathbf{D}_{K1}, \mathbf{D}_{K2}, \ldots, \mathbf{D}_{KR})\, \mathbf{A}_{RK\times K},\end{aligned} \tag{7}$$

where $\mathbf{A}$ is the RK rows and K columns matrix and $[\mathbf{I} \mid \mathbf{A}]$ is the generator matrix of the systematic $((R+1)K, RK)$-MDS code over the finite field. As described above, to finish the calculation over the finite field $\mathbf{D}_{ij}$ $(i = 1, \ldots, K,\ j = 1, \ldots, R)$ will be transformed to an n-dimensional column vector in the finite field. So, (7) can be rewrote as

$$(\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K) = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1,KR} \\ d_{21} & d_{22} & \cdots & d_{2,KR} \\ \vdots & \vdots & \cdots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{n,KR} \end{bmatrix} \mathbf{A}_{RK\times K}. \tag{8}$$

$(\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K)$ is the reference data in the finite field.

Step 2. Transform $\mathbf{C}_i$ $(i = 1, \ldots, K)$ to bit strings. The bit strings are denoted as $(\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K)$.

The generated reference data blocks $(\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_K)$ will be embedded as a part of watermark into the redundant space of the corresponding image block. From (4) and (7), it can be seen that the data block $\mathbf{C}_{ij}$ or $\mathbf{C}_i$ is the linear combination of all the representative data blocks. That means each data block carries the information of all the image blocks. The reference data block $\mathbf{R}_i$ embedded in any image block can provide the recovery information equally for any tampered image, or the reference data block $\mathbf{R}_i$ embedded in any image block will be shared by all the image blocks. By this way, a global reference share mechanism has been realized.

3.1.3. Authentication Data Generation. For the ith $(i = 1, \ldots, K)$ image block, the representative data $\mathbf{D}_i$ and the reference data $\mathbf{R}_i$ are connected and then fed into a hash function to generate the hash bits $\mathbf{H}_i$. The hash values $\{\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_K\}$ are the authentication data blocks which will be embedded into the redundant space of the image block as a part of the watermark. The redundant space of the image block is divided into two parts, one for the reference data and the rest for the hash data. So, the length of the hash data is equal to the length of the rest redundant space. In our experiment, we use the MD5 function; the output is shortened by exclusive disjunction on neighboring bit pairs to generate the required length hash data.

3.1.4. Watermark Embedding

Step 1. The reference data $\mathbf{R}_i$ and the authentication data $\mathbf{H}_i$ are connected and permuted based upon the secret key to generate the watermark $\mathbf{W}_i$ $(i = 1, \ldots, K)$.

Step 2. $\mathbf{W}_i$ is embedded into the redundant space of the ith image block. After all the image blocks have been processed, the watermarked image is produced. In our experiment, the 3 LSB of the ith image block is replaced by the watermark $\mathbf{W}_i$ to generate the watermarked image.

3.2. Tampering Detection and Content Recovery. Suppose the watermarked image has been altered without changing the size. For the receiver, the tampered image blocks will be identified and located firstly; then the tampered image blocks will be recovered. So, the tampering detection and content recovery procedure can be divided into two phases: tampered blocks detection and tampered blocks recovery.

3.2.1. Tampered Blocks Detection

Step 1. The received image is divided and the representative data of all the image blocks is collected as in the preprocessing.

Step 2. For the ith image block, the watermark is extracted from the redundant space, scrambled inversely using the same secret key and decomposed into two parts: the reference data block and the hash data.

Step 3. For each image block, input the representative data and the extracted reference to the HASH function to recalculate the hash value.

Step 4. Compare the recalculated hash value and the extracted hash data. If they are different, the image block is judged as a "tampered" image block; otherwise, it is judged as a "reserved" image block.

3.2.2. Tampered Blocks Recovery. As long as the tampering rate is not larger than the maximal tampering rate, we can perfectly recover the failed representative data of the tampered image blocks. The maximal tampering rate will be derived theoretically in Section 4. The procedure of tampered blocks recovery can be illustrated as follows. Take the case that the ration R is less than or equal to 1 as an example; the same result can be derived when R is greater than 1.

Step 1. Reconstruct the linear equations (4). Suppose there are t tampered image blocks. In order to explain the problem simply, it may be assumed that the front t blocks are tampered. The remaining $K - t$ blocks are the reserved image blocks. The reference data extracted from the tampered image blocks and the representative data of them are denoted as $(\mathbf{C}_1^*, \mathbf{C}_2^*, \ldots, \mathbf{C}_t^*)$ and $(\mathbf{D}_1^*, \mathbf{D}_2^*, \ldots, \mathbf{D}_t^*)$. The reference data extracted from the reserved image blocks and the representative data of them are denoted as $(\mathbf{C}_{t+1}, \mathbf{C}_{t+2}, \dots, \mathbf{C}_{K-t})$ and $(\mathbf{D}_{t+1}, \mathbf{D}_{\mathbf{t}+2}, \dots, \mathbf{D}_{K-t})$. Divide $\mathbf{C}_i^*$ and C_i into $1/R$ parts:

$$\begin{aligned} \mathbf{C}_i^* &= \left(\mathbf{C}_{i1}^*, \mathbf{C}_{i2}^*, \dots, \mathbf{C}_{i1/R}^*\right) \quad i = 1, 2, \dots, t, \\ \mathbf{C}_i &= \left(\mathbf{C}_{i1}, \mathbf{C}_{i2}, \dots, \mathbf{C}_{i,1/R}\right) \quad i = t+1, t+2, \dots, K. \end{aligned} \tag{9}$$

Then we can reconstruct the linear equations (4) as

$$\begin{aligned} &\left(\mathbf{C}_{11}^*, \mathbf{C}_{12}^*, \dots, \mathbf{C}_{11/R}^*, \dots, \mathbf{C}_{t1}^*, \mathbf{C}_{t2}^*, \dots, \mathbf{C}_{t1/R}^*, \mathbf{C}_{(t+1)1}, \mathbf{C}_{(t+1)2}, \right. \\ &\quad \left. \dots, \mathbf{C}_{(t+2)1/R}, \dots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \dots, \mathbf{C}_{K1/R}\right) = \left(\mathbf{D}_1^*, \mathbf{D}_2^*, \dots, \right. \\ &\quad \left. \mathbf{D}_t^*, \mathbf{D}_{(t+1)}, \dots, \mathbf{D}_K\right) \mathbf{A}_{K \times K/R} \end{aligned} \tag{10}$$

The representative data of the tampered image blocks $(\mathbf{D}_1^*, \mathbf{D}_2^*, \dots, \mathbf{D}_t^*)$ are the unknowns of the linear equations. We need to solve them by (10).

Step 2. Eliminate the equations that are invalid. $(\mathbf{C}_{11}^*, \mathbf{C}_{12}^*, \dots, \mathbf{C}_{11/R}^*, \dots, \mathbf{C}_{t1}^*, \mathbf{C}_{t2}^*, \dots, \mathbf{C}_{t1/R}^*)$ are from the tampered image blocks. The data may have been tampered. So, their equations are invalid. Cross out the invalid equations and reformulate the system of equations as a standard form of equations.

$$\begin{aligned} &\left(\mathbf{C}_{(t+1)1}, \mathbf{C}_{(t+1)2}, \dots, \mathbf{C}_{(t+2)1/R}, \dots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \dots, \mathbf{C}_{K1/R}\right) \\ &\quad = \left(\mathbf{D}_1^*, \mathbf{D}_2^*, \dots, \mathbf{D}_t^*, \mathbf{D}_{(t+1)}, \dots, \mathbf{D}_K\right) \mathbf{A}_{K \times (K-t)/R}^{E}, \end{aligned} \tag{11}$$

where $\mathbf{A}_{K \times (K-t)/R}^{E}$ is the $(K-t)/R$ columns taken from $\mathbf{A}_{K \times K/R}$ corresponding to the extracted correct reference data blocks $(\mathbf{C}_{(t+1)1}, \mathbf{C}_{(t+1)2}, \dots, \mathbf{C}_{(t+2)1/R}, \dots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \dots, \mathbf{C}_{K1/R})$.

Step 3. Rearranging (11) as the standard form and moving the portion with the unknowns to the right of the equations, we can reformulate (11) as follows:

$$\begin{aligned} &\left(\mathbf{C}_{(t+1)1}, \mathbf{C}_{(t+1)2}, \dots, \mathbf{C}_{(t+2)1/R}, \dots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \dots, \mathbf{C}_{K1/R}\right) \\ &\quad - \left(\mathbf{D}_{(t+1)}, \dots, \mathbf{D}_K\right) \mathbf{A}_{(K-t) \times (K-t)/R}^{(E,R)} \\ &\quad = \left(\mathbf{D}_1^*, \mathbf{D}_2^*, \dots, \mathbf{D}_t^*\right) \mathbf{A}_{t \times (K-t)/R}^{(E,T)}, \end{aligned} \tag{12}$$

where $\mathbf{A}_{(K-t) \times (K-t)/R}^{(E,R)}$ and $\mathbf{A}_{t \times (K-t)/R}^{(E,T)}$ are the rows of $\mathbf{A}_{K \times r/R}^{(E)}$ corresponding to the representative data blocks $(\mathbf{D}_{(t+1)}, \dots, \mathbf{D}_K)$ and $(\mathbf{D}_1^*, \mathbf{D}_2^*, \dots, \mathbf{D}_t^*)$, respectively.

Step 4. Solve the t unknowns $(\mathbf{D}_1^*, \mathbf{D}_2^*, \dots, \mathbf{D}_t^*)$ according to the equations. The calculation will be finished over the finite field. It can be demonstrated that if the tampering rate is not larger than the maximal tampering rate, the number of equations is more than the number of the unknowns. So, we can rewrite (12) as

$$\begin{aligned} S - \left(\mathbf{D}_{(t+1)}, \dots, \mathbf{D}_K\right) \mathbf{A}_{(K-t) \times t}^{(E,R,t)} \\ = \left(\mathbf{D}_1^*, \mathbf{D}_2^*, \dots, \mathbf{D}_t^*\right) \mathbf{A}_{t \times t}^{(E,T,t)}, \end{aligned} \tag{13}$$

where S is the front t data blocks of $(\mathbf{C}_{(t+1)1}, \mathbf{C}_{(t+1)2}, \dots, \mathbf{C}_{(t+2)1/R}, \dots, \mathbf{C}_{K1}, \mathbf{C}_{K2}, \dots, \mathbf{C}_{K1/R})$. $\mathbf{A}_{t \times t}^{(E,T,t)}$ is the $t \times t$ matrix whose columns are the first t columns of the matrix $\mathbf{A}_{t \times (K-t)/R}^{(E,T)}$. We can see that the matrix $\mathbf{A}_{t \times t}^{(E,T,t)}$ is the square submatrix of $\mathbf{A}$. From [24], we have known that $[\mathbf{I} \mid \mathbf{A}]$ is the generator matrix of the systematic MDS code if and only if any square submatrix of $\mathbf{A}$ is nonsingular. So, $\mathbf{A}_{t \times t}^{(E,T,t)}$ will be nonsingular because $[\mathbf{I} \mid \mathbf{A}]$ is the generator matrix of systematic MDS. Therefore, (13) has a unique solution. We can solve (13) over the finite field to retrieve the original values of $(\mathbf{D}_1^*, \mathbf{D}_2^*, \dots, \mathbf{D}_t^*)$. So, we can recover the representative data of tampered image blocks definitely. Similarly, the same result can be derived when $R > 1$.

The recovered representative data can be used to reconstruct the tampered image blocks. The quality of recovered content depends on the method of generating the representative data. Provided that the tampering rate is not larger than the maximal tampering rate, the quality of the reconstructed content does not degrade with the tampering area increasing.

4. The Upper Bound on the Tampering Rate

Suppose t image blocks are tampered. Because any square submatrix of the coding matrix $\mathbf{A}$ is nonsingular, as long as the number of equations is more than the number of the unknowns in (12), (13) will have the unique solution. From this the maximal tampering rate $\alpha_{\max}$ can be easily derived theoretically.

Case 1. The ration R is less than or equal to 1. If an image block is identified as a tampered block, there will be one data block and $1/R$ reference data blocks stored in the image block identified as the failed data blocks. After crossing out the invalid equations, there will be $(K-t)/R$ valid equations. In order to make (13) have unique solution, we should have $(K-t)/R \geq t$. From this inequality, we can work out $t/K \leq 1/(R+1)$. t/K is the ratio of the tampered image blocks to all the image blocks. So, in this case the maximal tampering rate $\alpha_{\max}$ is $1/(R+1)$.

Case 2. The ration R is greater than 1. In this case, the same conclusion can be drawn according to the discussion method in Case 1. So, In all cases, the maximal tampering rate

$$\alpha_{\max} = \frac{1}{R+1}. \tag{14}$$

Figure 1 shows the curve about the maximal tampering rate with respect to the ratio R. The curve indicate that the maximal tampering rate will decrease as the ratio R decreases. If we want to improve the restoration capability, we should try to reduce the ratio R by reducing the length of the representative data block or improve the size of the redundant space.

5. Experimental Evaluation and Comparisons

Experiments and comparison were conducted to demonstrate the effectiveness and evaluate the performance of the proposed scheme.

Figure 2(a) is the standard test gray scale image lake sized 512 × 512 which is used as the host image. The host image is

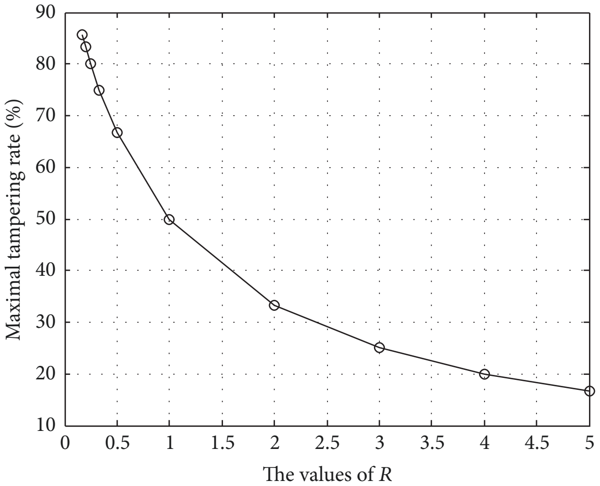


Figure 1: The maximal tampering rate with different R.

divided into blocks sized 8 × 8 pixels. So, the number of image blocks $K = 2^{12}$. The representative data of an image block is all the 5 MSB of the pixels in the image block. So, the length of the representative data of an image block is 320 bits. The redundant space of an image block is all the 3 LSB of the pixels in the image block. The size of it is 192 bits. We will use 160 bits to store the reference data and the remaining 32 bits to store the hash data. So, we can calculate the ratio R is 2. The coding matrix $\mathbf{A}$ should be a $2^{13} \times 2^{12}$ matrix and $[\mathbf{I} \mid \mathbf{A}]$ should be the systematic MDS code generator matrix. Here we generate the matrix $\mathbf{A}$ by constructing the $2^{13} \times 2^{12}$ Cauchy matrix over $G\ (2^{16})$ [24].

The 320-bit protected data of each image block will be divided into two smaller blocks with size of 160 bits. So, there are 2^{13} smaller data blocks in total. Each data block will be represented as a column vector of 10 elements in the finite field $G\ (2^{16})$. Then we calculate the 2^{12} reference data blocks according to (7). Each reference data block will be a column vector of 10 elements in the finite field $G\ (2^{16})$ and can be transformed into a bits string of length 160 bits. For each image block, the representative data will be linked with the corresponding reference data and then is fed into the MD5 function to produce the hash value. The output hash bits are shortened by exclusive disjunction on neighboring bit pairs to generate the 32 bits hash data. Then, the 160 bits reference data and the 32 bits hash data are linked and permuted. We permuted the 192 bits based on a pseudo-random sequence from the logistic chaotic system and use the initial condition as the secret key. The permuted 192 bits are embedded into the image block by replacing the three LSB planes of the block. This way, the watermarked image is produced. Figure 2(b) is the watermarked Lake. The values of PSNR due to watermark embedding are 37.9 dB. According to (14), it can be derived that the maximal tampering rate of the tested self-embedding scheme is 1/3.

Figure 3 shows the results of the meaningful tampering experiments. The watermarked lake is maliciously tampered with tampering rate $\alpha = 9.8\%$, 21.83%, and 32.69%. The tampered watermarked images are shown in Figure 3((a1)–(a3)) and their corresponding identification and restoration results are shown in Figure 3((b1)–(b3)) and Figure 3((c1)–(c3)). We can see all tampered blocks are located correctly. The tampered blocks are represented by the extreme white. The original MSB of tampered blocks were recovered without any error. In the three cases, PSNR values in the restored area are all 40.7 dB when regarding original image as reference. The quality of the recovered content does not degrade with the growth of tampering rate. We applied the method in [13] and forced the first and second LSB of the restored area as 0 and the third LSB as 1. The experiment demonstrates that if the ratio $R = 1/2$, the proposed scheme can perfectly recover the representative data of the tampered image blocks as long as the tampering rate $\alpha \leq 1/3$.

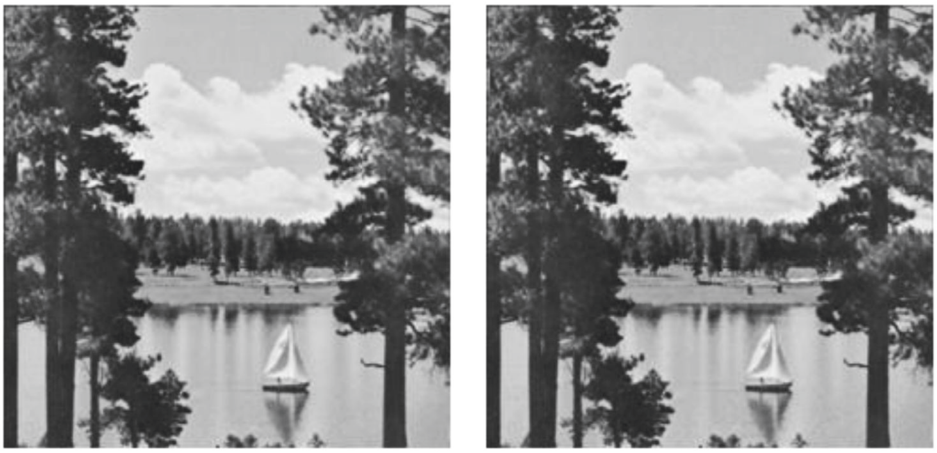
To evaluate the performance of the proposed scheme, we also conduct the random tampering experiments, seen in Figure 4. The gray scale image Lena, Baboon, Gold Hill, and Airplane in Figure 4((a1)–(d1)) are used as the host images. The watermarked images, shown in Figure 4((a2)–(d2)), are generated as the lake. The values of PSNR due to watermark embedding are 37.9. The watermarked images are tampered randomly with tampering rate $\alpha = 10\%$, 18%, 24%, 33%. The tampered images are shown in Figure 4((a3)–(d3)). The corresponding identification and restoration results are shown in Figure 4((a4)–(d4)) and Figure 4((a5)–(d5)). It can be seen that the perfect recovery has been realized in all the experiments. PSNR values in the restored area are all 40.7 dB when regarding original image as reference.

Another experiment was conducted to test the performance of the proposed scheme when the ratio R is less than 1. In the experiment, the standard test gray scale image lake sized 512 × 512 (shown in Figure 5(a)) is still used as the host image. The host image is divided into blocks sized 8 × 8 pixels. The representative data of an image block is the quantified and encoded DCT coefficients. The quantization procedure is the same as that employed in [21]. The quantified DCT coefficients in each block are converted to binary sequences by the following allocation vector:

$$\{8, 7, 4, 3, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\} \tag{15}$$

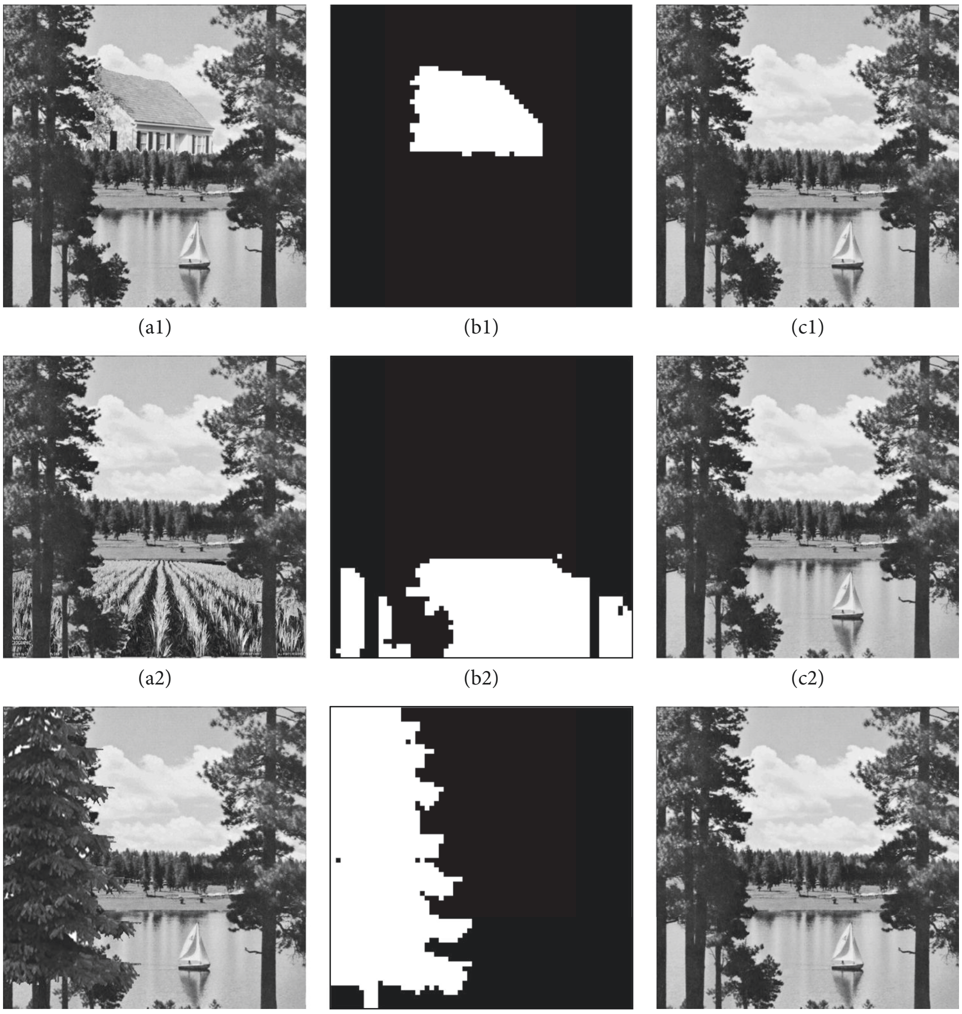
which results in a total of 56-bit representative data of an image block.

The redundant space of an image block is still all the 3 LSB of each pixel in the image block. So, the size of redundant space is still 192 bits. We will use 168 bits to store the reference data block and the remaining 24 bits to store the hash data. So the ratio R is 1/3. The coding matrix $\mathbf{A}$ will be 2^{12} rows and 3×2^{12} columns matrix and $[\mathbf{I} \mid \mathbf{A}]$ should be the generator matrix of a systematic MDS code. Just as in the first experiment, we generate the matrix A by constructing the 2^{12} rows and 3×2^{12} columns Cauchy matrix over $G\ (2^{14})$. The 56-bit representative data of each image block will be transformed into 4 elements in the finite field $G\ (2^{14})$. According to (4), the 3×2^{12} column vectors in the finite field $G\ (2^{14})$ are generated. For i $(i = 1, \ldots, 2^{12})$, we transform C_{ij} $(j = 1, 2, 3)$ to bit strings and then connect the three bit



(a) (b)

Figure 2: (a) Original image lake. (b) Watermarked lake.



(a1) (b1) (c1)

(a2) (b2) (c2)

(a3) (b3) (c3)

Figure 3: Results of the meaningful tampering experiments. ((a1)–(a3)) Tampered lake with α = 9.8%, 21.83%, 32.69%. ((b1)–(b3)) Tampered blocks identification result of ((a1)–(a3)). ((c1)–(c3)) Restored version of ((a1)–(a3)).

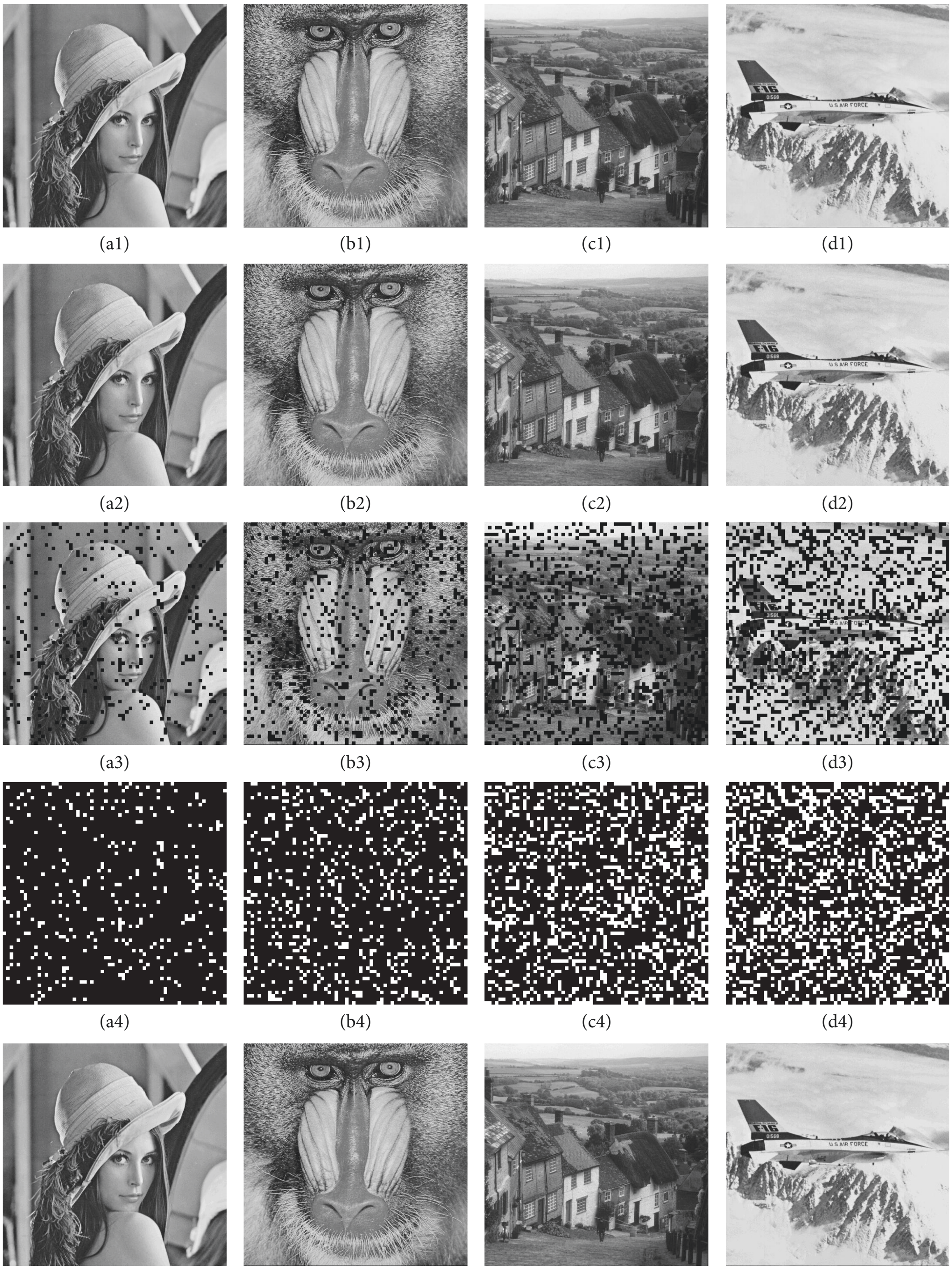


Figure 4: Results of the random tampering experiments. (a1)–(d1) are the original gray images; (a2)–(d2) are the watermarked images; (a3)–(d3) are the random tampered watermarked images with α = 10%, 18%, 24%, 33%; (a4)–(d4) are the tampered blocks identification results; (a5)–(d5) are the tampering restoration results.

strings to generate the 168 bits reference data R_i. The 24-bit hash data and the watermarked image (shown in Figure 5(b)) are generated as in the first experiment. The values of PSNR due to watermark embedding are 37.9 dB. According to (14), it can be calculated that the maximal tampering rate of the tested self-embedding scheme is 3/4.

Figure 6 shows three meaningful tampering experiments. The watermarked lake is maliciously tampered with tampering rate α = 32.69%, 54.35%, and 74.76%. The tampered watermarked images are shown in Figure 6((a1)–(a3)), and their corresponding identification and restoration results are shown in Figure 6((b1)–(b3)) and Figure 6((c1)–(c3)). Just

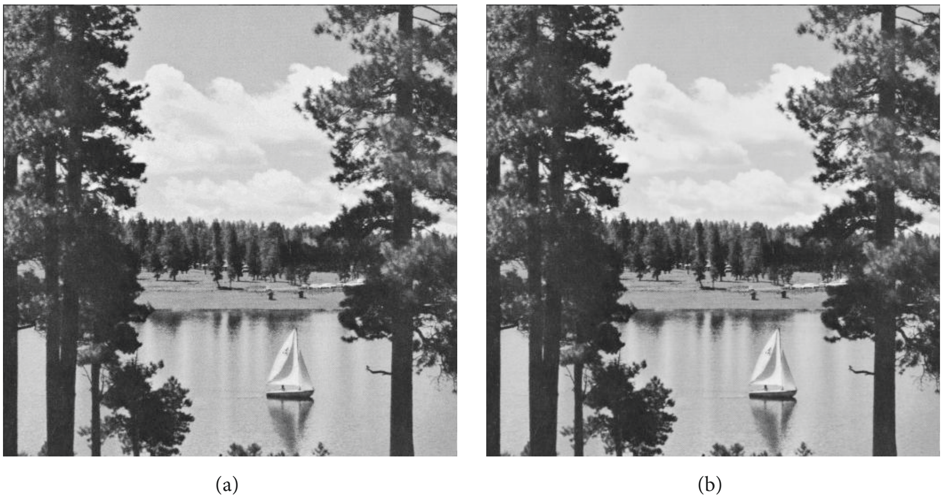


Figure 5: (a) Original image. (b) Watermarked.

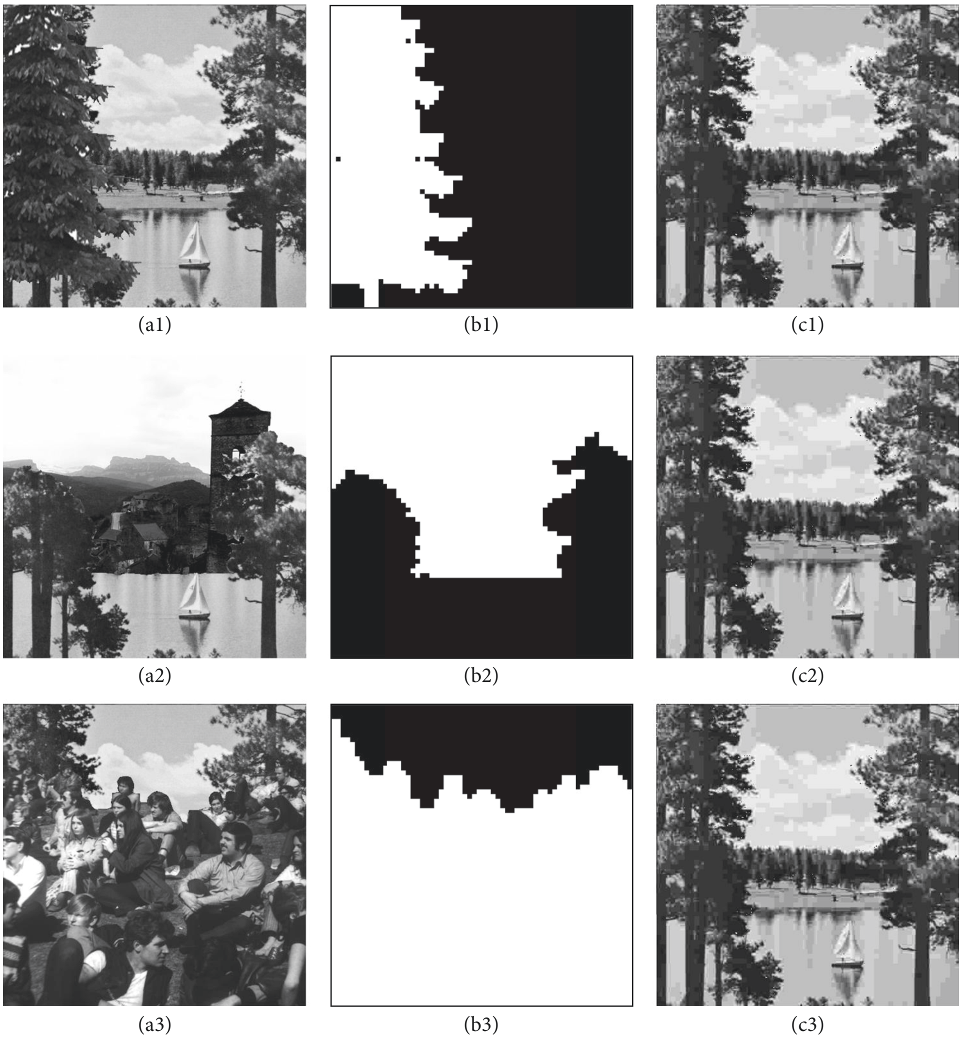
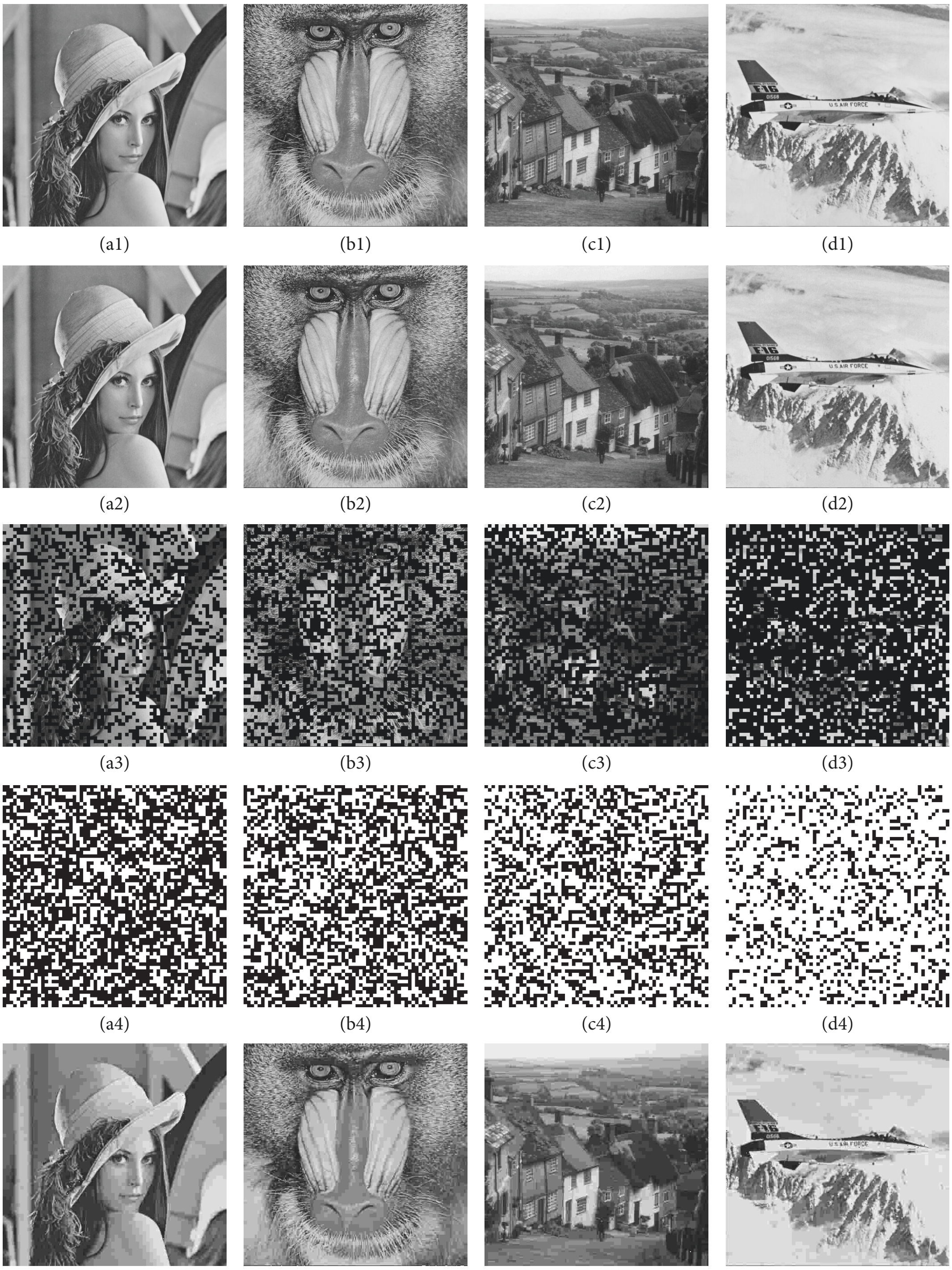


Figure 6: Results of the meaningful tampering experiments. ((a1)–(a3)) Tampered lake with α = 32.69%, 54.37%, 74.76%. ((b1)–(b3)) Tampered blocks identification result of (a1)–(a3). ((c1)–(c3)) Restored version of (a1)–(a3).



(a1) (b1) (c1) (d1)

(a2) (b2) (c2) (d2)

(a3) (b3) (c3) (d3)

(a4) (b4) (c4) (d4)

(a5) (b5) (c5) (d5)

Figure 7: Results of the random tampering experiments. (a1)–(d1) are the original gray images; (a2)–(d2) are the watermarked images; (a3)–(d3) are the random tampered watermarked images with α = 40%, 49%, 58%, 75%; (a4)–(d4) are the tampered blocks identification results; (a5)–(d5) are the tampering restoration results.

as in the first experiment, the tampered blocks are located correctly. The quantified and encoded DCT coefficients of tampered blocks were recovered without any error. In the three cases, PSNR values in the restored area are all 25.2 dB when regarding original image as reference. The experiment demonstrates that if the ratio $R = 3$, the proposed scheme can perfectly recover the tampered image as long as the tampering rate $\alpha \le 3/4$.

The random tampering experiments were also conducted (as seen in Figure 7). The gray scale images Lena, Baboon, Gold Hill, and Airplane in Figure 7((a1)–(d1)) are used as the host images. The watermarked images, shown in

Figure 7((a2)–(d2)), are generated as the lake. The values of PSNR due to watermark embedding are 37.9. The watermarked images are tampered randomly with tampering rate $\alpha = 40\%, 49\%, 58\%$, and 75%. The tampered images are shown in Figure 7((a3)–(d3)). The corresponding identification and restoration results are shown in Figure 7((a4)–(d4)) and Figure 7((a5)–(d5)). It can be seen that the recovery has been realized in all the experiments. PSNR values in the restored area are 27.8, 21.7, 26.9, and 26.9, respectively, when regarding original image as reference.

We compared the restoration capability of the proposed scheme with that of several other self-embedding watermark schemes. For the proposed scheme, the reference data embedded in one image block will be shared by all the image blocks. By using this global reference sharing mechanism and the special coding matrix, the problem of tampering coincidence is avoided absolutely. In the same experimental condition, the most extensive tampering area could be recovered and the recovery process is deterministic. Moreover, the quality of the restored content does not decrease as the percentage of tampering increases. However, the reference sharing mechanism had not been employed in the schemes [11, 20]. The reference data is embedded in another image block according to the block mapping. By this way the tampering coincidence cannot be avoided absolutely but only with the high probability. In the two schemes, if the tampering coincidence happened, the tampered blocks will be recovered with the neighborhood average. The maximal tampering rates are about 50% and 80%, respectively. But the data is obtained only by experiments, not by rigorous theoretical proof. Moreover, the PSNR of restored content in the two schemes decreases as the proportion of tampered area increases.

We also compare the restoration capability among different schemes based on the reference sharing mechanism. The PSNR between the original image and the watermarked image, the PSNR between the recovered image and the watermarked image or the original image, and the maximal tampering rate are considered. The experimental parameters of method 1 in [13] are the same as the proposed scheme when the ratio R is 2, which has been tested in the above experiment. All the two methods exploit 3 LSB watermark embedding. Therefore, the PSNR due to watermarking embedding is identical and equals 37.9 dB. The representative data blocks are all the 5 MSB of pixels in an image block and the length of the generated reference data blocks are all 160 bits. When the tampering rate is not larger than the maximal tampering rate, all the schemes can recover the representative data. PSNR values in restored area are identical and equal 40.7 dB when regarding original image as reference. But the maximal tampering rate of our proposed method is 33%, which is better than 24%, the maximal tampering rate of method 1 in [13]. The reason is the reference data embedded in one image block is shared by some image blocks but not all the image blocks. The local reference sharing method cannot get the maximum tamper ratio. The experimental parameters of the method in [21] are the same as the proposed scheme when the ratio R is 1/3, which has been tested in the above experiment. Both the proposed method and the method in [21] are based on the global reference sharing mechanism. They have the same restoration performance, while the encoding matrix applied to methods in [13, 21] is the random matrix. The random matrix can only promise the restoration can be successful with a great probability. In contrast the proposed method offers a deterministic self-embedding scheme by using the different encoding matrix.

6. Conclusion

In this paper, we proposed a self-embedding watermarking scheme based on the reference sharing mechanism. In the proposed scheme the special coding matrix is adopted and the global reference sharing mechanism is realized. The tampering coincidence and the reference waste are avoided. Based on our model, the maximal tampering rate can be derived in theory and considering the trade-off between the quality of recovered content and the restoration condition become more feasible. As long as the tampering rate is not larger than the maximal tampering rate, the representative data of the tampered image blocks can be recovered deterministically. The quality of the recovered content does not decrease as the proportion of tampered area increases. Moreover, the proposed scheme is reconfigurable. We experimentally evaluated the scheme in two sorts of configurations. Our experimental results demonstrate that the proposed method is effective.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper. The mentioned received funding in the "Acknowledgment" section did not lead to any conflict of interests regarding the publication of this manuscript.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant no. U1536110.

References

[1] Z. Zhou, Y. Wang, Q. M. J. Wu et al., "Effective and Efficient Global Context Verification for Image Copy Detection," *IEEE Transactions on Information Forensics Security*, vol. 12, no. 1, pp. 48–63, 2017.

[2] J. Wang, T. Li, Y. Shi, S. Lian, and J. Ye, "Forensics feature analysis in quaternion wavelet domain for distinguishing photographic images and computer graphics," *Multimedia Tools and Applications*, no. 76, pp. 23721–23737, 2017.

[3] J. Fridrich and M. Goljan, "Images with self-correcting capabilities," in *Proceedings of the International Conference on Image Processing (ICIP'99)*, pp. 792–796, October 1999.

[4] H. J. He, J. S. Zhang, and F. Chen, "Adjacent-block based statistical detection method for self-embedding watermarking techniques," *Signal Processing*, vol. 89, no. 8, pp. 1557–1566, 2009.

[5] X. Zhang, Z. Qian, Y. Ren, and G. Feng, "Watermarking with flexible self-recovery quality based on compressive sensing and

compositive reconstruction," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, pp. 1223–1232, 2011.

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

[7] C.-W. Yang and J.-J. Shen, "Recover the tampered image based on VQ indexing," *Signal Processing*, vol. 90, no. 1, pp. 331–343, 2010.

[8] C. Li, Y. Wang, B. Ma, and Z. Zhang, "A novel self-recovery fragile watermarking scheme based on dual-redundant-ring structure," *Computers Electrical Engineering*, vol. 37, no. 6, pp. 927–940, 2011.

[9] C. Qin, C.-C. Chang, and T.-J. Hsu, "Effective fragile watermarking for image authentication with high-quality recovery capability," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 11, pp. 2941–2956, 2013.

[10] 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.

[11] D. Singh and S. K. Singh, "DCT based efficient fragile watermarking scheme for image authentication and restoration," *Multimedia Tools and Applications*, vol. 76, pp. 1–25, 2015.

[12] X. Zhang, S. Wang, and G. Feng, "Fragile watermarking scheme with extensive content restoration capability," in *Proceedings of the International Workshop on Digital Watermarking*, pp. 268–278, 2009.

[13] 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.

[14] X. Zhang and S. Wang, "Fragile watermarking scheme using a hierarchical mechanism," *Signal Processing*, vol. 89, no. 4, pp. 675–679, 2009.

[15] 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.

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

[17] 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.

[18] 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.

[19] C. Qin, P. Ji, X. Zhang, J. Dong, and J. Wang, "Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy," *Signal Processing*, vol. 138, pp. 280–293, 2017.

[20] C. Qin, P. Ji, J. Wang, and C.-C. Chang, "Fragile image watermarking scheme based on VQ index sharing and self-embedding," *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 2267–2287, 2017.

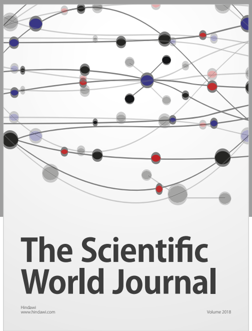
[21] P. Korus and A. Dziech, "Efficient method for content reconstruction with self-embedding," *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 1134–1147, 2012.

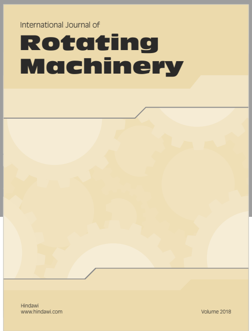
[22] P. Korus and A. Dziech, "Adaptive self-embedding scheme with controlled reconstruction performance," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 2, pp. 1134–1147, 2014.

[23] W.-C. Wu and Z.-W. Lin, "SVD-based self-embedding image authentication scheme using quick response code features," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 18–28, 2016.

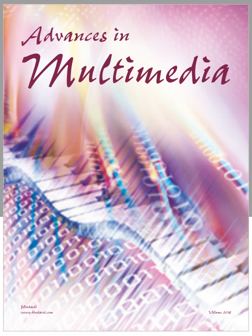
[24] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*, North Holland Publishing Co., Amsterdam, Netherlands, 1977.

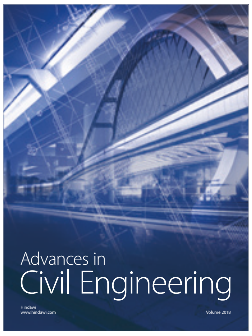
Journal of
Engineering

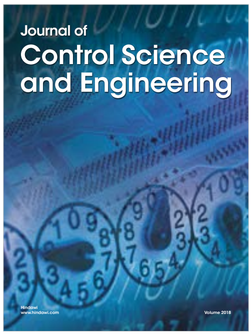

The Scientific
World Journal


International Journal of
Rotating
Machinery

Journal of
Sensors

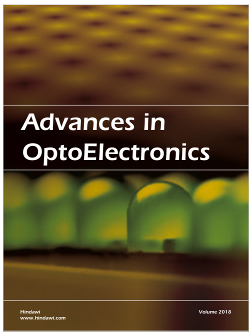

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

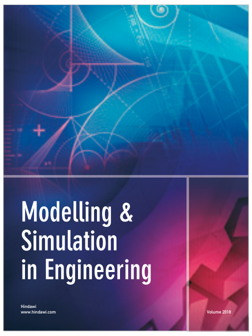
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

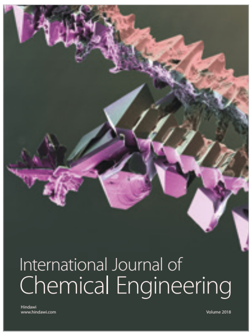

Advances in
OptoElectronics

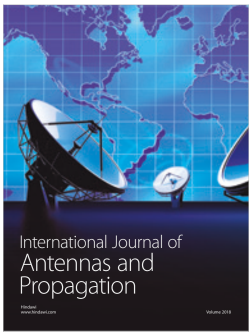
VLSI Design

International Journal of
Navigation and
Observation

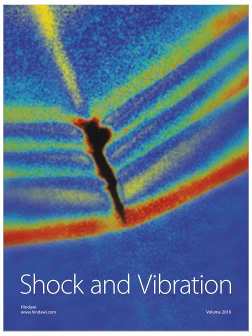

Modelling &
Simulation
in Engineering

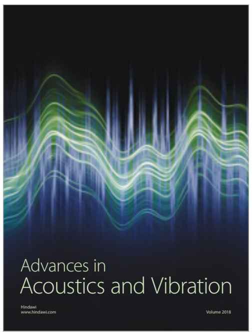
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration