Signal Processing 92 (2012) 1137–1150

Contents lists available at SciVerse ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

# Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism

Chuan Qin [a,b], Chin-Chen Chang [b,c,*], Pei-Yu Chen [d]

[a] School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
[b] Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan
[c] Department of Biomedical Imaging and Radiological Science, China Medical University, Taichung 40402, Taiwan
[d] Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 62102, Taiwan

ARTICLE INFO

*Article history:*
Received 4 July 2011
Received in revised form
25 September 2011
Accepted 12 November 2011
Available online 19 November 2011

*Keywords:*
Fragile watermarking
Self-embedding
Content restoration
Bit allocation
Nonsubsampled contourlet

ABSTRACT

In this paper, we propose a novel fragile watermarking scheme with content restoration capability. Authentication-bits are produced using the image hashing method with a folding operation. The low-frequency component of the nonsubsampled contourlet transform (NSCT) coefficients is used to encode the restoration-bits for each block by the adaptive bit allocation mechanism. During the bit allocation, all the blocks are categorized into different types according to their degree of smoothness, and, complex blocks, which are deemed to have higher priority than smooth blocks, are allocated more bits. Two algorithms are utilized to adjust the block classification and the binary representations in order to guarantee that the numbers of the self-embedding authentication-bits and restoration-bits are exactly suitable for 1-LSB embedding capacity. On the receiver side, the extracted authentication-bits and the decoded restoration-bits are used to localize and restore the tampered blocks, respectively. Due to the low embedding volume, the visual quality of the watermarked image is satisfactory. Experimental results also show that the proposed scheme provides high restoration quality.



## 1. Introduction

With the rapid development of networking and multimedia technologies, copyright protection for digital products becomes more and more important. Digital watermarking is one the information hiding technique [1–4] and it can be used as an effective tool for protecting digital contents, which has aroused significant interest among researchers [5]. Digital images are the most widely used, digital information medium in our Internet environment, attracting more research attention than any other digital medium. Currently, academic research efforts are focused mainly on three categories of digital image watermarking, i.e., robust watermarking, fragile watermarking, and semi-fragile watermarking. For robust watermarking, the watermark embedded in the cover image can be detected and identified, even if the watermarked image undergoes allowable and malicious attacks [6–9]. Therefore, this robustness can be used for ownership verification. Fragile watermarking is intended for checking the integrity and authenticity of digital images [10–14], but an embedded watermark is easily destroyed if the watermarked image undergoes any slight alteration. The robustness characteristic of semi-fragile watermarking is between that of robust watermarking and fragile watermarking [15–18]. The embedded watermark of the semi-fragile watermarking scheme is robust to the content-preserving manipulations, such as JPEG compression and

* Corresponding author at: Department of Information Engineering and Computer Science, Feng Chia University, 100 Wenhwa Road, Taichung 40724, Taiwan. Tel.: +886 4 24517250x3790; fax: +886 4 27066495.
*E-mail addresses:* qin@usst.edu.cn (C. Qin), alan3c@gmail.com (C.-C. Chang), petra.peiyu@gmail.com (P.-Y. Chen).

0165-1684/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2011.11.013

Gaussian low-pass filtering, but fragile to malicious attacks, such as cropping and content replacement. Our main focus is on fragile image watermarking.

Earlier research work on the fragile watermarking included attempts to design schemes for localizing the modified or tampered image regions, and many schemes usually divided a host image into small blocks and embedded the watermark into each block [10–12]. The embedded watermark is often a hash of the principal content about each cover image block. If an attacker tampers with the watermarked image, the image content and the extracted watermark corresponding to the tampered blocks cannot be matched so that the tampered blocks are detected and localized. The fragile watermarking method in [13] also was based on this block-wise technique, but it aimed at the image in JPEG format as the cover image and two bits were embedded into discrete cosine transform (DCT) coefficients of each $8\times 8$ block using the reversible data hiding technique. In addition to the localization of tampered blocks, this method can recover the intact region without any error. Recently, researchers have proposed some fragile watermarking methods to enhance those developed by earlier researchers, managing to restore the tampered regions after the tampering localization [19–25]. These kinds of methods often embed a compressed code of image content into the cover image. Once the watermarked image is tampered, this compressed code can be decompressed and used for content restoration. Since this compressed code is a compact representation of the cover image itself, these kinds of methods also are called self-embedding methods.

Fridrich et al. proposed two fragile watermarking approaches with recovery capability [19]. In the first method, the DCT coefficients of each $8\times 8$ block were quantized into 64 or 128 bits, which were used to replace the least significant bits (LSBs) of another block. In the second method, a low color depth version of the cover image generated by reducing the gray levels was cyclic-shifted and embedded into the pixel differences. After detecting the locations of the malicious modifications, the quantized DCT coefficients and the low color depth data extracted from the reserved regions can be exploited to recover the principal content of tampered regions. Instead of embedding the frequency coefficients, Yang et al. created an index table of the cover image via vector quantization (VQ) and embedded the VQ indices of all of the blocks together with authentication-bits generated by [10] into the cover image [20]. If the image is tampered, the index table can be extracted and used to restore the tampered regions, but the quality of the recovered image using this method is not high enough. In [22], a tailor-made watermark consisting of reference-bits and check-bits was embedded into the cover image using a reversible data hiding method. On the receiver side, the extracted and calculated check-bits can be compared to identify the tampered image blocks. The original image was reconstructed exactly using the reliable reference-bits extracted from other blocks. If the tampered area is not too extensive, this method can restore the cover image information with no errors, but the visual quality of the watermarked image of this method is relatively low. A reference-sharing mechanism for self-embedding fragile watermarking was proposed in [23]. The embedded watermark in this method was a reference derived from the original principal content in different regions and shared by these regions for content restoration. After detecting the tampered blocks, both reference data and the original content in the intact area were used to restore the tampered area. This method can restore the five most significant bits (MSBs) accurately, if the tampering rate is below 24%. Qian et al. managed to reduce the amount of the embedded data while maintaining good recovery quality [24]. A fast image inpainting technique was used in this method to restore the smooth blocks for decreasing the embedded data.

In this work, we propose a fragile watermarking scheme based on the nonsubsampled contourlet transform (NSCT) using an adaptive bit allocation mechanism. The proposed scheme utilizes two kinds of binary information, i.e., authentication-bits and restoration-bits, to authenticate integrity and restore content, respectively. The authentication-bits are generated using the image hashing strategy to localize the alterations [26,27]. The restoration-bits are produced by encoding the NSCT coefficients of the cover image. As we know, a good transform can capture the essence of an image with few basis elements. Due to the redundancy of the basis functions, the NSCT transform is more efficient in capturing and representing the image features than other transforms that use non-redundant basis functions, such as DCT. Therefore, we utilize the NSCT to represent the cover image and encode its coefficients for the restoration-bits. The NSCT construction consists of two parts, i.e., the nonsubsampled pyramid filter bank (NSPFB) and the nonsubsampled directional filter bank (NSDFB). The details concerning the implementation of NSCT can be found in [28,29].

Obviously, more restoration-bits will produce better reconstructed quality, but they could cause more serious damage to the cover image since more payloads are embedded. In order to balance the visual quality after embedding and the restoration capability, we encode the NSCT coefficients of the image blocks into restoration-bits with the dynamic lengths based on their degree of smoothness. By our adaptive bit allocation mechanism, the complex blocks are allocated more bits, while the smooth blocks are allocated fewer bits, and the same sized images have an equal volume of restoration-bits. The watermark consisting of authentication-bits and restoration-bits is only embedded into the 1-LSB plane of the cover image. Because modifications to the cover are small, the visual quality of watermarked image is satisfactory. On the receiver side, the tampered blocks can be detected accurately and restored with high quality after the watermark is extracted.

The rest of the paper is organized as follows. Section 2 describes the watermark self-embedding procedure of the proposed scheme. Section 3 presents the procedure of tampering localization and content restoration. Experimental results and analysis are given in Section 4, and Section 5 concludes the paper.

## 2. Watermark self-embedding procedure

The proposed scheme can localize and restore the tampered regions of the watermarked image by self-embedding the authentication-bits and the restoration-bits into the 1-LSB plane of the cover image. The authentication-bits for each cover image block are obtained by one image hashing algorithm that is integrated with a folding operation. The restoration-bits are generated by encoding the NSCT coefficients of the cover image. We use a bit allocation mechanism to adaptively represent the principle content of the image blocks with different types into a fixed length for 1-LSB embedding. The framework of the watermark self-embedding procedure is shown in Fig. 1.

### 2.1. Authentication-bits generation

First, the cover image $\mathbf{I}$ is divided into $n \times n$ non-overlapped blocks $\mathbf{B}_1$, $\mathbf{B}_2$, …, $\mathbf{B}_k$, and the 1-LSB plane of each block is set to zero, where $k$ is the number of total divided blocks. Then, we utilize the image hashing algorithm to calculate $m$-bits hash value for each image block. Image hashing can represent the image content in compact form, and visually different images have distinct hash values. In our research, we adopt an image hashing algorithm based on the DCT transform [27], and denote the DCT coefficients matrix of each block $\mathbf{B}_i$ as $\mathbf{C}_i$, where $i=1, 2,\ldots, k$. The size of $\mathbf{C}_i$ is also $n \times n$. To calculate the hash value of $\mathbf{B}_i$, $m$ low-pass, filtered, random masks, i.e., $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_m$, with the size of $n \times n$ are generated using a secret key. Therefore, the $m$-bits hash value $h_{i,1}, h_{i,2}, \ldots, h_{i,m}$ for $\mathbf{B}_i$ can be produced by Eq. (1). Note that the DC coefficient of $\mathbf{C}_i$ should be excluded in the summation for the performance of the image hashing algorithm. Any malicious tampering with the block may result in a serious change of the corresponding hash value

$$h_{i,j} = \begin{cases} 1, & \text{if } \sum_{x=1}^{n}\sum_{y=1}^{n} \mathbf{C}_i(x,y)\cdot\mathbf{M}_j(x,y) \geq 0 \quad i = 1,2,\ldots,k \\ 0, & \text{if } \sum_{x=1}^{n}\sum_{y=1}^{n} \mathbf{C}_i(x,y)\cdot\mathbf{M}_j(x,y) < 0 \quad j = 1,2,\ldots,m \end{cases} . \tag{1}$$

Inspired by the work of [13], we do not use the hash values directly as authentication-bits. Before embedding, the hash bits are folded in order to decrease the volume of the embedded data for the cover image. We generalize the folding method in [13] by randomly grouping the total $m \cdot k$ bits of the hash values into $\alpha \cdot k$ subsets, $\mathbf{S}_1$, $\mathbf{S}_2$, …, $\mathbf{S}_{\alpha \cdot k}$, using the secret key. Therefore, each subset contains $m/\alpha$ bits. After computing the modulo-2 sum of the $m/\alpha$ bits in each subset, $\alpha \cdot k$ bits are obtained for all $k$ blocks that are used as the final authentication-bits. The volume of hash bits is reduced significantly by this folding way.

Fig. 2 gives an example of the generation of the authentication-bits, in which the number of blocks $k$ is equal to 4, and the length $m$ of the hash value for each block is equal to 6. The value of $\alpha$ is set to 3 in this example. Therefore, after randomly grouping and the calculation of the modulo-2 sum, 12 authentication-bits are generated, which amounts to far less volume than the volume of the original hash bits. During the authentication procedure, these authentication-bits are used to judge the integrity of the blocks by a voting-based strategy, which is described in detailed in the next section.

### 2.2. Restoration-bits generation

We utilize the NSCT to decompose each cover image block into pre-determined levels. One low-frequency component and several high-frequency components with different directions are obtained for each block. Fig. 3(a)

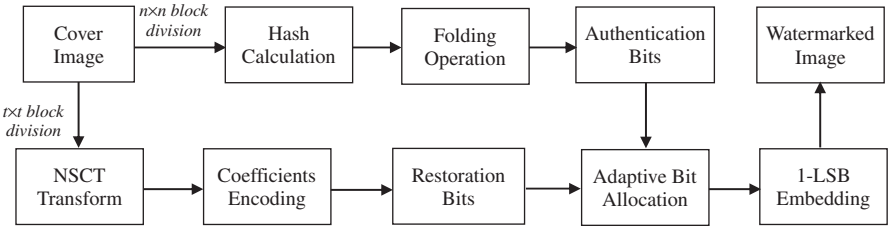


**Fig. 1.** Framework of the watermark self-embedding procedure.

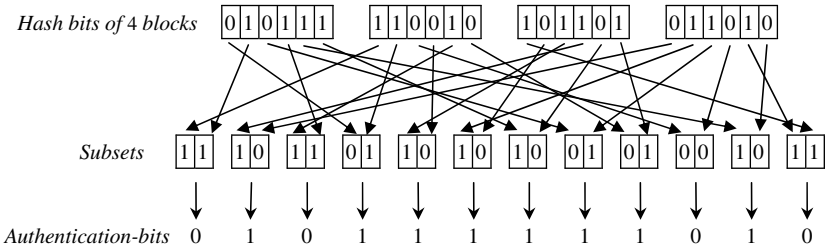


**Fig. 2.** Illustration of the generation of authentication-bits.

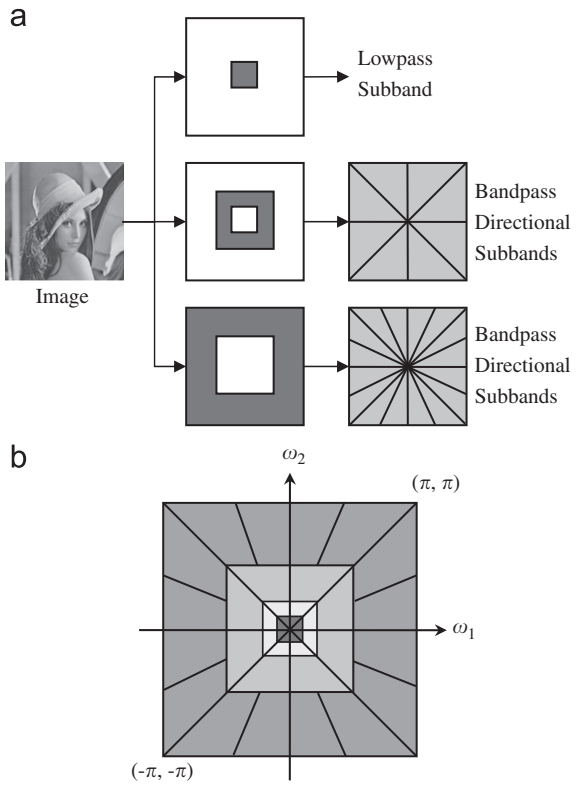


**Fig. 3.** Nonsubsampled contourlet transform. (a) Structure of nonsubsampled filter bank and (b) 2-D frequency partitioning.

gives an illustration of the NSCT. The filter bank in NSCT structure splits the 2-D frequency plane in the subbands shown in Fig. 3(b). The structures of NSPFB and NSDFB provide the multi-scale property and the directionality, respectively [28].

Different from the discrete wavelet transform (DWT), the decomposed components of NSCT all have the same size as the cover image blocks, which ensure the synchronization between the spatial domain and the frequency domain. This characteristic is very suitable for content restoration, because less image content is lost and no interpolation is needed. The high-frequency components consist mainly of noise, textures, and other details, while the low-frequency component reflects the principle structure and content of the image block. Therefore, we only use the low-frequency component of the NSCT decomposition results to generate the restoration-bits. The low-frequency component becomes smoother as the decomposition levels increase, and clearly, fewer bits are required to represent the smoother low-frequency component, which might also affect the restoration quality. Consequently, we can set the appropriate decomposition levels according to the requirement.

We denote the NSCT low-frequency components for the image blocks as $\mathbf{U}_i$, where $i=1, 2, \ldots, r$. The block size for NSCT decomposition is $t\times t$, which should not be smaller than 32 due to the property of NSCT and might not be equal to $n\times n$ for the calculation of authentication-bits. For simplicity, suppose that $t$ is divisible by $n$ and that $r$ is the number of all the blocks sized $t\times t$. In order to further decrease the volume of the embedded bits, we conduct the DCT transform to $\mathbf{U}_i$, and then, for each $\mathbf{U}_i$, the first $\beta$ percentage of the DCT coefficients in the zigzag scanning order are used for encoding the restoration-bits. The first one of these $\lfloor\beta\cdot t^2\rfloor$ coefficients is the DC value that reflects the information of mean illumination, which belongs to $[0,\ t\cdot 2^8-1]$. Thus, $8+\log_2 t$ bits are needed to represent the first coefficient losslessly. We group the rest $\lfloor\beta\cdot t^2\rfloor-1$ coefficients into $8+\log_2 t$ sets, i.e., $\Psi_j$, where $j=1, 2, \ldots, 8+\log_2 t$, according to the quota of bits needed for their binary representations. This indicates that the coefficients grouped in the set $\Psi_j$ all need $j$ bits to represent. Therefore, the quota of bits for losslessly representing the $\lfloor\beta\cdot t^2\rfloor$ coefficients of each $\mathbf{U}_i$ is:

$$W_i = 8+\log_2 t+\sum_{j=1}^{8+\log_2 t} Q_{i,j}\cdot j,\quad i=1,2,\ldots,r, \tag{2}$$

where $Q_{i,j}$ denotes the number of coefficients in the set $\Psi_j$ of $\mathbf{U}_i$, and it should satisfy

$$\sum_{j=1}^{8+\log_2 t} Q_{i,j}\equiv\lfloor\beta\cdot t^2\rfloor-1,\quad \text{for } \forall i\in\{1,2,\ldots,r\}. \tag{3}$$

### 2.3. Adaptive bit allocation mechanism

To allocate the bits quota to restoration-bits more efficiently, we divide all $r$ blocks $\mathbf{U}_i$ sized $t\times t$, into four types, $T_1$, $T_2$, $T_3$, and $T_4$, according to their values of $W_i$, where $i=1, 2,\ldots, r$. We assign two additional bits of {00, 01, 10, 11} to each $t\times t$ block to indicate the type of $\{T_1, T_2, T_3, T_4\}$ to which the block belongs. Because the bits of the type indicator are important in the decoding of restoration-bits, we duplicate the 2-bits of the type indicator into 4-bits to achieve decoding stability. We also use $\lceil\log_2(\lfloor\beta\cdot t^2\rfloor-1)\rceil$ bits to represent the number of the coefficients in set $\Psi_j$, i.e., $Q_{i,j}$, where $j=1, 2,\ldots, 8+\log_2 t$, for each $t\times t$ block $\mathbf{U}_i$. In the procedure of authentication-bits generation described in Section 2.1, $\alpha$-bits are produced for each $n\times n$ block. Therefore, the average spare bits quota for the restoration-bits in the 1-LSB plane of each $t\times t$ block is

$$\gamma = t^2-\alpha\cdot(t/n)^2-\lceil\log_2(\lfloor\beta\cdot t^2\rfloor-1)\rceil\cdot(8+\log_2 t)-4 \tag{4}$$

The process of classifying all $r$ blocks $\mathbf{U}_i$, where $i=1, 2, \ldots, r$, is assisted by setting three parameters, i.e., $L_1$, $L_2$, and $L_3$, where $L_1<L_2<L_3$

$$\begin{aligned}&\mathbf{U}_i\Rightarrow T_1 \quad \text{if } 8+\log_2 t<W_i\le L_1\\ &\mathbf{U}_i\Rightarrow T_2 \quad \text{if } L_1<W_i\le L_2\\ &\mathbf{U}_i\Rightarrow T_3 \quad \text{if } L_2<W_i\le L_3\\ &\mathbf{U}_i\Rightarrow T_4 \quad \text{if } W_i>L_3\end{aligned}\qquad i=1,2,\ldots,r, \tag{5}$$

where $\Rightarrow$ is the operation of classifying $\mathbf{U}_i$ into one of the four types, i.e., $T_1$, $T_2$, $T_3$, or $T_4$, and $L_3$ is equal to $\gamma$. Obviously, the blocks classified in $T_i$ must be more complex than the blocks in $T_{i-1}$, where $i=2, 3, 4$. We denote the numbers of the blocks in the four types after

classification as $N_1$, $N_2$, $N_3$, and $N_4$, respectively, and

$$\sum_{i=1}^{4} N_i \equiv r. \tag{6}$$

We manage to use $L_1$-bits, $L_2$-bits, $L_3$-bits, and $L_4$-bits ($L_4 > L_3$) to represent the blocks of the four types, i.e., $T_1$ through $T_4$, respectively. In other words, $L_i$ bits are allocated to represent the blocks in type $T_i$, where $i=1$, 2, 3, 4. The values of $L_1$, $L_2$, $L_3$, and $L_4$ are set to satisfy the following relationship:

$$\begin{matrix} L_3 - L_1 = \lambda_1 \cdot (L_4 - L_3) \\ L_3 - L_2 = \lambda_2 \cdot (L_4 - L_3) \end{matrix}, \quad \text{subject to } L_3 = \gamma, \text{ and } \lambda_1,\ \lambda_2 \in \mathbf{Z}^+, \tag{7}$$

where $\mathbf{Z}^+$ denotes the set of positive integers, $\lambda_1$ and $\lambda_2$ are the pre-determined parameters, and $\lambda_1$ is greater than $\lambda_2$. Before transforming all $r$ blocks $\mathbf{U}_i$, where $i=1, 2, \ldots, r$, into binary representations for restoration-bits, the blocks in the four types must be adjusted and reclassified in order to make the total number of bits suitable for the 1-LSB embedding capacity. The adjusting strategy is based on the fact that the quota of bits for the complex blocks should have higher priority than the smooth blocks to achieve better restoration quality. We utilize Algorithm 1 to adjust the block classification adaptively, and the pseudo code of Algorithm 1 is shown in Fig. 4.

The block numbers in the four types after reclassification must satisfy the relationship in Eq. (8), and the sum of the block number in each type must still be equal to $r$

$$L_1 \cdot N_1' + L_2 \cdot N_2' + L_4 \cdot N_4' = L_3 \cdot (N_1' + N_2' + N_4'), \quad \text{subject to } \sum_{i=1}^{4} N_i' \equiv r. \tag{8}$$

The number of bits for representing all $r$ blocks sized $t \times t$ in the four types is exactly suitable for the 1-LSB embedding capacity, see Eq. (9)

$$\sum_{i=1}^{4} L_i \cdot N_i' = r \cdot \gamma. \tag{9}$$

It should be noted that one can determine the appropriate values of $L_1$, $L_2$, and $L_4$ according to the requirement, since the different values of $L_1$, $L_2$, and $L_4$ correspond to different restoration particularities. Under the condition of $L_1 < L_2 < \gamma < L_4$ and Eq. (7), if the complex blocks are expected to be restored more effectively, a larger value for $L_4$ should be used, and larger $L_1$ and $L_2$ values ensure that the smooth blocks will have better restoration effects.

After the block reclassification using Algorithm 1, each $t \times t$ block is assigned with $L_1$-bits, $L_2$-bits, $L_3$-bits, or $L_4$-bits for binary representation according to the type $T_1$, $T_2$, $T_3$, or $T_4$, to which it belongs. We denote one $t \times t$ block as $\mathbf{U}_p$, which belongs to the type $T_q$, $q \in \{1, 2, 3, 4\}$. Because the needed bits $W_p$ for lossless representation of $\mathbf{U}_p$ calculated by Eq. (2) might not be equal to $L_q$, we utilize Algorithm 2 to adjust the binary representation for $\mathbf{U}_p$ to satisfy the relationship

$$L_q = 8 + \log_2 t + \sum_{j=1}^{8+\log_2 t} Q'_{p,j} \cdot j, \tag{10}$$

where $Q'_{p,j}$ is the adjusted number of the coefficients with $j$-bits representation in $\mathbf{U}_p$. The pseudo code of Algorithm 2 is shown in Fig. 5.

Consequently, after the adjustment using Algorithm 2, the length of the restoration-bits for each $t \times t$ block is equal to $L_1$, $L_2$, $L_3$, or $L_4$ and coincides with the type to which it belongs. Therefore, by our adaptive bit allocation mechanism, the complex blocks are assigned with a larger quota of bits, and the greater coefficients in each block are assigned with more bits for binary representation simultaneously.

### 2.4. LSB data embedding

After the authentication-bits and restoration-bits are generated, LSB embedding is conducted to hide the bits stream. According to the analysis in the previous Section 2.3, the length of the whole bits stream, consisting of authentication-bits, type indicators, number indicators, and restoration-bits, is fixed and is exactly equal to the 1-LSB embedding capacity, i.e., $r \cdot t^2$ bits. Table 1 lists the components and their corresponding numbers of bits in the bits stream. The type indicators, number indicators,

1 **If** ($N_4$ mod $\lambda_1$) mod $\lambda_2 \neq 0$; **then** $N_4' = N_4 + \lambda_2 - (N_4 \bmod \lambda_1) \bmod \lambda_2$; **else** $N_4' = N_4$; **end**.
2 **if** $N_4' \leq \lambda_1 \cdot N_1 + \lambda_2 \cdot N_2$;
3 **if** $N_1 \geq \lfloor N_4'/\lambda_1 \rfloor$; **then** $N_1' = \lfloor N_4'/\lambda_1 \rfloor$; **else** $N_1' = N_1$; $N_4' = N_4 + \lambda_2 - (N_4 - \lambda_1 \cdot N_1') \bmod \lambda_2$; **end**.
4 $N_2' = (N_4' - \lambda_1 \cdot N_1') / \lambda_2$; $N_3' = r - N_1' - N_2' - N_4'$.
5 **else**
6 find $N_1'$ and $N_2'$ to meet $N_4' = \lambda_1 \cdot N_1' + \lambda_2 \cdot N_2'$, and $N_1'$ is the maximum in its all possible solutions;
7 **if** $r - N_1' - N_2' - N_4' < 0$; **then** $N_4' = N_4' - 1$; go to the step in line 6;
8 **else** $N_3' = r - N_1' - N_2' - N_4'$; **end**.
9 **end**.
10 Sort all $r$ blocks $\mathbf{U}_i$ in the ascending order according to the corresponding $W_i$: $\mathbf{U}_{i'}$ ($i = 1, 2, \ldots, r$).
11 $\{ \mathbf{U}_{i'}, i \in [1, N_1'] \} \Rightarrow T_1$;
12 $\{ \mathbf{U}_{i'}, i \in [N_1' + 1, N_1' + N_2'] \} \Rightarrow T_2$;
13 $\{ \mathbf{U}_{i'}, i \in [N_1' + N_2' + 1, N_1' + N_2' + N_3'] \} \Rightarrow T_3$;
14 $\{ \mathbf{U}_{i'}, i \in [N_1' + N_2' + N_3' + 1, N_1' + N_2' + N_3' + N_4'] \} \Rightarrow T_4$.

**Fig. 4.** Algorithm 1 for the adjustment of block classification.

1 $D = L_q - W_p$; $Q'_{p,j} = Q_{p,j}$, $j = 1, 2, \ldots, 8+\log_2 t$;
2 **if** $D > 0$;
3 **Loop** and terminate until $D = 0$;
4 $Q'_{p,\eta} = Q'_{p,\eta} - 1$, where $\eta$ is the minimum of $j$, subject to $Q'_{p,\eta} \neq 0$;
5 $Q'_{p,\eta+1} = Q'_{p,\eta+1} + 1$; $D = D - 1$;
6 **end**.
7 **end**.
8 **if** $D < 0$;
9 **Loop** and terminate until $D = 0$;
10 $Q'_{p,\eta} = Q'_{p,\eta} - 1$, where $\eta$ is the minimum of $j$, subject to $Q'_{p,\eta} \neq 0$;
11 $Q'_{p,\eta-1} = Q'_{p,\eta-1} + 1$; $D = D + 1$;
12 **end**.
13 **end**.

**Fig. 5.** Algorithm 2 for the binary representation adjustment of each $t \times t$ block.

**Table 1**
Components of the bits stream for embedding.

| Components | Descriptions | Bit numbers |
|---|---|---|
| Authentication-bits | the authentication-bits of all the $k$ blocks sized $n \times n$ | $k \cdot \alpha$ |
| Type indicators | two duplicated type indicators for each $\mathbf{U}_i$ sized $t \times t$ ($i=1, 2, \ldots, r$) | $4r$ |
| Number indicators | the number indicator $Q'_{i,j}$ for $\lfloor \beta \cdot t^2 \rfloor - 1$ AC coefficients in each $\mathbf{U}_i$ ($j=1, 2, \ldots, 8+\log_2 t$) | $r \cdot (8+\log_2 t) \cdot \lceil \log_2 (\lfloor \beta \cdot t^2 \rfloor - 1) \rceil$ |
| Restoration-bits | One DC coefficient and $\lfloor \beta \cdot t^2 \rfloor - 1$ AC coefficients for each $\mathbf{U}_i$ | $r \cdot \left[ (8+\log_2 t) + \sum_{j=1}^{8+\log_2 t} Q'_{i,j} \cdot j \right]$ |

| TI | DC | $\mathbf{NI}_1$ | $\{\mathbf{AC}_1\}$ | $\mathbf{NI}_2$ | $\{\mathbf{AC}_2\}$ | ... | $\mathbf{NI}_{8+\log_2 t}$ | $\{\mathbf{AC}_{8+\log_2 t}\}$ |
|---|---|---|---|---|---|---|---|---|

**Fig. 6.** Structure of the reference-bits of each $t \times t$ block.

and restoration-bits are referred to as reference-bits in together. The structure of the reference-bits of each $t \times t$ block $\mathbf{U}_i$, where $i=1, 2, \ldots, r$, is illustrated in Fig. 6. **TI** expresses the 4-bits binary representation of the duplicated type indicators. **DC** represents the binary representation of the DC coefficient that occupies $8+\log_2 t$ bits. Each number indicator, $\mathbf{NI}_j$, occupies $\lceil \log_2 (\lfloor \beta \cdot t^2 \rfloor - 1) \rceil$ bits. $\{\mathbf{AC}_j\}$ denotes the $j$th set of binary representations for $\mathbf{NI}_j$ AC coefficients, and all binary representations in the $j$th set are commonly $j$ bits ($j=1, 2, \ldots, 8+\log_2 t$).

We first embed the $\alpha$ authentication-bits of each $n \times n$ block to the random locations in the 1-LSB plane of the block itself. As a result, each $n \times n$ block can be authenticated through the embedded authentication-bits on the receiver side. The procedure of tampered blocks localization is expressed in detail in the subsequent section. After embedding the authentication-bits, the remaining 1-LSB space of the image is used to hide the reference-bits for content restoration. The reference-bits of all $r$ blocks sized $t \times t$ are first concatenated, and then scrambled by a secret key. After substituting the remaining 1-LSB space of the cover image with the scrambled reference-bits, we can obtain the final watermarked image.

## 3. Authentication and restoration procedure

On the receiver side, first, the integrity of the received image should be identified, and the damaged regions should be localized. Then, the detected damaged regions are restored. The framework of the procedure for authentication and restoration is shown in Fig. 7. Assume that the size of the received image has not changed. The embedded bits are extracted from the 1-LSB plane of the received image, and reversed using the same secret key. Authentication and localization of the tampered blocks are achieved by comparing the extracted authentication-bits with the calculated authentication-bits. The NSCT reference image, which is obtained from the decoded restoration-bits, is utilized to restore the tampered blocks.

### 3.1. Tampered blocks localization

In the same way as the self-embedding stage described in Section 2.1, the watermarked image is divided into $k$ non-overlapped blocks sized $n \times n$. We can obtain $m$ hash bits for each $n \times n$ block using the same image hashing algorithm and compress the $m$ hash bits into $\alpha$ bits, called

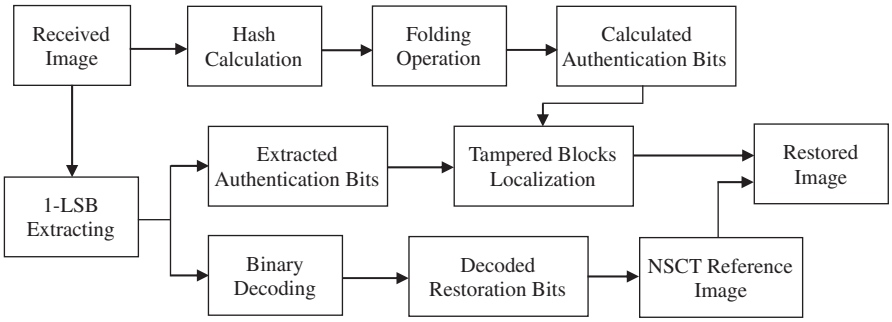


**Fig. 7.** Framework of the procedure for authentication and restoration.

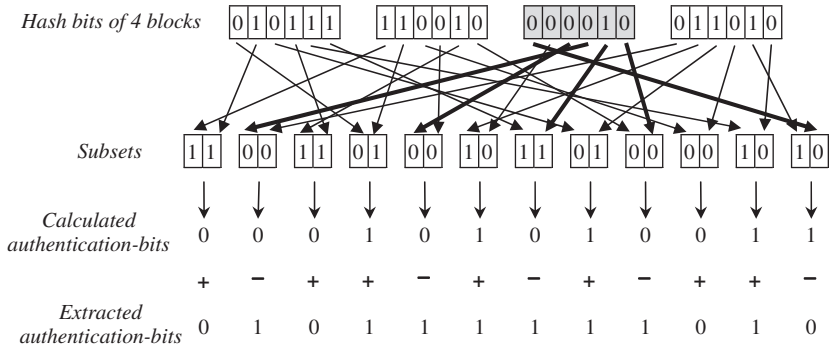


**Fig. 8.** Illustration of voting-based localization of tampered blocks.

calculated authentication-bits, by the folding operation. The extracted authentication-bits can be acquired easily after the 1-LSB extracting. Then, tampering localization can be achieved using a voting-based strategy. Each calculated authentication-bit is the modulo-2 sum of $m/\alpha$ bits in one subset, and the $m/\alpha$ bits in the subset are randomly distributed from all the hash bits of the $k$ blocks. Therefore, we define a voting parameter $\sigma_i$, where $i=1, 2, \ldots, k$, for each block to record the number of its hash bits distributed in the subsets whose calculated authentication-bits are not equal to the corresponding extracted authentication-bits.

Fig. 8 gives an example of the localization of tampered blocks using the voting strategy, in which the number of blocks, the length of the hash bits for each block, and the number of subsets are the same as the example shown in Fig. 2. Suppose the content of the third block is tampered. We can see that the mismatched indices between the calculated authentication-bits and extracted authentication-bits are 2nd, 5th, 7th, 9th, and 12th, which are generated from the corresponding subsets. Match and mismatch are indicated by '+' and '−', respectively. We count the voting parameter $\sigma_i$ for each block, which is equal to the number of its distributed subsets belonging to the mismatched subsets: 2nd, 5th, 7th, 9th, and 12th. The results are listed in Table 2. It can be found that the voting parameter $\sigma_3$ of the third block, i.e., the tampered block, is significantly larger than those of other intact blocks. Therefore, we can utilize the voting parameter $\sigma_i$ to identify and localize the tampered blocks by setting the appropriate threshold $\sigma$.

**Table 2**
Voting parameter $\sigma_i$ for each $n \times n$ block.

| Block index | Distributed subsets of hash bits | $\sigma_i$ |
|---|---|---|
| The 1st block | 4th, 1st, 8th, 3rd, 11th, 7th | $\sigma_1=1$ |
| The 2nd block | 1st, 4th, 10th, 5th, 3rd, 9th | $\sigma_2=2$ |
| The 3rd block | 12th, 6th, 5th, 2nd, 7th, 9th | $\sigma_3=5$ |
| The 4th block | 2nd, 6th, 8th, 10th, 12th, 11th | $\sigma_4=2$ |

### 3.2. Content restoration

In addition to the authentication-bits, the remaining portion of the extracted 1-LSB bits stream is the reference-bits, which include type indicators, number indicators, and restoration-bits, as shown in Table 1. To conduct content restoration for the tampered blocks, the NSCT reference image should be generated by decoding the restoration-bits from the reference-bits according to the bits structure in Fig. 6.

First, the reference-bits stream is segmented into $r$ parts using the type indicators, which indicate the different encoding lengths for different types of blocks. Each part corresponds to the reference-bits for one $t \times t$ block.

Then, the restoration-bits for each $t \times t$ block, consisting of the $8+\log_2 t$ bits DC coefficient and the $8+\log_2 t$ sets of AC coefficients, can be obtained according to the number indicators, which indicate the number of AC coefficients in each set. Finally, the NSCT reference image can be generated easily by conducting the inverse DCT on the obtained DC and AC coefficients of all $r$ blocks. Content restoration can be achieved by substituting the localized tampered blocks with the corresponding blocks of the NSCT reference image.

It should be noted that the reference-bits are capable of error correction, if the errors happen to occur in the sections of type indicators **TI** and number indicators $\mathbf{NI}_j$ in Fig. 6. In this case, the restoration-bits cannot be decoded accurately, because the lengths of the reference-bits and $\{\mathbf{AC}_j\}$ for each block are strictly consistent with **TI** and $\mathbf{NI}_j$, respectively. If the blocks, in which the bits of the sections **TI** and $\mathbf{NI}_j$ are embedded, are tampered, they can be localized by the authentication procedure. Therefore, exhaustive trials can be conducted on the error bits of the sections **TI** and $\mathbf{NI}_j$, until the restoration-bits can be decoded properly.

When the tampered image region is too extensive, it might occur that one block (denoted as A) and its corresponding block (denoted as B), in which the restoration-bits of block A are embedded, are destroyed simultaneously. In this scenario, the destroyed restoration-bits embedded in block B cannot be used to restore block A. To deal with this problem, we utilize the image inpainting technique to automatically repair the blocks whose corresponding restoration-bits are destroyed. The inpainting algorithm based on partial differential equation (PDE) proposed in [30] is adopted in this work, see Eq. (11)

$$\frac{\partial}{\partial t^*} I_i(x,y) = \nabla\left[\Delta I_i(x,y)\right] \cdot \nabla^{\perp} I_i(x,y), \quad \forall (x,y) \in \Omega, \tag{11}$$

where $I_i$ denotes the image in which the recoverable blocks are restored by their restoration-bits, $(x, y)$ is the coordinates of the image pixel, $\Omega$ is the set of the blocks that cannot be restored by the corresponding restoration-bits, $t^*$ denotes the time index, $\Delta$ and $\nabla$ are the Laplacian operator and the gradient operator, respectively, and $\nabla^{\perp}$ denotes the isophote direction that is normal to the gradient $\nabla$. The PDE-based model in Eq. (11) imitates the practice of a traditional art professional in the manual repairing process, and propagates the useful neighboring information of gray values along isophote direction into the damaged blocks, which cannot be restored by the restoration-bits, and completes the whole restoration procedure.

## 4. Experimental results

In the experiment, the sizes of the divided image blocks for the hashing calculation and the NSCT transform were $8 \times 8$ and $32 \times 32$, i.e., $n=8$ and $t=32$, respectively. The length $m$ of the hash bits for each $8 \times 8$ block was 33, and the threshold $\sigma$ for the voting-based tampering localization was 8. The parameters $\alpha$ and $\beta$ were set to 3 and 0.098, respectively. That is to say, the 33 hash bits of each $8 \times 8$ block produced 3 authentication-bits by the folding operation, and $\lfloor 0.098 \times 32^2 \rfloor = 100$ DCT coefficients for each $32 \times 32$ block of the NSCT low-frequency component were used to generate the restoration-bits. Larger $\alpha$ and $\beta$ values may enhance the accuracy of localization and restoration, but they also lead to embedding with more bits, which could affect the quality of the watermarked image.

Two standard gray images, Lena and Boat, both sized $512 \times 512$, were used as cover images for self-embedding by the proposed scheme. For the cover images sized $512 \times 512$, $L_3$ was always equal to 881, calculated by Eq. (4). In our experiment, all of the 256 blocks sized $32 \times 32$ were classified adaptively into four types, i.e., $T_1$, $T_2$, $T_3$, and $T_4$; $\lambda_1$ and $\lambda_2$ in Eq. (7) were equal to 5 and 2, respectively; $L_1$, $L_2$, and $L_4$ were set to 536, 743, and 950, respectively, which were satisfied with Eq. (7). As stated in Section 2.3, one can also set her or his own $L_1$, $L_2$, and $L_4$ values according to the requirement. By Algorithm 1, the types $T_1$ through $T_4$ of the Lena image consist of 4, 2, 226, and 24 blocks, respectively, while the types $T_1$ through $T_4$ of the Boat image consist of 8, 0, 208, and 40 blocks, respectively. This means that 536, 743, 881, and 950 bits are correspondingly used to represent the restoration-bits

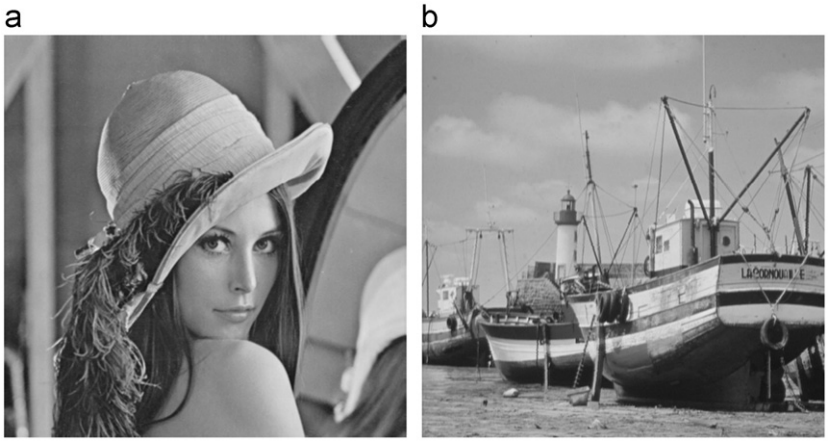


**Fig. 9.** Cover images of Lena and Boat.

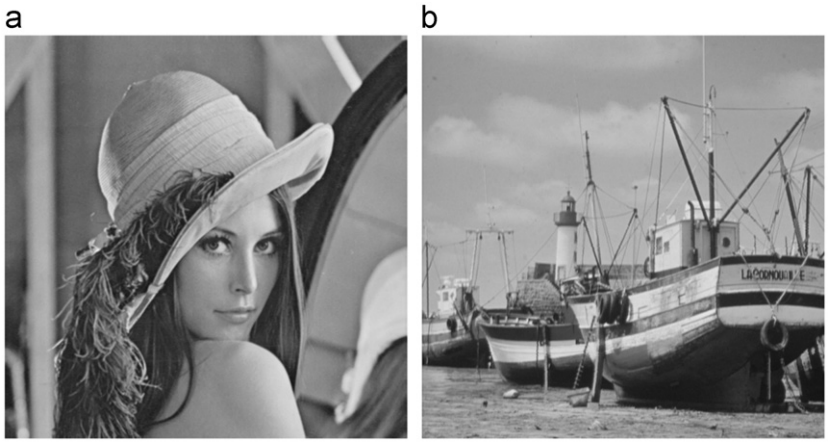


**Fig. 10.** Watermarked images of Lena and Boat.


**Fig. 11.** Tampered versions of Lena and Boat in Fig. 10.

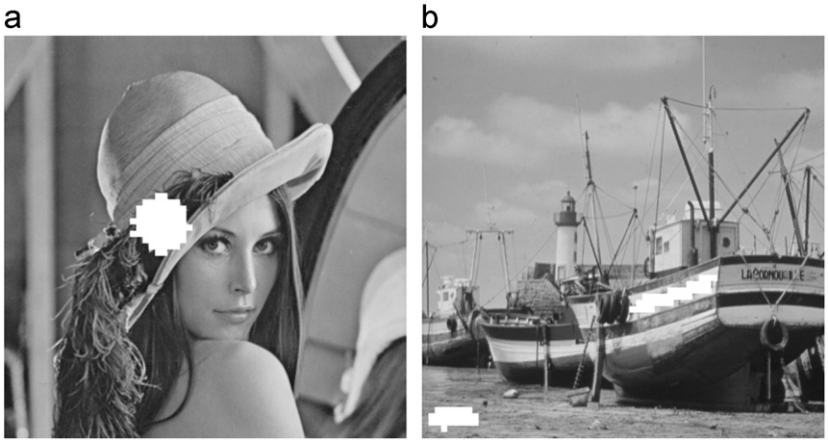


**Fig. 12.** Localization results for the tampered images Lena and Boat.

of 4, 2, 226, and 24 blocks for Lena, while 8, 0, 208, and 40 blocks for Boat, respectively.

Fig. 9 gives the cover images of Lena and Boat. Their watermarked versions are shown in Fig. 10(a) and (b), respectively. The values of the peak signal-to-noise ratio (PSNR) due to watermark embedding are 51.2 dB and 51.1 dB, respectively, and the visual distortions are imperceptible. The two watermarked images were altered by

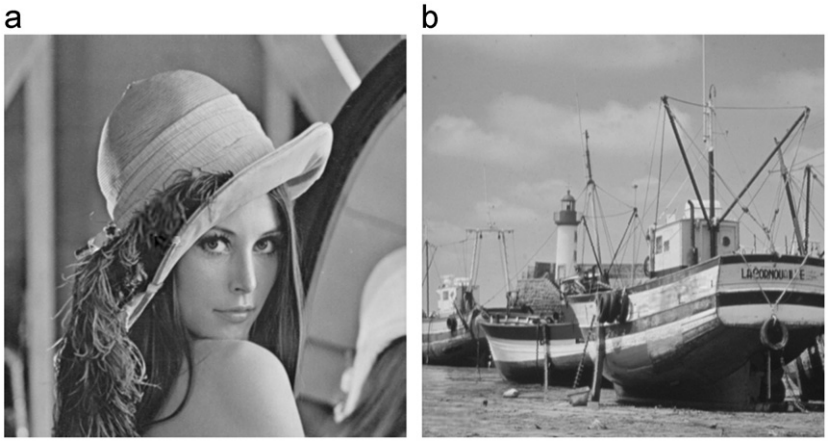


**Fig. 13.** Restored images for the tampered images Lena and Boat.


**Fig. 14.** Cover images of Gold and Girl and their tampered, watermarked versions.

substituting their partial contents with the fake information, shown as Fig. 11(a) and (b). According to the extracted and calculated authentication-bits, all fake blocks in the two tampered, watermarked images were localized and labeled with white, as shown in Fig. 12(a) and (b). The two restored images from the extracted, decoded restoration-bits are shown in Fig. 13(a) and (b), and the corresponding PSNR values are 41.8 dB and 44.8 dB, respectively.

Besides Lena and Boat, two other standard gray images, i.e., Gold and Girl, sized $512 \times 512$, were used to compare the content restoration performance between several self-embedding watermarking schemes. We generated the watermarked images using the proposed scheme and the


**Fig. 15.** Restored results of Gold with the methods in [20,23,24], and the proposed scheme.

methods in [20,23,24] and modified the contents of the watermarked images. Fig. 14(a) and (b) show the cover images Gold and Girl, and Fig. 14(c) and (d) are the corresponding tampered versions of the watermarked images by the proposed scheme. In Figs. 15 and 16, (a) through (d) show the restored results using the methods in [20,23,24], and the proposed scheme, respectively. Table 3 gives the PSNR values of the watermarked images and the restored images for Lena, Boat, Gold, and Girl. The methods in [20,23] used the 3-LSB planes to embed the watermark, while the proposed scheme and [24] only used the 1-LSB plane. Therefore, it can be observed that the PSNR values of watermarked images of the proposed scheme and [24] are higher than the values of [20,23]. The length of the codebook used in [20] was 256. Because the method in [23] can recover the 5-MSB of the tampered images with no error when the tampering rate is below 24%, the PSNR values of the images restored by this method are almost the same. In the results of Figs. 15 and 16 and Table 3, the methods in [20,23], and the proposed scheme just restore the localized tampered regions, while the method in [24] reconstructs the whole image region since this method does not emphasize on the tampering localization. Due to the good performance of adaptive bit allocation for the restoration-bits, it can be seen in Table 3 that the proposed scheme can restore the tampered images efficiently and that its restored images have higher PSNR values than those restored by [20,23,24]. Note that the tampering percentages in Figs. 11 and 14 were less than 5%.

We also conducted the experiments under the condition that the tampered image areas were relatively larger. As we stated in Section 3, the restoration-bits and the inpainting algorithm are jointly utilized to restore the extensive tampered region. First, the tampered blocks have the priority to be restored by their corresponding intact restoration-bits. Then, the inpainting algorithm in [30] is adopted to restore the remaining tampered blocks whose restoration-bits are destroyed due to the extensive tampering. Fig. 17(a) and (b) show the tampered, watermarked versions of Lena and Gold, and the tampering percentages are 7.0% and 27.3%, respectively. The restored images are presented in Fig. 17(c) and (d), and the corresponding PSNR values are 39.50 dB and 38.78 dB, respectively.

## 5. Conclusions

In this paper, we have proposed a self-embedding fragile watermarking scheme based on the adaptive bit

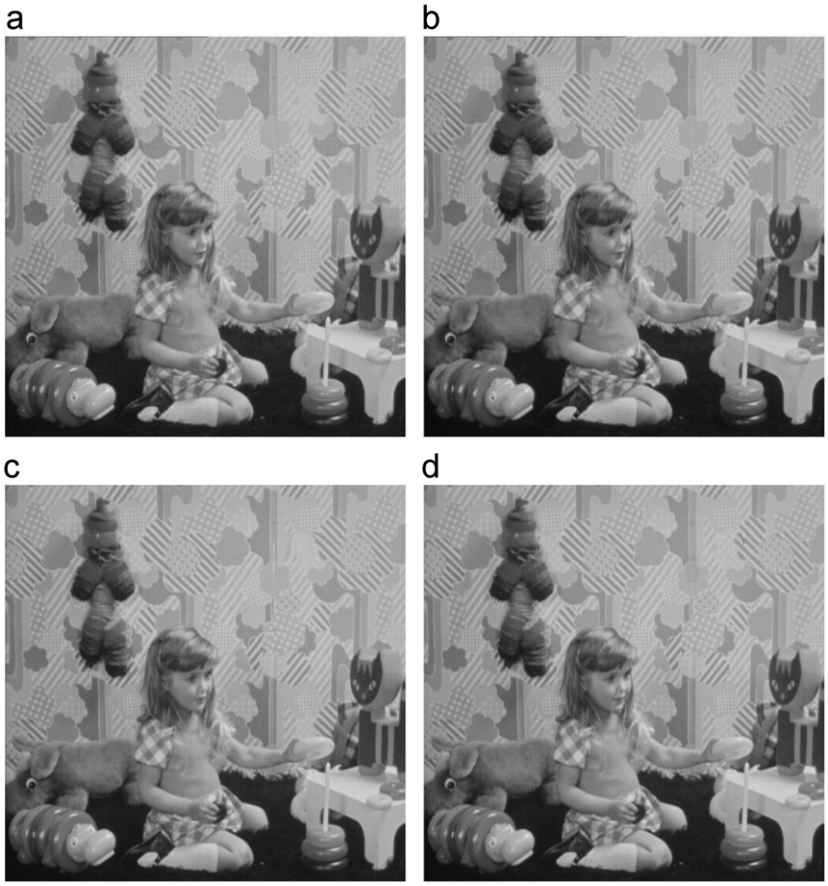


**Fig. 16.** Restored results of Girl with the methods in [20,23,24], and the proposed scheme.

**Table 3**
Comparisons of several self-embedding watermarking schemes.

| Watermarking schemes | PSNR of watermarked images (dB) | | | | PSNR of restored images (dB) | | | | Volume of embedded data |
|---|---|---|---|---|---|---|---|---|---|
| | Lena | Boat | Gold | Girl | Lena | Boat | Gold | Girl | |
| Method in [20] | 37.9 | 37.9 | 37.8 | 37.8 | 37.3 | 37.6 | 37.4 | 37.7 | 3-LSB |
| Method in [23] | 37.9 | 37.9 | 37.9 | 37.9 | 40.7 | 40.7 | 40.7 | 40.7 | 3-LSB |
| Method in [24] | 51.2 | 51.1 | 51.1 | 51.1 | 29.2 | 35.2 | 32.3 | 30.6 | 1-LSB |
| Proposed scheme | 51.2 | 51.1 | 51.1 | 51.1 | 41.8 | 44.8 | 45.1 | 48.4 | 1-LSB |

allocation mechanism. Authentication-bits and restoration-bits are embedded into the cover image to realize tampering localization and content restoration, respectively. A DCT-based image hashing algorithm is used to produce the hash bits for each cover block, and the hash bits are reduced to the authentication-bits by the folding operation. We utilize the NSCT transform to obtain the principle content of the image, and encode the NSCT coefficients into the restoration-bits using adaptive bit allocation. All cover blocks are categorized into four types according to the degree of smoothness. The type of complex blocks is allocated more bits, and the type of smooth blocks is allocated fewer bits. Two algorithms are used to adjust the block classification and binary representation for each block adaptively. By the adjustment of these two algorithms, the watermark bits number of total blocks is exactly suitable for embedding into the 1-LSB plane. On the receiver side, using a voting-based strategy, the calculated authentication-bits and the extracted authentication-bits are compared to localize the tampered blocks. After decoding, the restoration-bits are used to restore the localized, tampered blocks. When the tampered region is not too extensive, the PSNR values of the restored images are high due to the good performance of the adaptive bit allocation for restoration-bits.

Future work should include the development of a semi-fragile watermarking scheme that can simultaneously


**Fig. 17.** Restoration performance for larger tampered area.

provide good restoration capability and resist some kinds of content-preserving manipulations.

## References

[1] F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, Information hiding—a survey, Proceedings of the IEEE 87 (7) (1999) 1062–1078.
[2] C.D. Vleeschouwer, J.F. Delaigle, B. Macq, Invisibility and application functionalities in perceptual watermarking: an overview, Proceedings of the IEEE 90 (1) (2002) 64–77.
[3] W.L. Tai, C.M. Yeh, C.C. Chang, Reversible data hiding based on histogram modification of pixel differences, IEEE Transactions on Circuits and Systems for Video Technology 19 (6) (2009) 906–910.
[4] C.C. Chang, T.D. Kieu, A reversible data hiding scheme using complementary embedding strategy, Information Sciences 180 (16) (2010) 3045–3058.
[5] C.C. Chang, Y.S. Hu, T.C. Lu, A watermarking-based image ownership and tampering authentication scheme, Pattern Recognition Letters 27 (5) (2006) 439–446.
[6] I.J. Cox, J. Kilian, F.T. Leighton, T. Shamoon, Secure spread spectrum watermarking for multimedia, IEEE Transactions on Image Processing 6 (12) (1997) 1673–1687.
[7] X.G. Kang, J.W. Huang, Y.Q. Shi, Y. Lin, A DWT-DFT composite watermarking. scheme robust to both affine transform and JPEG compression, IEEE Transactions on Circuits and Systems for Video Technology 13 (8) (2003) 776–786.
[8] C.C. Chang, P.Y. Tsai, M.H. Lin, SVD-based digital image watermarking scheme, Pattern Recognition Letters 26 (10) (2005) 1577–1586.
[9] P.Y. Lin, J.S. Lee, C.C. Chang, Dual digital watermarking for internet media based on hybrid strategies, IEEE Transactions on Circuits and Systems for Video Technology 19 (8) (2009) 1169–1171.
[10] P.W. Wong, N. Memon, Secret and public key image watermarking schemes for image authentication and ownership verification, IEEE Transactions on Image Processing 10 (10) (2001) 1593–1601.
[11] S. Suthaharan, Fragile image watermarking using a gradient image for improved localization and security, Pattern Recognition Letters 25 (16) (2004) 1893–1903.
[12] H. Yang, A.C. Kot, Binary image authentication with tampering localization by embedding cryptographic signature and block identifier, IEEE Signal Processing Letters 13 (12) (2006) 741–744.
[13] X.P. Zhang, S.Z. Wang, Z.X. Qian, G.R. Feng, Reversible fragile watermarking for locating tampered blocks in JPEG images, Signal Processing 90 (12) (2010) 3026–3036.
[14] X.P. Zhang, S.Z. Wang, Fragile watermarking scheme using a hierarchical mechanism, Signal Processing 89 (4) (2009) 675–679.
[15] C.Y. Lin, S.F. Chang, A robust image authentication method distinguishing JPEG compression from malicious manipulation, IEEE Transactions on Circuits and Systems for Video Technology 11 (2) (2001) 153–168.
[16] C.Y. Lin, S.F. Chang, Semi-fragile watermarking for authenticating JPEG visual content, in: Proceedings of SPIE International Conference on Security and Watermarking of Multimedia Contents II, vol. 3971, Jan. 2000, pp. 140–151.
[17] X.Z. Zhu, A.T.S. Ho, P. Marziliano, A new semi-fragile image watermarking with robust tampering restoration using irregular sampling, Signal Processing: Image Communication 22 (5) (2007) 515–528.
[18] Z.M. Lu, C.H. Liu, D.G. Xu, S.H. Sun, Semi-fragile image watermarking method based on index constrained vector quantization, Electronics Letters 39 (1) (2003) 35–36.
[19] J. Fridrich, M. Goljan, Images with self-correcting capabilities, in: Proceedings of IEEE International Conference on Image Processing, 1999, pp. 792–796.
[20] C.W. Yang, J.J. Shen, Recover the tampered image based on VQ indexing, Signal Processing 90 (1) (2010) 331–343.

[21] H.J. He, J.S. Zhang, F. Chen, Adjacent-block based statistical detection method for self-embedding watermarking techniques, Signal Processing 89 (8) (2009) 1557–1566.
[22] X.P. Zhang, S.Z. Wang, Fragile watermarking with error-free restoration capability, IEEE Transactions on Multimedia 10 (8) (2008) 1490–1499.
[23] X.P. Zhang, S.Z. Wang, Z.X. Qian, G.R. Feng, Reference sharing mechanism for watermark self-embedding, IEEE Transactions on Image Processing 20 (2) (2011) 485–495.
[24] Z.X. Qian, G.R. Feng, Inpainting assisted self recovery with decreased embedding data, IEEE Signal Processing Letters 17 (11) (2010) 929–932.
[25] Z.X. Qian, G.R. Feng, X.P. Zhang, S.Z. Wang, Image self-embedding with high-quality restoration capability, Digital Signal Processing 21 (2) (2011) 278–286.
[26] A. Swaminathan, Y.N. Mao, M. Wu, Robust and secure image hashing, IEEE Transactions on Information Forensics and Security 1 (2) (2006) 215–230.
[27] J. Fridrich, M. Goljan, Robust hash function for digital watermarking, in: Proceedings of International Conference on Information Technology: Coding and Computing, 2000, pp. 173–178.
[28] A.L. Cunha, J.P. Zhou, M.N. Do, The nonsubsampled contourlet transform: theory, design, and applications, IEEE Transactions on Image Processing 15 (10) (2006) 3089–3101.
[29] A.L. Cunha, J.P. Zhou, M.N. Do, Nonsubsampled contourlet transform: filter design and application in denoising, in: Proceedings of IEEE International Conference on Image Processing, 2005, pp. 749–752.
[30] M. Bertalmio, A.L. Bertozzi, G. Sapiro, Navier–Stokes, fluid dynamics, and image and video inpainting, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2001, pp. 355–362.